Н. А. Молдовян, А. А. Молдовян, М. А. Еремеев

Криптография

От примитивов к синтезу алгоритмов

НАУЧНОЕ ИЗДАНИЕ

bhv®

Н.А. Молдовян, А.А. Молдовян, М.А. Еремеев

Криптография

От примитивов к синтезу алгоритмов

Санкт-Петербург
«БХВ-Петербург»
2004

УДК 681.3
ББК 32.81
М75

***Рецензенты*:**
доктор технических наук, профессор В.И. Коржик
доктор технических наук, профессор А.М. Чуднов

Молдовян Н. А., Молдовян А. А., Еремеев М. А.
М75 Криптография: от примитивов к синтезу алгоритмов. — СПб.: БХВ-Петербург, 2004. — 448 с.: ил.

ISBN 5-94157-524-6

В книге приводятся элементы математических основ криптографии. Раскрывается содержание симметричных и асимметричных шифров, систем цифровой электронной подписи и хэш-функций и основные требования к ним. Излагаются новые результаты в направлении проектирования скоростных шифров на основе управляемых преобразований.

Представлена классификация управляемых примитивов, на основе которых синтезируются новые классы операций, зависящих от преобразуемых данных. Анализируются основные свойства управляемых примитивов. Дается описание ряда новых криптографических примитивов и алгоритмов с оценкой их стойкости к дифференциальному, линейному и другим методам криптоанализа.

Для специалистов в области безопасности информации, криптографии, прикладной математики, информатики и электроники, а также для преподавателей, студентов и аспирантов инженерно-технических вузов.

УДК 681.3
ББК 32.81

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Компьютерная верстка	*Сергея Матвеева*
Корректор	*Евгений Камский*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.04.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 36,12.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-524-6

© Молдовян Н.А., Молдовян А.А., Еремеев М.А., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Введение

В последние годы интерес к криптографии пробудился у широких кругов специалистов, работающих в самых разнообразных областях информационных технологий и их приложений. Особый интерес к этому предмету проявляют специалисты, имеющие отношение к разработке систем и средств защиты информации, а также практическому использованию последних. Откликом на такую потребность явилось появление на книжном рынке большого числа отечественных и переводных книг, посвященных криптографии и ее приложениям. Несмотря на это, многие современные направления в области шифрования остаются слабо освещенными. Одним из таких направлений прикладной криптографии является имеющий большое практическое значение новый подход к построению скоростных шифров, который заключается в использовании так называемых управляемых операций в качестве криптографических примитивов (концепция управляемых преобразований). Впервые в цельном виде это направление прикладной криптографии было представлено в книге «Криптография: скоростные шифры» [14], изданной в 2002 г. Первый тираж этой книги быстро разошелся, свидетельствуя, что даже эта относительно узкая область прикладной криптографии вызывает интерес достаточно широкого круга читателей. За прошедший год у авторов настоящей книги, которые проводят исследования по развитию концепции управляемых преобразований, появились новые интересные результаты. Эти результаты позволяют более полно представить возможности применения управляемых операций и дают более полное понимание их места среди других криптографических примитивов. В частности, они позволяют построить достаточно прозрачные для оценки основных свойств шифрующие преобразования, дают возможность применить новые типы обобщенных криптосхем, обладающих свойством универсальности и являются удобным учебным объектом. Значительно расширился класс управляемых операций.

Новые результаты явились основным содержанием настоящей книги. Однако авторы сочли целесообразным включить три главы, отражающие общую картину современной криптографии. Это позволит читателю более широко ознакомиться с проблематикой криптографии и ее идеями, кроме того, это отражает место концепции управляемых преобразований. Такая практика была апробирована в книге [14] и нашла понимание читателей, поскольку позволяет в рамках одной книги ознакомиться с более широким кругом задач, решаемых криптографией, и более разнообразными ее методами и идеями. Поскольку настоящая книга задумана как продолжение монографии [14], то первые три главы написаны на основе материала, дополняющего и существенно расширяющего вводную главу из [14]. Материал, изложенный в главах 1-3 из настоящей книги и главах 1 и 2 из [14], представляет собой достаточно полное

введение в современную криптографию и ее проблематику. Аналогично этому основное содержание этих книг изложено независимо, но в то же время обе книги отражают разные грани единого подхода – концепции управляемых преобразований в синтезе блочных шифров – и могут рассматриваться как два тома, посвященных одному вопросу.

Главы с четвертой по шестую посвящены вопросам построения управляемых подстановочно-перестановочных сетей (УППС), основанных на элементарных управляемых подстановках трех различных типов: 2×2 с двумя состояниями, 2×2 с четырьмя состояниями и 3×3 с двумя состояниями. Интерес к использованию таких управляемых сетей возрожден предложением и обоснованием использования управляемых перестановочных сетей для построения криптосхем, основанных на переменных перестановках, реализуемых в форме битовых перестановок, зависящих от шифруемых данных. Поскольку УППС являются более общим вариантом управляемых сетей и включают перестановочные сети как частный случай, то в общем случае можно ожидать, что переменные подстановки, выполняемые с помощью УППС, являются более эффективным криптографическим примитивом. Результаты указанных глав подтверждают это предположение. С увеличением размера управляемого элемента резко возрастает число различных вариантов их реализации и появляется больше возможностей при выборе стойких криптосхем. Однако, при этом существенно возрастает сложность аппаратной реализации, поэтому основное внимание было уделено анализу управляемых элементов, которые легко реализуются в заказных и программируемых интегральных схемах.

Седьмая глава посвящена построению переключаемых операционных блоков, основанных на УППС. Переключаемые управляемые операции представлены как новый широкий класс криптографических примитивов. Рассматриваются различные варианты топологии переключаемых УППС. Вводится понятие переключаемой управляемой операционной подстановки (УОП) и обсуждаются различные варианты их экономичной реализации. Показана перспективность использования переключаемых УОП для построения блочных шифров с простым расписанием использования ключа свободных от слабых ключей и гомогенности шифрующих преобразований.

В восьмой главе описывается ряд шифров, построенных на основе управляемых операций, включая переключаемые переменные подстановки и перестановки. Достоинством этих криптосистем являются высокие значения стойкости и скорости шифрования при относительно низкой сложности аппаратной реализации. Для ряда шифров с переменными операционными подстановками приводится дифференциальный анализ и рассматриваются некоторые другие типы криптоаналитических атак.

В девятой главе рассматриваются вопросы построения алгоритмов вычисления криптографических контрольных сумм, в том числе хэш-функций, которые применяются для осуществления контроля целостности данных и в системах цифровой электронной подписи. Данные построения также основаны на применении новых примитивов – операционных подстановок, зависящих от преобразуемых данных.

Десятая глава представляет собой криптографический практикум, который включает методический материал к практическим занятиям и список задач по различным разделам криптографии. В целом весь материал настоящей книги и моно-

графии [14] очень подходит для формирования заданий по курсовому проектированию по криптографии. На его основе можно предложить достаточное число вариантов курсовых заданий, относящихся к проектированию, тестированию и выполнению дифференциального криптоанализа блочных шифров, построенных на основе новых криптографических примитивов – операций преобразования, зависящих от преобразуемых данных. Варианты проведения практических занятий относятся к основным разделам двухключевой криптографии. Приведенный перечень задач включает задачи различного уровня и различной тематики, существенная доля из которых относится к управляемым операциям, что позволяет более глубоко освоить вопросы проектирования блочных шифров с их использованием. В криптографический практикум включены также вопросы патентования изобретений в области криптографии, в частности обсуждается каким образом можно запатентовать изобретенный алгоритм.

Данная книга рассчитана на широкий круг читателей: студентов, преподавателей, инженеров, аспирантов, научных работников и профессионалов в области защиты информации и криптографии. Авторы надеются, что данная книга позволит каждому читателю найти для себя новые и интересные вопросы, относящиеся к современной криптографии, которые представлены в доступной форме изложения и с ориентацией на практическое применение.

Главы 1, 2, 3 написаны совместно Молдовяном Н.А. и Еремеевым М.А.

Главы 4, 5, 7 и Заключение написаны Молдовяном А.А.

Главы 6, 8 и Введение написаны совместно Молдовяном Н.А., Молдовяном А.А. и Еремеевым М.А.

Главы 9 и 10 написаны совместно Молдовяном Н.А. и Еремеевым М.А.

Авторы приносят искреннюю благодарность своим коллегам – Бодрову А.В., Гуцу Н.Д., Изотову Б.В., Молдовяну П.А. и Морозовой Е.В за участие в совместных работах, Матвееву С.А. за техническую помощь и В.П. Чернолесу за неоценимые консультации при патентовании новых способов криптографического преобразования. Особая благодарность директору СЦПС «Спектр» Долгиреву В.А. создавшему своим подчиненным все условия для творческой работы над книгой.

Авторы очень признательны Беляеву Е.А., Буренину Н.И., Гаскарову Д.В., Зиме В.М., Макарову В.Ф., Пальчуну Б.П., Пожарскому В.Н., Советову Б.Я., Стрельцову А.А., Хиже Г.С., Чижову В.А., Юсупову Р.М. и другим ученым и специалистам, существенно содействовавшим получению новых результатов в затронутом направлении исследований.

Глоссарий

Алгебраическая степень нелинейности — количество сомножителей в крайнем правом элементе алгебраической нормальной формы булевой функции.

Алгоритм защитного контрольного суммирования — алгоритм вычисления некоторого двоичного вектора сравнительно малого размера, зависящего от каждого бита сообщения произвольной длины. Важнейшим типом алгоритмов защитного контрольного суммирования являются хэш-функции.

Аппаратный шифр — шифр, ориентированный на реализацию в виде электронного устройства.

Аутентификация — процедура установления подлинности пользователя (абонента сети, отправителя сообщения), программы, устройства или данных (информации, получаемого сообщения, ключа). Частным вариантом аутентификации является установление принадлежности сообщения конкретному автору.

Блочный шифр — шифр, осуществляющий преобразование блоков данных фиксированного размера.

Вероятностное шифрование — процесс шифрования с использованием случайных параметров.

Вес двоичного вектора — число ненулевых битов двоичного вектора.

Вес Хемминга — вес двоичного вектора.

Гибкий шифр — шифр, описываемый как набор криптоалгоритмов, выбираемых в зависимости от секретного ключа.

Гибридная криптосистема — криптосистема, в которой распределение ключей осуществляется с помощью двухключевых криптоалгоритмов, а процесс шифрования информации – с помощью одноключевых. Гибридные криптосистемы сочетают в себе удобство распределения секретных ключей и высокую скорость шифрования.

Двухключевая криптография — направление исследований в области криптографии и разработки криптосистем, основанных на использовании двух ключей – открытого и закрытого. Открытый (публичный, общедоступный) ключ предполагается известным всем пользователям криптосистемы и потенциальному противнику. Закрытый ключ предполагается известным только одному пользователю.

Дифференциальный криптоанализ — метод анализа шифров, заключающийся в поиске наиболее вероятных разностей, получаемых на выходе шифра при заданной входной разности.

Доверительный центр — организация, осуществляющая регистрацию, хранение и распространение открытых ключей в двухключевых криптосистемах. Основным назначением доверительного центра является аутентификация открытых ключей пользователей. Для распространения открытых ключей используются 1) электронные справочники открытых ключей и 2) цифровые сертификаты. Справочники и сертификаты подписываются доверительным центром.

Зашифрование — процесс преобразования множества открытых сообщений во множество шифрованных сообщений с использованием ключа в целях защиты от несанкционированного доступа. Стойкость зашифрования (или уровень защищенности) определяется тем, что при зашифровании используется один или несколько секретных параметров, которые предполагаются известными только законным (легальным) пользователям.

Имитозащита — способ защиты от навязывания ложных сообщений путем передачи вместе с сообщением небольшой дополнительной информации, называемой имитовставкой.

Имитовставка — криптографическая контрольная сумма, зависящая от каждого бита сообщения и вычисляемая с использованием криптографических алгоритмов (например, алгоритмов шифрования) и ключа.

Инволюция — операция преобразования (отображение некоторого множества на себя), являющаяся обратной самой себе.

Ключ — параметр шифра, определяющий выбор конкретного варианта преобразования для зашифрования или расшифрования из множества преобразований, составляющих шифр.

Конфиденциальность — свойство информации, характеризующее ее защищенность от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней.

Корреляционная иммунность порядка k — свойство булевой функции, заключающееся в сбалансированности всех частных функций, полученных из исходной функции фиксированием любых ее k или менее переменных. Данное свойство позволяет обеспечить стойкость криптографических преобразований к статистическим атакам при фиксированных значениях битов на входе преобразования. Максимальный порядок корреляционной иммунности булевой функции от n переменных не превышает значения n–1.

Криптоанализ — процесс (алгоритм) получения исходного текста по зашифрованному без знания ключа или процесс вычисления ключа по исходному и шифрованному тексту. Криптоанализ выполняется противником с целью получения возможности осуществления несанкционированного доступа или разработчиком с целью оценивания стойкости шифра.

Криптографическое преобразование — процедура специального преобразования информации для решения одной из следующих криптографических задач: шифрования данных, формирования цифровой электронной подписи, вычисления специальных криптографических контрольных сумм и имитовставки.

Криптографический примитив — в широком смысле это операция или процедура, используемая в качестве элемента шифра, в узком смысле это операции и проце-

дуры, определяющие требуемые свойства криптосистемы (стойкость, возможность зашифрования и расшифрования с использованием различных ключей и т. п.).

Криптографический протокол — протокол, предусматривающий взаимодействие двух и более сторон с использованием криптографических алгоритмов.

Криптосистема — система обеспечения безопасности защищенной сети, использующая криптографические средства. В качестве подсистем может включать системы шифрования, идентификации, цифровой подписи и др., а также систему распределения ключей.

Криптостойкость (*стойкость шифра*) — способность криптосистемы противостоять атакам противника, направленным на получение ключа, открытого сообщения или навязывание ложного сообщения. Количественно выражается числом операций некоторого типа, которые необходимо выполнить для решения задачи криптоанализа. При этом указывается тип атаки на криптосистему, т. е. исходные данные для криптоанализа. Если исходные данные не указываются, то подразумевается стойкость к лучшему известному алгоритму криптоанализа (для атаки на основе специально подобранных текстов).

Лавинный эффект — свойство распространения влияния одного входного бита на многие выходные биты (свойство размножения ошибок).

Линейный криптоанализ — метод анализа шифров, заключающийся в поиске наилучших линейных аппроксимаций для отображений, выполняемых шифром.

Лобовая атака (силовая атака) — криптоанализ путем исчерпывающего перебора всех возможных ключей, т.е. методом подбора ключа. Для криптосистем с конечным ключом этот метод является универсальным, т. е. он применим к любому шифру такого типа. Однако, вероятность раскрытия ключа с помощью метода опробования всех возможных ключей является весьма низкой. Для задания низкой вероятности успеха такой атаки для симметричных криптосистем требуется использовать ключи размером не менее 80 бит. В настоящее время считается, что гарантированную стойкость к лобовому нападению обеспечивают равновероятно генерируемые случайные ключи размером 128 бит.

Модификация управляемой операции — преобразование, осуществляемое управляемой операцией при заданном фиксированном значении управляющего вектора.

Нелинейность — свойство булевой функции, показывающее степень удаленности булевой функции от множества аффинных или линейных булевых функций. Определяется значением минимального расстояния Хэмминга между заданной функцией и множеством аффинных функций. Характеризует стойкость к линейному методу криптоанализа.

Операционная подстановка — преобразование, осуществляемое с помощью операционного блока, представляющего собой подстановочно-перестановочную сеть.

Переключаемая операция — операция, выполняемая в зависимости от дополнительного однобитового управляющего вектора e, причем модификации, соответствующие значениям $e = 0$ и $e = 1$, являются взаимно-обратными. Частным случаем переключаемых операций являются переключаемые управляемые операции.

Перестановочная сеть — управляемая перестановочная сеть, состоящая из некоторого числа активных каскадов (активных слоев), между которыми расположены узлы фиксированной коммутации (фиксированные перестановки), причем каждый активный каскад состоит из совокупности управляемых элементарных переключателей, управляемых некоторым управляющим битом и выполняющих тождественное преобразование или перестановку двух битов в зависимости от значения управляющего бита.

Подстановочно-перестановочная сеть — узел преобразования, состоящий из некоторого числа каскадов (слоев), между которыми расположены узлы фиксированной коммутации (перестановки), причем каждый каскад состоит из совокупности блоков подстановки (обычно одного размера).

Управляемая подстановочно-перестановочная сеть — управляемая подстановочно-перестановочная сеть, состоящая из некоторого числа активных каскадов (слоев управляемых подстановок), между которыми расположены узлы фиксированной коммутации (фиксированные перестановки), причем каждый активный каскад состоит из совокупности управляемых блоков подстановок, управляемых некоторым управляющим двоичным вектором (или битом), задающим выбор текущей реализуемой подстановки.

Порядок управляемой перестановки — максимальное количество входных битов, которые могут быть переставлены в произвольные разряды выходного двоичного вектора при некотором значении управляющего вектора.

Поточный шифр — шифр, преобразующий последовательно отдельные биты или знаки исходного сообщения.

Преобразование Уолша-Адамара булевой функции — разновидность дискретного преобразования Фурье, заключающееся в линейном преобразовании векторного n-мерного пространства над двоичным полем в значение действительного числа и имеющее следующий вид: $U_{\alpha}\left(f(X)\right) = \sum_{X \in \mathrm{GF}(2)^n} f(X)(-1)^{\langle \alpha, X \rangle}$, где знак "$\langle\ \rangle$" обозначает скалярное произведение двух векторов.

Противник — субъект (оснащенная группа специалистов или физическое лицо), пытающийся преобразовать шифртекст в открытый текст без знания ключа или вычислить ключ по известным исходным и шифрованным текстам. Синонимами являются термины – атакующий, нападающий, криптоаналитик, криптоаналитик противника.

Предвычисления — процедуры преобразований или вычисления, выполняемые предварительно и используемые в дальнейшем при многократном выполнении некоторого алгоритма.

Программный шифр — шифр, ориентированный на реализацию в виде программы.

Простое расписание использования ключа — расписание использования ключа, при котором подключи, входящие в шифрующие преобразования, представляют собой непрерывные битовые цепочки, являющиеся частью секретного ключа. Представляет собой вариант отказа от сложных процедур преобразования сек-

ретного ключа. Применяется с целью уменьшения сложности схемотехнической реализации шифров и увеличения производительности криптосистем.

Протокол рукопожатия — протокол, позволяющий двум удаленным сторонам, владеющим некоторым общим секретом, осуществить взаимную проверку подлинности без раскрытия секрета.

Протокол тайного электронного голосования — криптографический протокол, обеспечивающий тайну голосования для каждого голосующего при возможности проверки со стороны последнего, как учтен его голос («за» или «против»). При этом никто, включая избирательный комитет, не сможет установить, как проголосовал избиратель.

Расшифрование — процесс преобразования множества шифрованных сообщений во множество открытых сообщений с использованием ключа. (Отметим, что вместо термина «расшифрование» иногда используется термин «дешифрование». В российской специальной литературе под термином «дешифрование» обычно понимают процесс восстановления открытого текста без знания ключа или вычисление ключа по открытым текстам и шифртекстам.)

Раскрытие шифра (криптосистемы, криптоалгоритма) — нахождение способа решения задачи криптоанализа за разумное время при использовании современных вычислительных средств. В качестве синонима используется термин «взлом шифра».

Регулярное отображение — такое отображение $Y=\varphi(X):GF(2)^n \to GF(2)^m$ при $n \geq m$, при котором Y ровно 2^{n-m} раза принимает все 2^m различных значений из $GF(2)^m$, в то время как X проходит 2^n значений из $GF(2)^n$. При $n=m$ каждое регулярное отображение является биективным. Необходимым условием регулярности отображения при $n \geq m$ является сбалансированность всех линейных комбинаций булевых функций, выполняющих данное отображение.

Режим шифрования — вариант осуществления криптографического преобразования информации (зашифрование и расшифрование) или вариант использования шифра. Например, блочные шифры могут быть использованы в режиме электронной кодовой книги, режиме сцепления блоков шифра, режиме выработки ключевой гаммы.

Резилентная (совершенная) булева функция — это корреляционно-иммунная степени k булева функция, которая является сбалансированной функцией.

Сбалансированная булева функция — булева функция, таблица истинности которой содержит одинаковое количество нулей и единиц. Данное понятие эквивалентно условию, что значение преобразования Уолша-Адамара на нулевом векторе равно нулю.

Симметричная криптосистема — криптосистема с секретным ключом. Симметричность означает, что ключи, используемые при зашифровании и в процессе расшифрования одинаковы, или могут быть получены один из другого с небольшой трудоемкостью.

Слепая подпись — протокол формирования цифровой электронной подписи, позволяющий сформировать правильную ЭЦП, соответствующую некоторому цифро-

вому сообщению, которое подписывающая сторона получает в зашифрованном (т. е. недоступном для нее) виде.

Совершенно нелинейная булева функция (*бент-функция*) — это булева функция от n переменных, для которой расстояние Хэмминга до множества всех аффинных функций над векторным n-мерным пространством максимально и имеет постоянную величину.

Статистические тесты — экспериментальные тесты, включающие статистическую обработку двоичных последовательностей или множеств двоичных векторов с целью проверки гипотез о свойствах последовательностей или алгоритмов, формирующих двоичные последовательности и вектора. Применяются для оценки качества алгоритмов шифрования или формирования ключей.

Стеганография — способ скрытной передачи информации или способ формирования скрытного канала. Часто используется для передачи шифртекстов. Широкое применение компьютерных технологий дало новый сильный импульс в развитии стеганографических методов. В настоящее время наблюдается тенденция использования криптографических идей в современной стеганографии, например, секретных ключей. Актуальным направлением стеганографии является создание цифровых водяных знаков.

Удостоверяющий центр — доверительный центр.

Управление ключами — совокупность мероприятий и процедур, обеспечивающих защищенную генерацию, распределение, хранение и уничтожение ключей.

Управляемые операции — операции, описываемые как множество некоторых более простых операций, выбираемых в зависимости от значения некоторого управляющего кода.

Управляемые инволюции — управляемые операции, каждая модификация которых является инволюцией.

Управляемые перестановочные инволюции — управляемые перестановки, каждая модификация которых является инволюцией.

Управляемый элемент — элементарный управляемый блок (узел) управляемой подстановочно-перестановочной или перестановочной сети; обозначается как блок $\mathbf{F}_{t/w}$, реализующий преобразование t-битовых двоичных векторов в зависимости от w-битового управляющего вектора; используется в качестве типового элемента при построении управляемых подстановочно-перестановочных сетей. Типичными значениями параметров w и t являются $w = 1, 2$ и $t = 2, 3, 4$.

Управляющий вектор — двоичный вектор, задающий выбор текущей модификации управляемой операции. Синонимом является термин «управляющий код».

Уравнение вычисления подписи — уравнение, задающее определенное соотношение между секретным ключом, хэш-функцией документа и цифровой подписью. Служит для формирования цифровой электронной подписи, подлинность которой может быть проверена с использованием открытого ключа.

Уравнение проверки подписи — уравнение, задающее определенное соотношение между открытым ключом, хэш-функцией документа и цифровой подписью. Служит для проверки подлинности подписи.

Хэширование — процесс вычисления значения хэш-функции.

Хэш-функция — криптографическая функция (процедура, алгоритм), аргументом которой являются произвольные сообщения (тексты, документы), представленные в виде последовательности битов, и значения которой лежат в области от 0 до $2^m - 1$, где m – размер хэш-кода (выходного значения хэш-функции). Хэш-функции представляют собой специальный класс криптографических контрольных сумм, вычисляемых обычно без использования секретных параметров и обеспечивающих сложную зависимость выходного значения от каждого бита входного сообщения.

Целостность информации — характеристика соответствия информации ее эталонному состоянию. Нарушение целостности информации есть ее несанкционированное модифицирование (умышленное или неумышленное).

Цифровой сертификат — электронный документ, содержащий информацию о владельце сертификата (ФИО, должность, организация, адрес, срок действия, открытый ключ и др.) и подписанный доверительным центром.

Цифровая подпись — электронная цифровая подпись.

Шифр с открытым ключом — двухключевая криптосистема. Особенностью данных криптосхем является наличие связанных между собой ключей для зашифрования и расшифрования, каждый из которых не может быть получен из другого с приемлемыми вычислительными и временными затратами. Синонимами являются термины «криптосистема с открытым ключом», «асимметричная криптосистема».

Шифр — семейство обратимых преобразований множества открытых сообщений в множество шифрованных сообщений и обратно, каждое из которых определяется некоторым параметром, называемым ключом.

Шифрование — процесс преобразования информации с использованием некоторой дополнительной информации, управляющей этим процессом и называемой ключом. Реализуется в виде процедуры расшифрования или зашифрования. Часто под шифрованием понимается зашифрование.

Шифратор — электронное устройство или программа, реализующая алгоритмы шифрования.

Электронная цифровая подпись — некоторая дополнительная информация, соответствующая данному электронному документу (сообщению), которая могла быть сформирована только владельцем некоторого секрета — закрытого ключа и которая позволяет с использованием специального алгоритма установить факт соответствия подписи закрытому ключу подписывающего. Под электронной цифровой подписью (ЭЦП) понимается также криптографическая система (совокупность алгоритмов и правил), позволяющая подписывать цифровые сообщения и проверять правильность формируемых цифровых подписей.

Обозначения

$\mathbf{Z}$ – множество целых чисел;

$E_Q(M)$ – функция преобразования сообщения M в зависимости от ключа Q;

$GF(q)$ – конечное поле или поле Галуа, $q = p^m$, где p – простое число, называемое характеристикой конечного поля, m – натуральное число, называемое степенью поля, q – порядок поля (количество элементов поля);

$GF(2)$ – конечное двоичное поле;

$GF(2)^n$ – n–мерное векторное пространство над двоичным полем;

$\mathbf{F}, \mathbf{F}^{-1}$ – прямая и обратная управляемые операции;

$\mathbf{F}_1 \bullet \mathbf{F}_2$ – суперпозиция операций $\mathbf{F}_1$ и $\mathbf{F}_2$;

$\oplus$ – операция поразрядного суммирования по модулю два;

$>>>k$ – операция циклического сдвига на k бит;

$\|$ – операция конкатенации двух двоичных векторов;

$a \leftarrow b$ – операция присваивания значения b переменной a;

$\#\, M$ *или* $\#\, \{M\}$ – мощность множества M;

$НОД(a, b)$ – наибольший общий делитель чисел a и b;

$\mathbf{Z}_p = \mathbf{Z}/p$ – множество классов вычетов целых чисел по модулю p (кольцо вычетов по модулю p);

$\phi(n)$ – функция Эйлера;

$GF(2)^n \rightarrow GF(2)^m$ – отображение пространства $GF(2)^n$ n-мерных векторов над полем $GF(2)$ в другое пространство $GF(2)^m$ m-мерных двоичных векторов;

$\langle a, b \rangle$ – скалярное произведение векторов a и b;

$S(f)$ – таблица истинности булевой функции $f : GF(2)^n \rightarrow GF(2)$;

$\boldsymbol{wt}(\alpha)$ – вес Хэмминга двоичного вектора α (количество единиц);

$d(f, g)$ – расстояние Хемминга между двумя двоичными векторами f и g (таблицами истинности булевых функций)

$r_\beta(f)$ – значение автокорреляционной функции булевой функции f относительно двоичного вектора β;

$U_\alpha(f)$ – значение преобразования Уолша-Адамара булевой функции f относительно вектора α;

$\lfloor n \rfloor$ – наибольшее целое число, меньшее либо равное значению аргумента n;

$\lceil n \rceil$ – наименьшее целое число, большее либо равное значению аргумента n;

deg(f) – алгебраическая степень нелинейности булевой функции (наибольшее количество сомножителей в алгебраической нормальной форме булевой функции);

NL(f) – нелинейность булевой функции f (значение минимального расстояния Хэмминга между функцией f и множеством всех аффинных функций);

$\mathbf{P}_{2/1}$ – элементарный блок управляемых перестановок, выполняющий или не выполняющий перестановки двух входных битов в зависимости от одного управляющего бита;

$\mathbf{P}_{m/n}(X, V)$ – блок управляемых перестановок, осуществляющий перестановку битов в n-разрядном входном векторе $X \in \mathrm{GF}(2)^n$ в зависимости от m-разрядного управляющего вектора $V \in \mathrm{GF}(2)^m$;

$\mathbf{F}^{(v)}$ – конкретная модификация управляемой операции, соответствующая управляющему вектору v;

$\mathbf{F}_{2/1}$ – элементарный управляемый элемент, выполняющий подстановку над двумя входными битами в зависимости от одного управляющего бита;

$\mathbf{F}_{n/m}(X, V)$ – управляемая подстановочно-перестановочная сеть (управляемая операционная подстановка), осуществляющая подстановочное преобразование n-разрядного входного вектора $X \in \mathrm{GF}(2)^n$ в зависимости от m-разрядного управляющего вектора $V \in \mathrm{GF}(2)^m$.

Сокращения

АКФ – автокорреляционная функция;
АНФ – алгебраическая нормальная форма;
АФРК – алгоритм формирования расширенного ключа;
БУОП – блок управляемой операционной подстановки;
БУП – блок управляемых перестановок;
БФ – булева функция;
ВБФ – векторная булева функция;
ВКФ – взаимная корреляционная функция;
ЗКС – защитная контрольная сумма;
ИА – итеративная архитектура;
ИОК – инфраструктура открытых ключей;
КА – конвейерная архитектура;
КВК – коэффициент взаимной корреляции;
КИ – корреляционная иммунность;
КР – критерий распространения;
КСЛЭ – критерий строгого лавинного эффекта;
ЛХ – линейная характеристика;
НОД – наибольший общий делитель;
ОПУА – обратное преобразование Уолша-Адамара;
ПЛИС – программируемая логическая интегральная схема;
ППС – подстановочно-перестановочная сеть;
ПРИК – простое расписание использования ключа;
ПРК – простое расписание ключа;
ПУА – преобразование Уолша-Адамара;
ПУО – переключаемая управляемая операция;
ПУОП – переключаемая управляемая операционная подстановка;
ПУСК – процедура усложнения секретного ключа;

СБИС – сверхбольшая интегральная схема;
СОС – список отозванных сертификатов;
Спектр УА – спектр Уолша-Адамара;
УОБ – управляемый операционный блок;
УОП – управляемая операционная подстановка;
УП – управляемая перестановка;
УППС – управляемая подстановочно-перестановочная сеть;
УПС – управляемая перестановочная сеть;
УЦ – удостоверяющий центр;
УЭ – управляемый элемент;
ЦР – центр регистрации;
ЦС – центр сертификации;
ЭВМ – электронная вычислительная машина;
ЭК – эллиптическая кривая;
ЭЦП – электронная цифровая подпись.

ГЛАВА 1

Вопросы одноключевой криптографии

1.1. Варианты реализации шифров

В настоящее время алгоритмы шифрования, представляющие практический интерес, являются для ручного шифрования достаточно сложными. Также в историю вошли алгоритмы, лежавшие в основе механических и электромеханических шифраторов. Современные алгоритмы шифрования обычно реализуются в виде некоторого электронного устройства, основной частью которого является крипточип – интегральная схема, реализующая алгоритм шифрования, и в виде программ для ЭВМ. Алгоритмы, разрабатываемые как стандарты для широкого использования, должны обеспечивать высокую производительность как при программной реализации, так и при недорогой аппаратной реализации. Однако для многих технологических применений предполагается использование либо устройств шифрования, либо программ шифрования. Очевидно, что в таких случаях разумно ориентироваться только на один из указанных вариантов реализации. Это позволит повысить производительность шифрующих программ, а при аппаратной реализации — существенно уменьшить стоимость криптографических устройств. Способы построения аппаратно-ориентированных шифров рассмотрены в книге [14].

К программным шифрам относятся криптосистемы, которые обеспечивают высокую производительность шифрования данных при их реализации в виде компьютерных программ. Система команд микропроцессоров является достаточно ограниченной, однако она позволяет реализовать достаточно производительные алгоритмы шифрования. В значительной степени это связано с тем, что в системе команд всегда присутствует команда пересылки (чтения) содержимого некоторой ячейки оперативной памяти в один из регистров процессора. Данная операция, реализующая выборку из некоторого массива в памяти, представляет собой не что иное, как нахождение значения некоторой функции, заданной табличным способом, по некоторому значению аргумента, заданному как адрес ячейки памяти. Табличным способом могут быть заданы произвольные функции, в том числе и операции подстановок, которые являются базовым криптографическим примитивом во многих современных криптосистемах.

В конечном счете, любой блочный шифр представляет собой чрезвычайно большое множество подстановок большого размера (число возможных входных значений 2^{64} для 64-битового шифра или 2^{128} для 128-битового), выбираемых в зависимости от секретного ключа. Однако такое непосредственное задание шифра практически нереализуемо, поскольку требует неимоверно большого объема памяти. Но такие под-

становки можно генерировать. Соответствующий генератор и представляет собой алгоритм шифрования. Если бы мы могли выбирать непосредственным образом некоторую секретную подстановку, то в принципе мы могли бы выбрать ее случайным образом. При использовании генератора формируемые подстановки не являются случайными, даже если мы выберем случайный секретный ключ, который задает выбор конкретной подстановки. Проблема разработки алгоритмов шифрования – это проблема задания такого множества подстановок, каждая из которых являлась бы псевдослучайной (практически неотличимой от случайной). Иными словами, разработка стойкого шифра связана с построением генератора псевдослучайных подстановок, управляемого секретным ключом.

Что касается подстановок малого размера (4×4, 8×8 и даже 16×16), то они легко реализуются программным и аппаратным способом и используются как базовые операции при проектировании шифров. При этом учитывается, что при увеличении размера подстановки резко возрастает ресурс, необходимый для их реализации. Наиболее эффективными для аппаратной реализации представляются подстановки размера 4×4, а для программной – 4×4 и 8×8. Подстановки размера 8×8 также имеют приемлемую сложность схемотехнической реализации. Именно эти варианты подстановок и представляют собой криптографический примитив, который позволяет сочетать эффективность программной и аппаратной реализации разрабатываемого шифра. Другой операцией, органически дополняющей операцию подстановки, является перестановка битов преобразуемых данных. Произвольная перестановка аппаратным путем реализуется практически без затрат ресурсов, а при программной реализации с использованием современных процессоров широкого назначения она вносит существенную задержку в процесс шифрования. Этот недостаток может быть полностью устранен путем расширения системы команд универсального процессора командой управляемой битовой перестановки, что и было предложено в работах [3, 20].

Выбор размера и конкретных таблиц подстановок при построении шифров является одной из главных задач, которые должны быть решены, но сама возможность эффективного осуществления операций подстановок при программной реализации обеспечивается стандартной системой элементарных команд процессора. Вместо перестановок произвольного типа в программных шифрах используются частные виды перестановок, например, операция циклического сдвига. Наряду с базовой операцией подстановки могут быть использованы другие элементарные команды процессора:

- операции циклического сдвига на фиксированное число двоичных разрядов;
- операции циклического сдвига на фиксированное число двоичных разрядов, зависящее от ключа;
- операции циклического сдвига на фиксированное число двоичных разрядов, зависящее от преобразуемых данных;
- поразрядное суммирование по модулю два;
- суммирование и/или вычитание по модулю 2^{32};
- операции поразрядного логического умножения и/или сложения двух n-битовых двоичных векторов и др.

В целом разработка программных шифров связана с учетом специфики обработки данных в компьютерных системах, что позволяет получить высокие скорости шифрования при использовании микропроцессоров широкого применения. Практическая потребность решения проблемы защиты электронной информации в массовом масштабе обусловливает актуальность разработки программных шифров и перспективы их широкого применения.

При выборе операций подстановок можно использовать следующие возможности, связанные с программной реализацией:

- можно использовать таблицы подстановок, зависящие от секретного ключа (в этом случае таблицы подстановок формируются по ключу на этапе предвычислений, выполняемых при инициализации криптосистемы);
- можно использовать таблицы подстановок достаточно большого размера;
- можно использовать табличные подстановки, зависящие от преобразуемых данных (данный тип подстановок реализуется с помощью некоторого множества пронумерованных таблиц подстановок).

Операции подстановок являются только частным случаем табличного задания функций отображения. Подстановки являются биективными отображениями, т. е. позволяют однозначно выполнить обратное преобразование. Это свойство необходимо для многих схем построения алгоритмов шифрования. Но оно не является необходимым условием для обеспечения возможности осуществления расшифрования закрытого сообщения. Например, в криптосхеме Фейстеля при построении раундовой функции могут быть использованы произвольные операции преобразования. Таким образом, представляют интерес не только подстановки, но и операции отображения более общего типа. Последние также могут быть эффективно реализованы программными средствами. Вместо операций подстановок или дополнительно к ним в программных шифрах можно использовать операции табличного отображения (фиксированные, зависящие от ключа и/или от преобразуемых данных). Одним из важных вариантов реализации операций табличного отображения является механизм выборки подключей в зависимости от преобразуемых данных, который успешно использован при разработке ряда скоростных программных шифров [93-96].

С программной реализацией криптосистем связана возможность применения достаточно сложных процедур предвычислений, реализуемых как этап инициализации криптосистемы, осуществляемый после ввода секретного ключа. В частности, инициализация может включать процедуру настройки алгоритма шифрования по секретному ключу [40]. При аппаратной реализации использование сложных предвычислений приводит к значительному повышению стоимости устройств шифрования и снижению их производительности при частой смене ключей. Шифры, в которых алгоритм шифрования формируется в зависимости от секретного ключа, называются гибкими или недетерминированными. В гибких шифрах каждому ключу соответствует уникальная модификация алгоритма шифрования. Поскольку множество ключей ограничено, то это означает, что гибкий шифр представляет собой множест-

во алгоритмов шифрования, описываемых с помощью некоторого алгоритма, который задает правило формирования алгоритма шифрования в зависимости от секретного ключа. Формирование секретных таблиц подстановок, таблиц операции отображения и настройка алгоритма шифрования предполагают использование этапа настройки шифра, выполняемой однократно при введении секретного ключа. После настройки криптосистема многократно выполняет процедуры шифрования и расшифрования данных. При сравнительно редкой смене ключей (например, ключ сменяется каждые 10 секунд) наличие этапа настройки практически не изменяет среднюю скорость шифрования.

В случае ограничения длины k секретного ключа (например, $k < 40$ бит) возможность значительного усложнения этапа предвычислений, используемого в программных шифрах, может быть использована для повышения стоимости (сложности) раскрытия ключа. Это позволит уменьшить вероятность раскрытия ключа со стороны большого числа потенциальных нарушителей. Еще более надежную страховку может обеспечить использование алгоритмов, настраиваемых по некоторому дополнительному параметру, который должен быть известен узкому кругу пользователей и иметь достаточный размер r (например, $r = 80$ бит). В этом случае для внешнего нарушителя переборная сложность задачи раскрытия криптосистемы может быть оценена как 2^{k+r} шифрований, что существенно превышает сложность задачи криптоанализа (2^k шифрований) для внутренних пользователей.

1.2. Повышение стойкости шифрования при ограничении длины секретного ключа

В случае коротких ключей наиболее результативным способом нападения на шифр может стать силовая атака, которая заключается в переборе всех возможных вариантов секретного ключа (или пароля, если последний используется в качестве ключа). Наличие этапа предвычислений служит определенным барьером против силовой атаки. Наличие предвычислений, которые необходимо выполнить для каждого испытываемого варианта ключа, существенно затрудняет такую атаку. Время выполнения процедур настройки t может быть задано достаточно большим путем усложнения алгоритма настройки или путем многократного его использования. Трудоемкость силовой атаки можно оценить по формуле:

$$W \approx 2^k W_t/2,$$

где W_t – трудоемкость алгоритма предвычислений, k – длина секретного ключа в битах (предполагается, что ключ является случайным и равновероятным по множеству всех ключей длины k). Мы полагаем, что атакующий имеет некоторый известный исходный текст и соответствующий ему шифртекст, причем процедура его зашифрования имеет сложность намного меньше значения W_t (т. е. время шифрования известного текста пренебрежимо по сравнению с t и составляет, например, одну миллисекунду). В случае пароля, состоящего из букв естественного языка или выбираемого из словаря, эта формула также может быть применена, если в качестве значения k использовать некоторую эффективную длину $k_e < k$.

Требуемое время для выполнения силовой атаки можно оценить по формуле:

$$T \approx 2^{k-1}t,$$

где t – время, затрачиваемое на выполнение алгоритма предвычислений. Для многих приложений значение t равное от 0.5 до 1 секунды, является вполне приемлемым, поскольку для законного пользователя эта задержка относится только к однократно выполняемой инициализации криптосистемы. Однако, для нарушителя при длине ключа, составляющей $k = 32$ бит, в среднем время силового взлома составит более 50 лет работы однопроцессорной ЭВМ широкого применения (для которой $t = 0.5$ с). Очевидно, что это делает стоимость раскрытия ключа для многих потенциальных нарушителей (например, хакеров) неприемлемой и заставит их отказаться от осуществления атаки. Для проведения атаки за разумное время потребуется использование очень большого числа ЭВМ. Например, для того чтобы раскрыть один ключ за один месяц, потребуется непрерывная работа более 500 компьютеров широкого применения либо применение специализированных многопроцессорных ЭВМ, которые есть в наличии только у весьма ограниченного числа организаций и являются весьма дорогостоящими, равно как и их эксплуатация.

Аналогичные оценки для секретных ключей, имеющих длину от 8 до 10 байт (т.е. от 64 до 80 бит), показывают, что силовой взлом шифров с предвычислениями легко сделать практически неосуществимым даже для нарушителя, обладающего очень мощными вычислительными ресурсами. Несомненно, противодействие силовой атаке путем увеличения длины ключа является наиболее эффективным общим приемом для всех криптосистем, допускающих произвольный выбор длины секретного ключа, однако при ограничении его длины этот прием не может быть использован в полной мере.

С пользовательской точки зрения, проблема выбора пароля (секретного ключа) и повышения стойкости к атакам на основе подбора пароля заслуживает внимания, поскольку для всех широко используемых систем защиты требуется выбирать хорошие (случайные) пароли, которые трудно запомнить. В средствах защиты можно использовать активный контроллер паролей, т.е. систему, блокирующую выбор плохих паролей. Другое возможное решение проблемы совмещения удобства пользователей с высоким уровнем секретности состоит в использовании паролей-фраз, т.е. не отдельных слов, а целых фраз, которые удобны для запоминания, но трудны для подбора из-за большой длины. Целесообразно построение таких механизмов парольного доступа к ресурсам ЭВМ, которые содержат встроенный механизм противодействия атакам на основе перебора возможных паролей (или парольных фраз). Такие механизмы могут состоять, например, в использовании достаточно сложных процедур вычисления однонаправленной функции от пароля, требующих времени около 1с для процессоров широкого применения. В постоянной памяти ЭВМ будет храниться таблица значений этой однонаправленной функции от паролей всех законных пользователей (таблица образов паролей), а проверка подлинности текущего пользователя будет осуществляться как вычисление значения этой функции от значения текущего пароля и сравнение полученного значения с соответствующим значением из таблицы образов паролей.

Применение долговременных ключевых элементов при ограничении длины секретного ключа

Наиболее результативным способом повышения стойкости криптосистем в условиях применения коротких секретных ключей (например, имеющих длину $k = 32$ бит) является использование долговременных ключей. Примером криптосистемы с долговременным ключом является российский стандарт шифрования ГОСТ 28147–89 [8], в котором таблицы подстановок являются секретными. В соответствии с общепризнанным принципом Керхкоффа при оценке стойкости алгоритм шифрования предполагается известным атакующему. В более общей трактовке этот принцип можно выразить так: *все долговременные элементы механизмов защиты информации необходимо считать известными потенциальному нарушителю*. Это связано с тем, что обычно долговременные элементы известны многим участникам разработки шифра.

Идея использовать долговременные ключи большого размера или долговременные секретные элементы алгоритма шифрования для повышения стойкости является отступлением от принципа Керхкоффа, но в случае ограничения длины секретного ключа этот прием является обоснованным тем, что этот элемент усиления не имеет цель обеспечить гарантированную стойкость. Преследуемая цель состоит только в существенном повышении сложности раскрытия короткого ключа, например, со стороны внешних нарушителей. Ограниченная длина ключа предполагает, что предусматривается защита от потенциального нарушителя с весьма ограниченными вычислительными ресурсами. При наличии у нарушителя больших вычислительных ресурсов подбор ключа реализуем за приемлемое время. Использование долговременного ключа для обоих типов нарушителей потребует определения долговременного ключа, а это может быть сделано либо путем более сложного криптоанализа, либо получением этого ключа «некриптографическими» методами. Долговременный ключ может представлять собой:

- секретные константы;
- секретные таблицы подстановок (отображений);
- секретный алгоритм предвычислений (настройки);
- секретный алгоритм шифрования.

При правильном построении криптосистемы короткий ключ вычислить практически невозможно (при использовании сколь угодно больших реально существующих вычислительных ресурсов) без знания перечисленных секретных элементов (если таковые используются). Это сделает невозможной задачу раскрытия короткого секретного ключа при атаке на криптосистему, осуществляемую субъектами, не владеющими долговременным ключом.

С точки зрения общей оценки криптографической стойкости, использование долговременных ключей не приводит к повышению секретности, поскольку долговременный ключ предполагается известным фиксированному кругу пользователей,

внутри которого реально действующим является только короткий ключ. Примером систем с долговременным ключом является стандарт шифрования ГОСТ 28147–89. Используемое в нем «заполнение таблиц блока подстановки», которое «является долговременным ключевым элементом, общим для сети ЭВМ», должно рассматриваться известным атакующей стороне в некоторых возможных ситуациях. Согласно описанию этого стандарта «заполнение таблиц блока подстановки является секретным элементом и поставляется в установленном порядке». Однако роль такого секретного параметра должна быть оценена с учетом упомянутых выше замечаний.

1.3. Криптосистемы с гибким алгоритмом и требования к алгоритму предвычислений

Использование долговременных (хотя даже и сменяемых) секретных частей шифрующих систем в общем случае не приводит к существенному повышению стойкости. Однако сама идея использования не только секретных параметров, но и секретных элементов алгоритма шифрования заслуживает внимания. Если секретные элементы алгоритма сделать легко сменяемыми, то в этом случае можно говорить о гибких криптосистемах или шифрах с гибким алгоритмом. Такие конструкции могут быть легко реализованы в программных шифрах. С примерами можно ознакомиться в работах [14, 42, 94]. В гибких шифрах долговременным элементом являются процедуры настройки алгоритма шифрования, а конкретная его модификация и ключ шифрования являются сменными элементами криптосистемы, которые автоматически заменяются одновременно со сменой паролей (ключей) и являются уникальными для каждого пользователя (или каждой пары абонентов защищенной сети связи).

Хранение описания секретной модификации алгоритма шифрования приводит к определенным неудобствам для пользователей. Кроме того, при очень большом числе таких модификаций возникают проблемы их хранения. Устранение этих проблем связано с использованием генератора модификаций алгоритма шифрования. Формируется алгоритм, генерирующий конкретную модификацию по конкретному вводимому секретному параметру. Этим параметром может быть дополнительный или основной секретный ключ. В последнем случае только длина секретного ключа будет определять переборную стойкость криптосистемы. Тем не менее, секретность конкретной модификации алгоритма приводит к существенному повышению стойкости к другим типам атак, которые являются наиболее опасными. Действительно, переборную атаку легко устранить простым удлинением ключа, тогда как противодействие некоторым другим типам атак требует тщательной проработки всех элементов криптосистемы.

Процедура формирования алгоритма шифрования по секретному ключу является существенно более сложной по сравнению с непосредственным шифрованием данных. Очевидно, что настройку алгоритма шифрования разумно выполнить как часть предвычислений. Алгоритм предвычислений является частью криптосистемы, поэтому он также вносит свой вклад в задание общей секретности (стойкости) шифра. Эта часть не является столь критичной как сам алгоритм непосредственного шифрования. Кроме того, для реализации предвычислений могут быть использованы значи-

тельные вычислительные ресурсы, что упрощает задачу построения необходимых процедур предвычислений. В общем случае секретная модификация алгоритма непосредственного шифрования и некоторые сгенерированные в зависимости от секретного ключа параметры или массивы данных могут быть рассмотрены как расширенный ключ. Рассмотрим общие требования к алгоритму предвычислений. Одним из требований к алгоритму формирования ключа шифрования является следующее: *количество возможных выходных последовательностей не должно быть существенно меньше числа возможных секретных ключей*. Желательно, чтобы число возможных расширенных ключей было равно числу различных значений секретного ключа. Это требование связано с тем, что число различных расширенных ключей может быть только равным или меньшим. Действительно, длина расширенного выходного ключа превышает длину секретного ключа, но для определенных процедур предвычислений может оказаться, что различным секретным ключам будут соответствовать одинаковые расширенные ключи.

В случае применения односторонних преобразований на этапе формирования ключей шифрования значительное сужение пространства ключей шифрования является маловероятным, однако желательно получение гарантии того, что мощность множества различных ключей шифрования равна мощности множества секретных ключей длиной $l < L$, где L – длина расширенного ключа в байтах. Последнее условие достигается, если используемые процедуры предвычислений на каждом шаге преобразований задают подстановку L-байтового блока данных M, полученного как первые L байт периодического ряда, представляющего собой многократное повторение секретного ключа. В качестве составной части алгоритма предвычислений можно использовать некоторый известный блочный шифр, используемый для преобразования сообщения M в режиме сцепления блоков шифра. При этом в качестве ключа можно использовать некоторое специфицированное значение Q. Криптограмма $C = \mathrm{E}_Q(M)$ может служить в качестве расширенного ключа. При этом выполняется также требование *псевдослучайности расширенного ключа*.

Рассмотренные выше два требования к процедурам формирования расширенного ключа представляются достаточными. При желании без труда можно предложить алгоритм настройки, удовлетворяющий некоторым другим (дополнительным) требованиям. Можно принять требование вычислительной сложности определения секретного ключа по расширенному ключу и известным процедурам предвычислений. Приемлемые алгоритмы построения расширенного ключа описаны в работах [14, 41]. Смысл использования сложных процедур настройки состоит в том, чтобы заставить нападающего отказаться от их рассмотрения и принять предположение о случайности ключа шифрования.

1.4. Управляемые операции как криптографический примитив

Управляемые операции давно привлекают внимание разработчиков алгоритмов шифрования. Одной из наиболее ранних работ, посвященных проектированию криптосистем на основе управляемых операций, является статья [78], в которой рассмот-

рено использование управляемой подстановочно-перестановочной сети. Другие попытки [105, 123] были связаны с применением управляемых перестановочных сетей в качестве криптографического примитива. Однако в предложенных схемах выбор конкретной модификации реализуемой операции осуществлялся в зависимости от секретного ключа. В таком применении управляемых операций фиксируется конкретная их модификация, которая не изменяется при шифровании большого числа блоков данных. В случае управляемых битовых перестановок мы имеем некоторую фиксированную перестановку, которая является линейной операцией, хотя и неизвестной атакующему. Детальные исследования стойкости различных вариантов криптосхем на основе управляемых операций, зависящих от секретного ключа, показало, что они не могут конкурировать с другими шифрами по производительности.

Другим типом управляемых операций являются операции, зависящие от преобразуемых данных. Их особенностью является изменчивость реализуемых модификаций, что позволяет использовать термин «переменные операции». Наиболее известными алгоритмами, использующими переменные операции в качестве базового криптографического примитива, являются итеративные блочные шифры DES [89, 113], RC5 [109], RC6 [103, 110].

Первый алгоритм использует управляемые табличные подстановки размера 4×4, реализованные как блоки подстановок размера 6×4, обеспечивающие выбор одной из четырех возможных подстановок 4×4. Однако размерность векторов, которые преобразуются этими блоками подстановок, невелика, а при увеличении размера табличных подстановок существенно возрастет как сложность выбора оптимальных табличных подстановок, так и сложность их реализации. В алгоритмах RC5 и RC6 в качестве управляемых операций применяются операции циклического сдвига на число битов, выполняемые в зависимости от преобразуемых данных. Несмотря на то, что упомянутые типы переменных операций обладают сравнительно малым числом различных реализуемых модификаций, они являются эффективным криптографическим примитивом. Таким образом, переменные операции с малым числом реализуемых модификаций оказываются более эффективными по сравнению с операциями, зависящими от ключа, хотя последние и обладают очень большим числом модификаций.

Это сопоставление приводит к идее применения операций с большим числом модификаций в качестве переменных операций [92, 97]. Наиболее детально эта идея проработана по отношению к управляемым битовым перестановкам, выполняемым над подблоками данных размером 32 и 64 бит [62, 72, 75, 82, 98]. Переход к произвольным перестановкам дал возможность существенно увеличить число различных модификаций, реализуемых переменной операцией (до $n!$, где n – длина преобразуемого вектора). Для обеспечения возможности выполнения расшифровывающего преобразования шифруемый блок данных разбивают на два подблока – управляемый и преобразуемый, которые обычно имеют одинаковый размер. В этом случае число изменяющихся модификаций ограничивается размером управляющего подблока данных и равно 2^n, где n – размер последнего. Очевидно, что перестановка, зависящая от преобразуемых данных, описывается как операция подстановки частного вида, выполняемая над всем преобразуемым блоком данных и оставляющая управляющий подблок без изменения. Эффективность переменной перестановки можно объ-

яснить тем, что это подстановка, выполняемая над всем преобразуемым блоком данных. Некоторые ее слабости (линейность суммы выходов, сохранение веса Хемминга) связаны именно с тем, что эта подстановка относится к специальному типу.

В монографии [14] приведены описание и анализ ряда скоростных блочных шифров, основанных на битовых перестановках, зависящих от преобразуемых данных. Несмотря на то, что переменные перестановки являются линейным криптографическим примитивом, их комбинирование с операциями, имеющими «небольшую» нелинейность, эффективно нейтрализуют линейный криптоанализ [62, 73, 82]. Это объясняется тем, что единственной линейной комбинацией выходов переменной перестановки является сумма всех выходных битов, а при наличии некоторой дополнительной нелинейной операции трудоемкость линейной атаки становится достаточно высокой.

Другим интересным типом управляемых операций, обладающих большим числом модификаций, являются управляемые сумматоры [9, 12], представляющие собой частный случай управляемых двухместных операций [27, 74]. Данные операции также могут быть применены в качестве переменных операций. Они представляют как самостоятельный интерес, так и для комбинирования с переменными перестановками в единой криптосистеме. В *главе 4* будет дано обобщение управляемых перестановок и построен класс управляемых операционных подстановок, реализуемых на основе подстановочно-перестановочных сетей с использованием управляемых элементов минимального размера. В *главах 5* и *6* будут рассмотрены управляемые операционные подстановки, реализуемые с использованием других типов управляемых элементов. Следует отметить, что в настоящее время имеется достаточно большое число различных типов управляемых операций, обладающих большим числом реализуемых модификаций и перспективных для использования в качестве переменных операций.

Эффективность аппаратной реализации шифров на основе переменных битовых перестановок показана в работах [38, 84]. При использовании сравнительно малых аппаратных ресурсов обеспечивается высокая производительность при реализации как в заказных, так и в программируемых СБИС [119]. Вопросы построения операционных блоков управляемых перестановок (БУП), обладающих заданными свойствами, рассмотрены в работах [10, 11]. На самом деле БУП представляют собой перестановочные сети, которые ранее были широко исследованы в области параллельных вычислений и телефонии [53, 54, 102, 124], однако криптографическое их применение требует рассмотрения некоторых других свойств таких сетей, например, линейных характеристик [2].

1.5. Аппаратная реализация шифров на основе битовых перестановок, зависящих от преобразуемых данных

Представляют интерес два следующих основных варианта аппаратной реализации алгоритмов шифрования:

- в заказных сверхбольших интегральных схемах (СБИС);

- в программируемых логических матрицах – программируемых логических интегральных схемах (ПЛИС).

Первый вариант используется для серийного производства шифраторов, поскольку в этом случае он обеспечивает более низкую стоимость единицы изделия. Другим преимуществом является более высокое быстродействие изготавливаемых специализированных СБИС. Второй вариант является предпочтительным при штучном производстве устройств шифрования. Кроме того, при разработке технологии производства заказных СБИС он применяется как предварительный для изготовления и тестирования экспериментальных устройств. Достоинствами использования ПЛИС являются малый срок разработки и возможность многократного перепрограммирования под другие алгоритмы или при модифицировании алгоритмов. Другими важными направлениями использования ПЛИС являются выполнение вычислительных экспериментов большого объема и решение криптоаналитических задач.

Аппаратную реализацию итеративных шифров осуществляют в соответствии с двумя основными типами архитектур:

- итеративной;
- конвейерной.

В первом случае схемно реализуется только один раунд шифрования. Раундовая функция шифрования используется многократно для выполнения всех раундов шифрования. Дополнительные схемные компоненты осуществляют смену раундовых ключей и другие функции, необходимые для правильного осуществления вычислений, предписываемых алгоритмом. Важнейшим достоинством итеративной архитектуры является обеспечение практически одинаковой производительности в режиме электронной кодовой книги (независимое шифрование блоков данных) и в режиме сцепления блоков шифра.

Конвейерная архитектура предполагает схемную реализацию всех раундов шифрования и возможность одновременного преобразования многих блоков данных. Обычно на выходе каждого раунда устанавливается регистр для хранения промежуточного значения преобразуемого блока данных и количество одновременно преобразуемых блоков данных равно числу раундов шифрования R. Раундовые ключи хранятся в регистре ключей постоянно при выполнении шифрования сообщения. Если схема одного раунда имеет достаточно малое время задержки, то за один такт работы устройства осуществляется преобразование сразу R блоков данных, причем один из них проходит последний раунд шифрования. Таким образом, в конвейерной архитектуре за время одного такта осуществляется преобразование одного блока данных и производительность устройства оказывается примерно в R раз более высокой по сравнению с итеративной архитектурой. Недостатками конвейерной архитектуры являются существенно более высокая стоимость реализации и невозможность сохранения высокой производительности в режиме сцепления блоков шифра.

В таблице 1.1 приводятся параметры реализации следующих шифров, основанных на переменных перестановках: CIKS-1 [92] и SPECTR-H64 [72], DDP-64 (*см. гл. 4*), COBRA-H64 [99] и COBRA-H128 (*см. гл. 8*). В таблице 1.2 дается сопоставление характеристик реализации шифров SPECTR-H64 и COBRA-H128 с широко из-

вестными криптосистемами DES [79], AES и IDEA для случая использования ПЛИС. Шифр SPECTR-H64 обеспечивает более высокую скорость при меньших затратах аппаратных ресурсов по сравнению с криптосистемами AES и IDEA. Стоимость реализации SPECTR-H64 несколько превышает стоимость реализации алгоритма DES, однако, первый обеспечивает многократное превышение по производительности. Сравнение 128-битовых шифров COBRA-H128 и AES показывает, что при примерно одинаковых аппаратных затратах первый из них обеспечивает существенно более высокую производительность.

Таблица 1.1

Характеристика аппаратной реализации шифров, основанных на переменных битовых перестановках, с использованием программируемых и заказных СБИС

Шифр	Арх-ра	ПЛИС (Xilinx Vitrex)			Заказные СБИС (0.33 мкм)		
		К-во блоков CLB*	Частота МГц	Скорость шифров. Гбит/с	Площадь, sqmil**	Частота, МГц	Скорость шифров. Гбит/с
CIKS-1	Итер.	907	81	0.648	3456	93	0.744
	Конв.	6346	81	5.184	21036	95	5.824
SPECTR-H64	Итер.	713	83	0.443	3194	91	0.485
	Конв.	7021	83	5.312	32123	94	6.016
DDP-64	Итер.	615	85	0.544	2620	92	0.589
	Конв.	3440	95	6.1	14050	101	6.5
COBRA-H64	Итер.	615	82	0.525	2694	100	0.640
	Конв.	3020	85	5.5	14640	110	7.1
COBRA-H128	Итер.	2364	86	0.917	6364	90	1.00
	Конв.	12080	90	11.5	48252	95	12.1

* CLB (Configurable Logic Blocks) – конфигурируемые логические блоки, являющиеся типовыми логическими элементами ПЛИС.

** Площадь используемой поверхности кремниевого кристалла указана в единицах sqmil; 1 sqmil = 7.45 10^{-4} мм^2

Таблица 1.2

Сравнение результатов аппаратной реализации различных шифров с использованием программируемых СБИС

Шифр	Размер входа, бит	Арх-ра	ПЛИС (Xilinx Vitrex)		
			К-во блоков CLB*	Частота, Мбит/с	Скорость шифрования Гбит/с
COBRA-H128	128	Итер.	2364	86	0.917
		Конв.	12080	90	11.5

Шифр	**Размер входа, бит**	**Арх-ра**	**ПЛИС (Xilinx Vitrex)**		
			К-во блоков CLB*	Частота, Мбит/с	Скорость шифрования Гбит/с
SPECTR-H64 [119]	64	Итер.	713	83	0.443
		Конв.	7021	83	5.312
AES [117]	128	Итер.	2358	22	0.259
		Конв.	17314	28.5	3.650
IDEA [66]	128	Итер.	2878	150	0.600
DES [79]	64	Итер.	722	11	0.181

Приведенные характеристики различных вариантов реализаций шифров, основанных на переменных перестановках, показывают, что использование операций, зависящих от преобразуемых данных, обеспечивает возможность создания высокоскоростных аппаратных шифраторов, которые могут быть изготовлены с минимальным потреблением аппаратных ресурсов.

1.6. Особенности проектирования блочных шифров на основе управляемых операций

1.6.1. Управляемые операции и отображения

Любую операцию, используемую при построении блочных шифров, можно представить как отображение векторного пространства h-мерных двоичных векторов $W = (w_1, w_2, \ldots, w_h)$ в векторное пространство n-мерных двоичных векторов $Y = (y_1, y_2, \ldots, y_n)$, где для всех $j \in \{1, \ldots, h\}$ и $i \in \{1, \ldots, n\}$ имеем $w_j, y_i \in \mathrm{GF}(2)$. Такое отображение можно записать в виде $\mathrm{GF}(2)^h \to \mathrm{GF}(2)^n$. В вероятностных шифрах используются отображения, относящиеся к случаю $h < n$, в котором выходное значение зависит от некоторого случайного значения. В таких случаях по выходному значению однозначно определяется входной двоичный вектор. При $h > n$ в общем случае нельзя определить однозначно входной вектор по выходному. Операции такого типа могут применяться при построении шифров на основе криптосхемы Фейстеля, которая для произвольной раундовой функции задает корректное построение блочного шифра, т. е. обеспечивает возможность правильного расшифрования. Операции, соответствующие случаю $h = n$ и устанавливающие взаимно-однозначное соответствие между входными и выходными векторами, задают преобразование исходного векторного пространства. Такие операции задают некоторую подстановку. Подстановками иногда называют также операции, соответствующие случаю $h > n$ (см., например, подстановки типа 6×4 в алгоритме DES).

При $h > n$ отображения можно интерпретировать как управляемые операции с размером управляющего входа равным $m = h - n$. Однако часто более удобным, наглядным и полезным для анализа оказывается рассмотрение операции как управляе-

мой, хотя всегда надо иметь в виду, что наиболее общим является описание в виде указанного отображения. Мы можем конструировать управляемые операции, исходя из тех или иных соображений и механизмов, оставаясь в каком-то частном классе отображений. При малых значениях m и n отображение можно задать табличным способом, который является наиболее общим. Но при больших значениях m и n (например, $m = 64$ и $n = 32$) табличный способ оказывается неприменимым. В этом случае можно строить некоторый генератор, который будет формировать отображения такого типа и называться операцией преобразования. В некоторых частных вариантах такой генератор можно назвать управляемой операцией.

Обычно в управляемой операции можно выделить информационный вход (или просто вход) и управляющий вход. Отображаемый вектор W длины h представляется в виде конкатенации (X, V) преобразуемого вектора X длины n и управляющего вектора длины $m = h - n$. Такие управляемые операции можно назвать одноместными, поскольку на информационный вход поступает только один операнд. Операция при фиксированном V называется модификацией управляемой операции. Если при каждом фиксированном V реализуется биективное (взаимно-однозначное) отображение пространства входных n-битовых векторов в выходное пространство n–битовых векторов, то можно говорить об управляемой операции преобразования или об управляемом преобразовании. При $2n < h$ можно говорить о двухместных управляемых операциях с размером управляющего входа $m = h - 2n$, в которых на информационный $2n$-разрядный вход поступают два n-битовых вектора. Целесообразным является синтез таких управляемых операций, множество модификаций которых можно было бы отнести естественным способом к некоторому классу. Характерным примером являются управляемые битовые перестановки.

Подход к разработке блочных шифров, опирающийся на использование переменных операций, связан с использованием управляемых операций с очень большим числом возможных модификаций, когда значение m в два и более раза превышает значение n. Однако в алгоритмах шифрования предполагается разбиение преобразуемого блока данных на равные подблоки. Обычно разбиение осуществляют на два подблока. Очевидно, что один из подблоков подлежит преобразованию (ведь мы хотим его зашифровать), а другой может быть использован для формирования управляющего вектора. Простейшим вариантом такого формирования является случай использования каждого бита управляющего подблока данных для задания нескольких битов управляющего вектора. При аппаратной реализации это реализуется простым разветвлением проводников и практически не требует затрат схемотехнических ресурсов. Блок такого разветвления будем называть блоком расширения E.

1.6.2. Расписание использования ключа

Ниже при рассмотрении нескольких типовых итеративных схем синтеза блочных шифров на основе управляемых операций мы остановимся на построениях, которые обеспечивают возможность зашифрования и расшифрования с помощью одной и той же электронной схемы. Смена режима шифрования в таких криптосистемах осуществляется изменением расписания использования ключа или простым обращением очередности использования раундовых ключей. В блочных шифрах использование

секретного ключа является важной частью криптосистемы в целом. Говорят о расписании использования ключа или просто о расписании ключа (key scheduling). Под этим термином понимаются все части механизма, определяющего вхождение ключевых элементов в шифрующие процедуры. В качестве ключевых элементов могут использоваться непосредственно некоторые части секретного ключа (подключи) или некоторые псевдослучайные значения, вырабатываемые в зависимости от секретного ключа по некоторым достаточно сложным процедурам. Данные процедуры называются процедурами усложнения секретного ключа (ПУСК), а полная совокупность вырабатываемых ими ключевых элементов – расширенным ключом. Если алгоритм формирования расширенного ключа выполняется до осуществления непосредственного шифрования данных, то говорят, что расширенный ключ формируется на этапе предвычислений.

Под расширенным ключом иногда понимают и совокупность частей секретного ключа, используемых во всех раундах или на всех шагах вхождения ключевых элементов в те или иные операции преобразования. Поскольку в этих случаях не выполняется преобразование секретного ключа или его частей, то говорят о простом расписании использования ключа. Как правило, в современных блочных шифрах используется процедура усложнения ключа, хотя российский стандарт шифрования ГОСТ 28147–89 обходится без этого, показывая пример стойкого шифра с простым расписанием ключа (ПРК). Разработка шифров с ПРК представляет интерес, поскольку при аппаратной реализации в этом случае экономятся ресурсы, которые бы потребовались для воплощения алгоритма усложнения ключа. Кроме того, при частой смене ключей ПУСК может приводить к снижению производительности шифратора. Последнее можно избежать, если ПУСК выполнять параллельно с осуществлением шифрующих процедур.

Это может быть сделано следующим образом. На первом раунде шифрования используется непосредственно какая-то часть секретного ключа. Пока выполняется процедура шифрования вычисляется ключевой элемент (раундовый ключ), используемый на втором раунде. В течение времени выполнения второго раунда вычисляется раундовый ключ для третьего раунда. При этом ПУСК должна быть такой, чтобы при зашифровании раундовый ключ, используемый в i-ом раунде, формировался таким образом, что при выполнении расшифровывающих преобразований он был бы сформирован к началу выполнения $(R - i + 1)$-го раунда. Если разрабатываемый блочный шифр предполагается использовать для построения итеративных хэш-функций, то следует учитывать следующее обстоятельство. Наличие сложной ПУСК приводит к увеличению стоимости аппаратной реализации и снижению производительности хэш-функций.

В итеративных шифрах, как правило, используется достаточно большое число раундов шифрования, при этом в одном раунде используется подключ (или раундовый ключ) длины от 32 до 256 бит. Если учесть, что длина секретного ключа обычно равна от 64 до 256 бит, то возникает проблема формирования расширенного ключа, представляющего собой совокупность всех раундовых ключей, а также ключей начального и конечного преобразования, если таковые используются в данном конкретном шифре. Различные варианты построения расписания использования секретного ключа имеют свои достоинства и недостатки. Дадим краткую характеристику применяемым подходам.

Использование предвычислений для формирования расширенного ключа позволяет обеспечить сложную зависимость раундовых ключей от секретного ключа. При этом расширенный ключ представляет собой псевдослучайную последовательность. Как правило, при использовании хорошего (криптографически сильного) алгоритма расширения ключа криптоанализ осуществляется в предположении независимости раундовых ключей. Недостатком этого подхода является снижение скорости шифрования в приложениях, требующих частой смены ключей. Кроме того, при аппаратной реализации потребляются существенные дополнительные схемотехнические ресурсы (часто превосходящие ресурсы, необходимые для реализации алгоритма шифрования).

Непосредственное использование секретного ключа заключается в использовании частей (размером 32 или 64 бита) секретного ключа в качестве раундовых ключей. Примером шифров, в которых используется такой подход, является российский стандарт ГОСТ 28147–89. Недостатком такого подхода к формированию раундовых ключей является то, что раундовые ключи являются явно зависимыми, что может быть использовано при криптоанализе. Кроме того, оценка стойкости шифра, выполняемая при его проектировании существенно усложняется необходимостью учета данного обстоятельства. Недостатком представляется также наличие большого числа слабых ключей, т. е. таких ключей, для которых процедура зашифрования совпадает с процедурой расшифрования. Несмотря на то, что доля слабых ключей чрезвычайно мала, разработчики блочных криптосистем стараются не допустить их наличия. Достоинством непосредственного использования частей секретного ключа в качестве раундовых ключей является то, что обеспечивается сохранение высокой скорости шифрования в режиме частой смены ключей и не требуется использования аппаратных ресурсов для реализации алгоритма расширения ключа.

Формирование раундовых подключей в процессе шифрования блока данных. В этом подходе при аппаратной реализации в качестве первого раундового ключа используется часть секретного ключа, а при выполнении первого раунда шифрования осуществляется формирование второго раундового подключа. При выполнении второго раунда шифрования вычисляется третий раундовый ключ и т. д. Такой ход формирования раундовых ключей имеет место как при выполнении зашифрования, так и при выполнении расшифрования. Учитывая связь между очередностью использования раундовых ключей в этих двух режимах, легко увидеть важность обеспечения формирования одинаковых раундовых ключей на i-том раунде расшифрования и $(R - i + 1)$-ом раунде зашифрования. Несмотря на то, что этот подход также требует дополнительных аппаратных ресурсов, он обеспечивает высокую производительность криптосистемы при частой смене ключей, что является важным в ряде сетевых приложений.

Преобразование подключей в зависимости от преобразуемых данных заключается в том, что части секретного ключа используются непосредственно, но перед их наложением на подблоки данных они преобразуются с помощью операций, зависящих от текущего значения одного из подблоков данных. Такое преобразование (механизм внутреннего усложнения ключа) может быть выполнено одновременно с преобразованием другого подблока данных, поэтому оно не приводит к снижению скорости шифрования, хотя обеспечивает существенное улучшение характеристик

раундового преобразования. Для этого подхода также имеет место проблема устранения слабых ключей. Данная проблема может быть решена путем построения универсальных криптосхем, в которых один раунд шифрования не является инволюцией и включает так называемую переключаемую операцию, которая является разновидностью управляемых операций. Переключаемой операцией называется управляемая операция, включающая две взаимообратные модификации и управляемая битом режима преобразования e ($e = 0$ соответствует зашифрованию, $e = 1$ – расшифрованию). Использование переключаемых операций в сочетании с механизмом внутреннего усложнения ключа представляется весьма перспективным для построения шифров с простым расписанием использования секретного ключа и являющихся свободными от наличия слабых ключей. Удачные решения в данном направлении имеют большое практическое значение, поскольку позволяют существенно упростить аппаратную реализацию и обеспечить высокое быстродействие при частой смене ключей.

1.6.3. Варианты криптосхем

В рассматриваемых ниже криптосхемах мы предполагаем, что используется ПРК. Они описывают некоторые типы общей структуры итеративного преобразования с использованием управляемых операций, условно обозначенных операционными блоками $\mathbf{F}$, $\mathbf{F}^{-1}$, $\mathbf{F_i}$ и $\mathbf{S}$. Блоки $\mathbf{F}$ и $\mathbf{F}^{-1}$ выполняют взаимно-обратные управляемые операции, т. е. при одном и том же значении управляющего вектора реализуемые ими модификации являются взаимно-обратными операциями. Блок $\mathbf{F_i}$ представляет собой управляемую инволюцию, т. е. для каждого из возможных значений управляющего вектора этот блок реализует операцию, являющуюся инволюцией. Таким образом, для используемых гипотетических управляемых операций для всех V и X выполняются соотношения

$$X = \mathrm{F}^{-1(V)}(\mathrm{F}^{(V)}(X)), \quad X = \mathrm{F}^{(V)}(\mathrm{F}^{-1(V)}(X)) \quad \text{и} \quad X = \mathbf{F_i}^{(V)}(\mathbf{F_i}^{(V)}(X)).$$

Криптосхема, представленная на рис. 1.1а, описывает один раунд шифрования, который можно представить в следующем виде:

$$(A_j, B_j) = \mathrm{Crypt}(A_{j-1}, B_{j-1}, K_j, Q_j),$$

где (A_{j-1}, B_{j-1}) и (A_j, B_j) – входной и преобразованный блоки данных, представленные в виде конкатенации подблоков одного размера, (K_j, Q_j) – j-ый раундовый ключ. Если выбрать эту схему, то дальнейшее проектирование шифра будет сводиться к разработке конкретной пары управляемых операций $\mathbf{F}$ и $\mathbf{F}^{-1}$, выбору числа раундов шифрования R и формированию расписания ключа. Данное раундовое преобразование не является инволюцией, но оно легко обращается простой перестановкой раундовых подключей K_j и Q_j:

$$\mathrm{Crypt}(A_j, B_j, Q_j, K_j) = (A'_j, B'_j);$$

$$\mathrm{Crypt}^{-1}(A'_j, B'_j, K_j, Q_j) = (A_j, B_j).$$

Легко показать, что при использовании ПРК, для которого выполняется условие $(K_j, Q_j) = (Q_{R-j+1}, K_{R-j+1})'$, где штрихом обозначен $(R - j + 1)$-ый раундовый ключ

процедуры расшифрования, шифрование является корректным, т. е. процедура расшифрования будет обратной процедуре зашифрования.

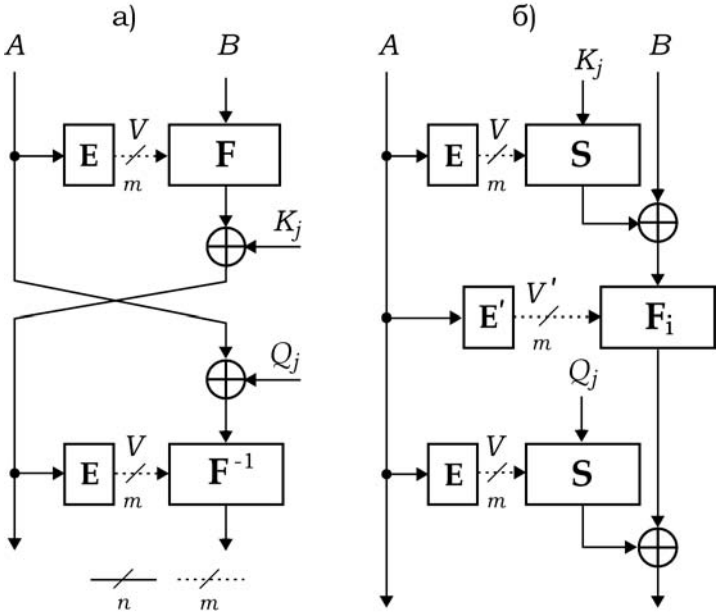

Рис. 1.1.

Один раунд шифрования, представленный на рис. 1.1б, также не является инволюцией, но легко обращается транспозицией раундовых подключей K_j и Q_j. Благодаря использованию управляемой инволюции нет необходимости выполнения двух взаимно-обратных операций при поочередном преобразовании подблоков данных, что упрощает структуру раундового преобразования. Особенностью данной криптосхемы является использование преобразования раундовых ключей с помощью управляемой операции **S**. Преобразование раундовых подключей в зависимости от преобразуемых данных можно назвать внутренним развертыванием ключа (internal key scheduling). Достоинством этой криптосхемы является возможность параллельного осуществления преобразования правого подблока с помощью операции $\mathbf{F_i}$ и подключа Q_j с помощью операции **S**. Таким образом, одна из двух операций преобразования подключей не вносит никакой временной задержки. Расшифрование выполняется таким изменением расписания ключа, при котором выполняется условие $(Q_{R-j+1}, K_{R-j+1})' = (K_j, Q_j)$. Укрупненная схема итеративного шифрования с использованием раундовых преобразований, показанных на рис. 1.1а и 1.1б, представлена на рис. 1.2а.

Возможно построение итеративных алгоритмов шифрования, в которых один отдельный раунд шифрования не является инволюцией и не может быть обращен перестановкой используемых в нем подключей. В таких случаях возможность легкой смены режима зашифрования на режим расшифрования обеспечивается использованием конечного преобразования, которое вносит необходимую симметрию в общую

процедуру шифрования. Благодаря этому становится возможным осуществить смену режима шифрования путем простого изменения расписания ключа.

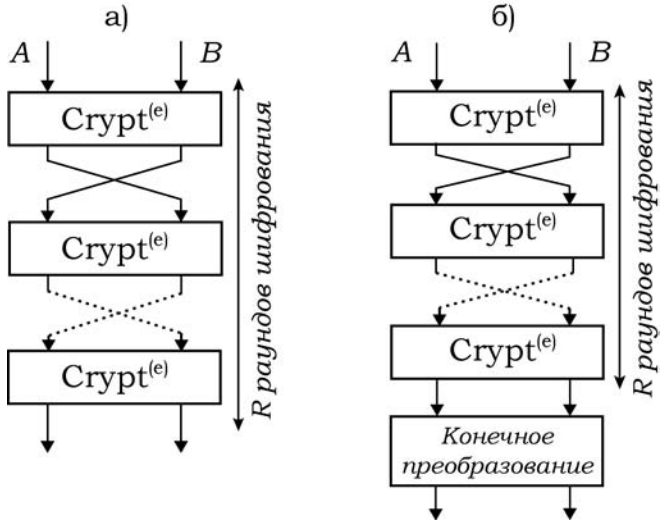

Рис. 1.2.

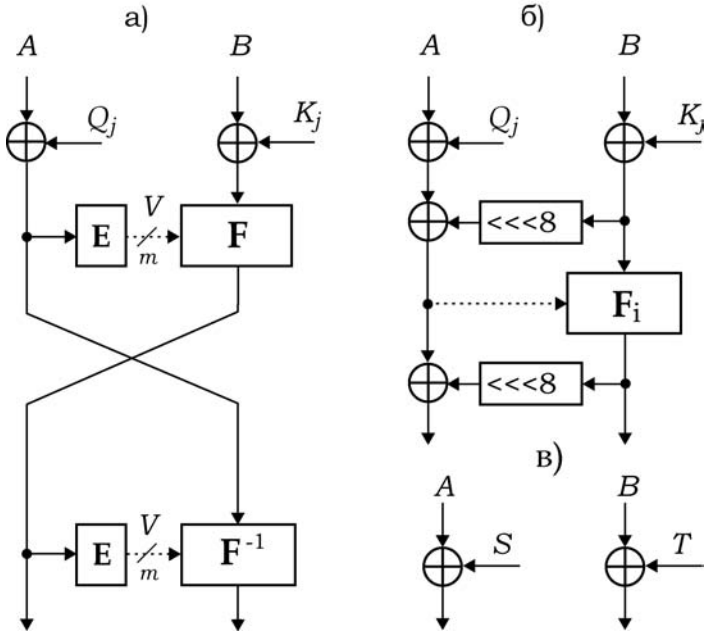

Рис. 1.3.

На рис. 1.3а и 1.3б показаны структуры раундового преобразования, которые требуют применения симметрирующего конечного преобразования. Первая схема характеризуется тем, что правый подблок преобразуется путем наложения на него подключей K_j, за которым следует выполнение прямой управляемой операции **F**, а левый – путем наложения на него подключей Q_j, за которым следует выполнение обратной управляемой операции $\mathbf{F}^{-1}$. После выполнения заданного числа раундов шифрования выполняется конечное преобразование (рис.1б.) с использованием подключей S и T:

$$(A', B') = (A_R \oplus S,\ B_R \oplus T).$$

Корректность шифрования обеспечивается таким изменением расписания ключа, при котором используемые при расшифровании подключи равны:

$$K'_1 = T;\quad K'_j = Q_{R-j+2} \text{ для } j = 2, 3, \ldots, R;\quad S' = Q_1;$$

$$Q'_1 = S;\quad Q'_j = K_{R-j+2} \text{ для } j = 2, 3, \ldots, R;\quad T' = K_1.$$

Раунд шифрования, показанный на рис. 1.3б, отличается более простым строением, что достигнуто использованием управляемой инволюции в качестве переменной операции. Корректность шифрования обеспечивается использованием указанных выше соотношений между подключами зашифрования и расшифрования:

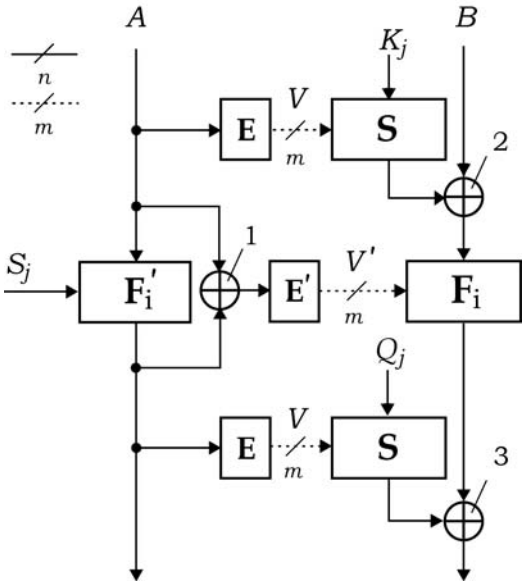

Рис. 1.4.

Представляет интерес криптосхема, показанная на рис. 1.4, где левый и правый подблоки данных преобразуются путем выполнения над ними управляемых инволю-

ций $\mathbf{F_i}'$ и $\mathbf{F_i}$, соответственно. Если временные задержки, вносимые управляемыми операциями $\mathbf{S}$, $\mathbf{F_i}'$ и $\mathbf{F_i}$, одинаковы, то при аппаратной реализации преобразование левого подблока и подключа K_j разумно выполнить одновременно, затем одновременно выполнить первую и вторую операции поразрядного суммирования по модулю два ($\oplus$), после чего выполнить параллельно преобразование правого подблока и подключа Q_j. Раундовое преобразование завершается выполнением третьей операции XOR. В результате оказываются преобразованными оба подблока данных, однако более сильному преобразованию подвергается правый подблок. Управляемая операция $\mathbf{F_i}'$, выполняемая над левым подблоком, зависит от подключа S_j. Вместо этой операции можно использовать каскад из 8 блоков подстановок размера 4×4 или из 4 блоков подстановок типа 8×8, в котором каждая подстановка является нелинейной инволюцией. Это устраняет необходимость использовать подключ S_j. Особенностью криптосхемы является суммирование входного и выходного значений операции $\mathbf{F_i}'$, что приводит к формированию одинаковых управляющих векторов операции $\mathbf{F_i}$ на соответствующих шагах процедур зашифрования и расшифрования. Для обеспечения корректности процедуры шифрования следует использовать ПРК, описываемое следующими формулами:

$$K'_j = Q_{R-j+1}; \quad Q'_j = K_{R-j+1}; \quad S'_j = S_{R-j+1}, \text{ где } j = 1, 2, \ldots, R.$$

Большое число других схем построения итеративных шифров на основе управляемых операций читатель может найти в монографии [14].

1.6.4. Этапы проектирования шифров

При проектировании шифров первостепенным вопросом является вопрос обеспечения стойкости. Этот вопрос одновременно является и наиболее сложным. Оценка стойкости является одним из наиболее длительных, трудоемких и разноплановых этапов. При этом важным является творческий подход, поскольку в настоящее время нет законченной теории блочных шифров, которая бы позволила выработать законченную методику оценивания стойкости. Однако уже предложены некоторые общие требования к качеству шифрующих преобразований. Если шифр удовлетворяет таким требованиям, то говорят о доказуемой стойкости (к известным методам криптоанализа) или о достижении гарантированных свойств шифрующих преобразований.

Необходимо иметь в виду, что достижение высокой стойкости не является самоцелью. Для криптографа, имеющего определенный опыт, не составляет труда разработать практически нераскрываемый шифр достаточно быстро как для аппаратной, так и для программной реализации. Важным является обеспечить параметры шифров (производительность, сложность реализации и др.), необходимые для конкретного применения. Эксплуатационные требования приводят к тому, что разработка подходящего шифра потребует большой искушенности и труда. Конкретные условия применения могут существенно влиять на выбор общей схемы построения шифра и на этап оценивания. Процесс синтеза блочных шифров связан со следующими этапами и соответствующими им вопросами:

1. ***Изучение области применения***. На этом этапе осуществляется выбор типа криптосистемы и формулирование требований к ее основным параметрам. Поточные шифры позволяют получить более высокие скорости шифрования и обеспечивают возможность независимого преобразования отдельных байтов и битов, что позволяет снизить влияние ошибок, возникающих при передаче зашифрованных сообщений по каналам с шумом. Однако для поточных шифров типичным является последовательная обработка знаков или битов, что затрудняет произвольный доступ к зашифрованным данным. Этот недостаток может быть устранен с использованием поточных шифров, элемент ключевой гаммы в которых вырабатывается в зависимости от секретного ключа и номера этого элемента. В таких шифрах уже имеются признаки блочного шифрования. Блочные шифры в настоящее время нашли более широкое применение. Они обеспечивают высокую стойкость в режиме независимого шифрования отдельных блоков, позволяя осуществлять произвольный доступ к зашифрованным массивам данных. В ряде средств защиты информации от несанкционированного доступа используется скоростное шифрование файлов в режиме on-line с использованием шифра комбинированного типа, имеющего признаки поточного и блочного шифрования. В подобных шифрах осуществляется независимое преобразование отдельных байтов, но они шифруются в зависимости от своего номера в файле и от специальной метки файла, вырабатываемой по случайному закону при создании файла. При выборе блочного варианта шифрования требуется выполнить обоснование длины используемого блока. С учетом областей применения определяются значения варианта реализации (программный, аппаратный, универсальный), сложности аппаратной реализации и скорости шифрования. Одновременно с выполнением шифрующих преобразований на криптографическое устройство может быть возложена задача вычисления хэш-функций от документов и сообщений. Реализация будет более экономичной, если хэш-функцию построить на основе алгоритма шифрования. Это вносит некоторые новые специфические вопросы, которые также должны быть учтены.
2. ***Выбор длины секретного ключа***. Нужно учитывать, что для любой криптосистемы с конечным ключом всегда существует возможность найти ключ методом полного перебора возможных ключей. В настоящее время для универсального случая рекомендуется длина ключа не менее 128 бит, но в отдельных случаях может применяться длина ключа 64 и даже 56 бит, если информация не представляет значительной ценности. В ситуациях, когда использование и разработка шифров с ограниченной длиной ключа (например, менее 40 бит) не ограничивается законодательно, может быть поставлена задача повышения стойкости таких криптосистем (см. раздел 1.2).
3. ***Выбор расписания использования ключа***. На этом этапе рекомендуется воспользоваться разъяснениями, приведенными в параграфе 1.6.2.
4. ***Выбор базовых криптографических примитивов и разработка криптосхемы***. Выполнение этого пункта требует знаний об основных подходах к построению шифров, типах блочных криптосистем, а также о конкретных шифрах и свойствах используемых операций, стоимости их аппаратной реализации и вносимом времени задержки.

5. ***Оценивание потребляемых ресурсов для реализации алгоритма шифрования***. Рассматриваются варианты программной и аппаратной реализации. Осуществляется реализация экспериментальных шифраторов и их программных моделей.
6. ***Оценивание производительности шифра***. Определяется значение скорости шифрования для различных вариантов программной и аппаратной реализации. Если оценки, полученные на шагах 5 и 6, не удовлетворяют значениям определенным на шаге 1, то повторяется этап 4 с учетом результатов, полученных на текущем этапе.
7. ***Рассмотрение стойкости к возможным типам криптоаналитических атак***. Рассматриваются основные варианты криптоанализа (на основе известного и специально подобранного текста), а также другие возможные варианты атак с использованием особенностей применения (инженерный анализ). Выявление наиболее эффективной атаки, трудоемкость которой и будет определять значение стойкости шифра. Разработчику полезно использовать критерий неотличимости блочного шифра от случайного преобразования. Достаточно определить трудоемкость для проведения атаки, позволяющей показать отличие шифра от случайного преобразования. Полученное значение не превышает трудоемкости вычисления ключа. Рассматривается объем необходимой памяти для выполнения различных вариантов атак и их трудоемкость.
8. ***Модифицирование алгоритма с учетом предварительного анализа***. С учетом результатов, полученных на этапе 7, осуществляется оптимизация основных узлов криптосхемы с целью повышения трудоемкости наиболее эффективной атаки. Затем выполняется предварительная оценка стойкости модифицированного алгоритма. Если необходимо, повторить этот пункт несколько раз до получения приемлемого предварительного значения стойкости.
9. ***Выполнение детального анализа модифицированного шифра***. Если детальный анализ выявил существенные слабости в модифицированной схеме, то требуется вернуться к этапу 8 или даже к этапу 4.
10. ***Выполнение статистических тестов и специальных экспериментов***. На этом шаге осуществляется программирование алгоритма шифрования или его реализация в ПЛИС и проводятся стандартные статистические тесты (см. [14]) и специальные эксперименты, планируемые с учетом результатов анализа для проверки полноты и адекватности теоретического анализа. Кроме того, спроектированный шифр (возможно уже вовлеченный в практическое использование) должен быть рассмотрен в плане возможных новых атак, основанных на последних криптоаналитических идеях. Этап анализа должен продолжаться и в процессе использования шифра.

1.7. Класс алгоритмов с выборкой подключей, зависящей от преобразуемых данных

В монографии [14] описан ряд скоростных программных алгоритмов шифрования, основанных на механизме выборки подключей (из расширенного ключа) в зави-

симости от преобразуемых данных. Причем их особенностью является то, что номера выбираемых подключей задаются не текущими значениями преобразуемых подблоков данных, а некоторыми аккумулирующими переменными. Причем значения последних формируются в зависимости от промежуточных значений большого числа подблоков преобразуемых данных. Эффективность такого механизма особенно выражено проявляется при большом размере входного блока данных (например, от 64 до 512 байт). С этим обстоятельством связано применение таких алгоритмов для шифрования данных, записываемых на встроенные магнитные диски ЭВМ, хотя они обеспечивают высокую стойкость и при сравнительно малом размере входного сообщения (например, 128 бит). Использование аккумулирующих переменных для непосредственной выборки подключей вносит существенный вклад в повышение стойкости, однако это также приводит в сложности анализа равномерности выборки различных подключей. Этот вопрос представляет определенный интерес, хотя в случае шифрования специально подобранных входных текстов, которые незначительно отличаются друг от друга, использование аккумулирующих переменных будет приводить к выравниванию потенциальной неравномерности, вносимой специально подобранными текстами.

Вопрос равномерности выборок подключей может быть проанализирован для другого класса программных шифров, в котором используется более простой механизм задания зависимости выбираемых подключей в зависимости от данных. Ниже рассматривается класс программных шифров с упрощенным механизмом выборки подключей. В нем в качестве номеров текущих выбираемых подключей используются непосредственно текущие значения преобразуемых подблоков данных. Этот класс программных шифров представляет как теоретический интерес в плане анализа равномерности ключевой выборки и наличия эквивалентных расширенных ключей, так и практический интерес, поскольку при обеспечении высокой стойкости они обладают высокой производительностью в случае шифрования блоков данных сравнительно малого размера (64 и 128 бит). Кроме того, показывая строгую равномерность выборки подключей в упрощенном механизме, эти алгоритмы дают косвенное теоретическое подтверждение достаточной равномерности выборки и для случая использования аккумулирующих переменных (в дополнение к экспериментальному подтверждению).

1.7.1. Формальное описание шифрующих процедур и свойство равномерности выборки

При описании класса программных шифров с упрощенной выборкой подключей будем использовать следующие обозначения:

- входной блок M представляется в виде конкатенации b-битовых подблоков $M^{(i)}$: $M = M^{(n-1)}||M^{(n-2)}||...||M^{(1)}||M^{(0)}$;
- $\# M$ – мощность множества всех возможных входных блоков, $\# M = 2^{nb}$;
- $\{Q_j\} = \{Q[j]\}$ – расширенный ключ шифрования, представленный в виде последовательности b-битовых подключей $Q[j]$, $j = 0, 1, 2, ..., 2^{\delta-1}$;

- Δ – двоичное число в интервале от 0 до $2^{\delta} - 1$, задаваемое управляющими двоичными разрядами подблока и определяющее текущий подключ $Q_{\Delta} = Q[\Delta]$, используемый в преобразованиях;
- δ – количество управляющих битов одного из подблоков данных, которые задают номер текущего выбираемого подключа;
- d – целое число, задающее число подраундов в одном раунде шифрования;
- s – номер текущего подраунда, $s = 1, 2, ..., b/d$;
- i – номер текущего шага шифрования в текущем раунде, $i = 1, 2, ..., nb/d$;
- $\alpha(g)$ – некоторая подстановка, определенная на множестве чисел $\{0, 1, ..., n - 1\}$;
- запись $M'^{>x>}$ ($M'^{<x<}$) обозначает операцию циклического сдвига вправо (влево) подблока M' на x двоичных разрядов.

Рассмотрим механизм преобразования. Один раунд шифрования описывается следующим выражением

$$C = \mathrm{E}(M) = \mathrm{e}_k(\mathrm{e}_{k-1}(...\mathrm{e}_2(\mathrm{e}_1(M)))),$$

где $k = nb/d$ – полное число некоторых элементарных функций шифрования $\mathrm{e}_1, \mathrm{e}_2, ..., \mathrm{e}_k$, C – выходной блок шифртекста. На i-ом шаге преобразуется только подблок $M^{(\alpha(g))}$, где $g = i \bmod n$:

$$M^{(\alpha(g))} := \mathrm{e}_i(M^{(\alpha(g))}, Q[\Delta(s, g')]),$$

где $\Delta(s, g') = (M^{(\alpha(g'))})^{>d(s-1)>} \bmod 2^{\delta}$, $g' = (i - 1) \bmod n$, e_i – некоторая элементарная шифрующая функция, которая для фиксированного $M^{(\alpha(g))}$ определяет некоторую подстановку $\gamma(Q_{\Delta})$ на множестве чисел $\{0, 1, ..., 2^b - 1\}$. Заметим, что номер выбираемого подключа Δ зависит от подблока с номером $\alpha(g')$ и этот подключ используется при преобразовании подблока данных с номером $\alpha(g)$. Причем функция преобразования такая, что подблок $M^{\alpha(g)}$ преобразуется к разным значениям при использовании подключей Q_{Δ}, имеющих разные значения.

Шифрование данного блока M можно представить в виде обобщенной формулы:

$$C = \mathrm{F}(M, Q_{h_1}, Q_{h_2}, ..., Q_{h_k}),$$

где $i = 1, 2, ..., k$ и Q_{h_i} – некоторый набор подключей, использующийся при шифровании данного подблока. Как видно из структуры функции шифрования, текущий набор $\{Q_{h_i}\}$ зависит как от ключа, так и от входного сообщения. Можно сказать, что для данного ключа и данной конкретной функции шифрования каждый входной блок генерирует некоторый набор индексов $\{h_i\}$. Соответствующий набор подключей $\{Q_{h_i}\}$ представляет собой некоторый виртуальный ключ для преобразования данного блока M. Существуют только $N_M = 2^{bn}$ различных входных блоков, тогда как число различных наборов $\{Q_{h_i}\}$ равно $N_h \approx (2^{\delta bn/d})^R$, где R – число раундов шифрования. Это

позволяет предположить, что вероятность генерации двух одинаковых виртуальных ключей для двух разных входных блоков является очень малой. При разработке конкретных шифров предпочтительно использование таких элементарных шифрующих функций, которые задают уникальность ключевых выборок для всех входных сообщений. Это требование ограничивает число возможных вариантов реализации подобных шифров с псевдовероятностной выборкой подключей, однако их число остается достаточно большим. Кроме того, могут быть разработаны гибкие шифры с числом потенциально реализуемых криптоалгоритмов более 10^{40} (следующие разделы), каждый из которых удовлетворяет данному требованию.

Процедуры расшифрования описываются следующей формулой:

$$M = D(C) = \mathrm{d}_k(\mathrm{d}_{k-1}(...\mathrm{d}_2(\mathrm{d}_1(C)))),$$

где d_i – некоторая элементарная функция, которая выполняется на i-ом шаге и является обратной по отношению к e_{n-i+1}. Элементарные дешифрующие функции имеют следующую структуру:

$$M^{(\alpha(q))} := \mathrm{d}_i(M^{(\alpha(q))}, \mathrm{Q}[\Delta(s, q')]),$$

где $\Delta(s, q') = (M^{(\alpha(q'))})^{>b-ds)>} \bmod 2^{\delta}$, $q = n + 1 - i \bmod n$, $q' = n - i \bmod n$. Легко установить, что для данного ключа шифрования $\{Q_j\}$ на соответствующих шагах шифрующих и дешифрующих преобразований используются одни и те же подключи.

Рассмотрим случай $d = \delta$ и такие элементарные шифрующие функции, которые удовлетворяют следующему условию:

Критерий 1.1

Все элементарные шифрующие функции задают подстановку старшей части преобразуемых подблоков.

В этом критерии предполагается, что текущий подблок представлен в виде $M' = H||L = H \cdot 2^{d(s'-1)} + L$, где старшей частью является H, а младшей – L, причем количество двоичных разрядов младшей части зависит от номера текущего подраунда преобразования и равно $l = d(s'-1)$, где $s' = s$ для всех элементарных шагов шифрования i, кроме $i = n, 2n, \ldots, nb/d$, и $s' = s + 1$ для $i = n, 2n, \ldots, n(b/d - 1)$. Критерий 1.1 определяет выбор элементарных функций со следующей структурой:

$$M' := \mathrm{e}(M') = \beta(H) \cdot 2^{d(s'-1)} + \mathrm{f}(L),$$

где $\beta(H)$ – подстановка на множестве чисел $\{0, 1, ..., 2^{b-d(s'-1)} - 1\}$ и $\mathrm{f}(L) \le 2^{\mathrm{d}(s'-1)} - 1$. При выполнении критерия 1.1 следующее утверждение является очевидным (этот критерий направлен на реализацию данного утверждения).

Утверждение 1.1

Для произвольного ключа шифрования при $d = \delta$ каждый входной блок генерирует уникальный набор индексов $\{h_i\}$.

Представляет интерес следующая теорема об уникальности ключевой выборки.

Теорема 1.1

Если $d = \delta$, то для произвольного ключа шифрования $\{Q_j\}$ и произвольного набора индексов $\{h_i\}$, где $i = 1, 2, ..., n/d$, существует единственный входной блок M, для которого генерируется заданный набор индексов.

Доказательство

Мощность множества всех возможных наборов из k индексов равна $\# \{h_i\} = (2^{\delta})^{nb/d}$. При $d = \delta$ имеем $\# \{h_i\} = 2^{nb} = \# M$. Учитывая утверждение 1.1, получаем взаимнооднозначное соответствие между элементами $\{h_i\}$ и множеством различных входных сообщений M, которое доказывает теорему 1.1 ◊

Теорема 1.1 отражает следующие два свойства, характерных для рассматриваемого класса функций шифрования:

1) неповторимость расписания использования подключей для всех возможных входных блоков данных
2) равномерность ключевых выборок.

Согласно этой теореме, для любого расширенного ключа шифрования и любой модификации шифрующих преобразований все возможные комбинации из k подключей реализуются одинаковое число раз при шифровании всего множества входных блоков. Для случая многораундовых функций, представляющих собой суперпозицию модификаций одноразундовых функций рассматриваемого класса, из теоремы об уникальности и равномерности ключевой выборки непосредственно следует, что при преобразовании всех возможных входных блоков каждый подключ будет выбран одинаковое число раз. При этом данное положение имеет силу для произвольного расширенного ключа шифрования.

Как известно, любой блочный шифр задает для заданного фиксированного ключа шифрования $\{Q_j\}$ некоторую подстановку $\varepsilon(M)$ на множестве двоичных чисел $\{0, 1, ..., 2^{nb} - 1\}$, где nb – длина входного блока. В общем случае различным ключам соответствуют разные подстановки $\varepsilon(M)$, хотя для некоторых случаев им может соответствовать одна и та же подстановка. Очевидным требованием к шифрам является задание очень большого значения мощности множества возможных подстановок $\varepsilon(M)$. Для дальнейшего изложения полезны следующие определения:

Определение 1.1

Для заданной функции шифрования два ключа $\{Q_j\}$ и $\{Q'_j\}$ называются эквивалентными, если они определяют одну и ту же подстановку $\varepsilon(M)$.

Определение 1.2

Функции шифрования называются эквивалентными в неформальном смысле, если они задают одинаковые подстановки $\varepsilon(M)$ для произвольного заданного ключа шифрования.

Определение 1.3

Функции шифрования называются эквивалентными в формальном смысле, если они задают одинаковые множества подстановок $\varepsilon(M)$ для всех возможных ключей шифрования.

Можно ожидать, что в гибких (т. е. с алгоритмом шифрования, зависящим от секретного ключа) шифрах с псевдослучайной выборкой подключей вероятность генерации эквивалентных модификаций или эквивалентных ключей для фиксированной функции шифрования является пренебрежимо малой. Однако формальное доказательство этого предположения в общем случае достаточно сложно. Дальше будут приведены конкретные алгоритмы, относящиеся к рассматриваемому классу недетерминированных шифрующих функций, для которых может быть дано строгое доказательство неэквивалентности всех возможных модификаций.

1.7.2. Алгоритмическая реализация

При построении шифрующей системы должно выполняться следующее условие: *все элементарные шифрующие функции должны задавать такое изменение текущего подблока данных, которое соответствует выполнению подстановки над преобразуемым подблоком.* В частности это условие выполняется для алгоритма 1.1. Если бы это условие не выполнялось, то было бы проблематичным составление алгоритма расшифрования. Но в случае хэширующих преобразований выполнение этого условия не является строго обязательным.

В общем случае в рассмотренном выше классе шифрующих преобразований подблоки данных могут быть преобразованы в различной последовательности, задаваемой подстановкой α. Задание различных очередностей преобразования подблоков может быть использовано при построении гибких алгоритмов, но в случае заранее известной очередности это не является принципиальным.

В представленных ниже алгоритмах шифрования последовательно преобразуемые блоки представляют собой 32-битовые слова. В качестве подстановки $\alpha(g)$ используется тождественная подстановка для всех подраундов. Для задания недетерминированности криптоалгоритма используется только способ настройки операций преобразований. В качестве зарезервированных операций, подлежащих настройке на этапе инициализации, используются простые арифметические операции: поразрядное сложение по модулю два ($\oplus$), сложение и вычитание по модулю 2^{32} («+» и «–», соответственно), а также циклический сдвиг на фиксированное число битов или на число битов, зависящее от преобразуемых данных. Предполагается, что используется случайно выбираемый 1024-байтовый ключ $\{Q_j\} = \{Q_\Delta\}$, где Q_j – 32-битовые подключи ($\Delta = j = 0, 1, ..., 255$).

При записи алгоритмов использованы следующие обозначения:

- X, Y, Z, W – 32-битовые переменные;
- $u = (s - 1) \bmod 4$;

- $Q(\Delta) = Q_\Delta$, где $\Delta = a_u, b_u, d_u, e_u$;
- $a_u = (X_s)^{>8u>} \bmod 2^8$;
- $b_u = (Y_s)^{>8u>} \bmod 2^8$;
- $d_u = (Z_s)^{>8u>} \bmod 2^8$;
- $e_u = (W_{s-1})^{>8u>} \bmod 2^8$;
- $X = x_4||x_3||x_2||x_1$; $Y = y_4||y_3||y_2||y_1$;
- $Z = z_4||z_3||z_2||z_1$; $W = w_4||w_3||w_2||w_1$.

Алгоритм 1.1: Зашифрование 128 -битовых блоков данных ($d = \delta = 8$; $b = 32$; $n = 4$)

ВХОД: 128-битовый блок исходных данных $M = D||B||A||E$, представленный в виде конкатенации четырех 32-битовых слов.

1. Установить: $R = 3$, $s = 1$, $X_0 = A$, $Y_0 = B$, $Z_0 = D$, и $W_0 = E$.
2. Выполнить преобразование подблока X: $X_s = X_{s-1} \oplus Q(e_u)$.
3. Преобразовать подблок Y: $Y_s = Y_{s-1} \oplus Q(a_u)$.
4. Преобразовать подблок Z: $Z_s = Z_{s-1} \oplus Q(b_u)$.
5. Преобразовать подблок W, выполнив следующие два шага:

$$W_s = W_{s-1} \oplus Q(d_u) \quad \text{и} \quad (w_4||w_3)_s := (w_4||w_3)_s^{>Y_s>}.$$

6. Прирастить $s := s + 1$ и преобразовать подблок X:

$$X_s = [X_{s-1} - W_{s-1}) \bmod 2^{32}] \oplus Q(e_u).$$

7. Преобразовать подблок Y: $Y_s = \{[Y_{s-1} + Q(a_u)] \bmod 2^{32}\} \oplus X_s$.
8. Преобразовать подблок Z: $Z_s = [(Z_{s-1} \oplus Y_s) - Q(b_u)] \bmod 2^{32}$.
9. Преобразовать подблок W, выполнив следующие два шага:

$$W_s = [W_{s-1} - Q(d_u)] \bmod 2^{32}. \quad \text{и} \quad (w_2||w_1)_s := (w_2||w_1)_s^{>Y_s>}.$$

10. Прирастить $s := s + 1$ и преобразовать подблок X:

$$X_s = [X_{s-1} - Q(e_u)] \bmod 2^{32}].$$

11. Преобразовать подблок Y: : $Y_s = \{[Y_{s-1} \oplus Q(a_u)] + W_{s-1}\} \bmod 2^{32}$.
12. Преобразовать подблок Z: $Z_s = \{[Z_{s-1} \oplus Q(b_u)] - X_s\} \bmod 2^{32}$.
13. Преобразовать подблок W: $W_s = W_{s-1} \oplus Q(d_u)$ и $s := s + 1$.
14. Преобразовать подблок X: $X_s = [X_{s-1} + Q(e_u)] \bmod 2^{32}$.
15. Преобразовать подблок Y: : $Y_s = Y_{s-1} \oplus Q(a_u)$.
16. Преобразовать подблок Z: $Z_s = [Z_{s-1} - Q(b_u)] \bmod 2^{32}$.
17. Преобразовать подблок W:

$$W_s = [(W_{s-1} \oplus Z_s)^{>Y_s>} + Q(d_u)] \bmod 2^{32}.$$

18. Если $s < 4R$, то прирастить s и перейти к шагу 2, в противном случае **СТОП**.

ВЫХОД: 128-битовый блок шифртекста $C = Z_{12}||Y_{12}||X_{12}||Z_{12}$.

Соответствующий алгоритм расшифрования можно легко составить, учитывая, что преобразования подблоков данных должны быть выполнены в обратном порядке с использованием соответствующих обратных операций. Допустимое число раундов шифрования равно $R \geq 2$. Заметим, что индекс s нумерует последовательные значения, принимаемые преобразуемыми подблоками данных на промежуточных шагах.

Элементарные шифрующие функции алгоритма 1.1 удовлетворяют критерию 1.1, следовательно, теорема 1.1 выполняется и мы имеем равномерную выборку подключей при использовании любого расширенного ключа. Заметим, что данный алгоритм работает с расширенным ключом достаточно большого размера, поэтому на практике он может быть использован совместно с некоторым алгоритмом предвычислений, который по секретному ключу формирует расширенный ключ. Варианты построения соответствующих алгоритмов предвычислений приведены в [14].

Рассмотрим алгоритм, в котором используются только два типа элементарных шифрующих функций, что делает запись более компактной.

Алгоритм 1.2: Зашифрование 64-битовых блоков данных ($d = \delta = 8$; $b = 32$; $n = 2$)

ВХОД: 128-битовый блок исходных данных $M = B||A$, представленный в виде конкатенации двух 32-битовых слов.

1. Задать число раундов $R = 3$ и установить начальные значения счетчика $s := 1$ и переменных $X := A$ и $Y := B$.

2. Выполнить преобразование переменной X: $X := [X^{>8>} + Q(y_1)] \bmod 2^{32}$.

3. Преобразовать переменную Y: $Y := \{[Y + Q(x_1)] \bmod 2^{32}\}^{>8>}$.

4. Если $s < 4R$, то прирастить s и перейти к шагу 2, в противном случае **СТОП**.

ВЫХОД: 64-битовый блок шифртекста $C = Y||X$.

1.7.3. Гибкие шифры с доказуемой неэквивалентностью всех модификаций криптоалгоритма

Уникальность выборки подключей для каждого преобразуемого сообщения способствует тому, что стойкость одноключевых криптосистем, основанных на механизме выборки подключей в зависимости от преобразуемых данных, не является критически чувствительной к конкретным наборам используемых элементарных шифрующих функций. В этом случае можно попытаться разработать гибкую криптосистему, в которой алгоритм непосредственного шифрования будет формироваться на этапе предвычислений в зависимости от секретного ключа. Например, в качестве

множества потенциально реализуемых модификаций криптоалгоритма можно выбрать некоторую совокупность алгоритмов, относящихся к классу, рассмотренному в разделе 1.7.1. В этом случае все потенциально реализуемые шифрующие функции будут обладать равномерностью выборки подключей. Рассмотрим некоторые возможные способы построения подобных недетерминированных шифров на основе выборки подключей в зависимости от преобразуемых данных.

1. Подстановка $\alpha(g)$ задает очередность преобразования подблоков входного сообщения. Ее формирование можно выполнить на этапе выполнения настройки шифра как один из шагов предвычислений. Этот прием дает при фиксированном наборе всех других элементов, формирующих функцию шифрования, число разных модификаций равное $n!$. Легко установить, что данная подстановка может быть различной для разных раундов и подраундов шифрования. Следовательно, используя модифицирование подстановки $\alpha(g)$, можно получить коэффициент размножения модификаций процедур преобразования равный $(n!)^{Rb/d}$, где R – число раундов шифрования. Для типичных значений $n = 2 – 4$, $R = 2 – 4$, $b = 32$ и $d = 8$ получаем значения этого коэффициента в интервале $10^2 – 10^{22}$.
2. Если элементарные функции e_i заданы некоторыми отдельными операциями преобразования или их комбинациями, то результирующая функция шифрования содержит в общем случае большое число операций нескольких типов. Применяя настройку (выбор конкретного вида) некоторого подмножества операций, можно получить очень большое число ($>10^{80}$) различных видов возможных криптоалгоритмов, реализуемых в единой криптосистеме. Для такого большого числа различных модификаций весьма актуальным является вопрос об оценке числа их подмножеств, задающих неэквивалентные шифрующие функции.

Во втором способе задания недетерминированности криптоалгоритма различные его модификации задаются отличающимися наборами операций преобразования, однако среди отличающихся по форме модификаций криптоалгоритма могут оказаться достаточно многочисленные подмножества, задающие одну и ту же функцию шифрования $C = E_Q(M)$ при произвольном заданном расширенном ключе Q. Можно предположить, что в некоторых случаях разные по форме представления функции E' и E'' могут задавать одинаковые зависимости $E'_{\mathbf{Q}_1}(M) = E''_{\mathbf{Q}_2}(M)$ при разных ключах $\mathbf{Q}_1$ и $\mathbf{Q}_2 \neq \mathbf{Q}_1$.

В приводимой ниже гибкой криптосистеме для задания многообразия вариантов представления криптоалгоритма использован только прием настойки операций преобразования. При использовании выбора подстановки $\alpha(g)$ по секретному ключу число возможных модификаций может быть увеличено в 10^3 раз и более, причем в этом случае не нарушается доказываемая ниже теорема о неэквивалентности (в неформальном смысле) каждой модификации алгоритма шифрования. Чрезвычайно большое число потенциально реализуемых модификаций криптоалгоритма делают криптосистему существенно более стойкой по сравнению со случаем фиксированно-

го алгоритма подобного типа. С целью исследования стойкости гибкого шифра могут быть проанализированы отдельные модификации, рассматриваемые как фиксированный известный алгоритм шифрования. Более того, при тестировании могут быть рассмотрены атаки на основе выбора слабейшей модификации, когда атакующему предоставляется возможность выбрать по его усмотрению модификацию и предпринять против нее криптоанализ на основе специально подобранных текстов. Если в этих условиях, которые благоприятствуют успеху атакующего, гибкий шифр оказывается стойким, то его можно считать стойким.

Приводимый ниже алгоритм 1.3 представляет собой гибкую криптосистему, основанную на механизме выборки подключей в зависимости от входных данных. Входной блок представляется в виде конкатенации двух 32-битовых подблоков. В качестве зарезервированных бинарных операций $*_p$ и $\lozenge_p$ (причем для любого заданного значения p операции $*_p$ и $\lozenge_p$ являются взаимно-обратными), подлежащих настройке на этапе инициализации, используются простые арифметические операции:

- поразрядное сложение по модулю два,
- сложение по модулю 2^{32},
- вычитание по модулю 2^{32}.

Кроме двухместных операций, зарезервированы также одноместные операции циклического сдвига вправо $>^{c_l}>$ на число битов равное $c_l = 1, 2, ..., 31$. Значения $p = 1, 2, ..., 8R$ и $l = 1, 2, ..., 11R$ являются порядковыми номерами двухместных и одноместных операций, соответственно.

При записи алгоритма, задающего гибкую криптосистему, использованы следующие обозначения:

- X, Y – 32-битовые переменные;
- $Q_j = \{Q(j)\}$ – ключ шифрования, представленный в виде последовательности 32-битовых подключей $Q(j)$, где $j = 0, 1, 2, ..., 255$;
- $X = x_4||x_3||x_2||x_1$ и $Y = y_4||y_3||y_2||y_1$, где x_i и y_i – 8-битовые подблоки преобразуемых данных.

Алгоритм 1.3 Гибкий шифр с 64-битовым входом (d = δ = 8; $b = 32$; $n = 2$)

ВХОД: 64-битовый блок данных, представленный как конкатенация 8-битовых слов m_i: $M = m_8||m_7||m_6||m_5||m_4||m_3||m_2||m_1$.

1. Установить режим преобразования: $e = 1$ (зашифрование) или $e = 0$ (расшифрование).
2. Установить число раундов шифрования $R = 3$ и значения: , $r := 1$, $X = m_8||m_7||m_6||m_5$ и $Y = m_4||m_3||m_2||m_1$.
3. Если $e = 0$, то перейти к шагу 11.
4. Преобразовать:

$$x_1 := x_1^{>c_{12r-11}>};\ x_2 := x_2^{>c_{12r-10}>};\ x_3 := x_3^{>c_{12r-9}>};\ x_4 := x_4^{>c_{12r-8}>};$$

$$y_1 := y_1^{>c_{12r-7}>};\ y_2 := y_2^{>c_{12r-6}>};\ y_3 := y_3^{>c_{12r-5}>};\ y_4 := y_4^{c_{12r-4}}.$$

5. Преобразовать:

$$X = [X *_{8r-7} Q(y_1)]^{>c_{12r-3}>};$$

$$Y = Y *_{8r-6} Q(x_1);$$

6. Преобразовать:

$$X = X *_{8r-5} Q(y_2);$$

$$(x_4||x_3) := (x_4||x_3)^{>c_{12r-2}>};$$

$$(y_4||y_3) := (y_4||y_3)^{>c_{12r-1}>}.$$

7. Преобразовать:

$$Y = Y *_{8r-4} Q(x_2).$$

8. Преобразовать:

$$X = X *_{8r-3} Q(y_3);$$

$$Y = Y *_{8r-2} Q(x_3).$$

9. Преобразовать:

$$X = X *_{8r-1} Q(y_4);$$

$$Y = [Y *_{8r} Q(x_4)]^{>c_{12r}>}.$$

10. Если $r < R$, то прирастить $r := r + 1$ и перейти к шагу 4, в противном случае перейти к шагу 18.

11. Преобразовать:

$$Y = [Y \lozenge_{8(R-r)+8} Q(x_4)]^{<c_{12(R-r)+12}<};$$

$$X = X \lozenge_{8(R-r)+7} Q(y_4).$$

12. Преобразовать:

$$Y = Y \lozenge_{8(R-r)+6} Q(x_3);$$

$$X = X \lozenge_{8(R-r)+5} Q(y_3).$$

13. Преобразовать: $Y = Y \lozenge_{8(R-r)+4} Q(x_2)$.

14. Преобразовать:

$$(y_4||y_3) := (y_4||y_3)^{<c_{12(R-r)+11}<};$$

$$(x_4||x_3) := (x_4||x_3)^{<c_{12(R-r)+10}<};$$

$$X = X \lozenge_{8(R-r)+3} Q(y_2).$$

15. Преобразовать:

$$Y = Y \lozenge_{8(R-r)+2} Q(x_1);$$

$$X = [X \lozenge_{8(R-r)+3} Q(y_1)]^{<c_{12(R-r)+9}<}.$$

16. Преобразовать:

$$y_4 := y_4^{<c_{12(R-r)+8}<};\ y_3 := y_3^{<c_{12(R-r)+7}<};\ y_2 := y_2^{<c_{12(R-r)+6}<};\ y_1 := y_1^{<c_{12(R-r)+5}<};$$

$$x_4 := x_4^{<c_{12(R-r)+4}<};\ x_3 := x_3^{<c_{12(R-r)+3}<};\ x_2 := x_2^{<c_{12(R-r)+2}<};\ x_1 := x_1^{<c_{12(R-r)+1}<};$$

17. Если $r < R$, то прирастить $r := r + 1$ и перейти к шагу 11.

18. СТОП.

ВЫХОД: 64-битовый блок данных $C = X||Y$.

Каждая модификация этого алгоритма обеспечивает скорость шифрования около 120/R Мбит/с (для процессора Pentium 200). Этот шифр инициализируется с помощью алгоритма предвычислений, задающего число зарезервированных операций в зависимости от установленного числа раундов шифрования R. Число потенциально реализуемых модификаций зависит от R и составляет примерно 10^{16R}. Допустимое число раундов шифрования равно $R \geq 3$.

Рассмотрим вопрос об эквивалентности модификаций алгоритма 1.3.

Неэквивалентность модификаций шифрующих функций E′ и E″ в неформальном смысле может быть установлена, если показать, что существует некоторый ключ и некоторое входное сообщение M, для которых выполняется соотношение E′(M) ≠ E″(M). Рассматривая структуру алгоритма 1.3, легко установить справедливость следующего:

Утверждение 1.2

Для произвольных двух модификаций функции шифрования **E**′ и **E**″, соответствующих различным наборам зарезервированных операций алгоритма 1.3, и произвольного ключа шифрования $\{Q_j\}$ существует значение M, для которого генерируются различные наборы индексов $\{h'_i(M)\}$ и $\{h''_i(\mathrm{i})\}$, i = 1, 2, ..., 8R, для **E**′ и **E**″, соответственно.

Действительно, если наборы операций в модификациях E′ и E″ различаются, то имеется некоторая первая (по порядку выполнения) зарезервированная позиция, в которой заданы различные операции преобразования. Пусть эта операция используется при формировании значения индекса $h_l(M)$. Рассматривая последовательность выполнения элементарных шагов преобразования для алгоритма 1.3, легко установить, что можно подобрать множество разных входных блоков, для которых будем иметь $h_l'(M) \neq h_l''(M)$. Докажем следующую теорему о неэквивалентности модификаций функций преобразования, задаваемых гибким алгоритмом 1.2.

Теорема 1.2

При $R \le 16$ все различные модификации алгоритма 1.3 задают неэквивалентные в неформальном смысле шифрующие функции.

Доказательство

Пусть E′ и E″ две различные модификации алгоритма 1.3.

Случай 1: Подмножества операций $\{*_1, *_2, ..., *_{8R-2}, {}^{>}c_1^{>}, {}^{>}c_2^{>}, ..., {}^{>}c_{12R-1}^{>}\}$ являются одинаковыми для E′ и E″. В этом случае подмножества операций $\{*_{8R-1}, *_{8R}, {}^{>}c_{12R}^{>}\}$ для E′ и E″ различны, так как в противном случае мы имели бы одну и ту же модификацию E = E′ = E″.

В этом случае для любого данного входного блока M легко указать множество различных ключей шифрования **Q**, удовлетворяющих условию $E'(M, \mathbf{Q}) \ne E''(M, \mathbf{Q})$.

Случай 2: Подмножества операций $\{*_1, *_2, ..., *_{8R-2}, {}^{>}c_1^{>}, {}^{>}c_2^{>}, ..., {}^{>}c_{12R-1}^{>}\}$ для E′ и E″ различаются.

Пусть r есть минимальное целое число, для которого подмножества операций $\{*_{8r-7}, *_{8r-6}, ..., *_{8r}, {}^{>}c_{12r-11}^{>}, {}^{>}c_{12r-10}^{>}, ..., {}^{>}c_{12r}^{>}\}$ для E′ и E″ различаются (иными словами r это номер раунда шифрования, на котором при выполнении модификаций E′ и E″ осуществляются различные операции, имеющие одинаковый номер). Функция шифрования E может быть представлена как суперпозиция одноразундовых шифрующих функций:

$$E = E_2(E_1(M)) = w_R(w_{R-1}(...w_1(M))),$$

где $E_1 = w_{r-1}(w_{r-2}(...w_1(M)))$, $E_2 = w_R(w_{R-1}(...w_r(M)))$ и w_r есть функция шифрования определяемая r-ым раундом. Поскольку $E_1'(M) = E_1''(M)$, мы будем рассматривать функции E_2' и E_2''. Существует минимальное $i = I$ $(8r - 7 \le I \le 8r)$, для которого мы имеем $h_I' \ne h_I''$. В соответствии с теоремой 1.2 существует входное сообщение M^*, которое задает генерацию подмножеств индексов

$$\{h'_{8r-7}, h'_{8r-6}, ..., h'_I, h'_{I+1}, ..., h'_{8r}\} \quad \text{и}$$

$$\{h''_{8r-7}, h''_{8r-6}, ..., h''_I, h''_{I+1}, ..., h''_{8r}\},$$

где $h'_{8r-7} = h'_{8r-6} = ... = h'_I \ne h'_{I+1}$; $h''_{8r-7} = h''_{8r-6} = ... = h''_{I-1} = h'_I \ne h''_I$ и $h''_I \ne h'_{I+1}$. Рассматриваемые индексы указывают номера выбираемых подключей, которые используются для преобразований соответствующих подблоков данных.

Легко установить, что, модифицируя подключ Q_{h_i}, можно задать произвольный индекс h_{i+1}. (После такого модифицирования мы будем иметь новый ключ шифрования, отличающийся от первоначального только подключом с номером h_i. При шифровании того же самого входного блока оба подключа будут генерировать одинако-

вые индексы $h_1, h_2, ..., h_i$.) Изменим подключи $Q_{h'_{I+1}}$ и $Q_{h''_I}$ таким образом, чтобы задать индексы h'_{I+2} и h''_{I+1}, отличающиеся от индексов h''_I, h'_I, и h'_{I+1}. Это модифицирование ключа шифрования не изменяет генерацию индексов h'_i для $i \le I + 1$ и индексов h''_i для $i \le I$, поскольку для E_2' и E_2'' подключи $Q_{h'_{I+1}}$ и $Q_{h''_I}$ при преобразовании сообщения M^* выбираются впервые. Если $R \le 16$, то мы можем последовательно модифицировать приводимые ниже подключи таким образом, чтобы каждый из их индексов выбирался впервые:

$$Q_{h'_{I+1}}, Q_{h''_I} \to Q_{h'_{I+2}}, Q_{h''_{I+1}} \to \dots \to Q_{h'_{8R-1}}, Q_{h''_{8R-2}} \to Q_{h''_{8R-1}}.$$

Таким образом, мы можем задать генерацию индексов

$$h'_{I+2}, h''_{I+1} \to h'_{I+3}, h''_{I+2} \to \dots \to h'_{8R}, h''_{8R-1} \to h''_{8R},$$

среди которых нет двух одинаковых. После всех модифицирований мы получим новый ключ шифрования $\mathbf{Q}'$. Если $E_2'(M^*, \mathbf{Q}') \ne E_2''(M^*, \mathbf{Q}')$, то мы имеем $E'(M^*, \mathbf{Q}') \ne E''(M^*, \mathbf{Q}')$. Если $E_2'(M^*, \mathbf{Q}') = E_2''(M^*, \mathbf{Q}')$, то мы можем модифицировать подключ $Q_{h''_{8R}}$, получая новый ключ шифрования $\mathbf{Q}''$, для которого выполняется соотношение

$$E_2'(M^*, \mathbf{Q}'') = E_2'(M^*, \mathbf{Q}') = E_2''(M^*, \mathbf{Q}') \ne E_2''(M^*, \mathbf{Q}''),$$

т. е. $E'(M^*, \mathbf{Q}'') \ne E''(M^*, \mathbf{Q}'')$, а следовательно модификации E' и E'' неэквивалентны.

Таким образом, для $R \le 16$ (в частности для $R = 3$) все возможные модификации являются неэквивалентными. Их число составляет примерно 10^{256} (10^{48}) при $R = 16$ ($R = 3$). Можно также предположить, что в рассмотренном гибком шифре *не существует двух эквивалентных в формальном смысле модификаций*, однако строгое доказательство этой гипотезы (или опровержение) дать довольно сложно. В плане оценки различия модификаций доказательство их неэквивалентности в неформальном смысле имеет определенное содержательное значение. Оно показывает, что в данной схеме построения гибких шифров настройка алгоритма непосредственного шифрования по ключу является эффективной.

По аналогии с алгоритмом 1.3 можно разработать и другие конкретные варианты гибких шифров с различным размером входа. Вариант построения 128-битового гибкого шифра с доказуемой неэквивалентностью модификаций рассмотрен в [42]. Построение гибких шифров на основе класса шифрующих функций, описанного в разделе 1.7.1, позволяет решить ряд вопросов, связанных с использованием формирования алгоритма непосредственного шифрования по секретному ключу. К ним относятся следующие вопросы, связанные с эффективностью реализации гибких шифров:

1. Обеспечение высокой стойкости любой из возможных модификаций к бесключевому чтению криптограмм. Это требование является очевидным, поскольку при таком способе атаки криптоаналитик не рассматривает непосредственно процедуры шифрования, а только анализирует статистику известных открытых текстов и соответствующих им шифртекстов.

2. Важным является получение требуемого минимального уровня стойкости по всему классу возможных модификаций криптоалгоритма.
3. Необходимо обеспечить примерно одинаковую вероятность выбора модификации из всех подмножеств эквивалентных модификаций.
4. Количество неэквивалентных модификаций должно быть большим, желательно >1030.
5. Необходимо показать, что для всех (или практически для всех) модификаций нельзя выделить некоторые группы раундов, суперпозиции которых задают взаимно-обратные функции преобразования. Данный пункт актуален для гибких шифров, в которых зарезервированные операции настраиваются в каждом раунде независимо (как это имеет место в алгоритме 1.3).

Пункты 1 и 2 находят свое решение в применении выборки подключей в зависимости от преобразуемых данных и использовании размера блоков данных ≥ 64 бит. При этом желательно обеспечить равномерность выборки подключей, т.е. примерно одинаковую частоту извлечения каждого подключа. Реализация требования 3 не представляет сложностей и связана с разработкой соответствующего алгоритма предвычислений, который должен выполнить инициализацию зарезервированных операций преобразования. Пункт 4 связан с анализом возможной эквивалентности различных модификаций алгоритмов непосредственного шифрования. Здесь могут оказаться полезными конструкции, обеспечивающие доказуемую неэквивалентность всех модификаций.

Вопрос 5 связан с обоснованием эффективности увеличения числа раундов шифрования для повышения стойкости. Проблема состоит в том, чтобы показать, что не существуют модификации, в которых шифрующее преобразование одной группы раундов будет сниматься после выполнения другой группы раундов, т. е. нужно показать, что добавление дополнительных раундов шифрования приводит к усилению, а не к ослаблению преобразования.

Таким образом, при разработке гибких шифров представляет интерес следующий вопрос. Могут ли быть выбраны такие модификации многораундовой функции шифрования, для которых несколько последующих раундов являются эквивалентными некоторой функции дешифрования, задающей обратные преобразования по отношению к нескольким предыдущим раундам шифрования? Этот вопрос является актуальным для шифров с формированием алгоритма преобразования по случайно выбираемому секретному ключу. Можно предположить, что для класса шифрующих функций, основанных на выборке подключей в зависимости от преобразуемых данных, такие ситуации не могут иметь места или, по крайней мере, вероятностью попадания на такие случаи можно пренебречь. В пользу этого косвенно свидетельствует теорема об уникальности каждой модификации недетерминированных алгоритмов, описанных выше.

Рассмотрим этот вопрос применительно к алгоритму 1.3. Используя технику доказательства, подобную доказательству теоремы 1.2, можно легко дать строгое доказательство, что для некоторого числа раундов из всех возможных модификаций алго-

ритма нельзя найти такую модификацию, в которой можно выделить некоторые группы раундов, задающих взаимно-обратные функции преобразования. Под взаимно-обратными модификациями будем понимать такие две модификации, одна из которых является эквивалентной модификации, являющейся обратной по отношению ко второй. Причем эквивалентность будем понимать в смысле определения неформальной эквивалентности (см. определение 1.2).

Рассмотрим гибкий алгоритм 1.3 при некотором заданном числе раундов равном $R \le 16$. Выберем две произвольных группы одноразундовых функций шифрования, причем в первой группе возьмем $R' \le 16$ функций, а во второй – $R'' \le 16 - R'$. Докажем, что не существует таких двух групп одноразундовых функций, которые бы определяли взаимно-обратные композиционные функции преобразования. Для этого укажем для произвольных наборов одноразундовых функций ключ шифрования, который задает разные подстановки для композиционной функции одной группы и функции обратной к композиционной функции другой группы (т.е. для композиционной функции дешифрования другой группы).

Выберем случайный ключ шифрования **Q**, а также входной блок M, для которого имеет место равенство

$$j'' = M \text{ div } 2^{56} \neq j' = M \bmod 2^8$$

Пусть E′ есть композиционная функция шифрования первой группы одноразундовых модификаций и D′′ есть композиционная функция дешифрования второй группы. Значение j' задает выбор первого подключа для E′ при шифровании M. Если на текущем шаге выбирается подключ, который ранее не выбирался, то мы его можем модифицировать таким образом, чтобы на следующем шаге преобразования для композиционной функции (E′ или D′′) выбирался другой новый подключ. Если первый подключ ранее не выбирался, то всегда имеется возможность так модифицировать текущие подключи, что все подключи будут выбираться не более одного раза (если общее количество выбираемых подключей не превысит 256).

Выполним процедуру шифрования E′(M) с однократной выборкой подключей, причем будем задавать такие номера выбираемых подключей, чтобы ни один из них не был равен j''. Пусть E′(M) = C. Осуществим теперь процедуру преобразования D′′(M). Значение j'' задает выбор первого подключа для D′′ при дешифровании M. Поскольку первый подключ, выбираемый при осуществлении процедуры D′′, ранее не использовался, то можно задать однократную выборку подключей, которые ранее не были использованы при выполнении процедуры E′.

На каждом раунде выбираются 8 подключей. При выполнении преобразований, задаваемых функцией E′, будет выбрано $8R'$ подключей, при осуществлении преобразований D′′ количество выбранных подключей составит $8R''$. Общее количество выбранных подключей равно $8(R' + R'') \le 256$, т.к. по условию $R'' \le 16 - R'$. Поскольку ключ шифрования содержит 256 подключей, то мы можем корректно выполнить процедуру выборки новых подключей на каждом шаге преобразования обеих рассматриваемых функций.

Очевидно, что последний выбираемый подключ для D″ (или для E′) можно так модифицировать, что будет выполняться соотношение $E'(M) \neq D''(M)$. Последнее означает, что композиционные функции E′ и D″ не являются эквивалентными в неформальном смысле. (Очевидно, что мы доказали также неэквивалентность композиционных функций D′ и E″, где D′ – функция обратная функции E′ и E″ – функция обратная функции D″). Поскольку мы рассматривали произвольные модификации, то это значит, что при $R \leq 16$ ни в одной из возможных модификаций алгоритма 1.3 невозможно выделить какие-то две группы однораундовых функций, для которых процедуры одной группы являются дешифрующими по отношению к другой группе процедур. Таким образом, мы доказали справедливость следующей теоремы о взаимно-обратных модификациях.

Теорема 1.3

При $R \leq 16$ ни в одну из модификаций алгоритма 1.3 не входят какие-либо взаимно-обратные суперпозиции однораундовых функций шифрования.

Эта теорема показывает, что при наращивании числа раундов преобразования нет опасности получить ослабленных модификаций за счет формирования суперпозиции новых раундовых функций, являющихся расшифровывающими по отношению к какой-либо предыдущей суперпозиции раундовых функций. При числе раундов $R > 16$ также представляются маловероятными случаи, нарушающие это положение, однако строгое общее доказательство при большом числе раундов проблематично.

В сравнение с программными шифрами, приведенными в *главе 2* книги [14], рассмотренный выше класс шифров отличается более простым заданием выборки подключей в зависимости от преобразуемых данных и непосредственным использованием подключей при выполнении преобразования подблоков данных. Это потребовало использования сравнительно большого числа шагов наложения подключей на каждый из подблоков данных, чтобы обеспечить стойкость шифрующего преобразования. Более простая конструкция позволила строго доказать свойство равномерности выборки подключей и возможность построить конкретные гибкие шифры, для которых может быть дано доказательство неэквивалентности всех модификаций и раундов зашифрования к раундам расшифрования. Эти положения косвенно характеризуют также и более сложные построения, для которых непосредственное доказательство является проблематичным (как раз из-за более эффективного механизма преобразования). Таким образом, доказанные результаты дополняют обоснование стойкости программных алгоритмов из [14].

Зависимость алгоритма непосредственного шифрования от секретного ключа обеспечивает существенное повышение стойкости к дифференциальному и линейному криптоанализу при обеспечении целостности шифратора. Но при технологических атаках (инженерном криптоанализе), когда данное условие не выполняется, этот элемент не является достаточным для предотвращения возможности взлома шифра. Чтобы он был эффективным и для таких атак, необходимо обеспечение стойкости к подобным нападениям каждой из модификаций. Например, гибкие и фиксированные

программные алгоритмы из [14] обеспечивают высокую стойкость к атакам на основе случайных аппаратных ошибок, а рассмотренный выше класс шифров, включая гибкие алгоритмы с очень большим числом модификаций, является уязвимым к такой атаке также как и DES, RC5, ГОСТ 28147–89 и многие широко используемые шифры (*см. гл. 6* в книге [14]).

ГЛАВА 2

Математические основы криптографии

2.1. Введение в конечные поля

Группы

Под ***алгебраической системой*** или ***алгебраической структурой*** понимается некоторое множество **S** с одной или несколькими операциями на нем. В элементарной арифметике используются две операции – сложение и умножение, важным свойством которых является ассоциативность. Наиболее изученными алгебраическими системами, имеющими одну ассоциативную операцию являются группы.

Группой <**G**, *> называется некоторое множество **G** с бинарной операцией ‘’*’’ на нем, для которого справедливы следующие условия:

- замкнутость – любой паре $(a, b) \in \mathbf{G}$ ставится в однозначное соответствие третий элемент $c \in \mathbf{G}$, возможно совпадающий с одним из исходных элементов a или b (алгебраическая система <**G**, *> с этим свойством называется ***группоидом***);
- операция ‘’*’’ ассоциативна, т.е. $\forall\ a, b, c \in \mathbf{G}$

$$a*(b*c) = (a*b)*c$$

(алгебраическая система <**G**, *> со свойствами замкнутости и ассоциативности – ***полугруппа***);

- в множестве **G** существует единичный элемент e, такой, что $\forall\ a \in \mathbf{G}$

$$a*e = e*a = a$$

(алгебраическая система <**G**, *>, удовлетворяющая свойствам замкнутости, ассоциативности и имеющая единичный элемент, называется ***моноидом***);

- $\forall\ a \in \mathbf{G}$ существует обратный элемент $a^{-1} \in \mathbf{G}$, такой, что

$$a*a^{-1} = a^{-1}*a = e.$$

Если группа **G** дополнительно удовлетворяет условию коммутативности

- $\forall\ a, b \in \mathbf{G}\ \ a*b = b*a,$

то она называется ***коммутативной*** или ***абелевой*** в честь выдающегося норвежского математика Н. Абеля.

В зависимости от используемых операций и обозначений различают аддитивные и мультипликативные группы, общая характеристика которых представлена в табл. 2.1.

Мультипликативная группа **G** называется ***циклической***, если в ней имеется такой элемент a, что каждый элемент $b \in \mathbf{G}$ является степенью элемента a, т.е. существует целое число k, такое, что $b = a^k$. Элемент a называется образующим элементом мультипликативной группы **G**. Из данного определения следует, что любая циклическая группа коммутативна.

Таблица 2.1

Вид группы	Вид Операции	Результат операции	Единичный элемент	Обратный элемент
Аддитивная	Сложение $+$	Сумма $a+b$	0	$-a$
Мультипликативная	Умножение $\cdot$	Произведение ab	1	a^{-1}

Выполнение операций в группах подчиняется следующим правилам (табл. 2.2).

Таблица 2.2

Мультипликативная группа	Аддитивная группа
$a^{-n}=(a^{-1})^n$	$(-n)a=n(-a)$
$a^m a^n=a^{m+n}$	$ma+na=(m+n)a$
$(a^m)^n=a^{mn}$	$m(na)=(mn)a$

Примеры аддитивных групп:

- множество целых чисел **Z**;
- множество рациональных чисел **Q**;
- множество действительных чисел **R**;
- множество многочленов степени не выше n с целыми коэффициентами.

Примеры мультипликативных групп:

- множество ненулевых рациональных чисел;
- множество корней n-й степени из единицы.

Группа называется ***конечной***, если она состоит из конечного числа элементов. Число элементов конечной группы называется ее ***порядком***. Для обозначения порядка группы используется обозначение $|\mathbf{G}|$ или $\#\mathbf{G}$.

Кольца

Кольцом <**R**, +, ·> называется множество **R** с двумя бинарными операциями + и ·, такими, что

- множество **R** является аддитивной абелевой группой;
- операция умножения ассоциативна, т.е. $\forall\ a, b, c \in \mathbf{R}$

$$a(bc) = (ab)c;$$

- выполняются законы дистрибутивности, т.е. $\forall\ a, b, c \in \mathbf{R}$

$$a(b+c) = ab+ac,\ (b+c)a = ba+ca.$$

***Примерами колец являются*:**

- кольцо целых чисел **Z**;
- кольцо целых чисел по модулю p $\mathbf{Z}/p$ (кольцо вычетов по модулю p, в котором операции + и · являются операциями сложения и умножения по модулю p).

Кольцо называется ***кольцом с единицей***, если оно имеет мультипликативную единицу, т.е. если существует такой элемент $1 \in \mathbf{R}$, что $a1 = 1a = a\ \forall\ a \in \mathbf{R}$.

Кольцо называется ***коммутативным***, если операция умножения коммутативна.

Кольцо называется ***целостным кольцом*** или областью целостности, если оно является коммутативным кольцом с единицей $e \neq 0$, в котором равенство $ab = 0$ влечет за собой $a = 0$ или $b = 0$.

Подмножество **S** кольца **R** называется ***подкольцом*** этого кольца, если оно замкнуто относительно операций сложения и умножения и образует кольцо относительно этих операций.

Пусть **R** — произвольное кольцо. Если существует натуральное число $p \in \{1,2,3,\ldots\}$, такое, что для каждого $r \in \mathbf{R}$ выполняется равенство $pr = 0$, то наименьшее из таких чисел называется ***характеристикой*** кольца **R**. Если таких натуральных чисел не существует, то характеристика кольца равна 0. ***Порядок*** кольца **R** определяется порядком аддитивной группы кольца и, следовательно, равен числу элементов кольца.

Поля

Полем <**F**, +, ·> называется множество **F** с двумя бинарными операциями + и ·, такими, что

- **F** является аддитивной абелевой группой с единичным элементом 0;
- элементы **F**, отличные от 0, образуют мультипликативную абелеву группу, единичным элементом которой является 1;
- операции сложения и умножения связаны законом дистрибутивности;

- для операций сложения и умножения существуют обратные операции: вычитание и деление (кроме деления на нуль).

Примеры полей:

- множество рациональных чисел **Q**;
- множество действительных чисел **R**;
- множество комплексных чисел **C**.

Если поле представляет собой конечное множество, состоящее из q элементов, то такое поле называется ***конечным полем*** или ***полем Галуа*** и обозначается GF(q) или $\mathbf{F}_q$. Конечные поля существуют только тогда, когда число элементов q является простым числом или степенью простого числа: $q = p^m$, где p – простое число, m – натуральное число. При этом p называется ***характеристикой*** конечного поля GF(q), q – ***порядком*** поля, а m – ***степенью*** поля GF(q) над его подполем GF(p). При $m = 1$ поле называется ***простым***, в противном случае поле называется ***расширенным***. Если же p – составное число, то алгебраическая система $<\mathbf{F}, +, \cdot>$, где сложение и умножение модулярные операции, полем не является, эта система образует кольцо, где деление даже на ненулевой элемент возможно не всегда.

В любом поле множество всех элементов образует по операции сложения аддитивную абелеву группу, аналогично множество всех ненулевых элементов поля образует по операции умножения мультипликативную циклическую группу.

Для каждого допустимого значения q существует ровно одно поле, т.е. все конечные поля порядка q изоморфны. Например, если $q = p$ – простое число, то элементами поля являются числа 0, 1, …, p–1, а сложение и умножение являются обычными сложениями и умножениями по модулю p, т.е. GF(p) = **Z**/p. Следовательно, кольцо вычетов по модулю простого числа есть простое поле. Если q является степенью простого числа, то элементами такого поля являются все многочлены степени m–1 и менее, коэффициенты которых лежат в простом поле GF(p). Правила умножения и сложения таких многочленов получаются из обычного умножения и сложения многочленов и последующего приведения результата по модулю некоторого специального многочлена $g(x)$ степени m. Этот многочлен обладает тем свойством, что его нельзя разложить на множители, используя только многочлены с коэффициентами из GF(p). Такие многочлены называются ***неприводимыми*** и по своей сути они аналогичны простым числам. Таблицы неприводимых многочленов над некоторыми конечными полями имеются в работе [33] и др.

Например, многочлен $g(x)=1+x+x^3$ является неприводимым над GF(2) и может использоваться для построения расширенного поля GF(8). Следует заметить, что приведение многочлена по модулю $g(x)$ эквивалентно делению на $g(x)$ и взятию остатка. С этим связано понятие сравнимости

$$a(x) \equiv b(x) \bmod g(x),$$

что интерпретируется как: $a(x)$ сравнимо с $b(x)$ по модулю $g(x)$, и означает, что оба многочлена имеют одинаковые остатки при делении на $g(x)$, или что $a(x)$–$b(x)$ делит-

ся на многочлен $g(x)$. Следовательно, понятие сравнимости многочленов аналогично понятию сравнимости целых чисел.

Мультипликативная группа ненулевых элементов произвольного конечного поля GF(q) является циклической. Образующий элемент α циклической группы называется ***примитивным элементом*** поля GF(q). Все элементы конечного поля можно представить следующим образом:

$$GF(q) = \{0, \alpha, \alpha^2, \ldots, \alpha^{q-2}, \alpha^{q-1} = \alpha^0 = 1\},$$

где α – примитивный элемент поля.

Другой способ выполнения умножения ненулевых элементов GF(q) состоит в том, чтобы найти их логарифмы, сложить их по модулю q–1 и перейти к антилогарифму. Существование логарифмов в конечном поле означает, что имеется представление элементов конечного поля, удобное для выполнения умножения, и другое представление, удобное для выполнения сложения. Наиболее удобный метод состоит, при выполнении сложения, в представлении элемента поля в виде вектора длиной m, компонентами которого являются коэффициенты соответствующего многочлена, и при выполнении умножения, в представлении элемента поля в виде степени примитивного элемента. В табл. 2.3 показана взаимосвязь таких представлений элементов поля GF(8).

Таблица 2.3

0	0 0 0
1	1 0 0
α^1	0 1 0
α^2	0 0 1
α^3	1 1 0
α^4	0 1 1
α^5	1 1 1
α^6	1 0 1

Используя такого рода таблицы, легко производить арифметические операции. Например, пусть требуется вычислить значение $f(\alpha) = (\alpha^2 + \alpha^5)\,\alpha + (1 + \alpha^3)\,\alpha$. С помощью табл. 2.3 получаем

$$\alpha^2+\alpha^5 = (0\ 0\ 1) + (1\ 1\ 1) = (1\ 1\ 0) = \alpha^3,$$

$$1+\alpha^3 = (1\ 0\ 0) + (1\ 1\ 0) = (0\ 1\ 0) = \alpha,$$

следовательно, $f(\alpha) = \alpha^3\alpha + \alpha\alpha = \alpha^4 + \alpha^2 = (0\ 1\ 1) + (0\ 0\ 1) = (0\ 1\ 0) = \alpha$.

Другим подходом выполнения операций в конечных полях является использование ***логарифма Зеча*** $z(n)$, задаваемого равенством

$$\alpha^{z(n)} = 1 + \alpha^n.$$

Используя этот метод, можно выполнять арифметические операции в конечном поле, работая только с логарифмами и не обращаясь к антилогарифмам. В этом случае сумма элементов α^m и α^n может быть вычислена следующим образом:

$$\alpha^m + \alpha^n = \alpha^m (1 + \alpha^{n-m}) = \alpha^{m+z(n-m)}.$$

Для поля GF(8) логарифмы Зеча приведены в табл. 2.4.

Таблица 2.4

n	$-\infty$	0	1	2	3	4	5	6
z(n)	0	$-\infty$	3	6	1	5	4	2

2.2. Элементы теории чисел

2.2.1 Некоторые определения и утверждения

Детальное рассмотрение приводимых здесь фрагментов теории чисел можно найти в работах [1, 4, 6].

Большую роль в теории чисел (и криптографии) играют простые числа. ***Простым числом*** называется число, которое делится без остатка только на единицу и само на себя. Иными словами простым называется число $p \geq 3$, которое не делится без остатка ни на одно из следующих чисел 2, 3, …, $p - 1$. Число 2 также является простым.

Важным является также понятие взаимной простоты двух натуральных чисел. ***Взаимно простыми*** называются два целых положительных числа, наибольший общий делитель которых равен 1.

Сокращение в сравнениях множителей, являющихся взаимно простыми с модулем, основано на следующем утверждении.

Утверждение 2.1

Если НОД(a, n) = 1 и $(a \times b) \equiv (a \times c) \bmod n$, то $b \equiv c \bmod n$.

Доказательство

Из сравнения $(a \times b) \equiv (a \times c) \bmod n$ следует: $ab - ac \equiv 0 \bmod n \Rightarrow a(b - c) = Qn$, где Q – целое положительное число или 0. Если $Q = 0$, то утверждение выполняется. Если $Q \neq 0$, из равенства $a(b - c) = Qn$ следует, что в правой части содержится множитель n, который может содержаться только в значении $b - c$, поскольку a и n являются взаимно простыми числами, т. е. не содержат общих множителей, кроме единицы. Таким образом, $b - c = qn$, где q – целое положительное число, т. е. $b \equiv c \bmod n$.

Пользуясь доказанным выше утверждением докажем следующее.

О существовании обратного элемента

Утверждение 2.2

Для любого целого числа $a > 0$ взаимно простого с модулем n существует обратное по mod n число, обозначаемое знаком a^{-1}, такое, что $a \times a^{-1} \equiv 1 \bmod n$. Число a^{-1} называется мультипликативно обратным по модулю n.

Доказательство

Рассмотрим множество значений $\{1, 2, \ldots, n-1\}$. Умножая каждое из них на a по mod n, получаем множество $\{(a \bmod n), (2a \bmod n), \ldots, ((n-1)a \bmod n)\}$, которое содержит по одному разу числа $1, 2, \ldots, n-1$, т. е. для некоторого значения i выполняется условие $ia \bmod n = 1$. Это вытекает из противоречия, возникающего при предположении о существовании двух одинаковых значений. Пусть, например, $ha \bmod n = ka \bmod n$. Тогда с учетом условия НОД$(a, n) = 1$ из последнего условия получаем $h \equiv k \bmod n \Rightarrow h = k$. Последнее противоречит тому, что мы умножали на a только различные числа. Указанное значение i и является мультипликативно обратным (к числу a) элементом по модулю n.

Следствие 2.1

Если модуль p является простым числом, то для любого числа $0 < a < p$ существует мультипликативно обратный элемент по модулю p.

Можно показать, что если НОД$(a, n) \neq 1$, то не существует числа b, такого, что $ab \equiv 1 \bmod n$. Действительно, пусть НОД$(a, n) = d$, т. е. $a = dq$ и $n = dp$. Умножая на a по модулю n каждое число из множества значений $\{1, 2, \ldots, dp-1\}$, получим следующий ряд остатков $\{(a \bmod n), (2a \bmod n), \ldots, ((dp-1)a \bmod n)\}$. Покажем, что каждый из этих остатков делится без остатка на d, т. е. содержит множитель d. Возьмем любое число b, такое, что $0 < b < dp$, и вычислим остаток от деления ab на n: $r = ab - Qn = dqb - Qdp = d(qb - Qp)$, где Q – некоторое натуральное число. Из равенства $r = d(qb - Qp)$ следует, что остаток r делится на d. (Одновременно мы доказали утверждение о делимости остатка, которое сформулировано ниже.) Таким образом, в указанном ряде $n-1$ остатков содержатся только остатки, делящиеся нацело на d. Поскольку среди них нет единицы, то это означает, что не существует числа b, при котором $ab \equiv 1 \bmod n$. Что мы и хотели показать.

О делимости остатка

Утверждение 2.3

Пусть для целых положительных чисел a и b имеем $a > b$ и НОД$(a, b) = d$, тогда для остатка r от деления a на b выполняется равенство НОД$(b, r) = d$.

Данное утверждение лежит в основе алгоритма Евклида, позволяющего быстро вычислять наибольший общий делитель двух целых положительных чисел.

Теорема Ферма

Теорема Ферма утверждает следующее: для любого простого числа p и любого положительного числа a, которое не делится на p, выполняется сравнение

$$a^{p-1} \equiv 1 \bmod p.$$

Доказательство

Рассмотрим множество чисел $\{1,2, \ldots, p-1\}$, которое обозначим как $\boldsymbol{Z}_p$. Если все элементы $\boldsymbol{Z}_p$ умножить на a по модулю p, то, используя утверждение 2.1, легко показать, что в результате получим некоторую перестановку элементов $\boldsymbol{Z}_p$. Иными словами в наборе $\{a \bmod p, 2a \bmod p, ..., (p-1)a \bmod p\}$ содержатся $(p-1)$ различных чисел, т. е. каждое из чисел 1, 2, ..., $p-1$ встречается ровно по одному разу. Перемножая числа a, $2a$, ..., $(p-1)a$, получаем:

$$a \times 2a \times \ldots \times (p-1)a = (p-1)!\, a^{p-1}.$$

Поделив на p левую и правую части последнего соотношения, получим

$$[a \times 2a \times \ldots \times (p-1)a] \bmod p \equiv (p-1)!\, a^{p-1} \bmod p;$$

$$[(a \bmod p) \times (2a \bmod p) \times \ldots \times ((p-1)a \bmod p)] \bmod p \equiv (p-1)!\, a^{p-1} \bmod p.$$

Поскольку произведение не изменяется от перестановки мест сомножителей, то левую часть последнего сравнения можно представить в виде

$$[(a \bmod p) \times (2a \bmod p) \times \ldots \times ((p-1)a \bmod p)] =$$

$$= 1 \times 2 \times \ldots \times (p-1) = (p-1)!$$

Из последних двух соотношений следует

$$(p-1)! \equiv (p-1)!\, a^{p-1} \bmod p.$$

Числа $(p-1)!$ и p являются взаимно простыми, поэтому в последнем выражении на основании утверждения 2.1 можно левую и правую части сократить на $(p-1)!$, откуда вытекает сравнение

$$a^{p-1} \equiv 1 \bmod p.$$

Теперь не представляет труда вывести следующие формулы:

$$a^p \equiv a \bmod p \text{ и}$$

$$b \equiv c \bmod (p-1) \Rightarrow a^b \equiv a^c \bmod p.$$

2.2.2. Функция Эйлера

В обобщении теоремы Ферма используется понятие функции Эйлера, которая часто будет нам встречаться в дальнейшем. Функция Эйлера играет важную роль в теории чисел. Она обозначается символом $\phi(n)$ и определяется как число положительных целых чисел, которые меньше n и являются взаимно простыми с n.

Очевидно, что для простого p имеем $\phi(p) = p - 1$. Используя утверждение о мультипликативности функции Эйлера, для числа $n = pq$, являющегося произведением двух простых числа p и q, можно легко получить

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p - 1) \times (q - 1).$$

Эту частную формулу легко получить и без использования свойства мультипликативности функции Эйлера. Действительно, рассмотрим множество чисел $\{1, 2, \ldots, (pq - 1)\}$. Нетрудно заметить, что все числа, которые не превышают значение pq и не являются взаимно простыми с $pq = n$, составляют следующие два множества $\{p, 2p, \ldots,(q - 1)p\}$ и $\{q, 2q, \ldots, (p - 1)q\}$. В первом содержится $q - 1$, а во втором $p - 1$ различных чисел. Вычитая из $n - 1$ значения $q - 1$ и $p - 1$, получим

$$\phi(n) = pq - 1 - (q - 1) - (p - 1) = pq - (p + q) + 1 = (p - 1) \times (q - 1).$$

Однако в общем случае, когда в каноническом разложении числа n содержится сравнительно большое число простых сомножителей и их степеней, при вычислении функции Эйлера используется то, что она является мультипликативной функцией, т. е. для двух взаимно простых чисел a и b выполняется соотношение $\phi(ab) = \phi(a) \times \phi(b)$.

Поскольку в каноническом разложении произвольного числа n содержатся только взаимно простые сомножители вида p^s, где $s \geq 1$, то для вычисления функции Эйлера достаточно научиться вычислять функцию Эйлера от чисел вида p^s и уметь разлагать число n на простые множители.

Пусть дано число p^s. Рассмотрим множество чисел

$$\{1, 2, \ldots, p, \ldots, 2p, \ldots, 3p, \ldots, p^{s-1}p\}.$$

Нетрудно видеть, что все числа, входящие в это множество и не являющиеся взаимно простыми с p^s, составляют подмножество $\{p, 2p, 3p, \ldots, p^{s-1}p\}$ из p^{s-1} чисел. Отсюда получаем

$$\phi(p^s) = p^s - p^{s-1} = p^{s-1}(p - 1).$$

Используя эту формулу и свойство мультипликативности, можно легко вычислить функцию Эйлера от произвольного заданного числа, если нам удастся разложить его на множители.

Теорема Эйлера

Теорема Эйлера, являющаяся обобщением теоремы Ферма, утверждает, что для любых взаимно простых чисел a и n выполняется сравнение

$$a^{\phi(n)} \equiv 1 \bmod n.$$

Доказательство

Доказательство аналогично доказательству теоремы Ферма. Поскольку для простого n имеем $\phi(n) = (n-1)$, то в этом случае сравнение $a^{\phi(n)} \equiv 1 \bmod n$ непосредственно следует из теоремы Ферма (является просто другой формой записи последней). Доказательство для произвольного n опирается на определение функции Эйлера: $\phi(n)$ равно числу положительных целых чисел, меньших n и взаимно простых с n. Множество таких целых чисел включает $\phi(n)$ значений, которые можно пронумеровать следующим образом:

$$\Phi = \{x_1, x_2, \ldots, x_{\phi(n)}\}.$$

Умножая каждый элемент этого множества на a по модулю n, получим множество

$$\Phi' = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots, (ax_{\phi(n)} \bmod n)\}.$$

Покажем, что последние два множества включают одни и те же элементы. Действительно, для всех значений $i = 1, 2, \ldots, \phi(n)$ числа a и x_i является взаимно простыми с n, поэтому число ax_i тоже является взаимно простым с n. Следовательно, все элементы S являются целыми числами, меньшими n (умножение мы выполняли по модулю n) и взаимно простыми с n. Во множестве Φ' нет повторений, так как если $(a \times b) \equiv (a \times c) \bmod n$, то $b \equiv c \bmod n$, поскольку по условию доказываемой теоремы a и n являются взаимно простыми числами. Действительно, из условия $ax_i \bmod n = ax_j \bmod n$, следует $x_i = x_j$, что противоречит тому, что в множестве Φ все числа различны. Перемножая все элементы множества Φ', а затем все элементы множества Φ, получаем равенство

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) \equiv \prod_{i=1}^{\phi(n)} x_i,$$

из которого следуют сравнения

$$\prod_{i=1}^{\phi(n)} ax_i \equiv \prod_{i=1}^{\phi(n)} x_i \,(\bmod n),$$

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i \,(\bmod n),$$

$$a^{\phi(n)} \equiv 1 \;\; (\bmod n).$$

Из последнего сравнения вытекают следующие соотношения:

$$a^{\phi(n)+1} \equiv a \bmod n \quad \text{и}$$

$$b \equiv c \bmod \phi(n) \Rightarrow a^b \equiv a^c \bmod n.$$

2.2.3. Алгоритм Евклида

Пусть **MOD**(a, b) есть операция взятия остатка от деления a на b, а **QUO**(a, b) есть частное от деления a на b. Алгоритм Евклида позволяет без разложения двух целых чисел на множители находить их наибольший общий делитель. В данном алгоритме используется следующее утверждение: если $a = bq + r$, где $b \neq 0$, и число d делит a и b, то оно делит и r, т. е. имеем $d \mid (a - bq)$. Это утверждение верно для любого делителя, включая наибольший общий делитель d = НОД(a, b). Отсюда следует, что НОД(a,b) = НОД [b, **MOD** (a, b)].

Для всякого a имеем также НОД(a,0) = $|a|$. Пусть заданы целое число a и ненулевое целое число b. Алгоритм Евклида предполагает выполнение следующей последовательности делений, где принято обозначение $a_0 = a$ и $a_1 = b$:

$$\begin{aligned}
&a_0 = a_1 q_1 + a_2, && 0 < a_2 < |a_1|,\\
&a_1 = a_2 q_2 + a_3, && 0 < a_3 < a_2,\\
&\ldots\\
&a_{k-2} = a_{k-1} q_{k-1} + a_k, && 0 < a_k < a_{k-1},\\
&a_{k-1} = a_k q_k + 0.
\end{aligned}$$

Процесс деления имеет конечное число шагов, поскольку остатки убывают по абсолютной величине $|a_1| > a_2 > a_3 > \ldots > 0$ (значение a_1 может быть отрицательным, поэтому оно взято по абсолютной величине; остальные остатки положительны). Значение a_k является наибольшим общим делителем. С учетом указанного выше утверждения имеем

$$\text{НОД}(a_0, a_1) = \text{НОД}(a_1, a_2) = \ldots = \text{НОД}(a_k, 0) = a_k,$$

т. е. на самом деле a_k является наибольшим общим делителем чисел a и b. Одновременно мы показали, что приведенный ниже алгоритм работает правильно. Обозначим операцию присвоения следующим образом: $x := y$ означает присвоение переменной x значения y, а $(x,y) := (x_1,y_1)$ означает выполнение операций $x := x_1$ и $y := y_1$.

Алгоритм Евклида (нахождение НОД(a, b))

ВХОД: a и $b \neq 0$.

1. [Инициализация] $(a_0, a_1) := (a, b)$.
2. [Основной цикл] Пока $a_1 \neq 0$ выполнять $(a_0, a_1) := [a_1,$ **MOD** $(a_0, a_1)]$
3. Вернуть $d := a_0$.

ВЫХОД: d = НОД(a, b).

2.2.4. Расширенный алгоритм Евклида

В любой строке описанного выше алгоритма Евклида каждый остаток представляет собой линейное представление делимого и делителя. Легко видеть, что, начиная с одного из ненулевых остатков (например, последнего, т. е. начиная с a_k) и выражая остатки, полученные на последующих шагах алгоритма Евклида, через остатки, полученные на предыдущих шагах, можно представить значение a_i (соответственно, a_k) в следующем виде $a_i = ax' + by'$ (соответственно, $a_k = ax + by$), где x и y некоторые целочисленные коэффициенты. Расширенный алгоритм Евклида позволяет вычислить значения коэффициентов x и y в указанном линейном представлении НОД(a, b), т. е. в выражении $a_k = ax + by$. Работа расширенного алгоритма Евклида организуется так, что значения x' и y' вычисляются в серии шагов, в каждом из которых мы выражаем a_i в виде указанного линейного представления.

Каждый остаток, вычисленный в процессе работы алгоритма Евклида, можно представить в виде $ax_i + by_i$.Рассмотрим следующую последовательность такого представления остатков:

$$
\begin{aligned}
&a_0 = a, && a_0 = ax_0 + by_0,\\
&a_1 = b, && a_1 = ax_1 + by_1,\\
&a_2 = a_0 - a_1\, q_1, && a_2 = ax_2 + by_2,\\
&\dots\\
&a_i = a_{i-2} - a_{i-1}\, q_{i-1}, && a_i = ax_i + by_i,\\
&\dots\\
&a_k = a_{k-2} - a_{k-1}\, q_{k-1}, && a_k = ax_k + by_k,\\
&0 = a_{k-1} - a_k\, q_k, && 0 = ax_{k+1} + by_{k+1}.
\end{aligned}
$$

В левом столбце фактически приведена последовательность делений, полученная в алгоритме Евклида, но записанная в виде выражения для вычисления остатков. В правом столбце каждый остаток выражен в интересующем нас виде $ax_i + by_i$. Пока мы не знаем коэффициентов. Нашей целью является вычисление x_i и y_i для любого i, а следовательно и для $i = k$. Очевидно, что x_0=1, y_0=0 и x_1=0, y_1=1. Сравнивая обе части на i-м шаге, получаем

$$a_i = ax_i+by_i = a_{i-2}-a_{i-1}q_{i-1} = (ax_{i-2}+by_{i-2}) - (ax_{i-1}+by_{i-1})q_{i-1} = a(x_{i-2}-x_{i-1}q_{i-1}) + b(y_{i-2}-y_{i-1}q_{i-1}),$$

откуда получается следующая индуктивная процедура вычисления x_i и y_i:

$$
\begin{aligned}
&q_{i-1} = \mathbf{QUO}(a_{i-2}, a_{i-1})\\
&a_i = a_{i-2} - a_{i-1}q_{i-1},\\
&x_i = x_{i-2} - x_{i-1}q_{i-1},\\
&y_i = y_{i-2} - y_{i-1}q_{i-1}.
\end{aligned}
$$

Отметим, что значение a_i может быть вычислено как MOD(a_{i-2}, a_{i-1}), но достоинство приведенного выше выражения заключается в том, что a_i вычисляется аналогично вычислению коэффициентов x_i и y_i. Этот факт используется в следующем алгоритме.

Расширенный алгоритм Евклида

ВХОД: a и $b \neq 0$

1. [Инициализация] $(a_0, a_1) := (a, b)$; $(x_0, x_1) := (1, 0)$; $(y_0, y_1) := (0, 1)$.
2. [Основной цикл] Пока $a_1 \neq 0$ выполнять

$$\{q := \mathbf{QUO}(a_0, a_1);$$
$$(a_0, a_1) := (a_1, a_0 - a_1 q);$$
$$(x_0, x_1) := (x_1, x_0 - x_1 q);$$
$$(y_0, y_1) := (y_1, y_0 - y_1 q)\}.$$

3. Вернуть $(d, x, y) := (a_0, x_0, y_0)$.

ВЫХОД: d, x, y, такие, что d = НОД(a, b) = $ax + by$.

2.2.5. Показатели и первообразные корни

Для взаимно простого с модулем n числа a, согласно теореме Эйлера, существует некоторое положительное $\gamma = \phi(n)$, для которого выполняется условие $a^\gamma \equiv 1 \pmod{n}$. В случае простого модуля p имеем $\gamma = p - 1$. Представляет интерес найти наименьшее из чисел γ, для которых выполняется указанное условие.

Определение

Пусть НОД(a, n) = 1. Наименьшее из чисел γ, для которых выполняется сравнение $a^\gamma \equiv 1 \pmod{n}$, называется ***показателем***, которому число a принадлежит по модулю n.

Утверждение 2.4

Если a по модулю n принадлежит показателю δ, то числа $a^0, a^1, \ldots, a^{\delta-1}$ по модулю n несравнимы.

Доказательство

Пусть $a^h \equiv a^k \pmod{n}$, $0 \le h < l < \delta$. Тогда получим $a^{h-k} \equiv 1 \pmod{n}$, где $0 < h - k < \delta$. Но по определению δ есть наименьшее из чисел γ, для которых выполняется сравнение $a^\gamma \equiv 1 \pmod{n}$. Противоречие доказывает утверждение.

Утверждение 2.5

а) Если a по модулю n принадлежит показателю δ, то $a^{\gamma} \equiv a^{\gamma'} \pmod{n}$ тогда и только тогда, когда $\gamma \equiv \gamma' \pmod{\delta}$.

б) Если $\gamma = 0$, то имеем сравнение $a^{\gamma} \equiv 1 \pmod{n}$, которое выполняется тогда и только тогда, когда γ делится на показатель δ.

Доказательство

Пусть r и r' есть наименьшие неотрицательные вычеты чисел γ и γ' по модулю δ. Тогда при некоторых q и q' имеем: $\gamma = \delta q + r$, $\gamma' = \delta q' + r'$. Поскольку $a^{\delta} \equiv 1 \pmod{n}$, то получаем $a^{\gamma} = (a^{\delta})^{q} a^{r} \equiv a^{r} \pmod{n}$ и $a^{\gamma'} = (a^{\delta})^{q'} a^{r'} \equiv a^{r'} \pmod{n}$.

Следовательно, $a^{\gamma} \equiv a^{\gamma'} \pmod{n}$ тогда и только тогда, когда $a^{r} \equiv a^{r'} \pmod{n}$, т.е. тогда и только тогда, когда $r = r'$. (Действительно, $r < \delta$ и $r' < \delta$ и из $a^{r} \equiv a^{r'} \pmod{n}$ следует $r = r'$. В противном случае имеем противоречие: $a^{|r - r'|} = 1 \pmod{n}$ при $0 < |r - r'| < \delta$.)

Следствие 2.2

Показатели, которым числа a принадлежат по модулю n, являются делителями $\varphi(n)$.

Действительно, пусть a по модулю n принадлежит показателю δ. Из $a^{\varphi(m)} \equiv 1 \pmod{n}$ следует, что $\varphi(n)$ делится на δ. Наибольший из этих делителей есть само $\varphi(n)$.

Первообразные корни

Интересен вопрос о существовании чисел, принадлежащих показателю $\varphi(n)$. Такие числа существуют и называются первообразными корнями по модулю n.

Утверждение 2.6

Если число a по модулю n принадлежит показателю $\varepsilon'\varepsilon$, то $a^{\varepsilon'}$ принадлежит показателю ε.

Доказательство

Действительно, путь $a^{\varepsilon'}$ принадлежит показателю δ. Тогда $(a^{\varepsilon'})^{\delta} \equiv 1 \pmod{n}$, т. е. $a^{\varepsilon'\delta} \equiv 1 \pmod{n}$, из чего следует, что $\varepsilon'\delta$ делится на $\varepsilon'\varepsilon$, т.е. δ делится на ε. С другой стороны, $a^{\varepsilon'\varepsilon} \equiv 1 \pmod{n}$, откуда $(a^{\varepsilon'})^{\varepsilon} \equiv 1 \pmod{n}$, откуда по утверждению 2.5 следует, что ε' делится на δ. Таким образом, $\varepsilon' | \delta$ и $\delta | \varepsilon'$, поэтому $\delta = b$.

Утверждение 2.7

Если a по модулю n принадлежит показателю u, а b – показателю v, причем НОД $(u, v) = 1$, то *ab принадлежит показателю uv.*

Доказательство

Действительно, пусть ab принадлежит показателю δ. Тогда $(ab)^{\delta} \equiv 1 \pmod{n} \Rightarrow (ab)^{v\delta} \equiv 1 \pmod{n} \Rightarrow a^{v\delta}b^{v\delta} \equiv 1 \pmod{n} \Rightarrow a^{v\delta} \equiv 1 \pmod{n}$, откуда следует, что $v\delta$ делится на u. Но поскольку НОД $(u, v) = 1$, то δ делится на u. Рассуждая аналогичным образом, получим, что δ делится на v. Ввиду того, что $u|\delta$, $v|\delta$ и НОД$(a, b) = 1$, то $(uv)|\delta$, т. е. δ делится и на uv. С другой стороны, из $a^u b^v \equiv (a^u)^v (b^v)^u \equiv (ab)^{uv} \equiv 1 \pmod{n}$ следует, что uv делится на δ. Поскольку $(uv)|\delta$ и $\delta|(uv)$, то $\delta = uv$.

В теории чисел доказываются теоремы о существовании первообразных корней по модулю p, по модулю p^k и по модулю $2p^k$, где p – простое нечетное число и k – произвольное положительное целое число.

Утверждение 2.8

Существуют первообразные корни по модулю p, где p – простое нечетное число.

Доказательство

Пусть $\delta_1, \delta_2, \ldots, \delta_k$ – все различные показатели, которым по модулю p принадлежат числа $1, 2, \ldots, (p-1)$. Пусть наименьшее общее кратное чисел $\delta_1, \delta_2, \ldots, \delta_k$ есть НОК$(\delta_1, \delta_2, \ldots, \delta_k) = \tau$, каноническое разложение которого имеет вид $\tau = q_1^{\alpha_1} q_1^{\alpha_2} \ldots q_k^{\alpha_k}$. Каждый множитель $q_s^{\alpha_s}$ этого разложения делит по меньшей мере один из показателей $\delta_1, \delta_2, \ldots, \delta_k$, например число δ_j (это как раз тот показатель, который содержит множитель $q_s^{\alpha_s}$ и вносит его в разложение τ). Показатель δ_j можно записать в виде $\delta_j = zq_s^{\alpha_s}$. Пусть a_j – одно из чисел ряда $1, 2, \ldots, p-1$, принадлежащих показателю δ_j. Согласно утверждению 2.6 число $h_j = a_j^z$ принадлежит показателю $q_s^{\alpha_s}$. Поскольку показатели $q_1^{\alpha_1}, q_1^{\alpha_2}, \ldots, q_k^{\alpha_k}$ являются попарно взаимно простыми, то согласно утверждению 2.7 произведение $g = h_1 h_2 \ldots h_k$ принадлежит показателю $q_1^{\alpha_1} q_1^{\alpha_2} \ldots q_k^{\alpha_k} = \tau$. В соответствии со следствием к утверждению 2.5 показатель τ является делителем числа $p-1$: $\tau|(p-1)$.

Поскольку показатели $\delta_1, \delta_2, \ldots, \delta_k$, к каждому из которых относится хотя бы одно из чисел ряда $\{1, 2, \ldots, p-1\}$, делят τ, то все значения $1, 2, \ldots, (p-1)$ являются решениями сравнения $x^{\tau} \equiv 1 \bmod p$ (см. утверждение 2.5). В теории чисел доказывается, что для простого p сравнение $a_n x^n + a_{n-1} x^{n-1} + \ \ldots \ + a_1 x + a_0 = 0 \bmod p$ имеет не более n решений, если хотя бы один из коэффициентов $a_1, a_2, \ldots, a_n$ не кратен p. Из этого утверждения следует, что $p - 1 \le \tau$. Поскольку $\tau|(p-1)$ и $p - 1 \le \tau$, то $\tau = p - 1$. Следовательно, g является первообразным корнем по модулю p.

Таким образом, существуют первообразные корни по модулю простого числа. Этот факт учитывается при рассмотрении метода открытого распределения ключей Диффи-Хеллмана. Представляют интерес также следующие утверждения о существовании первообразных корней по модулям p^{α} и $2p^{\alpha}$, где p – нечетное простое число, которые мы приводим без доказательства.

Утверждение 2.9

Пусть g – первообразный корень по модулю простого числа p. Можно указать t с условием, что u, определяемое равенством $(g + pt)^{p-1} = 1 + pu$, не делится на p. Соответствующее $g + pt$ будет первообразным корнем по модулю p^{α} при любом $\alpha > 1$.

Утверждение 2.10

Пусть $\alpha \geq 1$ и g_1 – первообразный корень по модулю p^{α}, где p – нечетное простое число. Нечетное из чисел g' и $g' + p^{\alpha}$ будет первообразным корнем по модулю $2p^{\alpha}$.

Представляет интерес задача разыскивания первообразных корней по модулю p. Эта задача решается методом, использующим следующее утверждение.

Утверждение 2.11

Пусть $q_1, q_2, \ldots, q_k$ – различные простые делители функции Эйлера $\varphi(n)$ от числа n. Для того чтобы число g, взаимно простое с n, было первообразным корнем по модулю n, необходимо и достаточно, чтобы это g не удовлетворяло ни одному из сравнений:

$$g^{c/q_1} \equiv 1 \pmod{n},\ g^{c/q_2} \equiv 1 \pmod{n},\ \ldots,\ g^{c/q_k} \equiv 1 \pmod{n}.$$

Индексы по модулям p^{α} и $2p^{\alpha}$

По отношению к первообразным корням g вводится понятие индекса (при основании g) по модулю.

Утверждение 2.12

Пусть p – простое нечетное, $\alpha \geq 1$; n – одно из чисел p^{α} и $2p^{\alpha}$; $c = \varphi(n)$, g – первообразный корень по модулю n. Если γ пробегает наименьшие неотрицательные вычеты $\gamma = 0, 1, \ldots, c - 1$ по модулю c, то g^{γ} пробегает приведенную систему вычетов по модулю n.

Действительно, g относится к показателю c и g^{γ} пробегает c чисел, взаимно простых с n, которые являются попарно несравнимыми по модулю n.

Для чисел a, взаимно простых с n, рассматривается понятие об индексе (дискретном логарифме), представляющее аналогию понятию о логарифме. Если $a \equiv g^{\gamma} \pmod{n}$ (предполагается, что $\gamma \geq 0$), то γ называется индексом числа a по модулю n при основании g и обозначается символом $\gamma = \mathrm{ind}_g a$ (или просто $\gamma = \mathrm{ind}\, a$). Из утверждения 2.12 следует, что всякое a, взаимно простое с n, имеет некоторый единственный индекс среди чисел ряда $\gamma = 0, 1, \ldots, c - 1$. Зная γ', такое, что $\gamma' = \mathrm{ind}_g a$, мы можем указать все индексы числа a: это будут все неотрицательные числа класса $\gamma \equiv \gamma' \pmod{c}$. Действительно, имеем $a \equiv g^{\gamma} \pmod{n}$ и $a \equiv g^{\gamma'} \pmod{n}$, поэтому $g^{\gamma - \gamma'} \equiv 1 \pmod{n}$ и, поскольку g относится к показателю $c = \varphi(n)$, то $c | (\gamma - \gamma')$, т. е. $\gamma \equiv \gamma' \pmod{c}$.

Из определения индекса следует непосредственно, что числа с данным индексом γ образуют класс чисел по модулю n.

Рассмотрим следующие свойства индексов:

$$\mathrm{ind}_g\, ab\ldots l \equiv \mathrm{ind}_g\, a + \mathrm{ind}_g\, b + \ldots + \mathrm{ind}_g\, l \pmod{c}$$

и, в частности,

$$\mathrm{ind}_g\, a^n \equiv n\, \mathrm{ind}_g\, a \pmod{c}.$$

Действительно, перемножая сравнения $a \equiv g^{\mathrm{ind}\, a} \pmod{m}$, $b \equiv g^{\mathrm{ind}\, b} \pmod{m}$, …, $l \equiv g^{\mathrm{ind}\, l} \pmod{m}$, находим $ab\ldots l \equiv g^{\mathrm{ind}\, a + \mathrm{ind}\, b + \ldots + \mathrm{ind}\, l} \pmod{m}$. Следовательно, сумма $\mathrm{ind}_g\, a + \mathrm{ind}_g\, b + \ldots + \mathrm{ind}_g\, l$ является один из индексов произведения $ab\ldots l$.

2.3. Булевы функции и свойства криптографических примитивов

Безопасность симметричных блочных алгоритмов шифрования зависит от свойств булевых функций (БФ), реализующих различные криптографические примитивы, входящие в их состав.

Преобразование информации, осуществляемое криптографическими примитивами, можно формализовать в виде отображения некоторого пространства $\mathrm{GF}(2)^n$ n-мерных векторов над полем GF(2) $X = (x_1, x_2, \ldots, x_n)$ в другое пространство $\mathrm{GF}(2)^m$ m-мерных двоичных векторов $Y = (y_1, y_2, \ldots, y_m)$, где для любого $i \in \{1, \ldots, n\}$ и любого $j \in \{1, ..., m\}$, $x_i \in \mathrm{GF}(2)$, $y_j \in \mathrm{GF}(2)$. Отображения такого рода будем задавать в виде векторной булевой функции (ВБФ) $Y = \varphi(X)$: $\mathrm{GF}(2)^n \to \mathrm{GF}(2)^m$, которая является объединением компонентных БФ $f_i(X)$, осуществляющих отображение $\mathrm{GF}(2)^n \to \mathrm{GF}(2)$, то есть $\varphi(X) = \{f_1(X), f_2(X), \ldots, f_m(X)\}$.

Для описания БФ будем использовать их представление в виде таблицы истинности и в виде алгебраической нормальной формы (АНФ). АНФ предполагает следующее описание БФ:

$$\begin{aligned} f(X) = a_0 \oplus \sum_{1 \le i \le n} a_i x_i \oplus \sum_{1 \le i_1 < i_2 \le n} a_{i_1 i_2} x_{i_1} x_{i_2} \oplus \ldots \oplus \\ \sum_{1 \le i_1 < \ldots < i_k \le n} a_{i_1 \ldots i_k} x_{i_1} \ldots x_{i_k} \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n \end{aligned} \qquad (2.1)$$

где $X \in \mathrm{GF}(2)^n$, а все коэффициенты $a \in \mathrm{GF}(2)$. Таким образом, АНФ БФ над полем $\mathrm{GF}(2)^n$ представляет собой сумму взятых с определенными коэффициентами всевозможных произведений переменных. Количество перемножаемых переменных в крайнем правом элементе АНФ является алгебраической степенью нелинейности БФ $f(X)$ и обозначается как $deg(f)$.

2.3.1. Преобразование Уолша-Адамара

При проведении дифференциального и линейного криптоанализа, а также исследовании корреляционных свойств булевых функций (БФ), описывающих примитивы блочных алгоритмов шифрования, основным аппаратом анализа является преобразование Уолша-Адамара – разновидность дискретного преобразования Фурье над конечным полем Галуа. Данное преобразование позволяет непосредственно оценить такие показатели качества БФ, как:

- сбалансированность;
- нелинейность;
- корреляционная иммунность.

Кроме того, с помощью преобразования Уолша-Адамара можно получить оценки автокорреляционных свойств БФ и различных показателей распространения изменений.

Можно ожидать, что БФ не могут удовлетворять всем показателям одновременно, поэтому возникает задача некоторого компромисса (оптимизации) в выборе БФ, удовлетворяющих тем или иным показателям.

Под ***преобразованием Уолша-Адамара*** (ПУА) функции $f(X)$, $X \in \mathrm{GF}(2)^n$, относительно вектора $\alpha \in \mathrm{GF}(2)^n$ понимается линейное преобразование $\mathrm{GF}(2)^n \to \mathbf{Z}$, принимающее значение в области действительных чисел и имеющее следующий вид [31, 35, 100]:

$$U_\alpha(f) = \sum_{X \in \mathrm{GF}(2)^n} f(X)(-1)^{\langle \alpha, X \rangle} \qquad (2.2)$$

где знак "$\langle\ \rangle$" обозначает скалярное произведение двух векторов.

Для сопряженной функции $\hat{f}(X) = (-1)^{f(X)}$, осуществляющей отображение из $\mathrm{GF}(2)^n$ в множество $\{-1, 1\}$, ПУА преобразуется к следующему виду:

$$\hat{U}_\alpha(f) = \sum_{X \in \mathrm{GF}(2)^n} (-1)^{f(X)}(-1)^{\langle \alpha, X \rangle} = \sum_{X \in \mathrm{GF}(2)^n} (-1)^{f(X) \oplus \langle \alpha, X \rangle} \qquad (2.3)$$

Очевидно, что при ПУА БФ $f(X)$, данная функция и функция $\langle \alpha, X \rangle = \alpha_1 x_1 \oplus \ldots \oplus \alpha_n x_n$, принимают действительные значения в поле GF(2), а при преобразовании сопряженной функции $\hat{f}(X)$ действительные значения принимает функция $f(X) \oplus \langle \alpha, X \rangle$. Исходя из взаимосвязи БФ $f(X)$ и сопряженной ей функции

$\hat{f}(X)$ следует, что $U_\alpha(f) = \left\langle S(f),\ \hat{S}(\langle \alpha, X \rangle) \right\rangle$ и $\hat{U}_\alpha(f) = \left\langle \hat{S}(f),\ \hat{S}(\langle \alpha, X \rangle) \right\rangle$, где скалярное произведение характеристических последовательностей $\langle S',\ S'' \rangle = \sum_{i=1}^{n} s'_i\, s''_i$ и принимает значения из **Z**, а сами преобразования связаны соотношениями [31, 115, 125]:

$$\hat{U}_\alpha(f) = 2^n \delta(\alpha) - 2U_\alpha(f), \tag{2.4}$$

$$U_\alpha(f) = 2^{n-1}\delta(\alpha) - \frac{\hat{U}_\alpha(f)}{2}, \tag{2.5}$$

где $\delta(\alpha) = 1$ для $\alpha = (0, \ldots, 0)$ и $\delta(\alpha) = 0$ для $\alpha \neq (0, \ldots, 0)$.

Обратное преобразование Уолша-Адамара (ОПУА) задается выражениями:

$$f(x) = \frac{1}{2^n} \sum_{\alpha \in GF(2)^n} U_\alpha(f)(-1)^{\langle \alpha, X \rangle},\ \hat{f}(x) = \frac{1}{2^n} \sum_{\alpha \in GF(2)^n} \hat{U}_\alpha(f)(-1)^{\langle \alpha, X \rangle} \tag{2.6}$$

Если 2^n значений таблицы истинности $S(f)$ БФ $f(X)$ и 2^n значений действительной функции $U_\alpha(f)$ записать в виде вектор-столбцов $[S(f)]$ и $[U_\alpha(f)]$, то линейное преобразование Уолша-Адамара задается матрицей Адамара $\mathbf{H}_n$ в следующем виде:

$$[U_\alpha(f)] = \mathbf{H}_n[S(f)]. \tag{2.7}$$

Аналогично $[\hat{U}_\alpha(f)] = \mathbf{H}_n[\hat{S}(f)]$.

Матрица $\mathbf{H}_n$ формируется итеративно:

$$\mathbf{H}_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \mathbf{H}_{n-1} = \begin{bmatrix} \mathbf{H}_{n-1} & \mathbf{H}_{n-1} \\ \mathbf{H}_{n-1} & -\mathbf{H}_{n-1} \end{bmatrix},\ \mathbf{H}_0 = [1], \tag{2.8}$$

где $\otimes$ – означает кронекеровское произведение матриц.

Кронекеровское произведение матрицы **A** размера $m \times n$ и матрицы B размера $s \times t$ – это матрица размера $ms \times nt$ задаваемая как

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \ldots & a_{mn}\mathbf{B} \end{bmatrix}, \tag{2.9}$$

где a_{ij} – элемент i-ой строки и j-го столбца матрицы **A**.

Матрицей ***Адамара*** называется матрица $\mathbf{H}$ с элементами из множества {–1, 1}, если справедливо следующее равенство $\mathbf{H}\mathbf{H}^{\mathrm{T}} = n\mathbf{I}_n$, где $\mathbf{H}^{\mathrm{T}}$ – транспонированная матрица $\mathbf{H}$, $\mathbf{I}_n$ – единичная матрица порядка n [32].

Основными свойствами матриц Адамара являются:

1. $\forall\ s, t \,|\, (s+t) = n\ \ \mathbf{H}_n = \mathbf{H}_s \otimes \mathbf{H}_t$;
2. $\mathbf{H}_n^2 = 2^n \mathbf{I}_n$;
3. Если записать матрицу с помощью ее строк h_i

$$\mathbf{H}_n = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{2^n-1} \end{bmatrix},$$

то каждая строка h_i матрицы $\mathbf{H}_n$ является характеристической последовательностью линейной функции $\langle \alpha_i,\ X \rangle$, где вектор $\alpha_i \in \mathrm{GF}(2)^n$, а его целочисленное представление равно i. В свою очередь каждая характеристическая последовательность $\hat{S}(f_{лин})$ линейной функции от n переменных является строкой матрицы $\mathbf{H}_n$. Откуда следует, что строки матриц $\mathbf{H}_n$ и $\overline{\mathbf{H}_\mathbf{n}}$ охватывают последовательности всех аффинных функций над $\mathrm{GF}(2)^n$. Здесь $\overline{\mathbf{H}_\mathbf{n}}$ – матрица, все элементы которой являются инверсными по отношению к элементам матрицы $\mathbf{H}_n$.

В соответствии со свойством 2 обратное ПУА задается соотношением:

$$[S(f)] = 2^{-n}\mathbf{H}_n[U_\alpha(f)],\ [\hat{S}(f)] = 2^{-n}\mathbf{H}_n[\hat{U}_\alpha(f)].$$

Алгоритм преобразования Уолша-Адамара имеет сложность $O(n2^n)$ операций.

2.3.2. Сбалансированность БФ

Во избежании прямых статистических атак на примитивы криптоалгоритма необходимо, чтобы выполнялись следующие условия:

- все компонентные БФ, реализующие преобразование, должны быть сбалансированы;
- преобразование в целом должно быть регулярным.

БФ $f(X)$, $X \in \mathrm{GF}(2)^n$ называется ***сбалансированной***, если количество единиц в ее таблице истинности равно количеству нулей, т.е. $\#\{X \mid f(X)=0\} = \#\{X \mid f(X)=1\} = 2^{n-1}$.

Отображение $Y=\varphi(X) : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^m$ при $n \geq m$ называется ***регулярным***, если Y ровно 2^{n-m} раза принимает все 2^m различных значений из $\mathrm{GF}(2)^m$, в то время как X

проходит 2^n значений из $GF(2)^n$. Очевидно, что каждое регулярное отображение $Y=\varphi(X) : GF(2)^n \to GF(2)^m$ при $m=n$ является биективным отображением.

Необходимым условием регулярности отображения $Y=\varphi(X) : GF(2)^n \to GF(2)^m$ при $n \geq m$ является сбалансированность любых линейных комбинаций компонентных БФ, реализующих векторную БФ $\varphi(X) = \{f_1(X), f_2(X), \ldots, f_m(X)\}$. Под линейной комбинацией элементов вектора $\varphi(X) = \{f_1(X), f_2(X), \ldots, f_m(X)\}$ для любых $\lambda=(\lambda_1,\ldots,\lambda_m) \in GF(2)^m$ понимается сумма $\sum_{i=1}^{m} \lambda_i f_i(x)$.

Сбалансированность БФ в терминах ПУА эквивалентна условию $\hat{U}_\alpha(f) = 0$, $U_\alpha(f) = \frac{1}{2^{n-1}}$, где $\alpha=(0, \ldots, 0)$, т.е. значение ПУА в точке "0" определяет степень отклонения БФ $f(X)$ от сбалансированности.

Например, БФ $f(X)$, $X \in GF(2)^6$, имеющая вид $f(X) = x_1x_2 \oplus x_3x_4 \oplus x_5x_6$ является несбалансированной, так как значение компоненты спектра ПУА данной функции, представленного на рис. 2.1, в точке 0 $\hat{U}_0(f) = 8$. На рис. 2.1 значения i, отложенные по оси абсцисс, являются целочисленным представлением вектора α. Таблица истинности данной БФ имеет вид

$S(f)$ = [0001000100011110000100010001111000010001000111101110111011100001].

Для построения сбалансированных БФ можно использовать следующее свойство: если одна из функций $f_1(X)$ или $f_2(Z)$, где $X, Z \in GF(2)^n$ является сбалансированной БФ, то функция $f(X, Z) = f_1(X) \oplus f_2(Z)$ также будет сбалансированной БФ.

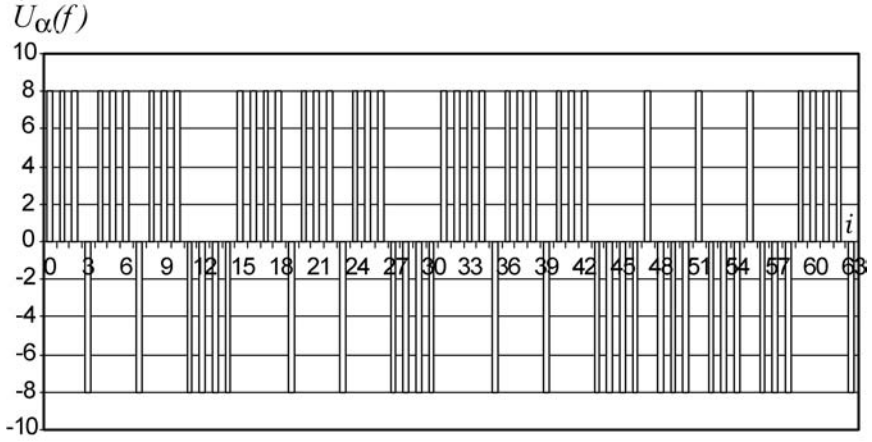

Рис. 2.1.

Показатель сбалансированности БФ является самым простым и, как правило, рассматривается в сочетании с другими показателями.

2.3.3. Корреляционные свойства БФ

Существенным усилением свойства сбалансированности БФ $f(X)$ является требование сбалансированности всех частных функций, полученных из исходной функции фиксированием любых ее k или менее переменных. Указанное требование позволяет обеспечить стойкость криптографических преобразований к статистическим атакам при фиксированных значениях битов на входе преобразования. Данное свойство связано с показателем корреляционной иммунности (КИ). При этом БФ $f(X)$, $X \in \mathrm{GF}(2)^n$ называется ***корреляционно-иммунной порядка k*** (CI(k)), $1 \le k < n$, если значение компонентов спектра УА $\hat{U}_\alpha(f) = 0$ для всех $\alpha \in \mathrm{GF}(2)^n$, вес Хэмминга которых $1 \le \boldsymbol{wt}(\alpha) \le k$ [35, 37, 51, 52, 125].

Следовательно, БФ является КИ порядка k, если значения функции $Y=f(X)$ статистически независимы от любого набора из k компонентов произвольного вектора-аргумента $X \in \mathrm{GF}(2)^n$. Если некоторая функция является корреляционно-иммунной порядка k, то она будет иметь корреляцию с некоторыми наборами компонентов вектора X размером большей чем k, т. е. существуют векторы $\alpha \in \mathrm{GF}(2)^n$ такие что $\boldsymbol{wt}(\alpha) > k$, $U_\alpha(f) \ne 0$. Отсюда следует, что максимальный порядок КИ БФ $Y=f(X)$, $X \in \mathrm{GF}(2)^n$ не превышает значения n–1. Единственными функциями, достигающими максимального порядка $k=n$–1, являются аффинные БФ: $f(x_1, x_2, \ldots, x_n)=x_1 \oplus x_2 \oplus \ldots \oplus x_n$ и $f(x_1, x_2, \ldots, x_n)=x_1 \oplus x_2 \oplus \ldots \oplus x_n \oplus 1$. Для $n = 6$ таблица истинности БФ подобного вида имеет следующие значения

$S(f)$ = [01101001100101101001011001101001100101100110100101101001100101 10],

а спектр УА представлен на рис. 2.2.

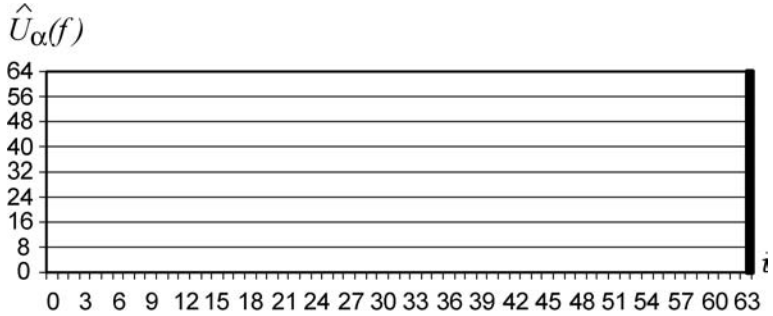

Рис. 2.2.

Достижение максимума показателя КИ БФ является достаточно сильным требованием, поэтому введем альтернативное понятие – корреляционно-эффективных БФ,

для которых не менее чем на половине векторов веса $1 \le w \le q$ значения компонентов спектра УА равны 0. Например, спектр УА (рис.2.3) БФ $f(X)$, $X \in \mathrm{GF}(2)^n$, $n=7$, $f(X)=x_1x_5x_7 \oplus x_2x_7 \oplus x_4x_6 \oplus x_4 \oplus x_3$ имеет 88 нулевых значений (≈69%), что позволяет отнести данную БФ к классу корреляционно-эффективных БФ.

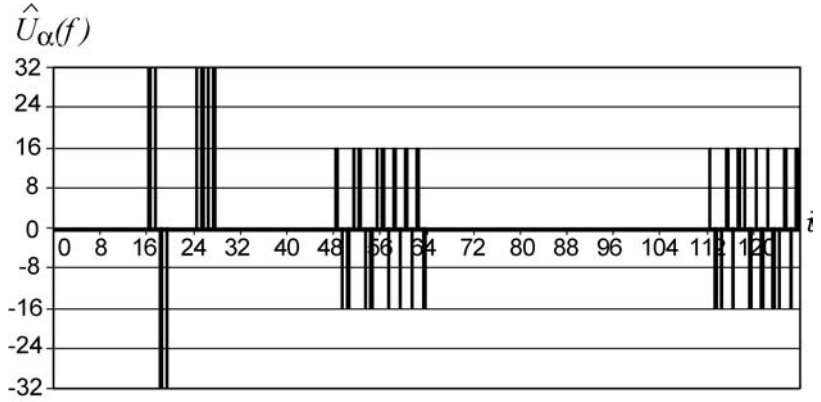

Рис. 2.3.

Корреляционно-иммунная степени k БФ $f(X)$ называется совершенной корреляционно-иммунной степени k или ***резилентной*** функцией, если она является сбалансированной функцией.

При синтезе БФ, обеспечивающих высокую стойкость к разностному, линейному и корреляционному криптоанализу, большое значение имеет автокорреляционная функция (АКФ) БФ. Под ***автокорреляцией*** БФ $f(X)$ относительно вектора $\beta \in \mathrm{GF}(2)^n$ понимается функция вида [31, 35, 51, 107]:

$$r_\beta(f) = \frac{1}{2^n} \sum_{X \in \mathrm{GF}(2)^n} \hat{f}(X)\, \hat{f}(X \oplus \beta) \qquad (2.10)$$

В случае обычного преобразования Фурье энергетический спектр (квадрат модуля преобразования Фурье) и АКФ играют важную роль и имеют широкую область приложений. Связь между энергетическим спектром и АКФ выражает известная теорема Винера-Хинчина, доказывающая, что обратное преобразование энергетического спектра реализуется в автокорреляционной функции. Данное соотношение можно перенести на случай ПУА. АКФ БФ $f(X)$, $X \in \mathrm{GF}(2)^n$ связана с энергетическим спектром $\hat{U}_\alpha^{\,2}(f)$ функции $f(X)$ через прямое и обратное ПУА [107]:

$$\hat{U}_\alpha(r_\beta(f)) = \sum_{\beta \in \mathrm{GF}(2)^n} r_\beta(f)(-1)^{\langle \beta, \alpha \rangle} = \frac{1}{2^n} \hat{U}_\alpha^{\,2}(f), \quad \alpha \in \mathrm{GF}(2)^n, \qquad (2.11)$$

$$r_\beta(f)=\frac{1}{2^{2n}}\sum_{\alpha\in \mathrm{GF}(2)^n}\hat{U}_\alpha^2(f)(-1)^{\langle\beta,\alpha\rangle},\quad \beta\in\mathrm{GF}(2)^n, \tag{2.12}$$

т.е. ПУА АКФ БФ $f(X)$ совпадает с энергетическим спектром этой функции. Для АКФ БФ $f(X)$, $X\in\mathrm{GF}(2)^n$ справедливы следующие соотношения [107, 111]:

$$\Pr\{f(X)\neq f(X\oplus\beta)\}=\Pr\left\{\hat{f}(X)\neq\hat{f}(X\oplus\beta)\right\}=\frac{1}{2}-\frac{1}{2}r_\beta(f), \tag{2.13}$$

$$\sum_{\beta\in\mathrm{GF}(2)^n,\beta\neq 0}\Pr\left\{\hat{f}(X)\neq\hat{f}(X\oplus\beta)\right\}=2^{n-1}-\frac{\hat{U}_0^2(f)}{2^{n+1}} \tag{2.14}$$

Данные свойства позволяют записывать основные показатели качества БФ в терминах АКФ.

2.3.4. Критерии распространения изменений для БФ

Важными характеристиками качества криптографических преобразований являются понятия критерия строгого лавинного эффекта – КСЛЭ (SAC – strict avalanche criterion) и критерия распространения – КР (PC – propagation criterion). Их основная сущность заключается в оценивание вероятности изменения значения БФ в зависимости от изменения части бит аргументов этих функций.

Пусть разность $\Delta f(X,\beta)=f(X)\oplus f(X\oplus\beta)$, где $X,\beta\in\mathrm{GF}(2)^n$; $f(X)\in\mathrm{GF}(2)$, тогда БФ $f(X)$ удовлетворяет [51, 71]:

- КСЛЭ, если разность $\Delta f(X,\beta)$ является сбалансированной функцией для любых α, вес которых $\boldsymbol{wt}(\alpha)=1$;
- КСЛЭ порядка k (SAC(k)), если любая функция, полученная из $f(X)$ путем подстановки констант на места произвольных ее k переменных, удовлетворяет КСЛЭ.

Переходя к спектральным свойствам, БФ $f(X)$ удовлетворяет КСЛЭ, если АКФ $r_\beta(f)=0$ для любых векторов $\beta\in\mathrm{GF}(2)^n$, вес которых $\boldsymbol{wt}(\beta)=1$, что в соответствии с (2.11) равносильно выполнению условия: обратное преобразование над $\hat{U}_\alpha^2(f)$ должно удовлетворять равенству $\sum_{\alpha\in\mathrm{GF}(2)^n}\hat{U}_\alpha^2(f)(-1)^{\langle\alpha,\beta\rangle}=0$.

Булева функция $f(X)$, $X\in\mathrm{GF}(2)^n$ удовлетворяет [31, 85]:

- КР степени $l \le n$ (РС(l)), если при замене $1 \le i \le l$ произвольных входных битов на свои дополнения функция $f(X)$ изменяется с вероятностью $\Pr\{f(X) \ne f(X \oplus \beta)\} = 0.5$, что эквивалентно сбалансированности разности $\Delta f(X, \beta)$ для любых β, вес которых $1 \le \boldsymbol{wt}(\beta) \le l$;
- КР степени $l \le n$ и порядка $k \le n$-1 (РС(l)/k), если любая функция, полученная из БФ $f(X)$ фиксацией любых ее k переменных, удовлетворяет РС(l).

В терминах спектральных свойств БФ удовлетворяет КР степени l (РС(l)), если ее АКФ $r_\beta(f) = 0$ на всех векторах $\beta \in GF(2)^n$, вес Хэмминга которых лежит пределах $1 \le \boldsymbol{wt}(\beta) \le l$. Очевидно, что КР является обобщением КСЛЭ, точнее, КСЛЭ эквивалентен КР РС(1).

Булевы функции вида $f(X) = \bigoplus_{i=1}^{n-1} \bigoplus_{j=i+1}^{n} x_i x_j$ удовлетворяют следующим показателям:

- SAC(n–2);
- РС(k)/m при $k+m \le n-1$ или ($k+m$=1 и k–чётное);
- РС(2)/(n–2);
- РС(n) при чётном n;
- РС(n–1) при нечётном n и $f(X)$ будет сбалансированной БФ.

Например, над векторным пространством $GF(2)^n$, при n=8, таблица истинности БФ $f(X) = \bigoplus_{i=1}^{7} \bigoplus_{j=i+1}^{8} x_i x_j$ имеет вид

$S(f)$ = [00010111011111100111111011101000011111101110100011101000100000010111111011101000111010001000000111101000100000011000000100010111111010001000000110000001000101111100000010001011100010111011111110].

Спектр ПУА БФ представлен на рис.2.4, а АКФ $r_\beta(f)$=0 на всех векторах β, вес которых $\boldsymbol{wt}(\beta) \ge 1$.

Рис. 2.4.

Таким образом, рассмотренная БФ имеет следующие показатели: PC(8), PC(6)/2, PC(2)/6.

Как и в случае с корреляционной иммунностью, критерия распространения ненулевой степени (а тем более ненулевого порядка) достичь довольно трудно.

2.3.5. Исследование нелинейности БФ

При анализе эффективности криптографических преобразований, в частности в отношении линейного криптоанализа, одной из основных характеристик является нелинейность БФ, реализующих данное преобразование, которая показывает степень удаленности БФ от множества аффинных или линейных БФ [31, 34-36, 44, 50, 100, 114, 121]. Нелинейность БФ является важным показателем, поскольку линейные функции сравнительно легко вскрываются криптоаналистическими методами.

Под ***аффинными*** функциями $f_{афф}(X) \in \mathcal{A}$, где $\mathcal{A} = \{f_{афф}(X)\}$, $X \in \mathrm{GF}(2)^n$ понимаются БФ следующего вида

$$f_{афф}(X) = \alpha_1 x_1 \oplus \alpha_2 x_2 \oplus \ldots \oplus \alpha_n x_n \oplus c\,, \qquad (2.15)$$

где $\alpha, c \in \mathrm{GF}(2)$. При $c = 0$ аффинные функции образуют подмножество ***линейных*** функций $\mathcal{L} = \{f_{лин}(X)\}$, каждая из которых является скалярным произведением вида $f_{лин}(X) = \langle \alpha, X \rangle$.

Удаленность БФ от множества аффинных $\mathcal{A}$ или линейных $\mathcal{L}$ БФ оценивается через расстояние Хемминга $d(f, g)$ между двумя БФ $f(X)$ и $g(X)$, которое определяет количество отличающихся значений функций:

$$d(f, g) = \#\{f(X) \neq g(X) : X \in \mathrm{GF}(2)^n\} = \boldsymbol{wt}(S(f) \oplus S(g)). \qquad (2.16)$$

В работе [107] показано, что расстояние (2.16) можно определить через сопряженные функции

$$d(f, g) = 2^{n-1} - \frac{1}{2} \sum_{X \in \mathrm{GF}(2)^n} \hat{f}(X)\,\hat{g}(X) = 2^{n-1} - \frac{1}{2}\left\langle \hat{S}(f),\ \hat{S}(g) \right\rangle \qquad (2.17)$$

Поскольку взаимная корреляционная функция (ВКФ) (функция кросс-корреляции) $R_t(f, g)$ между БФ $f(X)$ и $g(X)$ определяется выражением:

$$R_t(f,\ g) = \sum_{X \in \mathrm{GF}(2)^n} (-1)^{f(X)} (-1)^{g(X \oplus t)} = \sum_{X \in \mathrm{GF}(2)^n} \hat{f}(X)\,\hat{g}(X \oplus t)\,, \qquad (2.18)$$

а значение ВКФ в точке 0 является коэффициентом взаимной корреляции (КВК) [121]

$$r(f, g)=\frac{R_0(f,\ g)}{2^n}=\frac{1}{2^n}\sum_{X\in \mathrm{GF}(2)^n}\hat{f}(X)\hat{g}(X)\ =$$

$$\frac{1}{2^n}\left(\sum_{\substack{X\in \mathrm{GF}(2)^n,\\ f(X)=g(X)}}\hat{f}(X)\hat{g}(X)-\sum_{\substack{X\in \mathrm{GF}(2)^n,\\ f(X)\neq g(X)}}\hat{f}(X)\hat{g}(X)\right), \qquad (2.19)$$

т. е.

$$r(f, g) = \Pr\{f(X) = g(X)\} \ - \ \Pr\{f(X) \neq \mathrm{g}(X)\}, \qquad (2.20)$$

то КВК и расстояние Хемминга связаны между собой следующим соотношением:

$$r(f, g) = \frac{2^n-d(f,g)}{2^n}-\frac{2^n-d(f,\bar{g})}{2^n}=\frac{1}{2^n}(d(f,\bar{g})-d(f,\ g))\ = 1-\frac{d(f,\ g)}{2^{n-1}}. \qquad (2.21)$$

Для суммарной корреляции любой БФ $f(X)$ с набором всех линейных функций справедливо равенство Парсеваля:

$$\sum_{f_{лин}\in\mathcal{L}} r^2(f, f_{лин})=\frac{1}{2^{2n}}\sum_{\alpha\in \mathrm{GF}(2)^n}\hat{U}_\alpha^{\,2}=1. \qquad (2.22)$$

Следовательно, суммарная корреляция не зависит от вида функции $f(X)$, и для любой БФ $f(X)$ всегда существует корреляция с какой-либо линейной функцией, т. е. существует такая $f_{лин}\in\mathcal{L}$, что $r(f, f_{лин})\neq 0$.

На основании вышеизложенного, дадим определение ***нелинейности*** БФ $f(X) : \mathrm{GF}(2)^n\to\mathrm{GF}(2)$, под которой понимается значение минимального расстояния Хэмминга между функцией $f(X)$ и функциями $f_{афф}(X)\in\mathcal{A}$:

$$\mathrm{NL}(f) = \min_{f_{афф}\in\mathcal{A}} d(f,\ f_{афф}). \qquad (2.23)$$

Аналогично нелинейность $\mathrm{NL}(f)$ можно выразить через расстояние до линейных функций:

$$\mathrm{NL}(f) = \min_{f_{лин}\in\mathcal{L}}\min\{\ d(f, f_{лин}), 2^n - d(f, \overline{f}_{лин})\ \} =$$

$$2^{n-1} - \frac{1}{2}\max_{f_{лин}\in\mathcal{L}}\left|\, d(f,\ f_{лин}) - d(f,\ \overline{f}_{лин})\,\right| = \qquad (2.24)$$
$$2^{n-1} - 2^{n-1}\max_{f_{лин}\in\mathcal{L}}\left|\, r(f,\ f_{лин})\,\right|$$

где

$$r(f, f_{лин}) = \Pr\{f(X) = f_{лин}(X)\} - \Pr\{f(X) \neq f_{лин}(X)\} =$$

$$2\left(\Pr\{f(X) = \langle \alpha, X\rangle\} - \frac{1}{2}\right) = \frac{1}{2^n}\left\langle \hat{S}(f), \hat{S}(\langle \alpha, X\rangle)\right\rangle = \frac{1}{2^n}\hat{U}_\alpha(f). \qquad (2.25)$$

Функция $r(f, f_{лин})$ используется в качестве линейной характеристики функции $f(X)$ по линейной функции $f_{лин}(X) = \langle \alpha, X\rangle$ и применяется при проведении линейного криптоанализа.

Подставив соотношение (2.25) в выражение (2.24), получим значение нелинейности функции $f(X)$ в терминах ПУА:

$$\mathrm{NL}(f) = 2^{n-1} - \frac{1}{2}\max_{\alpha \in \mathrm{GF}(2)^n} |\hat{U}_\alpha(f)|, \qquad (2.26)$$

т.е. для определения нелинейности БФ следует определить максимальное значение компоненты спектра УА. Используя данное соотношение, можно достаточно просто определять нелинейность БФ при малых значениях n.

На основании теоремы Парсеваля верхняя граница значения нелинейности $\mathrm{NL}(f)$ для любых БФ $f(X)$, $X \in \mathrm{GF}(2)^n$, определяется неравенством [37, 50, 121]

$$\mathrm{NL}(f) \leq \begin{cases} 2^{n-1} - 2^{\frac{n}{2}-1}, & \text{для } n \text{ – четного,} \\ \left\lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \right\rfloor, & \text{для } n \text{ – нечетного,} \end{cases} \qquad (2.27)$$

где $\lfloor \bullet \rfloor$ означает максимальное четное целое, меньшее либо равное значению аргумента. Аналогично для любой сбалансированной БФ справедливы неравенства:

$$\mathrm{NL}(f) \leq \begin{cases} 2^{n-1} - 2^{\frac{n}{2}-1} - 2, & \text{для } n \text{ – четного,} \\ \left\lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \right\rfloor, & \text{для } n \text{ – нечетного.} \end{cases} \qquad (2.28)$$

Противоположным понятию нелинейности является ***линейность*** БФ $f(X)$: $\mathrm{GF}(2)^n \to \mathrm{GF}(2)$, которая определяется из следующего выражения [101]:

$$\mathrm{L}(f) = \max_{f_{лин} \in \mathcal{L}} |r(f, f_{лин})|. \qquad (2.29)$$

Таким образом, в соответствии с выражениями (2.24, 2.29) линейность и нелинейность БФ $f(X)$ связаны равенствами:

$$\mathrm{NL}(f) = 2^{n-1}(1 - \mathrm{L}(f)), \tag{2.30}$$

$$\mathrm{L}(f) = 1 - \frac{1}{2^{n-1}}\mathrm{NL}(f). \tag{2.31}$$

Из равенства Парсеваля (2.22) следует, что линейность находится в пределах

$$2^{-\frac{n}{2}} \le \mathrm{L}(f) \le 1. \tag{2.32}$$

Для подстановочного преобразования $Y = \varphi(X), X \in \mathrm{GF}(2)^n, \ Y \in \mathrm{GF}(2)^m$, где $\varphi(X)$ – векторная БФ, понятия нелинейности и линейности определяются следующими соотношениями [101]:

$$\mathrm{NL}(\varphi(X)) = \min_{\substack{c\in \mathrm{GF}(2)^m, \\ c\neq 0}} \mathrm{NL}\langle c, \varphi\rangle, \tag{2.33}$$

$$\mathrm{L}(\varphi(X)) = \max_{\substack{c\in \mathrm{GF}(2)^m, \\ c\neq 0}} \mathrm{L}\langle c, \varphi\rangle, \tag{2.34}$$

где минимум и максимум находятся не только относительно всех компонентных БФ, но и любых линейных комбинаций данных БФ.

Достичь наилучших показателей нелинейности и линейности для векторной БФ сложно, поэтому на практике используется понятие средней нелинейности и линейности векторной БФ $\varphi(X)$, которые определяются из выражений:

$$\overline{\mathrm{NL}}(\varphi(X)) = \frac{1}{2^n}\sum_{\substack{c\in \mathrm{GF}(2)^n, \\ c\neq 0}} \mathrm{NL}\langle c, \ \varphi\rangle, \tag{2.35}$$

$$\overline{\mathrm{L}}(\varphi(X)) = \frac{1}{2^n}\sum_{\substack{c\in \mathrm{GF}(2)^n, \\ c\neq 0}} \mathrm{L}\langle c, \ \varphi\rangle. \tag{2.36}$$

Пределы нелинейности и линейности для векторных БФ $\varphi(X)$ совпадают с пределами для компонентных БФ, т.е. определяются выражениями (2.27, 2.28, 2.32).

2.3.6. БФ, достигающие максимальной нелинейности

Булева функция является ***совершенно нелинейной***, если $\forall\beta\in\mathrm{GF}(2)^n$, $\beta\neq 0$, выполняется равенство

$$\#\{X\in \mathrm{GF}(2)^n : f(X) = f(X\oplus\beta)\} = \#\{X\in \mathrm{GF}(2)^n : f(X) \neq f(X\oplus\beta)\} = \frac{2^n}{2},$$

что эквивалентно равенству $r_\beta(f) = 0$.

В терминах ПУА БФ $f(X)$, $X\in \mathrm{GF}(2)^n$ является совершенно нелинейной или иначе ***бент-функцией***, если $\forall\ \alpha\in \mathrm{GF}(2)^n$ выполняется равенство [31, 34, 50, 61, 100, 114]

$$\hat{U}_\alpha(f) = \pm 2^{n/2}. \qquad (2.37)$$

То есть, это БФ, для которых расстояние Хэмминга до множества всех аффинных функций $\mathcal{A}$ над $\mathrm{GF}(2)^n$ максимально и имеет постоянную величину $d\left(f_{бент},\ f_{афф}\right)^{\forall f_{афф}\in\mathcal{A}} = const$.

Основными свойствами множества бент-функций $\mathcal{B} = \{ f_{бент}(X) \}$ являются [34, 114]:

- количество переменных n для любой $f_{бент}(X)\in\mathcal{B}$, $X\in \mathrm{GF}(2)^n$ четно, а алгебраическая степень $deg(f_{бент}) \le n/2$;

- бент-функции $f_{бент}(X)\in\mathcal{B}$ – несбалансированные функции, такие что $\#\{ f_{бент}(X) = 1\} = 2^{n-1} \pm 2^{\frac{n}{2}-1}$, а $\#\{ f_{бент}(X) = 0\} = 2^{n-1} \mp 2^{\frac{n}{2}-1}$.

- бент-функции $f_{бент}(X)\in\mathcal{B}$ имеют максимальное значение нелинейности и, соответственно, минимальное значение линейности:

$$\mathrm{NL}(f_{бент}(X)) = 2^{n-1} - 2^{\frac{n}{2}-1},\ \mathrm{L}(f_{бент}(X)) = 2^{-n/2}; \qquad (2.38)$$

- единственным множеством функций, удовлетворяющих КР степени n - РС(n), является множество бент-функций $\mathcal{B}$, т.е. $\forall\beta\in \mathrm{GF}(2)^n$, вес Хэмминга которого $1\le \boldsymbol{wt}(\beta)\le n$, разностная функция $\Delta f(X, \beta) = f(X)\oplus f(X\oplus\beta)$ является сбалансированной функцией;

- коэффициент взаимной корреляции между бент-функцией и множеством всех линейных БФ имеет постоянное значение по абсолютной величине $\left| r\left(f_{бент}, f_{лин}\right)\right|^{\forall f_{лин}\in\mathcal{L}} = \frac{1}{2^{n/2}}$.

Существуют следующие, простейшие, методы построения бент-функций [31, 34, 61, 114, 121]:

- БФ $f(X)$, $X \in \mathrm{GF}(2)^n$ вида $f(X) = \bigoplus_{i=1}^{n-1} \bigoplus_{j=i+1}^{n} x_i x_j$ является бент-функцией (пример рассмотрен на рис. 2.4).
- если $f(X)$ и $g(X)$ есть соответственно бент– и аффинная БФ, то $h(X) = f_{бент}(X) \oplus g_{афф}(X)$ является бент-функцией;
- если g – произвольная БФ от m переменных, тогда функции $f(X)$, $X \in \mathrm{GF}(2)^n$, где $n=2m$, следующего вида

$$f(X) = g(x_1, x_2, \ldots, x_m) \oplus x_1 x_{m+1} \oplus x_2 x_{m+2} \oplus \ldots \oplus x_m x_n$$

являются бент-функциями.

Аналогично, если перейти к векторным БФ, то функция $\varphi(X) = \{f_1(X), \ldots, f_m(X)\}$ является векторной бент-функцией, если для любых $\lambda \in \mathrm{GF}(2)^m$ БФ $\phi(X) = \langle \lambda, \varphi(X) \rangle$ также есть векторная бент-функция, т.е. каждая нетривиальная линейная комбинация компонентных функций является бент-функцией. Векторные бент-функции существуют только для четного значения $n \geq 2m$.

2.3.7 Обобщение показателей качества подстановочных преобразований

В целях упрощения анализа показателей качества криптографических преобразований над исходными векторами следует проводить аффинное преобразование координат, так как алгебраическая степень, сбалансированность, нелинейность и количество векторов, на которых выполняется критерий нераспространения – все инвариантны относительного аффинного преобразования координат [31, 115]. В качестве аффинного преобразования координат можно использовать умножение вектора $X \in \mathrm{GF}(2)^n$ на невырожденную матрицу порядка n над полем GF(2).

Таким образом, обобщая изложенные результаты, сформулируем необходимые требования ко всему подстановочному преобразованию и к формирующим его компонентным БФ:

1. Регулярность отображения $Y=\varphi(X) : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^m$ при $n \geq m$, т.е. сбалансированность всех компонентных БФ и их любых линейных комбинаций.
2. Высокая алгебраическая степень нелинейности $deg(f)$ компонентных БФ.
3. Соответствие БФ показателям корреляционной иммунности.
4. Соответствие БФ КСЛЭ и КР из множества $\mathrm{SAC}(k)$, $\mathrm{PC}(k)/l$, т.е. низкие автокорреляционные показатели.

5. Высокая нелинейность $\mathrm{NL}(f)$ компонентных БФ.

6. Высокая нелинейность $\mathrm{NL}(\varphi(X))$ или средняя нелинейность $\overline{\mathrm{NL}}(\varphi(X))$ векторного преобразования.

ГЛАВА 3

Двухключевые криптосистемы

3.1. Сравнительная характеристика одноключевых и двухключевых шифров

Криптографические системы (шифры) с секретным ключом, называемые также симметричными, решают проблемы сохранения информации в тайне и обеспечения контроля целостности информации. Первая из этих задач определяет существо одноключевой криптографии.

Главным требованием, предъявляемым к симметричным шифрам, является их стойкость. Абсолютно (безусловно) стойкими криптосистемами являются шифры с ключом однократного использования [116]. Такие шифры достаточно просты в плане осуществляемых процедур зашифрования и расшифрования, однако они требуют распределения чрезмерно большого объема ключевого материала, что делает их использование очень дорогим.

Задача распределения ключей по защищенным каналам является фундаментальной [55, 89, 113]. При ее решении одновременно решается и задача аутентификации ключей. Действительно, получатель должен не только получить секретный ключ, но также и убедиться, что именно этот ключ был отправлен законным отправителем (в частности, центром распределения ключей). Проблема распределения секретных ключей в определенной степени затеняет проблему их аутентификации. Следует иметь в виду, что и вторая проблема является фундаментальной. Более того, она является фундаментальной и в двухключевой криптографии, которая предоставляет наиболее экономичные и удобные механизмы распределения секретных ключей. Эти вопросы будут обсуждаться ниже.

Абсолютная стойкость шифра рассматривается только по отношению к атакам на основе шифртекста. Действительно, если ключ используется однократно, то атаки на основе известного или специально подобранного текста теряют смысл. На практике наибольшее распространение получили шифры другого типа, а именно шифры с конечным ключом (обычно от 64 до 256 бит). Такие шифры являются условно стойкими. Всегда имеется теоретическая возможность определения ключа шифрования путем полного перебора всего пространства ключей, если имеется шифртекст, превышающий интервал единственности. Однако на практике такая атака несостоятельна ввиду чрезвычайно большого времени, которое требуется для перебора. В связи с этим такие шифры называют также практически стойкими.

Если рассматривать атаки на основе известных или специально подобранных текстов, то в принципе могут быть найдены некоторые способы вычисления ключа. Алгоритмы шифрования должны быть такими, чтобы обеспечить высокую вычислительную сложность наилучшего алгоритма атаки. В связи с этим шифры с конечным ключом называют также вычислительно стойкими.

Вычислительно стойкие шифры нашли на практике существенно более широкое применение по сравнению с теоретически (безусловно) стойкими криптосистемами, поскольку они свободны от ряда существенных для практического использования недостатков, связанных со стремлением достигнуть абсолютной стойкости. Кроме устранения технологических недостатков, переход к вычислительно стойким шифрам дает возможность построить качественно новые криптосистемы – двухключевые шифры, называемые также криптосистемами с открытым ключом. В последнем термине подчеркивается принципиальное значение использования общеизвестного ключа. Необходимость использования секретного ключа очевидна, поскольку он необходим в шифрах любого типа.

Принципиально новым, что вносится двухключевыми шифрами, является *возможность построения криптографических протоколов, решающих задачи взаимодействия сторон, которые не доверяют друг другу*. Появление такой возможности связано с тем, что в двухключевых шифрах секретный ключ, вырабатываемый некоторым пользователем (абонентом), остается известным только ему. В протоколах, основанных на одноключевых (симметричных) криптосистемах, решаются задачи, в которых взаимодействующие стороны *доверяют* друг другу, что связано с тем, что секретный ключ должен быть известным, по крайней мере, двум сторонам. Для реализации задач протоколов, основанных на двухключевых шифрах, требуется знание другими сторонами только открытого ключа. Например, секретный ключ может быть использован для выработки цифровой электронной подписи к некоторому документу, которая будет представлять собой некоторое число, зависящее от секретного ключа и от каждого бита документа (или от хэш-функции последнего). Если хотя бы еще один участник протокола знал бы секретный ключ, то и он смог бы сформировать такую же подпись, т. е. невозможно было бы однозначно установить участника, подписавшего данный документ (т. е. выработавшего некоторый двоичный вектор, интерпретируемый как число или сообщение со специальной структурой и служащий электронной цифровой подписью к данному документу). Вопрос проверки правильности подписи решается с использованием общедоступного открытого (публичного) ключа. Таким образом, подписать документ может только единственная сторона, являющаяся владельцем секретного ключа, а проверить подпись может любой желающий.

Очевидно, что реализация такого обобщенного механизма функционирования системы цифровой электронной подписи (ЭЦП) требует наличия определенной связи между секретным и открытым ключами, т. е. они должны быть зависимыми между собой. Ниже будут рассмотрены несколько вариантов того, как это может быть реализовано.

3.2. От открытого распределения ключей до электронной цифровой подписи

3.2.1. Система распределение ключей Диффи-Хеллмана

Зарождение двухключевой криптографии и основные криптосистемы с открытым ключом связаны с использованием функции возведения в большую дискретную степень по модулю большого простого числа

$$f(x) = \alpha^x \pmod p,$$

где x – целое число, $1 \le x \le p-1$, p – k-битовое простое число, α – первообразный корень по модулю p.

Используя данную функцию, учеными Диффи и Хеллманом [69] была показана возможность построения практически стойких секретных систем, которые *не требуют передачи секретного ключа*. Предложенная ими система получила название метода открытого распространения ключей. В ней каждый абонент выбирает случайным образом секретный ключ x и вырабатывает открытый ключ y, соответствующий выбранному секретному ключу, в соответствии с формулой

$$y = \alpha^x \pmod p.$$

Системой Диффи-Хеллмана называется следующий способ использования дискретного возведения в степень для обмена секретными ключами между пользователями сети с применением только открытых сообщений. Выбирается большое простое число p и соответствующий ему первообразный корень $\alpha < p$. (Для обеспечения стойкости рассматриваемой системы открытого шифрования на число p накладывается следующее условие: разложение числа $p-1$ на множители должно содержать, по крайней мере, один большой простой множитель; размер числа p должен быть не менее 512 бит.)

Механизм распределения секретных ключей по открытому каналу состоит в следующем. Каждый абонент выбирает случайный секретный ключ x и вырабатывает открытый ключ y, соответствующий выбранному секретному ключу, в соответствии с формулой

$$y = \alpha^x \pmod p.$$

Два абонента **A** и **B** могут установить секретную связь без передачи секретного ключа следующим образом. Абонент **A** берет из справочника открытый ключ y_B абонента **B** и, используя свой секретный ключ x_A, вычисляет общий секретный ключ:

$$Z_{AB} = (y_B)^{x_A} = (\alpha^{x_B})^{x_A} = \alpha^{x_B x_A} \pmod p.$$

Аналогично поступает абонент **B**:

$$Z_{AB} = (y_A)^{x_B} = (\alpha^{x_A})^{x_B} = \alpha^{x_B x_A} \pmod p.$$

Таким образом, оба абонента сформировали одинаковый секретный ключ Z_{AB} без использования какого-либо заранее оговоренного общего секрета. Владея только им известным секретом и используя его в качестве мастер-ключа, данная пара абонентов может зашифровывать направляемые друг другу сообщения. Указанные выше вычисления легко осуществимы для достаточно больших значений p, α, y и x (например, имеющих в двоичном представлении длину 4096 бит и более). Атакующему известны значения $y_B = \alpha^{x_B} \pmod p$ и $y_A = \alpha^{x_A} \pmod p$, но для того чтобы вычислить Z_{AB}, он должен решить задачу дискретного логарифмирования и определить либо x_A, либо x_B. Легко найти большие значения p (более 1024 бит), для которых задача дискретного логарифмирования является трудно решаемой. Если будут найдены вычислительно эффективные методы решения задачи дискретного логарифмирования, то метод Диффи-Хеллмана окажется несостоятельным — в связи с этим говорят, что данный метод открытого распределения ключей основан на сложности дискретного логарифмирования. В настоящее время в общем случае задача дискретного логарифмирования практически неразрешима, что дает возможность широкого практического применения метода Диффи-Хеллмана и многочисленных систем ЭЦП, основанных на сложности вычисления дискретных логарифмов.

Не следует упускать из виду проблему аутентификации открытых ключей. Корректность протоколов с использованием асимметричных шифров может быть обеспечена только в случае, если все открытые ключи в справочнике открытых ключей являются подлинными. Если открытый ключ какого-либо абонента нарушителю удастся подменить, то секретные сообщения, посланные данному абоненту, будут доступны нарушителю. Таким образом, двухключевые шифры обеспечивают решение проблемы распределения секретных ключей, однако проблема аутентификации сохраняется и имеет фундаментальный характер, хотя требование подтверждения подлинности (аутентификации) относится уже к открытому, а не к секретному ключу. В неявном виде аутентификация открытого ключа включает в себя аутентификацию секретного ключа.

3.2.2. Открытый шифр Эль-Гамаля

В рассмотренной выше двухключевой криптосистеме осуществляется открытое распределения ключей, но сами процедуры зашифрования и расшифрования осуществляются с помощью симметричного шифра. Существуют двухключевые шифры, в которых зашифрование осуществляется по открытому ключу, а расшифрование по секретному. Такой шифр может быть построен на основе метода Диффи-Хеллмана путем использования разового открытого ключа. Рассмотрим способ шифрования, предложенный Эль-Гамалем [70]. Чтобы послать пользователю i секретное сообщение T, отправитель действует в соответствии со следующим алгоритмом:

1. Выбрать случайное число x' (по своей сути x' является разовым секретным ключом).
2. Вычислить значение $y' = \alpha^{x'} \pmod p$, которое фактически является разовым открытым ключом.

3. Используя открытый ключ y получателя, вычислить значение $Z = y^R \pmod p$, которое есть разовый общий секретный ключ отправителя и i-го пользователя.
4. Осуществить зашифрование сообщения T с помощью умножения по модулю p сообщения на разовый общий секретный ключ: $C = y^R T \pmod p$.
5. Отправить криптограмму (y', C) i-му пользователю.

Легко видеть, что суть открытого шифрования Эль-Гамаля состоит в использовании умножения по модулю сообщения на разовый общий секрет. Для того чтобы i-ый абонент мог правильно расшифровать сообщение, в криптограмму вместе с шифртекстом C включается также и разовый открытый ключ отправителя, что приводит к увеличению размера криптограммы примерно в 2 раза по сравнению с исходным текстом. Легко показать, что i-ый пользователь, получив криптограмму (y', C), может вычислить исходный текст: $T = C/(y')^x = C/Z \pmod p$.

Важным моментом в открытом шифре Эль-Гамаля является использование разового открытого ключа. Этот прием является принципиальным при построении систем ЭЦП, основанных на сложности дискретного логарифмирования и являющихся дальнейшим развитием криптографических применений функции возведения в большую степень по модулю большого простого числа.

3.3. Системы ЭЦП на основе задачи дискретного логарифмирования

3.3.1. Общие положения

Пусть дан некоторый документ. Имея криптографически стойкую хэш-функцию, мы можем отождествить подписывание хэш-функции h от документа с подписыванием самого документа. В системах ЭЦП это делается для того, чтобы подпись к документу имела сравнительно небольшой фиксированный размер независимо от размера самого документа. Это имеет важное технологическое значение, однако не является принципиальным для понимания механизма действия системы ЭЦП. Подпись должна зависеть от секретного ключа и от значения h. Попытаемся задать уравнение проверки подписи s в следующем виде:

$$h = y^s \pmod p.$$

Если владельцу секретного ключа удастся для данного значения h вычислить подпись s, то мы бы имели уже систему ЭЦП. По этой формуле, используя открытый ключ подписывающего, любой желающий может проверить справедливость подписи. Однако легко прийти к выводу, что в общем случае владелец секретного ключа не сможет вычислить значение s, которое будет удовлетворять условию $h = y^s = \alpha^{xs} \pmod p$, поскольку для нахождения s потребуется представить значение хэш-функции h в виде степени α, а именно $h = \alpha^w \pmod p$, т. е. для вычисления под-

писи потребуется решить задачу дискретного логарифмирования. Значит надо найти некоторое другое уравнение проверки подписи, например, следующее:

$$\alpha^h = y^s \pmod p.$$

Очевидно, что, зная только открытый ключ y, найти значение s, которое будет удовлетворять второму уравнению проверки подписи, вычислительно сложно. При этом владелец секретного ключа x может легко вычислить подпись s, поскольку он может воспользоваться соотношением $\alpha^h = y^s = \alpha^{xs} \pmod p$, из которого вытекает уравнение вычисления подписи:

$$h = xs \pmod{(p-1)}, \text{ т. е. } s = h/x \pmod{p-1}.$$

Следовательно требуется осуществить умножение по модулю $p-1$ на элемент обратный значению секретного ключа. Он существует и является единственным, если x является взаимно простым с $p-1$. Это некоторое несущественное ограничение на выбор секретного ключа, с которым можно было бы смириться, однако все же данный вариант ЭЦП несостоятелен, поскольку по однажды предоставленной для проверки подписи проверяющий может не только убедиться, что значение s было сформировано владельцем секретного ключа, но и вычислить сам секретный ключ. Секретный ключ стал известен более чем одному пользователю. Системы ЭЦП не должны допускать такой возможности. Несмотря на текущую неудачу, мы можем уяснить для себя полезный факт: *значение хэш-функции документа целесообразно использовать как степень числа α или как множитель этой степени*. Это предоставит возможность вычисления подписи по известному секретному ключу. Теперь наша задача состоит в том, чтобы сделать вычислительно невозможным нахождение секретного ключа по сформированной подписи. Попытаемся обеспечить это свойство путем использования разового дополнительного открытого ключа r, формируемого подписывающим по формуле

$$r = \alpha^k \pmod p,$$

где k случайно выбираемое число. Теперь у нас появился дополнительный параметр, который дает надежду на новые возможности. Попробуем следующее уравнение проверки подписи

$$\alpha^h = y^s r \pmod p.$$

Ему соответствует уравнение формирования подписи $s = x/kh \pmod{p-1}$. Теперь уже по значениям s, r и h вычисление секретного ключа является вычислительно сложной задачей. Легко заметить, что значение разового секретного ключа k необходимо держать в секрете (при знании k проверяющий без труда сможет вычислить секретный ключ). Так как оно нужно только для формирования подписи, то его лучше *уничтожить сразу после вычисления подписи*. Замечаем также, что при формировании параметра r нужно выбрать такое число k, которое является взаимно простым с числом $p-1$. Тогда при h, взаимно простом с $p-1$, существует и можно найти число, обратное к kh и вычислить s. Однако значение хэш-функции является

случайным и в общем случае может содержать общий множитель с $p-1$. Это показывает еще одно обстоятельство, которое нужно учитывать при построении системы ЭЦП. А именно, желательно получить такое уравнение формирования подписи, в котором знаменатель содержит параметры, которые мы можем выбрать таким образом, чтобы они имели значения взаимно простые с $p-1$.

В рассматриваемом примере подписью служит пара чисел (s, r), т. е. подпись стала вдвое длиннее, но мы существенно продвинулись в том, что вычисление секретного ключа стало невозможным по значениям параметров, используемым для проверки подписи. Однако важно, чтобы наличие подписи к одному документу не давало возможность сформировать подпись к другому документу. Может оказаться, что это вычислительно осуществимо и без вычисления секретного ключа. Анализ данного вопроса показывает, что это действительно так. Например, пусть дана подпись (s, r) к документу, соответствующему значению хэш-функции h. Рассмотрим документ, хэш-функция от которого равна h', и попытаемся сформировать правильную подпись (s', r') к нему. Нам необходимо выбрать такие значения s' и r', чтобы выполнялось условие $\alpha^{h'} = y^{s'} r' \pmod p$. Левую часть последнего соотношения можно преобразовать следующим образом:

$$\alpha^{h'} = (\alpha^h)^{h'/h} = (y^s r)^{h'/h} = (y^s)^{h'/h} r^{h'/h} = y^{sh'/h} r^{h'/h} \pmod p.$$

Из последнего соотношения видно, что можно взять пару чисел $s' = sh'/h \pmod{(p-1)}$ и $r' = r^{h'/h} \pmod p$, которые будут являться действительной подписью ко второму документу. Таким образом, мы приходим к выводу, что последнее уравнение проверки подписи все еще требует модернизации, поскольку, имея один образец подписи, можно легко вычислить правильную подпись к любому другому документу, хотя до появления первой правильной подписи это сделать вычислительно сложно. Проведя аналогичный анализ следующих уравнений проверки подписи

$$\alpha^h = (yr)^s \pmod p \quad \text{и} \quad (\alpha r)^h = y^s \pmod p,$$

обнаруживаем, что они также допускают формирование новых подписей без знания секретного ключа при условии, что имеется одна правильная подпись, сформированная с использованием секретного ключа. При проведении такого анализа легко прийти к выводу, что относительно простым вариантом модернизации является использовании значения разового открытого ключа не только как основания одного из сомножителей, фигурирующих в уравнении проверки подписи, но также и в качестве степени того же самого или другого сомножителя.

Проанализируем следующие уравнения проверки и формирования подписи:

$$\alpha^{hr} = y^s r \pmod p, \quad \text{и} \quad s = (hr - k)/x \pmod{p-1}.$$

Для корректного формирования подписи необходимо выбрать случайный секретный ключ взаимно простой с $p-1$. Для заданной хэш-функции h сформировать подпись без знания секретного ключа представляется вычислительно сложной задачей, связанное с решением проблемы дискретного логарифмирования. Данная пара уравнений может использоваться в качестве системы ЭЦП, хотя некоторым недос-

татком последней является указанное ограничение (НОД(x, $p-1$) = 1) на выбор секретного ключа.

С целью устранить ограничения на выбор секретного ключа испытаем другой вариант ЭЦП, получаемый из предыдущего путем перестановки параметров s и h:

$$\alpha^{sr} = y^h r \pmod p, \text{ и } s = (hx + k)/r \pmod{p-1}.$$

Теперь требуется сформировать значение r, взаимно простое с $p-1$. До выполнения этой проверки нужно выполнить операцию возведения в большую целую степень, что несколько усложняет нахождение значения, взаимно простого с $p-1$. Кроме того, эта операция проверки на взаимную простоту должна теперь выполняться при каждой процедуре подписывания сообщений (если использовать *одно и то же* значение r для подписывания *двух* различных сообщений, то можно составить два различных уравнения формирования подписи с двумя неизвестными x и k, что позволит легко вычислить секретный ключ x). С этими незначительными недостатками можно было бы смириться, но окончательная дискредитация нашего усовершенствования связана с тем, что без знания секретного ключа можно сформировать правильную подпись к любому заданному документу. Действительно, пусть h есть хэш-функция от него. Выберем произвольное k и вычислим значение

$$r = \alpha^k y^{-h} \pmod p.$$

Если НОД (r, $p-1$) > 1, то выберем другое k и будем повторять вычисления до тех пор, пока не получим НОД (r, $p-1$) = 1. После этого вычисляем подпись s по формуле $s = k/r \pmod{p-1}$. Легко проверить, что полученные значения (r, s) удовлетворяют уравнению проверки подписи.

В случае системы ЭЦП с уравнением проверки подписи $\alpha^{hr} = y^s r \pmod p$ для произвольного значения s можно аналогичным образом без знания секретного ключа сформировать значение хэш-функции $h = k/r$, для которого (r, s) является правильной подписью. Однако теперь это не является принципиальным недостатком, поскольку при стойкой хэш-функции вычислительно невозможно подобрать документ, хэш-функция от которого равна наперед заданному случайному h. (В рассматриваемом случае это связано с тем, что при формировании r произвольно выбираются k и s, но не представляется возможным получить наперед заданные значения для r и h, т. е. последнее является вычислительно сложным.)

Если бы вместо хэш-функции использовалось значение сообщения m (где, допустим, $m < p$), то мы могли бы говорить, что в последнем примере имеется возможность сформировать случайное сообщение m и подпись к нему. Только в редких случаях использования систем ЭЦП такое обстоятельство является существенным недостатком, делающим недопустимым применение подобной ЭЦП. Использование значения хэш-функции (в уравнении проверки подписи) от сообщений любого (в том числе и малого) размера снимает эту проблему в случае ЭЦП, которые допускают возможность без знания секретного ключа найти некоторое случайное h и правильную к нему подпись (r, s). Многие варианты ЭЦП, рассматриваемые как стойкие, имеют эту особенность.

Применяя возведение значения r в степень s, h или r, можно получить очень большое число различных вариантов стойких ЭЦП, некоторые из которых заданы ниже соответствующими уравнениями проверки и вычисления подписи:

$$\alpha^h = y^s r^r \pmod p, \quad \text{и} \quad s = (h - kr)/x \pmod{p-1},$$

$$\alpha^h = y^r r^s \pmod p, \quad \text{и} \quad s = (h - xr)/k \pmod{p-1},$$

$$\alpha^s = y^r r^h \pmod p \quad \text{и} \quad s = xr + kh \pmod{p-1},$$

$$\alpha^r = y^h r^s \pmod p \quad \text{и} \quad s = (r - xh)/k \pmod{p-1},$$

$$\alpha^r = y^s r^h \pmod p \quad \text{и} \quad s = (r - kh)/x \pmod{p-1}.$$

Определенный интерес представляет ЭЦП с уравнением вычисления подписи $s = xr + kh \pmod{p-1}$, в котором отсутствует операция деления, т. е. не требуется вводить ограничение выбора значений взаимно простых с модулем $p - 1$. В первой и пятой системах ЭЦП секретный ключ надо выбирать таким образом, чтобы он был взаимно простым с $p - 1$, но это требуется выполнить только один раз до очередной смены секретного ключа. Во второй и четвертой системах ЭЦП на выбор секретного ключа специальных ограничений не накладывается, но при формировании подписи к каждому новому сообщению потребуется осуществить проверку выполнения условия НОД(k, $p - 1$) = 1. Во всех пяти указанных выше вариантах стойких ЭЦП можно без знания секретного ключа вычислить тройку чисел h, r и s – таких, что (r, s) является правильной подписью к h, но ситуацию спасает то, что для заранее заданного h вычислительно сложно подобрать соответствующую ему правильную подпись (r, s). Если нарушитель может предъявить подписанное значение хэш-функции h, то проверяющий запросит также и исходное сообщение, которое якобы подписано, а нахождение сообщения, хэш-функция от которого была бы равна наперед заданному случайному значению, является вычислительно невыполнимой задачей при использовании стойкой хэш-функции.

Для каждой из перечисленных выше пар уравнений сложность формирования правильной подписи без знания секретного ключа имеет тот же порядок сложности, что и задача дискретного логарифмирования. Вторая пара уравнений описывает систему ЭЦП Эль-Гамаля, которая представляется сейчас только частным вариантом ЭЦП, основанных на сложности задачи дискретного логарифмирования.

Заметим, что вместо степени r можно использовать степень $F(r)$, где $F(r)$ есть произвольная функция от r. Например, можно использовать такие пары уравнений:

$$\alpha^h = y^{F(r)} r^s \pmod p \quad \text{и} \quad s = \big(h - xF(r)\big)/k \pmod{p-1},$$

$$\alpha^{F(r)} = y^h r^s \pmod p \quad \text{и} \quad s = \big(F(r) - xh\big)/k \pmod{p-1}.$$

Однако это не снимает перечисленные замечания к рассматриваемым системам ЭЦП. Устранение недостатков может быть достигнуто применением в качестве степени некоторой стойкой хэш-функции $F'(r, h)$ от двух аргументов – h и r (легко видеть, что s должно входить в уравнения непосредственно, поскольку при использова-

нии хэш-функции от s даже при известном секретном ключе вычисление правильной подписи будет вычислительно неосуществимо). Эта модернизация вводит дополнительную процедуру – вычисление функции F или F', что несущественно увеличивает сложность формирования и проверки подписи. Возможны варианты устранения упомянутых недостатков и другими способами, например, использованием в качестве степеней обоих функций F и F' или функции $F''(h)$ совместно с исходным значением h.

Использование только значения $F''(h)$ в качестве степени (или сомножителя степени) вводит определенную специфику, например, может оказаться, что, подбирая совместно r, $F''(h)$ и s, можно найти такие значения этих трех параметров, которые удовлетворяют уравнению проверки подписи. Однако при использовании стойкой хэш-функции в этом случае практически неосуществимо нахождение значения h, которое бы давало нужное значение $F''(h)$. Если бы в уравнение проверки подписи входило непосредственно значение подписываемого сообщения m, то введение вместо m значения $F''(m)$ имело бы принципиальный характер. При изначальном использовании значения стойкой хэш-функции $h(m)$ от сообщения m введение функции $F''(h)$ принципиальных изменений не вносит, поскольку значения некоторых подобранных h, s и r, которые удовлетворяют уравнению проверки подписи, являются случайными, а вычисление сообщения, которое якобы подписано, есть практически неразрешимая задача. Поэтому замена $h(m)$ на $F''(h(m))$ в определенном смысле эквивалентна простому изменению алгоритма хэширования исходного документа, т. е. это не затрагивает свойств уравнений проверки и формирования подписи. Сразу заметим, что введение функций F, F' и F'' по отдельности или всех сразу не приведет автоматически к устранению всех слабостей системы ЭЦП. После их введения требуется провести соответствующий анализ вновь сформированной системы ЭЦП.

Например, уравнение проверки подписи $\alpha^{sF(r,h)} = y^h r \pmod p$ соответствует слабой ЭЦП. Покажем, как можно подделать подпись без знания секретного ключа. Берем некоторый документ и вычисляем от него хэш-функцию h. Далее действуем по следующему алгоритму.

1. Выбрать некоторый параметр z, такой, что $z < p - 1$.
2. Вычислить $r = \alpha^z y^{-h}$ и $F(r,h)$.
3. Если НОД($F(r,h)$, $p - 1$) > 1, то повторить шаги 1 и 2 пока не выполнится условие НОД($F(r,h)$, $p - 1$) = 1.
4. Вычислить $s = z/F(r, h)$.
5. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в левую часть уравнения проверки подписи значение $s = z/F(r, h)$, а в правую значение $r = \alpha^z y^{-h}$, легко показать, что оно выполняется:

$$\alpha^z = y^h \alpha^z y^{-h} \pmod p.$$

3.3.2. Сокращение длины подписи

Кроме рассмотренных выше вариантов стойких ЭЦП, основанных на задаче дискретного логарифмирования, существует большое число других. Для практического использования можно рекомендовать те из них, которые требуют выполнения проверки и формирования подписи с наименьшей трудоемкостью, но стойкость которых не ниже сложности задачи дискретного логарифмирования. Заметим также, что уравнение проверки подписи и соответствующее ему уравнение формирования подписи могут быть заданы в различных формах, например, представленных в табл. 3.1.

Таблица 3.1

Варианты задания систем ЭЦП

Уравнение проверки подписи	**Уравнение для вычисления *s***	
	По модулю $p-1$ (α – первообразный корень по mod p)	По модулю q (α – число, принадлежащее показателю q по mod p)
$\alpha^a = y^b r^c \pmod p$	$a = bx + ck \pmod{p-1}$	$a = bx + ck \mod q$
$y^a = \alpha^b r^c \pmod p$	$ax = b + ck \pmod{p-1}$	$ax = b + ck \mod q$
$r^a = \alpha^b y^c \pmod p$	$ak = b + cx \pmod{p-1}$	$ak = b + cx \mod q$
$r^a \alpha^b = y^c \pmod p$	$ak + b = cx \pmod{p-1}$	$ak + b = cx \mod q$
$r^a \alpha^b y^c = 1 \pmod p$	$ak + b + cx = 1 \pmod{p-1}$	$ak + b + cx = 1 \mod q$

Уравнение проверки подписи может быть задано в двух вариантах: по модулю $p-1$ или по модулю некоторого числа q, являющегося делителем числа $p-1$ и имеющего размер 160 и более бит. Известно, что любой делитель числа $p-1$ является показателем по модулю простого p. Для любого показателя существуют числа β, не превосходящие $p-2$, для которых выполняются следующие условия: 1) $\beta^q = 1 \pmod p$ и 2) все числа $\beta, \beta^1, \beta^2, \ldots, \beta^q$ являются несравнимыми между собой по модулю p. Эти условия обеспечивают возможность использования уравнений формирования подписи по модулю q, который по размеру значительно меньше p, что приводит к получению значений $s < q$. Причем для формирования подписи без знания секретного ключа требуется решить задачу дискретного логарифмирования по модулю p, т.е. такое сокращение размера числа s не снижает исходной стойкости системы ЭЦП.

Таким образом, использование уравнения вычисления подписи по модулю делителя числа $p-1$ позволяет сократить длину одного из параметров подписи, а именно значения s. При этом независимо от длины простого модуля p, последний всегда можно выбирать таким образом, чтобы длина q не превосходила 160 – 256 бит. Это позволяет сократить длину подписи (s, r) почти в два раза по сравнению со случаем использования уравнения вычисления подписи по модулю $p-1$. Действительно, значение r ($r < p$) имеет длину примерно равную длине p, которая по соображениям

обеспечения высокой стойкости должна быть около 1000 бит или более. Для указанной длины модуля p при использовании 160-битового показателя q можно оценить, что длина подписи сокращается примерно в 1.7 раза. При этом с целью повышения стойкости ЭЦП можно увеличивать размер модуля p, сохраняя размер показателя q. Стойкость ЭЦП будет определяться только длиной простого числа p и правильностью выбора уравнения проверки подписи.

При выборе варианта с вычислением подписи по модулю q мы будем полагать, что в качестве α выбирается некоторое число, относящееся к показателю q по модулю p, т. е. такое число, для которого выполняется соотношение

$$\alpha^q = 1 \pmod{p},$$

где q является делителем числа $p - 1$ требуемого размера. Предполагается, что при формировании простого числа p обеспечивается наличие делителя, имеющего нужный размер, например 160 или 256 бит. При этом в разложении числа $p - 1$ на множители желательно иметь один из делителей большого размера, существенно превышающего размер q, поскольку наличие большого простого делителя в существенной степени сложность задачи дискретного логарифмирования по модулю p. Существуют различные способы формирования простых чисел, удовлетворяющих этим условиям.

В табл. 3.2, составленной по данным [113], приводятся приемлемые для применения варианты ЭЦП, заданные уравнениями проверки и формирования подписи.

Таблица 3.2

Системы ЭЦП с сокращенной длиной подписи,
где $r' = r \pmod{q}$ и $\alpha^q = 1 \pmod{p}$

Уравнение для вычисления s	Уравнение проверки подписи
$r'k = s + hx \pmod{q}$	$r^{r'} = \alpha^s y^h \pmod{p}$
$r'k = h + sx \pmod{q}$	$r^{r'} = \alpha^h y^s \pmod{p}$
$sk = r' + hx \pmod{q}$	$r^s = \alpha^{r'} y^h \pmod{p}$
$sk = h + r'x \pmod{q}$	$r^s = \alpha^h y^{r'} \pmod{p}$
$hk = s + r'x \pmod{q}$	$r^h = \alpha^s y^{r'} \pmod{p}$
$hk = r' + sx \pmod{q}$	$r^h = \alpha^{r'} y^s \pmod{p}$

В системах ЭЦП, представленных в табл. 3.2, в качестве r' можно взять значение некоторой хэш-функции F от r, т. е. взять $r' = F(r)$.

Имеется также возможность сократить второй параметр подписи, а именно, значение r. Идея такого сокращения опирается на тот факт, что, если правая и левая часть уравнения проверки подписи сравнимы по модулю p, то остатки от деления левой и правой частей на p будут сравнимы по модулю q. Осуществляется это сле-

дующим образом. Выбирается уравнение проверки подписи типа $R^a = \alpha^b y^c \pmod p$ и преобразуется к эквивалентному виду

$$R \equiv \alpha^u y^v \pmod p, \text{ т. е. } (R \bmod p) = (\alpha^u y^v \bmod p),$$

где $u = ba^{-1} \pmod q$, $v = ca^{-1} \pmod q$, а R формируется с использованием случайно выбираемого числа k, которое является разовым секретным ключом, в соответствии с формулой $R = \alpha^k \bmod p$. Затем уравнение проверки преобразуют к виду

$$(R \bmod p) \equiv (\alpha^u y^v \bmod p) \pmod q, \text{ т. е.}$$

$$\big((R \bmod p) \bmod q\big) = \big((\alpha^u y^v \bmod p) \bmod q\big).$$

В последней формуле следует задать зависимость пары чисел u и v от хэш-функции подписываемого документа h и подписи (r, s), где $r = \big((R \bmod p) \bmod q\big)$. Теперь уже оба элемента подписи (т. е. r и s) имеют значение, не превышающее q. Длина подписи сокращается в 4 – 6 раз (в зависимости от длины модулей q и p). Подобную конструкцию имеют первые варианты американского и российского стандартов цифровой подписи, принятые в 90–х годах.

В качестве параметра r можно взять значение хэш-функции: $r = F(R \bmod p)$. В этом случае уравнение проверки подписи имеет вид

$$r = F(\alpha^u y^v \bmod p).$$

Заметим, что в последнем уравнении пара чисел u и v зависит от h и $r = F(R \bmod p)$. Таблица 3.3 показывает возможные варианты представления параметров u и v через h, r и s. Заметим, что в этих вариантах нет проблемы подбора значений обратных числам h, r и s, связанной с обеспечением взаимной простоты с модулем, поскольку выражение для формирования подписи записывается по модулю простого числа q. Это является еще одним преимуществом перехода к модулю q.

Таблица 3.3

Варианты задания ЭЦП с сокращенной длиной подписи (r, s).

Параметр *u*	**Параметр *v***
$r^{-1}s \bmod q$	$r^{-1}h \bmod q$
$r^{-1}h^{-1}s \bmod q$	$h^{-1}r^{-1} \bmod q$
$r^{-1}s \bmod q$	$h^{-1}r^{-1} \bmod q$
$h^{-1}s \bmod q$	$h^{-1}r^{-1} \bmod q$
$h^{-1}r \bmod q$	$h^{-1}s^{-1} \bmod q$

Весьма существенным достоинством ЭЦП, основанных на сложности задачи дискретного логарифмирования, является то, что они позволяют разработать вариан-

ты со сравнительно малым размером подписи. При этом если будут предложены новые методы дискретного логарифмирования, которые потребуют увеличения размера модуля p, то последнее можно сделать без увеличения размера подписи. В случае цифровой подписи RSA увеличение размера модуля приводит к увеличению размера подписи.

3.3.3. Примеры анализа слабых ЭЦП

Под слабыми системами ЭЦП будем понимать те из них, которые допускают подделку подписи. Последнее означает формирование подписи к некоторому априори заданному сообщению без знания секретного ключа. Как мы уже упоминали, многие системы ЭЦП, рассматриваемые как стойкие, допускают возможность сформировать без знания секретного ключа случайные значения h и (r, s), которые удовлетворяют уравнению проверки подписи, т. е. (r, s) является правильной подписью к h. Однако эти же системы ЭЦП практически не позволяют вычислить правильную подпись к заданному документу, или заданной хэш-функции, если секретный ключ неизвестен.

Рассматриваемые ниже примеры слабых ЭЦП составлены в близкой аналогии к построению стойких ЭЦП. Однако некоторые внешне незначительные отличия вносят недопустимую слабость – возможность формирования потенциальным нарушителем (который не знает секретного ключа) правильной подписи к заданному значению h. В основе такой подделки лежит идея представления параметра r в виде $r = \alpha^z y^w \bmod p$, где значения степеней w и z выбираются заранее (фиксируются), а затем выражаются через значения h, s и r (вместо r может фигурировать r', $F(r)$ или $F'(h,r)$) в зависимости от конкретного вида уравнения проверки подписи. При этом получается такая ситуация: значения r, r', $F(r)$ и $F'(h,r)$ зависят от одного или обоих значений w и z, а последние зависят от r. Это противоречие снимается тем, что величины w и z рассматриваются как зафиксированные, что достигается подгонкой за счет выбора значений h и s с учетом вычисленных (по фиксированным w и z) значений r, r', $F(r)$ и $F'(h,r)$.

Пример 3.1. Уравнение подписи вида $r = y^h \alpha^{rs} \pmod p$. Осуществим подделку подписи без знания секретного ключа. Пусть дан некоторый документ, хэш-функция от которого равна h. Далее действуем по следующему алгоритму.

1. Выбрать некоторое значение z.
2. Вычислить $r = \alpha^z y^h \pmod p$.
3. Если НОД$(r, p-1) > 1$, то повторить шаги 1 и 2, пока не выполнится условие НОД$(r, p-1) = 1$.
4. Вычислить $s = z/r \pmod{p-1}$, т. е. получаем $z = sr \pmod{p-1}$.
5. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в левую часть уравнения проверки подписи значение $r = \alpha^{sr} y^{h} (\mathrm{mod}\ p)$, видим, что правая и левая части совпадают, т. е. сформированная подпись является правильной.

Пример 3.2. Уравнение подписи вида $\alpha^{r} = y^{hs} r^{s} (\mathrm{mod}\ p)$. Пусть дан некоторый документ, хэш-функция от которого равна h. Злоумышленник может действовать по следующей схеме.

1. Выбрать некоторое значение z, такое, что НОД$(z, p-1) = 1$.
2. Вычислить $r = \alpha^{z} y^{-h} \ (\mathrm{mod}\ p)$.
3. Вычислить $s = r/z \ (\mathrm{mod}\ p-1)$.
4. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в правую часть уравнения проверки подписи значения $sz = r \ (\mathrm{mod}\ p-1)$ и $r = \alpha^{z} y^{-h} \ (\mathrm{mod}\ p)$, получаем $y^{hs} \alpha^{zs} y^{-hs} = \alpha^{r} \ (\mathrm{mod}\ p)$, т. е. правая часть совпадает с левой. Это означает, что сформированная подпись правильная.

Пример 3.3. Уравнение подписи вида $\alpha^{hs} = y^{r} r^{s} (\mathrm{mod}\ p)$. Пусть дан некоторый документ, хэш-функция от которого равна h. Злоумышленник может действовать по следующей схеме.

1. Выбрать некоторое значение w, такое, что НОД$(w, p-1) = 1$.
2. Вычислить $r = \alpha^{h} y^{-w} \ (\mathrm{mod}\ p)$.
3. Вычислить $s = r/w \ (\mathrm{mod}\ p-1)$.
4. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в правую часть уравнения проверки подписи значение $r = ws \ (\mathrm{mod}\ p-1)$ и $r = \alpha^{h} y^{-w} \ (\mathrm{mod}\ p)$, получаем $y^{ws} \alpha^{hs} y^{-ws} = \alpha^{hs} \ (\mathrm{mod}\ p)$. Совпадение левой и правой частей уравнения проверки показывает, что сформированная (без знания секретного ключа) подпись является правильной.

Пример 3.4. Уравнение подписи вида $\alpha^{hs} = y^{r f(h)} r^{s} (\mathrm{mod}\ p)$, где $f(h)$ есть произвольная функция от h. Пусть дан некоторый документ, хэш-функция от которого равна h. Подделка подписи осуществляется в соответствии со следующим алгоритмом.

1. Выбрать некоторое значение w, такое, что НОД$(w, p-1) = 1$.
2. Вычислить $r = \alpha^{h} y^{-w} \ (\mathrm{mod}\ p)$.
3. Вычислить $s = rf(h)/w \ (\mathrm{mod}\ p-1)$.
4. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в правую часть уравнения проверки подписи значение произведения $rf(h) = ws \ (\mathrm{mod}\ p-1)$ и $r = \alpha^{h} y^{-w} \ (\mathrm{mod}\ p)$, получаем $y^{ws} \alpha^{hs} y^{-ws} = \alpha^{hs} \ (\mathrm{mod}\ p)$. Совпадение левой и правой частей уравнения проверки показывает, что сформированная подпись (r, s) является правильной.

Пример 3.5. Уравнение подписи вида $r^{f(h)} = \alpha^{h} y^{rs} \ (\mathrm{mod}\ p)$, где $f(h)$ есть произвольная функция от h. Для того чтобы подделать подпись к некоторому документу, нарушитель может действовать следующим образом.

1. Вычислить хэш-функцию h от документа.
2. Если НОД($f(h)$, $p-1$) > 1, то модифицировать документ, сохраняя его смысловое содержание, и повторить шаг 1, в противном случае перейти к следующему шагу алгоритма.
3. Выбрать значения $z = h/f(h) \pmod{p-1}$ и $w = rs/f(h) \pmod{p-1}$.
4. Вычислить $r = \alpha^z y^w \pmod{p}$.
5. Вычислить $s = wf(h)/r \pmod{p-1}$.
6. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Подставляя в левую часть уравнения проверки подписи значение $r = \alpha^z y^w \pmod{p}$, получаем $r^{f(h)} = \alpha^{zf(h)} y^{wf(h)} = \alpha^h y^{rs} \pmod{p}$. Совпадение левой и правой частей уравнения проверки показывает правильность, сформированной подписи (r, s) к документу, подобранному нарушителем.

Пример 3.6. Уравнение подписи вида $r^{F(r)} = \alpha^{hs} y^{sf(h)} \pmod{p}$, где $f(h)$ и $F(r)$ есть произвольные функции от h и r, соответственно. Для того чтобы подделать подпись к некоторому документу, нарушитель может действовать следующим образом.

Пусть дан некоторый документ. Подделка подписи осуществляется в соответствии со следующим алгоритмом.

1. Вычислить хэш-функцию h от документа.
2. Если НОД($f(h)$, $p-1$) > 1, то модифицировать документ, сохраняя его смысловое содержание, и повторить шаг 1, в противном случае перейти к следующему шагу алгоритма.
3. Выбрать пару значений z и w таких, что $z/w = h/f(h)$.
4. Вычислить $r = \alpha^z y^w \pmod{p}$ и $F(r)$.
5. Если НОД($F(r)$, $p-1$) > 1, то повторить шаги 3 и 4, в противном случае перейти к следующему шагу алгоритма.
6. Вычислить $s = wF(r) / f(h) \pmod{p-1}$.
7. Предъявить в качестве подписи к хэш-функции h пару чисел (r, s).

Из выражения в п. 6 легко получить

$$w = sf(h)/F(r) \pmod{p-1} \quad \text{и} \quad z = wh/f(h) = sh/F(r) \pmod{p-1}.$$

Подставляя в левую часть уравнения проверки подписи значение $r = \alpha^z y^w \pmod{p}$, получаем $r^{F(r)} = \alpha^{zF(r)} y^{wF(r)} = \alpha^{sh} y^{sf(h)} \pmod{p}$. Совпадение левой и правой частей уравнения проверки показывает правильность сформированной подписи (r, s) к документу.

Способы подделки подписи, рассмотренные в этих примерах, могут быть использованы и в схемах ЭЦП с сокращенной подписью, причем в некоторых случаях задача формирования подписи без знания секретного ключа упрощается, поскольку не потребуется подбирать некоторые значения таким образом, чтобы они были вза-

имно простыми с $p-1$, поскольку модуль q, используемый вместо $p-1$, является простым.

3.3.4. Системы ЭЦП с дополнительными свойствами

Представляет интерес построить такую систему ЭЦП, в который было бы вычислительно сложным сформировать правильную подпись не только к заранее заданному значению хэш-функции, но и к значению h, которое подбирается наряду со значениями r и s. При наличии трех параметров подгонки задача формирования подписи без знания секретного ключа сильно упрощается, однако в этом случае все три указанных значения заранее не предопределены и получаются случайными. При такой "подгонке" нарушитель может найти "правильную" пару h и (r, s), но подобрать сообщение (документ), соответствующий полученному значению h, он не сможет (предполагается, что используется стойкая хэш-функция).

Рассмотрим, как можно получить "правильную" пару h и (r, s) в случае ЭЦП Эль-Гамаля, уравнение проверки подписи которой имеет вид:

$$\alpha^h = y^r r^s \pmod{p}.$$

Зададим следующую структуру числа $r = \alpha^z y^{-w} \pmod{p}$. Выберем случайные значения z и w такие, что НОД$(z, p-1) = 1$ и НОД$(w, p-1) = 1$, и вычислим значение $r = \alpha^z y^{-w} \pmod{p}$. Значения h и s определим по формулам:

$$h = sz \pmod{p-1} \quad \text{и} \quad s = r/w \pmod{p-1}.$$

Найденные значения h и (r, s) удовлетворяют уравнению проверки подписи. Действительно, правая часть уравнения Эль-Гамаля переписывается в следующем виде $y^r \alpha^{zs} y^{-ws} = y^r \alpha^h y^{-wr/w} = \alpha^h \pmod{p}$. Этот анализ показывает, что в уравнении Эль-Гамаля нежелательно использовать значение сообщения m, поскольку в этом случае имеется возможность сформировать подпись к каким-то случайным сообщениям. Данный факт имеет место и в ЭЦП RSA, но не является катастрофическим для подавляющего большинства применений систем цифровой подписи. Однако в некоторых случаях такая особенность может оказаться принципиальной слабостью.

Рассмотрим еще одно интересное сравнение с системой RSA. В последней проверка подписи заключается в расшифровании (восстановлении) подписанного сообщения. Это означает, что для осмысленных сообщений (т. е. имеющих некоторую ожидаемую структуру) достаточно передать проверяющему только подпись. Процедура проверки подписи (расшифрование по открытому ключу) восстанавливает сообщение. Как мы видели выше в рассмотренных системах ЭЦП, основанных на задаче дискретного логарифмирования, требуется предъявить одновременно и подпись, и подписываемое сообщение. Представляет интерес преобразовать последний тип ЭЦП таким образом, чтобы процедура проверки подписи восстанавливала сообщение. Это можно сделать, используя следующее уравнение проверки подписи:

$$m = \alpha^s y^{F(r)} r \pmod{p},$$

где $F(r)$ – некоторая функция от r, например $F(r) = r' = r \bmod q$. Формирование цифровой подписи к сообщению m осуществляется следующим образом.

Выбрать случайное число k и вычислить значение r по формуле:

$$r = m\alpha^k \pmod p.$$

Значение s вычисляется из соотношения

$$s = -xF(r) - k \pmod{p-1}.$$

Проверка подписи (r, s) заключается в подстановке данных значений в уравнение проверки подписи, приводящей к получению «осмысленного» сообщения с ожидаемой структурой. Фактически проверка состоит в расшифровании криптограммы (r, s). Очевидно, что подстановка в уравнение проверки любой пары случайных значений r и s приведет к получению некоторого значения m, которое будет случайным. Если в некотором конкретном протоколе этого достаточно, чтобы сделать заключение, что это незаконная подпись, то такую систему ЭЦП с восстановлением сообщения допустимо использовать. Однако в ряде приложений требуется подписывать случайные сообщения. В таких случаях можно использовать следующую схему:

- К сообщению m, которое необходимо подписать и передать по открытому каналу, присоединяется заранее условленный двоичный вектор v, например имеющий размер $l = 64$ бит: $M \to m \mid v$.
- Вырабатывается подпись (r, s) для сообщения M.
- Проверяющая сторона по значению подписи (r, s) вычисляет текст M', соответствующее ему значение двоичного вектора $v' = M' \bmod 2^l$ и сообщение $m' = M' \operatorname{div} 2^l$.
- Если v' равно заранее условленному значению v, т.е. если выполняется условие $v' = v$, то принимается решение, что сообщение m' подписано владельцем секретного ключа, соответствующего открытому ключу, с помощью которого выполнялась проверка подписи. (Вероятность того, что случайное сообщение может быть принято за подписанное достаточно мала и составляет 2^{-l}.)

Другим способом обеспечить возможность стойкого подписывания произвольных сообщений является использование значения некоторой хэш-функции $h(m)$ от сообщения m в качестве степени при параметре r в уравнении проверки подписи. В такой схеме ЭЦП с восстановлением сообщения параметр r формируется в соответствии с формулой

$$r = m^{1/h}\alpha^k \pmod p.$$

В этом случае уравнение проверки подписи имеет вид

$$m = \alpha^s y^r r^h \pmod p.$$

Значение параметра вычисляется из соотношения

$$s = -xr - kh \pmod{p-1}.$$

Недостатком этого варианта ЭЦП является то, что должно выполняться условие НОД(h, $p-1$). Поэтому некоторые сообщения m потребуется модифицировать с сохранением смыслового содержания несколько раз, пока не будет получено от него значение h, взаимно простое с $p-1$. Для практики это является существенным изъяном. В этой схеме в качестве подписи фигурируют уже три параметра (r, s, h), что является определенным недостатком, связанным с увеличением ее размера. Этот недостаток несколько сглаживается, если использовать схемы ЭЦП с сокращенной длиной подписи. Вместо функции $h(m)$ можно использовать само значение m, но тогда исчезает свойство восстановления сообщения из подписи, что являлось исходной целью видоизменения ЭЦП.

В табл. 3.4 приведены более сложные схемы построения систем ЭЦП с восстановлением сообщения, в которых значение r вычисляется по формуле $r = m\alpha^k \pmod{p}$, а α относится к показателю q|($p-1$), т.е. $\alpha^q \bmod p = 1$. В этих схемах обеспечивается уменьшение размера параметра s.

Таблица 3.4

Системы ЭЦП с восстановлением сообщения, где α целое число – такое, что $\alpha^q = 1 \pmod{p}$, $r' = r \bmod q$ и $f(h)$, $f(r)$ и $f(h, r)$ есть функции с областью значений $0 < f < q$

Уравнение для вычисления значения s	**Уравнение проверки подписи**	**Подпись к m**
$s + xrf(h) + k = 0 \pmod{q}$	$m = \alpha^s y^{rf(h)} r \pmod{p}$	(r, s, h)
$s + xf(h, r) + k = 0 \pmod{q}$	$m = \alpha^s y^{f(h,r)} r \pmod{p}$	(r, s, h)
$r' + xsf(h) + k = 0 \pmod{q}$	$m = \alpha^{r'} y^{sf(h)} r \pmod{p}$	(r, s, h)
$r' + xs + k = 0 \pmod{q}$	$m = \alpha^{r'} y^{s} r \pmod{p}$	(r, s)
$r' + xs\, f(r) + k = 0 \pmod{q}$	$m = \alpha^{r'} y^{sf(r)} r \pmod{p}$	(r, s)
$s\, f(r) + xr' + k = 0 \bmod q)$	$m = \alpha^{sf(r)} y^{r'} r' \pmod{p}$	(r, s)
$r's + x\, f(h, r) + k = 0 \pmod{q}$	$m = \alpha^{r's} y^{f(h,r)} r \pmod{p}$	(r, s, h)
$rf(h) + xs + k = 0 \pmod{q}$	$m = \alpha^{rf(h)} y^{s} r \pmod{p}$	(r, s, h)

Выше мы описали системы ЭЦП, основанные на сложности задачи дискретного логарифмирования по модулю простого числа. Очевидно, что для каждого из рассмотренных вариантов ЭЦП данного типа может быть предложен аналог для составного модуля N. Уравнения проверки и формирования подписи сохраняют свой вид полностью, если учесть что $p-1$ есть функция Эйлера от модуля. При этом все ва-

рианты сокращения размера элементов подписи r и s и уравнение проверки подписи с восстановлением сообщения могут быть также реализованы. Отметим некоторые особенности, связанные с использованием составного модуля. При специальном выборе составного модуля можно сделать так, что функция Эйлера от него будет секретным элементом, т. е. частью секретного ключа, который известен только одному пользователю. С этой целью можно предусмотреть следующие действия со стороны владельца секретного ключа. Он выбирает два больших простых числа p и q и перемножает их, получая модуль $N = pq$. Значение N принимается в качестве части открытого ключа (y, N), а значения простых множителей держатся в секрете или уничтожаются после вычисления $\varphi(N) = (p - 1)(q - 1)$. В этом случае формирование подписи к сообщению m для случая уравнения проверки вида $\alpha^h = y^s \pmod p$ не позволит проверяющему вычислить секретный ключ, что было возможным при простом модуле p. Сложной проблемой для проверяющего является вычисление значения $\varphi(N)$, поскольку он не знает разложения модуля на множители. Систему ЭЦП с таким уравнением проверки можно назвать одноразовой ЭЦП, поскольку при формировании подписей s_1 и s_2 к двум различным сообщениям возникают предпосылки для вычисления $\varphi(N)$ и секретного ключа x. Действительно при наличии двух подписей имеется следующая система из двух уравнений с неизвестными x и $\varphi(N)$:

$$h(m) = xs_1 \bmod \varphi(N),$$
$$h(m) = xs_2 \bmod \varphi(N).$$

Замечание

Практические неудобства разовой подписи очевидны: после подписания одного документа текущий открытый ключ не должен быть использован для формирования подписи ко второму документу (хотя для формирования общего секрета в соответствии с системой Диффи-Хеллмана он может использоваться далее). Ее рассмотрение имеет только методическое значение. В этом плане следует также указать, что значение α для некоторых N (разные пользователи формируют разные значения модуля) уже не будет первообразным корнем, однако на корректность работы рассматриваемой системы разовой подписи и ее стойкости это практически не влияет (вероятность того, что для какого-то N число α окажется относящимся к показателю малого размера является ничтожно малой). Длина числа α может быть выбрана сравнительно небольшой (меньше размера используемых значений p и q), что позволяет не рассматривать случаи, когда НОД$(\alpha, p) \neq 1$ или НОД$(\alpha, q) \neq 1$.

Для обеспечения возможности подписывания многих сообщений без смены секретного ключа x потребуется ввести использование разового открытого ключа $r = \alpha^k \bmod N$. В результате приходим к примерным вариантам ЭЦП, представленных в табл. 3.5. В данных примерах одна и та же хэш-функция h используется для хэширования сообщения m и значения r.

Таблица 3.5

Системы ЭЦП с составным модулем $N = pq$

Уравнение для вычисления s	**Уравнение проверки подписи**	**Показатель, к которому относится α**	**Подпись к m**
$h(m) = xr + ks \pmod{\varphi(N)}$	$\alpha^{h(m)} = y^r r^s \bmod N$	$\varphi(N)$	(r, s)
$h(m) = xr + ks \pmod{\varphi(q)}$	$\alpha^{h(m)} = y^r r^s \bmod N$	$\varphi(q)$	(r, s)
$h(m) = xr + ks \pmod{\delta}$	$\alpha^{h(m)} = y^r r^s \bmod N$	$\delta \mid \varphi(q)$	(r, s)
$h(r) + xs + k = 0 \pmod{\delta}$, где $r = m\alpha^k \bmod N$	$m = \alpha^{h(r)} y^s r \bmod N$	$\delta \mid \varphi(q)$	(r, s)
$h(r)k = s + h(m)x \pmod{\delta}$	$h(r) = h(\alpha^{s/h(r)} y^{h(m)/h(r)} \bmod N)$	$\delta \mid \varphi(q)$	$(h(r), s)$
$r'k = s + h(m)x \pmod{\delta}$, где $r' = (\alpha^k \bmod N) \bmod \delta$	$r' = (\alpha^{s/r'} y^{h(m)/r'} \bmod N) \bmod \delta$	$\delta \mid \varphi(q)$	(r', s)

3.3.5. Слепая подпись

Ряд важных для практического применения криптографических систем включают в себя как составную часть протокол слепой подписи. К ним относятся системы тайного электронного голосования и электронных денег. Суть слепой подписи заключается в том, что владелец секретного ключа должен иметь возможность осуществить подписывание сообщения, представленного в зашифрованной форме. Необходимо, чтобы сторона, подготовившая сообщение (документ), была уверена в том, что подписавший не прочтет его. Как это ни странно на первый взгляд, но целесообразность в системах слепой подписи имеется. Более того, без них не обойтись. Остроумное и красивое решение построения слепой подписи на основе использования системы RSA было предложено Чаумом [63-65], оно основано на использовании особенностей преобразований системы RSA. Возможно ли построение некоторого варианта слепой подписи с использованием ЭЦП, основанных на сложности дискретного логарифмирования? Легко построить протокол слепой подписи с использованием любой системы ЭЦП. Это решение подсказывает сам принцип вычисления подписи к сообщениям большого размера, используемый на практике. Как известно, документы (сообщения) большого размера подписываются следующим образом.

Пусть дано сообщение m. От сообщения m вычисляется хэш-функция $h(m)$, значение которой имеет размер 160- или 256-битового числа. Формируется подпись S к значению хэш-функции. Если используемая хэш-функция является криптографически стойкой, то можно считать, что подписывание значения хэш-функции эквивалентно подписыванию самого сообщения m. Учитывая, что по значению хэш-функции невозможно восстановить само сообщение, приходим к следующему варианту универсальной слепой подписи. Абонент X подготавливает некоторое сообщение m, которое ему нужно подписать у абонента Y таким образом, чтобы последний не знал его содержания, но в то же время подпись была правильной (действительной). Затем X вычисляет по заранее оговоренному алгоритму значение хэш-функции $h(m)$, которое он и предоставляет абоненту Y для подписывания, а само сообщение m

держит в секрете. Абонент Y подписывает значение $h = h(m)$, т. е. вычисляет значение S. Естественно, что по значению h он не может определить m. Абонент X получает значение S, после чего, когда потребуется, он может представить сообщение m и правильную к нему подпись S. Достоинством этого универсального варианта слепой подписи является то, что он работает эффективно с сообщениями любого размера.

Данный вариант универсальной слепой подписи подчеркивает проблему обеспечения анонимности в полном объеме. Предназначением систем слепой подписи является обеспечение анонимности (неотслеживаемости), но подписывающий, получив возможность ознакомиться с некоторым подписанным им «вслепую» документом, имеет возможность вычислить значение хэш-функции и сравнить со списком хэш-функций с указанием лиц, которые представляли эти хэш-функции для подписывания документов «вслепую». Таким образом, решение задачи обеспечения анонимности в полном объеме на основе описанной системы универсальной слепой подписи с очевидностью требует использования других дополнительных механизмов.

Анонимность необходима, например, в системах электронных денег, где подписывающим выступает банк, который подписывает вслепую электронные банкноты одним клиентам (покупатели), тогда как представляют электронные деньги для начисления на свой счет другие лица (продавцы). Анонимность электронных денег заключается в том, что банк не должен иметь возможность определить то лицо, от которого продавец получил деньги при продаже своего товара.

На основе сложности задачи дискретного логарифмирования могут быть разработаны различные частные варианты слепой подписи. Рассмотрим систему ЭЦП с восстановлением сообщения, со следующим уравнением проверки подписи:

$$m = \alpha^s y^r r \pmod{p},$$

где $r = m\alpha^k \pmod{p}$. На основе этой ЭЦП легко осуществить подписывание «вслепую». Для этого приготовим документ $m' = m\alpha^t \pmod{p}$, где t – случайное число, не превышающее $p - 1$. Представляем m' для подписывания. От подписывающего получаем подпись (s', r). На основе полученной подписи формируем новую подпись (s, r), где $s = s' - t \pmod{p-1}$, которая является правильной для сообщения m. Действительно, подпись (s', r) удовлетворяет уравнению

$$m' = \alpha^{s'} y^r r = \alpha^{s'} y^r r \pmod{p}.$$

Разделив по модулю p левую и правую часть уравнения на α^t, получаем

$$m = m'\alpha^{-t} = \alpha^{s'-t} y^r r = \alpha^s y^r r \pmod{p}.$$

Следовательно, сформированная нами подпись (s, r) является верной для сообщения m. Если значение $s = s' - t \pmod{p-1}$ держать в секрете от подписывающего, то он не сможет определить значения m. В противном случае он может вычислить $m = m'\alpha^{-t}$. Проблема обеспечения анонимности остается открытой и для этой схемы. Действительно, подписывающий может вести учет значений m' и r. (На самом деле для того, чтобы нарушить анонимность достаточно фиксировать только r.)

Свободной от указанного недостатка является схема слепой подписи Чаума (см. 10.1.2). Действительно, некоторое лицо формирует значение m', которое связано с документом m, который нужно подписать «вслепую», следующим соотношением $m' = k^e m \pmod{N}$, где k – некоторое случайное число, неизвестное подписывающему,

и e – экспонента открытого ключа (e, N) подписывающего. Последний, используя свой секретный ключ, формирует к сообщению m' подпись $s' = (k^e m)^d = ks \pmod N$, где $s = m^d \bmod N$ есть правильная подпись к документу m. Подписавший не может определить значение s по s', так как он не знает значения k. Даже при ведении им учета подписанных значений m' с указанием лиц, которые эти значения формировали для подписывания, при предъявлении ему в дальнейшем соответствующих друг другу значений M и S он не сможет доказательно выявить значение M', содержавшее в себе момент подписывания документ M. Действительно, для любой пары m' и s' из учетного списка существует некоторое k, такое, что $m' = k^e M \pmod N$. При этом для соответствующих значений s' и S будет выполняется условие $s' = kS$. Более того, подписывающий не сможет даже убедительно доказать, что на момент подписывания он не был ознакомлен с документом M.

3.3.6 Проблема бесключевого шифрования

В криптоанализе известно понятие бесключевого чтения, заключающееся в чтении шифртекста без восстановления секретного ключа, с использованием которого была получена криптограмма. А возможно ли осуществить «бесключевое шифрование», т. е. некоторое преобразование исходного текста таким образом, чтобы получатель смог его восстановить, а нарушитель, перехвативший преобразованный текст, не смог? До открытия Диффи и Хеллманом двухключевой криптографии никто всерьез бы не воспринял постановку такой задачи. Двухключевая криптография по своей природе связана с взаимодействием пользователей криптосистемы. Пожалуй, упомянутую задачу невозможно решить с использованием только одной посылки по каналу связи. А вот с использованием протоколов можно попытаться. Уточним постановку задачи: разработать протокол «бесключевого шифрования», в котором не используется передача ключа (секретного или открытого), но который обеспечивает защищенную передачу сообщений по открытому каналу.

Двухключевая (асимметричная) криптография опирается на распределение открытых ключей и решение проблемы их аутентификации другими методами, т. е. для того, чтобы некий двухключевой шифр обеспечивал решение задачи организации секретной связи с использованием открытых каналов, необходимо предварительно решить задачу аутентификации распределяемых открытых ключей. Огромные преимущества шифров с открытым ключом обусловлены тем, что задача аутентификации открытых ключей решается намного проще и намного дешевле, чем задача распределения секретных ключей в одноключевых криптосистемах, которые требуют использования защищенных каналов. В последнем типе криптосистем аутентификация ключей совмещена с процедурой их распределения. В асимметричных криптосистемах нет проблемы распределения секретных ключей и эффективно решается проблема аутентификации открытых ключей.

После того как Диффи и Хеллман опубликовали свою парадоксальную идею построения криптографического протокола, позволяющего осуществить передачу секретного ключа по открытому каналу, интерес к разработке различного рода протоколов и использованию функции возведения в большую дискретную степень по модулю большого простого числа пробудился у широких кругов исследователей. В частности одним из первых интригующих протоколов явился протокол «бесключевого

шифрования», т. е. система, позволяющая передать секретное сообщение по открытому каналу вообще без использования передачи секретного ключа. С точки зрения классической одноключевой криптографии, сама постановка такой задачи носит оттенок абсурдности, но именно решение нестандартных задач и является центральным содержанием двухключевой криптографии. Поскольку двухключевые шифры решают такую задачу с использованием шифрования по открытому ключу, то система «бесключевого шифрования» представляет интерес как протокол, решающий поставленную задачу без использования передачи не только секретного, но и открытого ключа.

Рассмотрим вариант построения системы «бесключевого шифрования», называемый по имени своего изобретателя трехпроходным протоколом Шамира. На самом деле применяются даже два ключа шифрования, но ни один из них не передается по какому-либо каналу, т. е. они используются локально каждым из взаимодействующих абонентов. В этом протоколе используется коммутативный симметричный шифр, для которого выполняется условие:

$$\mathbf{E}_A(\mathbf{E}_B(m)) = \mathbf{E}_B(\mathbf{E}_A(m)),$$

где $\mathbf{E}_K$ есть функция шифрования по ключу K, а параметры A и B являются секретными ключами двух взаимодействующих абонентов A и B, соответственно. Протокол включает следующие шаги:

1. Абонент A шифрует сообщение m, получает шифртекст $c_1 = \mathbf{E}_A(m)$ и посылает c_1 абоненту B.
2. Абонент B зашифровывает сообщение c_1 (теперь сообщение m зашифровано дважды с использованием двух различных ключей), получает шифртекст $c_2 = \mathbf{E}_B(c_1) = \mathbf{E}_B(\mathbf{E}_A(m))$ и посылает c_2 абоненту A.
3. Абонент A, используя процедуру расшифрования **D**, преобразует сообщение c_2, получает шифртекст $c_3 = \mathbf{D}_A(c_2) = \mathbf{D}_A(\mathbf{E}_B(\mathbf{E}_A(m))) = \mathbf{D}_A(\mathbf{E}_A(\mathbf{E}_B(m))) = \mathbf{E}_B(m)$ и посылает c_3 абоненту B.

Получив значение c_3, абонент B без труда восстанавливает сообщение $m = \mathbf{D}_B(\mathbf{E}_B(m))$. Этот протокол вообще не требует обмена ни секретными, ни открытыми ключами. Наиболее сложной проблемой является построение шифрующих преобразований, обладающих свойством коммутативности и обеспечивающих высокую криптостойкость этого протокола. Свойство коммутативности обеспечивается процедурой шифрования, заключающейся в наложении с помощью операции XOR ($\oplus$) на сообщение m ключа, длина которого равна длине m. Пусть ключи A и B являются случайными равновероятными ключами, тогда в отдельности каждая из процедур шифрования $c_A = m \oplus A$ и $c_B = m \oplus B$ обеспечивает абсолютную стойкость криптографического преобразования. Однако такой способ шифрования неприемлем в рассматриваемом протоколе. Действительно, в этом случае на шагах 1, 2 и 3 по открытому каналу пересылаются сообщения $c_1 = m \oplus A$; $c_2 = m \oplus A \oplus B$; $c_3 = m \oplus B$, а следовательно, потенциальный нарушитель может легко вычислить $m = c_1 \oplus c_2 \oplus c_3$.

На самом деле в этой системе передатчиком и приемником используются ключи индивидуального использования, которые не требуют того, чтобы их знала какая-

либо другая сторона. Но поскольку ключевой обмен отсутствует, то мы говорим о «бесключевом шифровании». Суть состоит в том, что и передатчик, и приемник формируют некоторые секретные параметры, но при этом совсем не требуется их передача партнеру по сеансу. Передатчик зашифровывает сообщение и посылает его приемнику. Приемник еще раз зашифровывает сообщение и возвращает двукратно зашифрованное сообщение передатчику. Передатчик расшифровывает сообщение, превращая его в однократно зашифрованное по ключу приемника, и отправляет его еще раз приемнику. Теперь приемник, зная свой секрет, по которому он осуществлял шифрование, выполняет процедуру расшифрования и восстанавливает сообщение, которое и хотел ему переслать передатчик. Схема достаточно проста, но требуется найти такие процедуры шифрующих преобразований, выполняемых двумя сторонами, которые были бы прозрачны друг относительно друга. Никакого труда не представляет найти некоторое прямое преобразование исходного сообщения и обратное ему, но совсем не очевидно, что между прямым и обратным преобразованием можно выполнить некоторое другое преобразование и получить тот же результат, который бы получался, если бы другое преобразование выполнялось бы сразу над исходным сообщением. По крайней мере, такие коммутирующие преобразования надо специально подбирать.

Независимо друг от друга А. Шамир и Дж. Омура описали алгоритм шифрования, пригодный для использования в описанном выше протоколе и использующий операцию возведения в большую дискретную степень по модулю большого простого числа p в качестве шифрующей процедуры. Причем при зашифровании и расшифровании осуществляется возведении в различную степень. Пусть зашифрование сообщения $m < p$ состоит в возведение в степень, т. е. значение шифртекста равно $c = m^e \pmod p$, тогда для правильного расшифрования нужно найти такую степень d, что будет выполняться условие $m = c^d = m^{ed} \pmod p$. Из теории чисел известно, что последнее условие справедливо для любого m, если имеет место условие $ed = 1 \pmod{p-1}$. Также известно, что если выбрать e взаимно простым с $p-1$, то для такого e существует и с помощью расширенного алгоритма Евклида легко находится соответствующее ему обратное (по модулю $p-1$) число d, удовлетворяющее последнему условию.

Таким образом, приходим к работающему протоколу «бесключевого шифрования», включающему передачу следующих значений c_1, c_2 и c_3:

$c_1 = m^{e_A} \pmod p$, где e_A есть ключ зашифрования абонента А;

$c_2 = c_1^{e_B} = m^{e_A e_B} \pmod p$, где e_B есть ключ зашифрования абонента В;

$c_3 = c_2^{d_A} = m^{e_A e_B d_A} = m^{e_B} \pmod p$, где d_A есть ключ расшифрования абонента А.

Получив шифртекст c_3, абонент В легко расшифровывает сообщение: $m = c_3^{d_B}$. Действительно, имеем $c_3^{d_B} = m^{e_B d_B} = m \pmod p$. В данном случае по значениям c_1, c_2 и c_3 нарушитель не может восстановить передаваемое сообщение, поскольку для этого ему придется решить задачу дискретного логарифмирования, что является вычислительно неосуществимым при правильном выборе простого числа p. Напри-

мер, нарушитель по значениям c_2 и c_3 попытается вычислить d_A и восстановить сообщение $m = c_1^{d_A} \pmod p$.

Таким образом, мы имеем систему, в которой используются следующие параметры:

p – большое простое число, такое, что разложение числа $p - 1$ содержит, по крайней мере, один большой простой множитель;

α – первообразный корень по модулю p.

Также предполагается, что передатчик А формирует пару чисел e_A и d_A, взаимно простых с $p - 1$, таких, что $e_A d_A = 1 \pmod{p-1}$, а приемник В – пару чисел e_B и d_B, взаимно простых с $p - 1$, таких, что $e_B d_B = 1 \pmod{p-1}$. Протокол передачи секретного сообщения m от А к В включает следующие шаги:

1. Передатчик вычисляет значение $c_1 = m^{e_A} \pmod p$ и отправляет c_1 приемнику. (Этот шаг соответствует вложению сообщения в кейс и закрытию кейса на первый замок.)

2. Приемник, получив значение c_1, вычисляет $c_2 = c_1^{b_e} = m^{e_a s_e} \pmod p$ и отправляет c_2 передатчику. (Этот шаг соответствует дополнительному запиранию кейса на второй замок.)

3. Передатчик, получив значение c_2, вычисляет $c_3 = c_2^{a_d} = m^{e_a r_e d} = m^{b_e} \pmod p$ и отправляет c_3 приемнику. (Этот шаг соответствует открытию первого замка. Кейс с сообщением остался закрытым только на замок, установленный получателем.)

4. Приемник, получив значение c_3, вычисляет $c_4 = c_3^{d_B} = m^{e_B d_B} = m \pmod p$, т. е. восстанавливает сообщение, которое ему направлено передатчиком.

В результате выполнения протокола секретное сообщение оказывается переправленным от одного абонента к другому по открытому каналу, причем для этого не потребовался обмен ключами (ни секретными, ни открытыми). На самом деле можно все-таки говорить о пересылке ключа в неявном виде. Действительно, пересылаемое зашифрованное сообщение содержит в себе информацию как о сообщении, так и об использованных ключах, однако разделение этой информации представляет собой вычислительно сложную задачу. Только владелец пары соответствующих друг другу ключей зашифрования и расшифрования может накладывать и полностью снимать «шифрующий эффект». В качестве сообщения этим способом может быть передан по открытому каналу секретный ключ, который затем может быть использован для выполнения шифрования передаваемых сообщений с использованием некоторого симметричного шифра. Иными словами рассмотренный протокол может использоваться как система открытого распределения ключей (но для этого еще надо решить проблему аутентификации передаваемых сообщений).

Однако следует иметь в виду, что в подобной системе открытого распределения ключей решение проблемы аутентификации сообщений является существенно более сложной задачей по сравнению с системой Диффи–Хелмана. Это обусловлено тем,

что в протоколе «бесключевого шифрования» отсутствуют открытые ключи, аутентификацию которых можно обеспечить, не раскрывая секретного ключа (тот, кто сформировал пару соответствующих друг другу открытого и секретного ключей, должен оставаться единственным владельцем секретного ключа). Протокол «бесключевого шифрования» сталкивается с серьезной проблемой аутентификации. При установлении сеанса связи законные абоненты каким-то образом должны подтвердить свою подлинность. А это как раз весьма удобно делать с использованием аутентифицированных ключей (открытых или закрытых). Поскольку сообщение может быть подменено в процессе пересылки (например, по разветвленной компьютерной сети), то каждое сообщение должно быть аутентифицировано. Если этого не выполнять, то ни о какой серьезной стойкости протокола «бесключевого шифрования» говорить не приходится.

Таким образом, этот красивый протокол демонстрирует новые возможности и идеи, связанные с применением криптографических протоколов, но для практического применения требует дальнейшего усовершенствования. Например, он может быть превращен в систему открытого распределения ключей, в которой какими-то дополнительными методами решается проблема аутентификации открытых ключей. В рассмотренном выше протоколе каждый из пользователей формировал два различных ключа – один для зашифрования, а другой для расшифрования, но оба ключа являются секретными, поскольку по одному из них можно легко вычислить другой. Чтобы один из них можно было бы сделать открытым, требуется превратить процедуру вычисления одного из них по другому в вычислительно сложную задачу. Вспомним, что один из ключей выбирается случайно, а второй вычисляется как число взаимнообратное значению первого ключа по модулю $p - 1$. Последнее значение представляет собой функцию Эйлера от простого числа p. Если взять вместо простого числа p некоторое составное n, то рассмотренная выше система также работает. Корректность работы системы теперь будет основываться на теореме Эйлера, которая утверждает, что для любого m, взаимно простого с n, выполняется соотношение

$$m^{\varphi(n)} = 1 \pmod{n} \text{ или } m^{k\varphi(n)} = 1 \pmod{n} \text{ или } m^{k\varphi(n)+1} = m \pmod{n},$$

где k есть произвольное натуральное число. Из последнего соотношения видно, что для вычисления пары ключей требуется использовать соотношение

$$ed = 1 \bmod \varphi(n).$$

При этом появилась дополнительная забота – вычисление функции Эйлера от составного модуля n. Для чисел сравнительно малого размера данная задача не представляет проблемы, но для больших чисел эта задача может оказаться вычислительно невыполнимой. Действительно, для вычисления функции Эйлера предварительно выполняется разложение n на простые множители, но может оказаться так, что в разложении n будут присутствовать два или более простых сомножителя большой длины, например два множителя длиной не менее 500 бит. Тогда нам вряд ли удастся выполнить разложение полностью, а, следовательно, мы не сможем найти значение $\varphi(n)$. Но этого не сможет сделать и нарушитель, а, следовательно, даже зная один из парных ключей, он не сможет вычислить второй.

Нетрудно теперь догадаться, что законному пользователю следует выбрать два больших (длиной около 500 бит) простых числа p и q и сформировать n как произведение этих чисел: $n = pq$. Для выполнения зашифрования путем возведения в некоторую целую степень по модулю n не требуется знать разложения n, но для вычисления парного ключа без этого не обойтись. Таким образом, законный пользователь может вычислить значение функции Эйлера $\varphi(n) = (p - 1)(q - 1)$, а затем сформировать пару ключей e и d, такую, что $ed = 1 \bmod \varphi(n)$. Один из ключей можно предоставить для общего использования, например ключ e, который называется ключом зашифрования (encryption key) или открытым ключом. Второй ключ – ключ расшифрования (decryption key) или закрытый ключ – пользователь должен держать в секрете, поскольку именно он позволяет прочитать все то, что зашифровано с использованием открытого ключа. Выбор модуля n, представляющего собой произведение двух больших простых чисел, определяет также и тот факт, что вероятность того, что какое-то сообщение m окажется не взаимно простым с n, является пренебрежимо малой. Но можно строго показать, что даже и в этом пренебрежимом случае расшифрование по ключу d выполняется корректно.

Рассмотренное видоизменение системы «бесключевого шифрования» не спасло ее как систему, не требующую передачи никаких ключей вообще, поскольку для решения проблемы аутентификации потребуется использовать открытые ключи и осуществить их аутентификацию. Однако, используя вычисления по составному модулю n, становится возможным решить проблему аутентификации передаваемых сообщений и протокол «бесключевого шифрования» будет работать в том смысле, что он не требует передачи секретного ключа или формирования общего секрета для двух пользователей, как это имеет место в системе Диффи-Хеллмана.

Наиболее существенным результатом введения вычислений по составному модулю является возможность открытого использования одного из ключей. В модернизированной системе открытый ключ дает возможность расшифрования сообщений, которые будут зашифрованы закрытым ключом. Тот факт, что некоторая криптограмма $s = m^d \bmod n$ правильно расшифровывается по открытому ключу в исходный текст, т. е. имеет место соотношение $m = s^e \bmod n$, может служить подтверждением того, что данный исходный текст был зашифрован с использованием секретного (закрытого) ключа. Поскольку секретный ключ известен только законному пользователю, а именно тому, кто сгенерировал пару соответствующих друг другу взаимнообратных по модулю $\varphi(n)$ ключей, то факт правильного расшифрования криптограммы s свидетельствует, что текст был зашифрован пользователем, которому принадлежит открытый ключ e, приводящий к правильному расшифрованию m. Значение s фактически является подписью (или печатью) рассматриваемого пользователя к сообщению m. Появление возможности аутентифицировать источник сообщения (документа) представляет собой важнейший результат перехода от простого к составному модулю со специальной структурой. Модифицированная система получила название системы цифровой подписи RSA, которая нашла чрезвычайно широкое практическое применение. Авторами системы RSA являются Р. Ривест (R. Rivest), А. Шамир (A. Shamir) и Л. Адлеман (L. Adleman) [108]. Процедура шифрования сообщения по закрытому ключу называется подписыванием. Шифрование подписи по открытому ключу называется процедурой проверки подписи.

Скорость шифрования, обеспечиваемая двухключевыми (асимметричными) шифрами, на несколько порядков ниже скорости, которой обладают одноключевые (симметричные) криптосистемы. Поэтому наиболее эффективны гибридные криптосистемы, в которых информация шифруется с помощью одноключевых шифров, а распределение сеансовых ключей осуществляется по открытому каналу с помощью двухключевых шифров. Например, используя криптосистему RSA, можно легко обменяться сеансовым ключом с любым абонентом, зашифровав сеансовый ключ с помощью его открытого ключа. Зашифрованный сеансовый ключ можно безопасно передать по открытому каналу связи, поскольку необходимым для расшифрования секретным ключом обладает только абонент, открытый ключ которого был использован для зашифрования. Для непосредственного засекречивания информации двухключевые шифры находят ограниченное применение. Незаменимое их значение заключается в том, что они являются основой технологий электронного документооборота.

3.4. ЭЦП на эллиптических кривых

3.4.1 Основные свойства эллиптических кривых

В целях применения эллиптических кривых (ЭК) в криптографических приложениях необходимо рассмотреть ряд их свойств и найти такие семейства кривых, которые пригодны для построения эффективных криптографических конструкций.

Пусть GF(q) представляет собой конечное поле с характеристикой p мощности $q=p^n$. Уравнение эллиптической кривой $E(\mathrm{GF}(q))$ в аффинных координатах, вместе с некоторой специальной точкой **O**, которую назовем «точкой в бесконечности» или нулевой точкой, имеет вид [29, 46]:

$$E(\mathrm{GF}(q)):\ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\,. \qquad (3.1)$$

Если характеристика поля $p>3$, тогда $E(\mathrm{GF}(p)):\ y^2 \equiv x^3 + ax + b (\mathrm{mod}\ p)$.

Наиболее важными параметрами кривой $E(\mathrm{GF}(p))$ являются [29, 60]: дискриминант $\Delta = -16(4a^3 + 27b^2)$ и инвариант $j = \dfrac{1728(4a)^3}{\Delta}$. Коэффициенты a, b ЭК $E(\mathrm{GF}(q))$ по известному инварианту j определяются следующим образом:

$$\begin{cases} a \equiv 3k (\mathrm{mod}\ p) \\ b \equiv 2k (\mathrm{mod}\ p) \end{cases},$$

где $k \equiv \dfrac{j}{1728 - j}\ (\mathrm{mod}\ p)$, $j\neq 0$, $j\neq 1728$.

Уравнения ЭК над конечными полями GF(q) с различными значениями характеристик p представлены в табл. 3.6.

Таблица 3.6

p	$E(\mathrm{GF}(q))$	j
2	$y^2 + ay = x^3 + bx + c$	0
2	$y^2 + xy = x^3 + ax^2 + b$	$1/b$
3	$y^2 = x^3 + ax + b$	0
3	$y^2 = x^3 + ax^2 + b$	$\neq 0$
$\neq 2, \neq 3$	$y^2 = x^3 + ax + b$	$\dfrac{1728\,(4a)^3}{\Delta}$

Рассмотрим дополнительные свойства ЭК:

- если дискриминант $\Delta \neq 0$, кривая называется неособой;
- если две кривые имеют одинаковый инвариант j, тогда они являются изоморфными кривыми;
- для каждой ЭК $E(\mathrm{GF}(p))$ существует так называемая скрученная ЭК $\tilde{E}(\mathrm{GF}(p))$, определяемая выражением [29]:

$$\tilde{E}(\mathrm{GF}(p)):\ y^2 \equiv x^3 + ac^2 x + bc^3 \pmod{p}, \qquad (3.2)$$

где c – квадратичный невычет по модулю p. Кривые $E(\mathrm{GF}(p))$ и $\tilde{E}(\mathrm{GF}(p))$ изоморфны, поэтому скрученные кривые могут использоваться для построения ЭК заданного порядка;

- две ЭК: $(y^2=x^3+ax+b)_{(1)}$ и $(y^2=x^3+ax+b)_{(2)}$ изоморфны, если $y_{(1)} = u^3 y_{(2)}$, $x_{(1)}=u^2 x_{(2)}$, $a_{(1)}=u^4 a_{(2)}$, $b_{(1)}=u^6 b_{(2)}$, где $u \in \mathrm{GF}(q)$, $u \neq 0$.

Если многочлен от x в правой части уравнения ЭК над полем характеристики $p \neq 2$ и $p \neq 3$ не имеет кратных корней, тогда кривая является неособой [49]. Для конечных полей Галуа с характеристиками $p=2$ и $p=3$ требования неособости выполняются автоматически.

Множество точек ЭК вместе с нулевой точкой **O** порождает конечную абелеву группу A относительно операции сложения, вводимой ниже.

3.4.2 Групповой закон сложения точек на ЭК

При характеристике поля $p>3$ уравнение ЭК, согласно табл. 3.6, имеет вид

$$E(\mathrm{GF}(p)): y^2 \equiv x^3 + ax + b \ (\mathrm{mod}\ p). \tag{3.3}$$

Операция обращения точки кривой эквивалентна записи: $-(x, y)=(x, -y)$. Операция сложения точек кривой $P_1=(x_1, y_1)$ и $P_2=(x_2, y_2)$ в виде $P_3= P_1 + P_2$ определяется следующими сравнениями [29, 46]:

$$\begin{cases} x_3 \equiv \lambda^2 - x_2 - x_1 \ (\mathrm{mod}\ p) \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \ (\mathrm{mod}\ p) \end{cases}, \tag{3.4}$$

где $\lambda \equiv \dfrac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\ p)$.

При $P_1=P_2=(x_1, y_1)$, $y_1 \neq 0$, результат удвоения точки P_1 получим из выражения $2P_1=(x_3, y_3)$, где

$$\begin{cases} x_3 \equiv \lambda^2 - 2x_1 \ (\mathrm{mod}\ p) \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \ (\mathrm{mod}\ p) \end{cases}, \tag{3.5}$$

где $\lambda \equiv \dfrac{3x_1^2 + a}{2y_1} (\mathrm{mod}\ p)$.

Непосредственно из выражений (3.4, 3.5) видно, что точка в бесконечности получается при удвоении точки P_1 с нулевой координатой y либо при сложении двух различных точек с одинаковой координатой x.

Точка в бесконечности является точкой перегиба для правой части уравнения ЭК. В общем случае единичным элементом группы может быть любая точка перегиба, а групповой закон сложения точек будет задан другими формулами. Однако все операции сложения в группе изоморфны. На рис. 3.1 – 3.3 представлена геометрическая интерпретация группового закона сложения точек на ЭК. Рис. 3.1 соответствует случаю сложения двух различных точек $P_1 + P_2 = (x_3, -y_3)$. На рис. 3.2 происходит удвоение точки $2P_1=(x_2, -y_2)$ и, наконец, рис. 3.3 соответствует случаю сложения двух различных точек, но с одинаковой координатой x: $P_1 + P_2 = \mathbf{O}$.

Ряд криптографических алгоритмов с открытым ключом строится на основе таких функций, в которых вычисляется кратное некоторого элемента абелевой группы **A** [43, 47, 83, 91, 113]. Как известно, абелевой группой называется группа элементов множества **A**, удовлетворяющая коммутативному закону $ba=ab$ для всех $a, b \in \mathbf{A}$. Из определения следует, что всякая циклическая группа является абелевой, поэтому в дальнейшем будем использовать термин циклическая группа.

Групповой закон сложения точек ЭК, как элементов циклической группы, обладает следующим криптографическим свойством. Пусть P и G – элементы (точки)

циклической подгруппы **A** кривой $E(\mathrm{GF}(q))$, причем G является примитивным элементом (генератором) этой подгруппы. Тогда, если $P = n * G$, где $n \in \mathrm{GF}(q)$ – секретный ключ (случайное число), "$*$" – означает многократное сложение точки G, то нахождение числа n по двум заданным элементам $P \in \mathbf{A}$ и $G \in \mathbf{A}$ при $n \to \infty$ является вычислительно сложной задачей [83, 91]. Задача нахождения индекса n по двум заданным элементам группы становится вычислительно невыполнимой при размерностях n в 120 бит и более. Далее для простоты будем использовать обозначение $nG = n * G$.

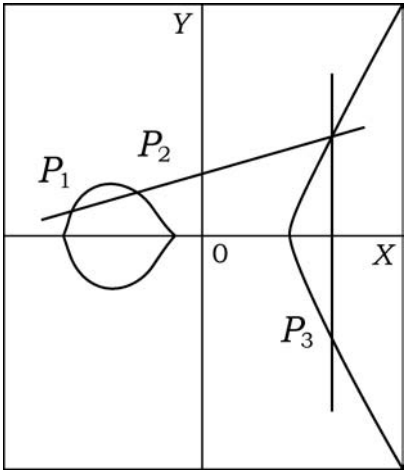

Рис. 3.1.

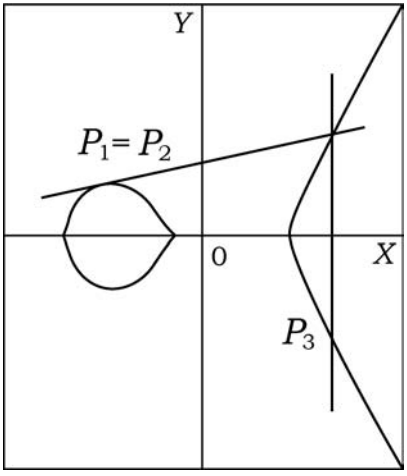

Рис. 3.2.

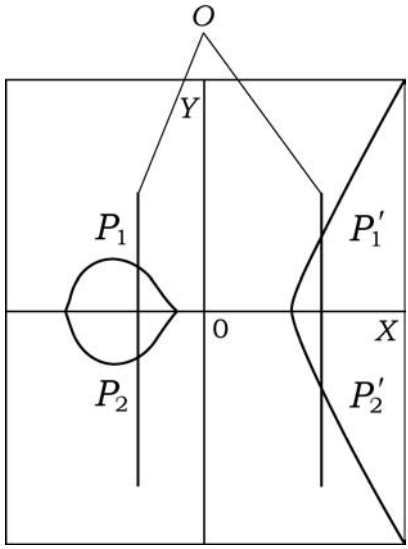

Рис. 3.3.

При построения эллиптических криптографических конструкций используется циклическая подгруппа группы точек эллиптической кривой E, определенная над конечным полем GF(q).

3.4.3 Групповой закон сложения точек ЭК над конечными полями с различной характеристикой *p*

Найдем выражения, описывающие группой закон сложения точек ЭК над полями с характеристикой p=2 и p=3. В этом случае уравнения кривых принимают вид, представленный в табл. 3.6 (выражения 1–4). Будем рассматривать кривые с инвариантом $j\neq 0$. Для таких кривых над расширенным конечным полем GF(2^n) получены формулы, определяющие координаты x_3, y_3 [29]:

$$\begin{cases} x_3 = \lambda^2 + \lambda - a - x_1 - x_2 \\ y_3 = -(\lambda + 1)x_3 - \nu \end{cases}, \tag{3.6}$$

где для случая $P_1(x_1, y_1) \neq P_2(x_2, y_2)$ $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$, $\nu = \dfrac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$. Если P_1=P_2=(x_1, y_1), имеем $\lambda = \dfrac{\partial y}{\partial x}$, дифференцируя данное выражение, получим $\lambda = \dfrac{3x_1^2 + 2x_1 a}{2y_1 + x_1}$,

$\nu = \dfrac{3x_1^3 + 2x_1^2 - 2y_1^2 - y_1x_1}{2y_1 + x_1}$. Следует отметить, что в этом случае обращение точки имеет вид $P_3=(x_3, -y_3-x_3)$. Аналогично для кривой с инвариантом $j\neq 0$, над конечным полем GF(3^n):

$$\begin{cases} x_3 = \lambda^2 - a - x_1 - x_2 \\ y_3 = -x_3\lambda - \nu \end{cases}, \qquad (3.7)$$

где для $P_1(x_1, y_1) \neq P_2(x_2, y_2)$: $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$, $\nu = \dfrac{y_1x_2 - y_2x_1}{x_2 - x_1}$, а в случае $P_1 = P_2 = (x_1, y_1)$: $\lambda = \dfrac{3x_1^2 + 2x_1a}{2y_1}$ и $\nu = \dfrac{3x_1^3 + 2x_1^2 - 2y_1^2}{2y_1}$.

Операции умножения в конечных полях GF(p^n) с характеристикой p=2 или p=3, проводятся по модулю порождающего многочлена поля, а сложения – по модулю 2 или 3, соответственно.

Приведенные групповые законы сложения для кривых над полями с различными характеристиками можно использовать в качестве функций криптографического преобразования для построения различных протоколов защищенного обмена информацией.

3.4.4 Способы повышения быстродействия вычислений в циклической группе точек ЭК

Преобразование уравнения кривой в проективные координаты

В кольце целых чисел $\boldsymbol{Z}_p$ и в полях Галуа GF(q) наиболее трудоемкой операцией является инверсия. Чтобы исключить данную операцию, необходимо перейти из аффинных координат в проективные и сделать координату Z кратной знаменателю λ в выражениях (3.4-3.7). Рассмотрим уравнение кривой E(GF(q)): $Y^2Z = X^3 + aXZ^2 + bZ^3$ над полем с характеристикой $p\neq 2$ и $p\neq 3$, получаемое из выражения (3.8) переходом в проективные координаты. В этом случае E(GF(q)) можно рассматривать как эквивалентный класс точек (X, Y, Z) на проективной плоскости $\boldsymbol{P}^2$ над $GF(q)$. Точка **O** в $\boldsymbol{P}^2$(GF(q)) представляет собой все ненулевые точки с отношением эквивалентности (X, Y, Z) ~(αX, αY, αZ). Обозначим эти точки как (X, Y, Z). Отметим, что существует только одна точка с координатой Z=0 — это точка (0, 1, 0). Если Z=0, то и X=0 и существует только один класс эквивалентности с X=0 и Z=0, а именно (0, 1, 0*)*. Как было показано ранее, это не что иное, как точка в бесконечности **O** $\in E$(GF(q)). Если $P = (X, Y, Z) \notin \mathbf{O}$, тогда выполнив замену (X, Y, Z) = $\left(\dfrac{X}{Z}, \dfrac{Y}{Z}, 1\right)$, получим однозначное

соответствие точки (X, Y, Z) точке $\left(x = \frac{X}{Z}, y = \frac{Y}{Z}\right)$. Формулы удвоения точки для кривой над полем с $p \neq 2$ и $p \neq 3$ можно получить из следующего представления

$$\begin{cases} X_3 = Z_3 x_3 \\ Y_3 = Z_3 y_3 \end{cases}, \tag{3.8}$$

где координата $Z_3 = 8Y_1^3 Z_1^3$ кратна λ. После соответствующих преобразований получим формулы группового закона в проективных координатах. При удвоении точки они имеют вид

$$\begin{cases} X_3 = 2Y_1 Z_1 \left[\left(3X_1^2 + aZ_1^2\right)^2 - 8X_1 Y_1^2 Z_1 \right] \pmod p \\ Y_3 = 4Y_1^2 Z_1 \left[3X_1 \left(3X_1^2 + aZ_1^2\right) - 2Y_1^2 Z_1 \right] - \left(3X_1^2 + aZ_1^2\right)^3 \pmod p \\ Z_3 = 8Y_1^3 Z_1^3 \pmod p \end{cases} \tag{3.9}$$

При сложении двух различных точек примем $Z_3 = Z_1 Z_2 (X_2 X_1 - X_1 Z_2)^3 \pmod p$, откуда для других координат точки $P_3(X_3, Y_3, Z_3)$ получим:

$$\begin{cases} X_3 = \left(X_2 Z_1 - X_1 Z_2\right) \left[Z_1 Z_2 \left(Y_2 Z_1 - Y_1 Z_2\right)^2 - \left(X_2 Z_1 + X_1 Z_2\right) \cdot \right. \\ \left. \left(X_2 Z_1 - X_1 Z_2\right)^2 \right] \pmod p \\ Y_3 = \left(X_2 Z_1 - X_1 Z_2\right)^2 \left[Y_2 Z_1 \left(Y_2 Z_1 + 2Y_1 Z_2\right) - Y_1 Z_2 \left(X_1 Z_2 + 2X_2 Z_1\right) \right] \\ -Z_1 Z_2 \left(Y_2 Z_1 - Y_1 Z_2\right)^3 \pmod p \\ Z_3 = Z_1 Z_2 \left(X_2 Z_1 - X_1 Z_2\right)^3 \pmod p \end{cases} \tag{3.10}$$

Для кривых из табл. 3.6 с инвариантом $j \neq 0$, над расширенным полем $GF(2^n)$, после осуществления аналогичного перехода к проективным координатам, получены следующие выражения, описывающие групповой закон сложения точек:

при $P_1 = P_2$

$$\begin{cases} X_3 = X_1^4 + Z_1 X_1^3 \pmod{g(x)} \\ Y_3 = \left(Z_1 X_1^6 + Z_1^3 X_1^4 + X_1^3 + X_1 Y_1 Z_1\right) X_1 \pmod{g(x)}, \\ Z_3 = Z_1^2 X_1^2 \pmod{g(x)} \end{cases} \tag{3.11}$$

при $P_1 \neq P_2$

$$\begin{cases} X_3 = (X_2Z_1 + X_1Z_2)\Big\langle Z_1Z_2\Big[Z_1^2\left(aX_2^2 + Y_2^2\right) + Z_2^2\left(aX_1^2 + Y_1^2\right) + \left(Y_2Z_1 + Y_1Z_2\right)\cdot \\ \left(X_2Z_1 + X_1Z_2\right)\Big] + X_2^2Z_1^2 + X_1^2Z_2^2\Big\rangle \left(\mathrm{mod}\, g(x)\right) \\ Y_3 = Z_1Z_2\Big[\left(Y_2Z_1 + Y_1Z_2\right)^3 + a\left(X_2Z_1 + X_1Z_2\right)^3 + \left(X_2^2Z_1^2 + X_1^2Z_2^2\right)\Big\{\left(Y_2Z_1 + Y_1Z_2\right)\cdot \\ (1+a) + Y_1X_2\left(1 + X_2Z_1\right) + Y_2X_1\left(1 + X_1Z_2\right)\Big\}\Big] + X_2^2Y_2Z_1^3 + X_1^2Y_1Z_2^3 \left(\mathrm{mod}\ g(x)\right) \\ Z_3 = Z_1Z_2\left(X_2Z_1 + X_1Z_2\right)^3 \left(\mathrm{mod}\ g(x)\right) \end{cases} \quad (3.12)$$

Для кривых $E(\mathrm{GF}(q))$ над расширенным полем $\mathrm{GF}(3^n)$ с характеристикой p=3 и инвариантом $j \neq 0$, при $P_1 = P_2$ получим:

$$\begin{cases} X_3 = Y_1Z_1^3\left[Y_1^2\left(aZ_1 - X_1\right) - a^2X_1^2Z_1\right]\left(\mathrm{mod}\ g(x)\right) \\ Y_3 = aX_1Z_1^2\left[\left(aX_1Z_1^2\right)^2 - Y_1^2Z_1\left(aZ_1 - X_1\right)\right] - Y_1^2Z_1^2\left(Y_1^2 - aX_1^2Z_1\right)\left(\mathrm{mod}\ g(x)\right) \\ Z_3 = -Y_1^3Z_1^4 \left(\mathrm{mod}\ g(x)\right) \end{cases} \quad (3.13)$$

а при $P_1 \neq P_2$:

$$\begin{cases} X_3 = Z_1Z_2\left(Y_2Z_1 - Y_1Z_2\right)^2 - \left(X_2Z_1 - X_1Z_2\right)^2\left(X_1Z_2 + X_2Z_1 + a\right)\left(\mathrm{mod}\ g(x)\right) \\ Y_3 = -\Big\{\left(Y_2Z_1 - Y_1Z_2\right)^3 Z_1Z_2 + \left(X_2Z_1 - X_1Z_2\right)^2\Big[2Z_1Z_2\left(Y_1X_2 - X_1Y_2\right) \\ + Z_1^2\left(Y_1X_1 - Y_2X_2\right) + aY_1Z_1\Big]\Big\}\left(\mathrm{mod}\, g(x)\right) \\ Z_3 = Z_1Z_2\left(X_2Z_1 - X_1Z_2\right)^2 \left(\mathrm{mod}\ g(x)\right) \end{cases} \quad (3.14)$$

Известно, что сложность выполнения умножения в конечных полях оценивается величиной $log^2(q)$, а деления - $log^3(q)$, где q – количество элементов (мощность) поля $\mathrm{GF}(q)$. Следовательно, несмотря на большее количество операций умножения для группового закона в проективных координатах, результирующее количество необходимых операций для выполнения преобразования будет меньше примерно в $log(q)$ раз по сравнению с преобразованиями в аффинных координатах. После выполнения сложения в проективных координатах переход к аффинным координатам осуществляется путем деления X_3 и Y_3 на координату Z_3.

Использование комплексного умножения на ЭК

Для ЭК $E(\mathrm{GF}(q))$, $q=p^n$, определенной уравнением (3.2), существует отображение, называемое эндоморфизмом кривой $\phi : (x, y) \to (f(x), h(x, y))$, где $(x, y) \in E(\mathrm{GF}(q))$,

$\phi(x, y) \in E(\mathrm{GF}(q))$, $\phi(\mathbf{O}) = \mathbf{O}$, $f(x)$ и $h(x, y)$ – рациональные функции [49]. Если группа точек ЭК содержит циклическую подгруппу **A**, такую, что порядок подгруппы ξ – простое число и не выполняется условие делимости $\#E(\mathrm{GF}(q)) \mid \xi^2$, тогда отображение ϕ переводит подгруппу **A** в себя или в бесконечно удаленную точку. В первом случае эндоморфизм ЭК задает автоморфизм подгруппы **A**, для которого существует натуральное наименьшее число l – такое, что $\phi^l(P) = P$, $P \in \mathbf{A}$, $l \mid (\xi - 1)$. Так как ϕ является автоморфизмом, тогда

$$\phi(P) = eP, \tag{3.15}$$

где $e < \xi$ - легко вычислимое число. Данное отображение задает комплексное умножение на ЭК [67]. Например, ЭК (3.2) с коэффициентами $a_1, a_2, \ldots, a_6 \in \mathrm{GF}(p)$ имеет комплексное умножение (3.15) вида $\phi(x, y)=(x^p, y^p)$ с соответствующим квадратным характеристическим уравнением. В табл. 3.7 представлены типы комплексного умножения для различных уравнений ЭК. Отображение вида $\phi(x, y)=(x^2, y^2)$ называется эндоморфизмом Фробениуса [46, 67, 88].

Таблица 3.7

Уравнение ЭК	**Вид комплексного умножения**
(3.3), a=0, $p \equiv 1 \pmod 6$	$\phi(x, y) = (\omega x, -y)$, $\omega^3 \equiv 1 \pmod p$, $\phi^2+\phi+1=0$, $e = \dfrac{1+\sqrt{-3}}{2}$
(3.3), b=0, $p \equiv 1 \pmod 4$	$\phi(x, y) = (-x, iy)$, $i^2 \equiv -1 \pmod p$, $\phi^2+1=0$, $e = \dfrac{\sqrt{-1}}{2}$
$y^2+xy=x^3+1$, p=2, $n \geq 2$	$\phi(x, y) = (x^2, y^2)$, $\phi^2+\phi+2=0$, $e = \dfrac{-1+\sqrt{-7}}{2}$

Применение свойств комплексного умножения для вычисления отображений в циклической подгруппе **A** ЭК $E(\mathrm{GF}(q))$ позволяет ускорить процедуру многократного сложения точки $P \in \mathbf{A}$ **(**умножения точки на число) примерно в два раза [43, 67]. В этом случае любое число m, связывающее две точки, можно представить в виде

$$m = m_0+m_1e, \tag{3.16}$$

где $m_0 < \sqrt{q}$, $m_1 < \sqrt{q}$. После этого находятся точки m_0P, m_1P и на основе комплексного умножения (3.15) точка $e(m_1P)$. Окончательно получим $mP = (m_0P) + (e(m_1P))$.

Первоначально комплексное умножение было определено для ЭК над полем комплексных чисел $\boldsymbol{C}$, которые определяются своим инвариантом j. Такого рода кривых очень мало. Однако при переходе к конечным полям, используя метод построения циклов изогений, предложенный в [68], можно расширить класс кривых, обладающих комплексным умножением.

Использование комплексного умножения для отображения точек ЭК, позволяет понизить сложность отображения почти в два раза по сравнению с методом удвоений, основанным на двоичном представлении числа, связывающего точки кривой.

3.4.5 Исследование стойкости алгоритмов защиты информации, использующих эллиптические криптографические конструкции

В п.п. 3.4.1 было показано, что стойкость определяется сложностью нахождения индекса, связывающего два элемента абелевой группы точек на кривой. В качестве показателя эффективности защиты информации (стойкости) при ее преобразовании на основе применения группового закона сложения в аддитивной абелевой группе на ЭК целесообразно применять асимптотическую оценку сложности алгоритма анализа в терминах O - символики.

При использовании криптоалгоритмов, основанных на дискретном логарифмировании в конечных абелевых группах, сложность раскрытия существенно зависит от порядка группы [77, 83, 86, 91, 112]. При этом алгоритм является стойким, если в разложении порядка группы на простые множители встречается большой простой делитель. В противном случае путем перехода к подгруппам и факторгруппам сложность раскрытия удается значительно понизить. Поэтому при оценивании сложности анализа криптоалгоритмов важным является вычисление порядка группы с последующим разложением его на простые множители.

Для нахождения дискретного логарифма на ЭК можно применять методы «встречи посередине», Сильвера-Полига-Хеллмана (СПХ), Полларда [43, 104]. Наиболее эффективными из них являются алгоритмы СПХ и Полларда. Алгоритм СПХ имеет сложность по времени $S_t = O\left(\sqrt{r}\right)$, а по памяти $S_c = O\left(\sqrt{r}\right)$, алгоритм Полларда имеет аналогичную временную сложность и меньшую сложность по памяти: $S_c = O(\log p)$. В обоих случаях r означает наибольший простой множитель порядка мультипликативной группы поля: $p - 1$.

Применяя данные алгоритмы для аддитивной циклической группы точек ЭК, придем к выводу, что стойкость подобных криптографических конструкций определяется как сложность используемого алгоритма по определению индекса и равна $O\left(\sqrt{\zeta_p}\right)$, где ζ_p - наибольший простой множитель порядка группы точек ЭК. Кроме того, для исключения отображения циклической подгруппы группы точек ЭК в мультипликативную группу некоторого расширения GF(p^n), которое есть GF(p^{n^i}), необходимо, чтобы для выбранной кривой не выполнялось следующее условие делимости [45, 90]:

$$\#E(GF(p^n)) \Big| \left(p^{n^i} - 1\right), \tag{3.17}$$

где i=1, 2, ..., k. Следует использовать кривые, для которых это свойство делимости не выполняется при $k \geq 30$. В ином случае задача нахождения индекса на ЭК сводится к задаче дискретного логарифмирования в конечном поле. Отображение аддитивной абелевой группы ЭК в мультипликативную группу конечного поля основано на использовании отображений Вейля [46]. Это приводит к субэкспоненциальному алгоритму дискретного логарифмирования в группе точек на кривой и соответственно к снижению стойкости криптосистемы, поскольку дискретные логарифмы в конечных полях GF(p^n) могут быть определены с помощью алгоритма Копперсмита со сложностью $O\left(\exp\left(cn^{1/3}\right)\cdot\left(\ln n\right)^{2/3}\right)$, где c – небольшая константа [90].

Такого рода отображение неприменимо для невырожденных кривых, инвариант j которых не равен нулю. Исходя из этого факта, в п.п.3.4.2 рассматривались только кривые с ненулевым инвариантом. Кроме того, атаки подобного рода на криптосистему можно усложнить посредством построения кривых с нулевым инвариантом, группу которых можно отобразить в мультипликативную группу расширенного поля: $E(\mathrm{GF}(p^n)) \mapsto \mathrm{GF}(p^{ln})$, где l - степень расширения исходного поля. Сложность нахождения индекса в зависимости от исходного поля и степени расширения поля, куда отображается группа кривых, представлена в табл. 3.8. Из таблицы видно, что степень расширения l = 2 обеспечивает сравнимую сложность при малых размерах ключа, поэтому необходимо отображать аддитивную группу ЭК с нулевым инвариантом в мультипликативную группу поля со степенью расширения $l \geq 4$.

Таблица 3.8

Исходное поле GF(2^n)	**Сложность нахождения индекса**	**Поле GF(2^{2n})**	**Сложность нахождения индекса**	**Поле GF(2^{4n})**	**Сложность нахождения индекса**
GF(2^{100})	10^{15}	GF($_2{}^{2\cdot100}$)	10^{17}	GF($2^{4\cdot100}$)	10^{25}
GF(2^{200})	10^{30}	GF($2^{2\cdot200}$)	10^{25}	GF($2^{4\cdot200}$)	10^{38}
GF(2^{300})	10^{45}	GF($2^{2\cdot300}$)	10^{32}	GF($2^{4\cdot300}$)	10^{45}

В качестве примера рассмотрим две кривые над кольцом целых чисел: $y^2 = x^3 + b \pmod p$, где $p \equiv 5 \pmod 6$, и $y^2 = x^3 + ax \pmod p$, где $p \equiv 3 \pmod 4$. Данные кривые, согласно п.п. 3.4.2, имеют порядок группы #E(GF(p)) = p + 1, делящий в соответствии с (3.27) число $p^2 - 1$. Поэтому задача дискретного логарифмирования в группе на этих ЭК сводится к задаче дискретного логарифмирования в конечном поле GF(p^2). Аналогично в [56] показано, что группу кривой $y^2 + y = x^3 + ax + b$ над полем GF(2^n) можно отобразить в мультипликативную группу поля GF(2^{4n}).

Если сравнить сложность задач факторизации целых чисел, дискретного логарифмирования в мультипликативных группах и дискретного логарифмирования в аддитивной абелевой группе точек ЭК, то последние выглядят предпочтительнее. Это показано в табл. 3.9, где представлено сравнение приблизительных оценок сложности криптоанализа методов, основанных на разложении целых чисел (РЦЧ), дискретном логарифмировании в конечном поле (ДЛКП) и дискретном логарифмировании в группе точек (ДЛГТ) ЭК для различных характеристик полей и в зависимости от длины ключа.

Таблица 3.9

Длина ключа (бит)	**Сложность анализа**			
	РЦЧ $\forall p$	**ДЛКП для p=2**	**ДЛКП для $p\neq 2$**	**ДЛГТ $\forall p$**
100	$1.3\cdot10^{7}$	$1.3\cdot10^{7}$	$1.6\cdot10^{11}$	$1.1\cdot10^{15}$
200	$7.2\cdot10^{9}$	$7.2\cdot10^{9}$	$9.6\cdot10^{16}$	$1.3\cdot10^{30}$
300	$7.1\cdot10^{11}$	$7.1\cdot10^{11}$	$3.8\cdot10^{21}$	$1.4\cdot10^{45}$
400	$3\cdot10^{13}$	$3\cdot10^{13}$	$3.4\cdot10^{25}$	$1.6\cdot10^{60}$
500	$7.5\cdot10^{14}$	$7.5\cdot10^{14}$	$1.2\cdot10^{29}$	$1.8\cdot10^{75}$
600	$1.3\cdot10^{16}$	$1.3\cdot10^{16}$	$2.1\cdot10^{32}$	$2\cdot10^{90}$
700	$1.7\cdot10^{17}$	$1.7\cdot10^{17}$	$2.1\cdot10^{35}$	$2.3\cdot10^{105}$
800	$1.8\cdot10^{18}$	$1.8\cdot10^{18}$	$1.4\cdot10^{38}$	$2.6\cdot10^{120}$
900	$1.7\cdot10^{19}$	$1.7\cdot10^{19}$	$6.5\cdot10^{40}$	$2.9\cdot10^{135}$
1000	$1.3\cdot10^{20}$	$1.3\cdot10^{20}$	$2.3\cdot10^{43}$	$3.3\cdot10^{150}$

Принимая во внимание, что сложность выполнения преобразования в абелевой группе ЭК оценивается величиной $O(log^2 q)$, а в мультипликативной группе поля – $O(log^3 q)$, преимущества использования первых для построения криптосистем становятся очевидными. Также необходимо отметить, что криптографические конструкции, сложность анализа которых превышает значение 10^{50}, нецелесообразно применять на практике [47], так как данные значения превосходят возможности современных технологий по обработке информации. Поэтому следует ограничиваться длиной ключа до 400 бит.

Таким образом, стойкость методов криптографического преобразования, основанных на использовании группового закона сложения элементов аддитивной абелевой группы на ЭК, существенно превосходит стойкость аналогичных методов, основанных на использовании мультипликативных полей. Выигрыш в стойкости особенно заметен при больших размерах ключа. Данное обстоятельство позволяет исполь-

зовать криптографические конструкции подобного типа для построения криптографических протоколов различного назначения.

3.4.6 Алгоритмы выбора ЭК

Синтез криптографических конструкций на ЭК, удовлетворяющих показателям стойкости, требует, в первую очередь, выбора следующих параметров:

- вида конечного поля;
- характеристики поля и (или) его расширения;
- уравнения ЭК;
- порядка циклической подгруппы точек ЭК;
- генератора подгруппы точек ЭК.

От выбора данных параметров существенно зависит стойкость криптографических конструкций и безопасность протоколов на ЭК. Одним из главных условий является то, что подгруппа группы точек выбранной кривой должна быть циклической с точкой, играющей роль примитивного элемента (генератора) подгруппы. Если порядок группы – простое число, тогда любой элемент группы может служить ее генератором.

Подходы к определению порядка группы

Порядок группы точек ЭК является зависимым параметром и определяется видом конечного поля, характеристикой конечного поля, степенью его расширения, а также коэффициентами уравнения кривой. Известно [1], что число решений сравнения $x^n=a$ находится из выражения $N\left(x^n=a\right)=\sum_{\chi^n=\varepsilon}\chi(a)$, где сумма берется по всем мультипликативным характерам χ, порядок которых делит n. Следовательно, порядок группы точек можно найти через суммы мультипликативных характеров [46]:

$$\#E(\mathrm{GF}(p))=p+1+\sum_{x\in \mathrm{GF}(p)}\chi\left(x^3+ax+b\right). \qquad (3.18)$$

Либо, иначе [21] – $\#E(\mathrm{GF}(p))=1+N(y^2-x^3-ax-b=0)$, где N – число решений уравнения ЭК. Далее правую часть данного равенства можно разбить на два сомножителя

$$\#E(\mathrm{GF}(p))=1+\sum_{\varsigma-\eta=0}N\left(y^2=\varsigma\right)N\left(x^3+ax+b=\eta\right), \qquad (3.19)$$

где

$$N\left(y^2=\varsigma\right)=\sum_{i=0}^{1}\chi^i(\varsigma),$$

$$N\left(x^3+ax+b=\eta\right)=\sum_{\nu+\frac{\xi-b}{a}=\eta}N\left(x^3=\nu\right)N\left(x=\frac{\xi-b}{a}\right).$$

С учетом того, что $N\left(x=\frac{\xi-b}{a}\right)=1$ для всех простых p [1, 33], получим

$$N\left(x^3+ax+b=\eta\right)=\sum_{\nu+\frac{\xi-b}{a}=\eta}N\left(x^3=\nu\right)=\sum_{\nu+\frac{\xi-b}{a}=\eta}\sum_{j=0}^{2}\chi^j(\nu).$$

Подставляя выражения для множителей в равенство (3.19) и изменяя порядок суммирования, можно получить:

$$\#E(\mathrm{GF}(q))=1+\sum_{i=0}^{1}\sum_{j=0}^{2}\sum_{\nu+\frac{\xi+b}{a}=\eta}\sum_{\varsigma-\eta=0}\chi^i(\varsigma)\chi^j(\nu), \tag{3.20}$$

где χ^i и χ^j – мультипликативные характеры порядка 2 и 3, соответственно.

Кроме того, согласно теореме Хассе [29] порядок группы оценивается сверху – снизу следующей формулой

$$q+1-2\sqrt{q}\le\#E(\mathrm{GF}(q))\le q+1+2\sqrt{q}\,. \tag{3.21}$$

В общем случае, сложность вычисления порядка группы по выражению (3.18) для ЭК оценивается величиной $O(log^9 q)$. Следовательно, при $log_2 q$ = 40÷50 требуется выполнение 10^{14}-10^{15} операций, необходимых для поиска порядка группы, что требует существенных вычислительных и временных ресурсов и приводит к невозможности применения данного подхода.

Порядок группы для кривых над кольцом целых чисел Z_p с неполными уравнениями правой части

Для кривых над кольцом Z[p] с характеристикой $p\neq2$ и $p\neq3$, имеющих неполные уравнения функции $f(x)$, найдены частные случаи, для которых порядок группы вычисляется со сложностью $S < O(log^9 q)$ [21, 22, 43]. При $p\neq2$ и $p\neq3$ такого рода ЭК задаются уравнением

$$y^2 \equiv x^3 + ax + b \pmod p,$$

где хотя бы один из коэффициентов a или b равен 0.

Если характеристика поля $p \equiv 2 \pmod 3$ и $p \equiv 5 \pmod 6$, а также коэффициент $a=0$, происходит взаимно-однозначное отображение поля. Так как количество ненулевых элементов поля четное и ровно половина из них является квадратичными вычетами, то сумма характеров по всем x будет равна нулю и группа будет циклической с порядком $\#E(\mathrm{GF}(p)) = p + 1$. Впервые такого вида кривые для синтеза криптосистем были предложены в работе [91]. При $p \equiv 3 \pmod 4$ и $b=0$ порядок группы также будет равен $\#E(\mathrm{GF}(p)) = p + 1$, однако в этом случае нет взаимно-однозначного отображения поля. Здесь число –1 не является квадратическим вычетом. Следовательно, если число δ является квадратическим вычетом, то $-\delta$ им уже не является. Разобьем множество ненулевых элементов поля на две половины, положительную и отрицательную, по правилу, если $\delta < \dfrac{p+1}{2}$, то δ положительное, иначе – отрицательное. Функция ax^3+b является нечетной, поэтому сумма характеров от ax^3+b и от $-ax^3-b$ будет нулевой. Так как все множество ненулевых элементов поля разбивается на две половины, то сумма характеров по всем ненулевым x также будет нулевой.

Для кривых над полями с $p \equiv 1 \pmod 6$ и $p \equiv 1 \pmod 4$ с неполными уравнениями правой части порядок группы точек кривой $\#E(\mathrm{GF}(p))$ определяется разложением чисел p в кольцах алгебраических $\mathbf{Z}[\omega]$ и комплексных $\mathbf{Z}[i]$ чисел соответственно, где $i = \sqrt{-1}$, $\omega = \dfrac{\left(-1+\sqrt{-3}\right)}{2}$.

Подставив уравнение ЭК $y^2 \equiv x^3+b \pmod p$, где $p \equiv 1 \pmod 6$, в выражение (3.20), получим:

$$\#E\left(\mathrm{GF}(p)\right) = 1 + \sum_{\varsigma+\eta=b} N\left(y^2=\varsigma\right) N\left(x^3=-\eta\right). \qquad (3.22)$$

Выражение (3.22) можно преобразовать: $\#E(\mathrm{GF}(p)) = 1 + \sum_{\varsigma+\eta=b} \sum_{i=0}^{1} \chi^i(\varsigma) \sum_{j=0}^{2} \chi^j(-\eta)$, а с учетом того, что $\sum_{\varsigma+\eta=b} \chi^0(\varsigma)\chi^0(\eta) = p$,

$$\#E(\mathrm{GF}(p)) = p + 1 + \sum_{\varsigma+\eta=b} \chi^1(\varsigma)\chi^1(\eta) + \sum_{\varsigma+\eta=b} \chi^1(\varsigma)\chi^2(\eta). \qquad (3.23)$$

Известно [1], что сумма Якоби $J\left(\chi^1(\varsigma), \chi^1(\eta)\right) = \sum_{\varsigma+\eta=1} \chi^1(\varsigma)\chi^1(\eta)$, тогда осуществив замену переменных: $\varsigma = \varsigma b$, $\eta = \eta b$, и переобозначив $\chi^1(\varsigma) = \varphi(\varsigma)$, $\chi^1(\eta) = \chi(\eta)$, получим

$$\sum_{\varsigma+\eta=b} \varphi(\varsigma)\chi(\eta) = \sum_{\varsigma'+\eta'=1} \varphi(\varsigma' b)\chi(\eta' b) = \varphi(b)\chi(b) \sum_{\varsigma'+\eta'=1} \varphi(\varsigma')\chi(\eta'),$$

где согласно свойствам мультипликативных характеров $\varphi(b)\chi(b) = \varphi\chi(b)$. Следовательно,

$$\sum_{\varsigma+\eta=b} \varphi(\varsigma)\chi(\eta) = \varphi\chi(b) J\left(\varphi(\varsigma), \chi(\eta)\right). \tag{3.24}$$

Подставим выражение (3.24) в (3.23):

$$\#E(\mathrm{GF}(p)) = p + 1 + \varphi\chi(b) J\left(\varphi(\varsigma), \chi(\eta)\right) + \overline{\varphi\chi(b) J\left(\varphi(\varsigma), \chi(\eta)\right)}. \tag{3.25}$$

Используя свойство [43] $J\left(\varphi(\varsigma), \chi(\eta)\right) = \chi(4) J\left(\chi(\eta), \chi(\eta)\right)$, выражение (3.25) можно преобразовать к виду

$$\#E(\mathrm{GF}(p))=p+1+\varphi\chi(4b) J\left(\chi(\eta), \chi(\eta)\right) + \overline{\varphi\chi(4b) J\left(\chi(\eta), \chi(\eta)\right)}.$$

Характеристика конечного поля p раскладывается на множители в кольце алгебраических чисел $\mathbf{Z}[\omega]$ следующим образом:

$$\mathrm{p}= \pi\overline{\pi} = (d + e\omega)\overline{(d + e\omega)} = (d + e\omega)(d + e\omega^2) = \mathrm{d}^2 - \mathrm{de} + \mathrm{e}^2, \tag{3.26}$$

где $d \equiv 2 \pmod 3$, $e \equiv 0 \pmod 3$. Тогда, если в качестве характера порядка 6 взять символ вычета степени 6 – $\left(\frac{d}{\pi}\right)_6$, можно получить $\varphi\chi(d) = \overline{\left(\frac{d}{\pi}\right)_6} = \left(\left(\frac{d}{\pi}\right)_6\right)^5$. В рассматриваемом случае сумма Якоби является комплексным числом, модуль которого равен π.

Таким образом, для кривой $y^2 \equiv x^3 + b \pmod p$, где $p \equiv 1 \pmod 6$ имеет место равенство $\#E(GF(p)) = p + 1 + \overline{\left(\frac{4b}{\pi}\right)_6}\,\pi + \left(\frac{4b}{\pi}\right)_6 \overline{\pi}$. Так как символ вычета степени 6 может принимать значения $\{\pm 1, \pm\omega, \pm\omega^2\}$, то, в зависимости от разложения характеристики поля p, количество точек ЭК:

$$\#E(\mathrm{GF}(p)) = p + 1 \pm s, \tag{3.27}$$

где $s = \begin{cases} d + e \\ 2d - e \\ d - 2e \end{cases}$, знаки “+”, “–” выбираются в зависимости от того, является или нет коэффициент уравнения ЭК b квадратическим или кубическим вычетом.

Анализ выражения (3.27) приводит к следствию.

Следствие 3.1

a) Если коэффициент b является кубическим и квадратическим вычетом одновременно, то с учетом точки в бесконечности $\#E(\mathrm{GF}(p)) = 6m$, где здесь и далее m – простое число;

b) Если коэффициент b является только кубическим вычетом, то с учетом точки в бесконечности $\#E(\mathrm{GF}(p)) = 3m$;

c) Если коэффициент b является только квадратическим вычетом, то с учетом точки в бесконечности $\#E(\mathrm{GF}(p)) = 2m$;

d) Если коэффициент b является кубическим и квадратичным невычетом одновременно, то $\#E(\mathrm{GF}(p)) = 6m + 1$, где дополнительное слагаемое означает точку в бесконечности, следовательно, $\#E(\mathrm{GF}(p)) \equiv 1 \pmod 6$.

Порядок группы точек ЭК вида $y^2 \equiv x^3 + ax \pmod p$ при $p \equiv 1 \pmod 4$ определяется разложением p в кольце комплексных чисел $\mathbf{Z}[i]$:

$$\mathrm{p} = \pi\overline{\pi} = (d + ei)\overline{(d + ei)} = (d + ei)(d - ei) = \mathrm{d}^2 + \mathrm{e}^2. \qquad (3.28)$$

Проведя преобразования, аналогичные (3.23–3.26), можно получить [1, 33, 46] $\#E(GF(p)) = p + 1 - \overline{\left(\frac{a}{\pi}\right)_4}\,\pi - \left(\frac{a}{\pi}\right)_4\overline{\pi}$, где $\left(\frac{a}{\pi}\right)_4$ – символ вычета четвертой степени (биквадратичного вычета). Подставив в данное выражение значения символа вычета четвертой степени $\{\pm 1, \pm i\}$, получим

$$\#E(\mathrm{GF}(p)) = p + 1 \pm h, \qquad (3.29)$$

где $h = \begin{cases} 2d \\ 2e \end{cases}$. Если значение коэффициента a, взятое с обратным знаком, т.е. $(p–a)$, есть квадратический вычет в поле $\mathrm{GF}(p)$, то $\#E(\mathrm{GF}(p)) = p + 1 \pm 2d$. Если значение $(p–a)$ является квадратическим невычетом, тогда $\#E(\mathrm{GF}(p)) = p + 1 \pm 2e$.

Анализируя выражение (3.29), получим следствие.

Следствие 3.2

a) Если значение коэффициента a уравнения ЭК $y^2 = x^3 + ax \pmod p$, взятое с обратным знаком: $(p–a)$, при $p \equiv 1 \pmod 4$ является квадратическим невычетом в поле $\mathrm{GF}(p)$, то $\#E(\mathrm{GF}(p)) \equiv 2m$, где m-простое число;

b) Если значение $(p–a)$ является квадратическим вычетом, то порядок группы точек $\#E(\mathrm{GF}(p)) \equiv 4m$.

Рассмотренные уравнения кривых исчерпывают все неизоморфные случаи ЭК с неполными уравнениями правой части.

Кривые с нулевым дискриминантом

Кривые с дискриминантом $\Delta = -2^4\left(2^2 a + 3^3 b\right) (\mathrm{mod}\ p) \equiv 0\ (\mathrm{mod}\ p)$ являются вырожденными кривыми и имеют кратные корни правой части уравнения ЭК. В общем случае их не рекомендуется использовать в качестве основы для создания криптографических конструкций, так как группа точек таких кривых изоморфна либо аддитивной, либо мультипликативной группе кольца $\mathbf{Z}_p$. Первый вариант приводит к тому, что сложность дискретного логарифмирования в группе точек кривой эквивалентна сложности деления целых чисел, что неприемлемо при синтезе криптографических конструкций. Второй вариант приводит к тому, что стойкость криптографической конструкции будет определяться сложностью дискретного логарифмирования в кольце целых чисел, что эквивалентно известным криптосистемам, и может использоваться в эллиптических криптографических конструкциях в случае, когда порядок группы точек кривой достаточно велик.

Для поиска порядка группы точек на ЭК такого рода предлагается следующее утверждение.

Утверждение 3.1

Если $\Delta \equiv 0\ (\mathrm{mod}\ p)$, а коэффициенты $a \neq 0$ и $b \neq 0$, тогда $\#E(\mathrm{GF}(p)) = p + 1 \pm 1$, в зависимости от того, является или нет коэффициент b квадратическим вычетом поля $\mathrm{GF}(p)$. Утверждение верно при выполнении условий:

$\left\{\left[p \equiv 3\ (\mathrm{mod}\ 3)\right] \bigcap \left[\left(p \neq 1\ (\mathrm{mod}\ 4)\right) \bigcup \left(p \equiv 3\ (\mathrm{mod}\ 4)\right)\right]\right\}$, а также

$\left\{\left[p \equiv 1\ (\mathrm{mod}\ 3)\right] \bigcap \left[\left(p \equiv 1\ (\mathrm{mod}\ 4)\right) \bigcup \left(p \neq 3\ (\mathrm{mod}\ 4)\right)\right]\right\}$.

Напротив, если выполняется условие:

$\left\{\left[p \equiv 2\ (\mathrm{mod}\ 3)\right] \bigcap \left[\left(p \equiv 1\ (\mathrm{mod}\ 4)\right) \bigcup \left(p \neq 3\ (\mathrm{mod}\ 4)\right)\right]\right\}$, а также, когда

$\left\{\left[p \equiv 1\ (\mathrm{mod}\ 3)\right] \bigcap \left[\left(p \neq 1\ (\mathrm{mod}\ 4)\right) \bigcup \left(p \equiv 3\ (\mathrm{mod}\ 4)\right)\right]\right\}$ – порядок группы тот же самый, но в зависимости от того, не является или является коэффициент b квадратическим вычетом.

Доказательство

Рассмотрим случай, когда характеристика поля $p \equiv 2\ (\mathrm{mod}\ 3)$ и одновременно $p \equiv 3\ (\mathrm{mod}\ 4)$. Поскольку в этом случае число –1 не является квадратическим вычетом, то функция $x^3 + ax$ будет нечетной и сумма характеров данной функции для всех $x \in \mathrm{GF}(p)$ равна 0, поэтому число решений уравнения ЭК будет определяться в соответствии с формулой (3.18) мультипликативным характером коэффициента b (символом Лежандра $\left(\frac{b}{p}\right)$). Используя определение символа Лежандра, получаем, если ко-

эффициент b является квадратическим вычетом, то $\left(\frac{b}{p}\right) = 1$ и порядок группы $\#E(\mathrm{GF}(p)) = p + 1 + 1$, если b не является квадратическим вычетом, то $\left(\frac{b}{p}\right) = -1$ и $\#E(\mathrm{GF}(p)) = p + 1 - 1$. По аналогии с этим доказываются остальные случаи данного утверждения.

ЭК над полями Галуа

Для вычисления порядка группы кривой над расширенным полем Галуа $\mathrm{GF}(p^k)$ достаточно найти число точек над простым полем $\mathrm{GF}(p)$. Если $\#E(\mathrm{GF}(p)) = p + 1 - 2d$ — порядок группы над простым полем, то можно найти комплексные числа α и β такие, что $p = \alpha\beta$, а $-2d = \alpha + \beta$. Тогда порядок группы для ЭК над полем степени расширения k с коэффициентами кривой $a, b \in \mathrm{GF}(p)$ можно определить из равенства

$$\#E(\mathrm{GF}(p^k)) = p^k + 1 + \alpha^k + \beta^k. \qquad (3.30)$$

Выражение (3.30) справедливо для полей Галуа с любой характеристикой p. На его основе предлагаются два утверждения.

Утверждение 3.2

Если над полем GF(2) ЭК имеет вид $y^2 + xy = x^3 + ax^2 + b$ с коэффициентами $a, b \in \mathrm{GF}(2)$, тогда порядок группы точек $\#E(\mathrm{GF}(2)) = \{2, 4\}$, а порядок группы точек $\#E(\mathrm{GF}(2^k))$ над расширениями поля GF(2) можно определить из выражения:

$$\#E\left(\mathrm{GF}(2^k)\right) = 2^k + 1 - 2^{\frac{k}{2}+1}\left\lceil \cos\left(k \arctan\left(\pm\sqrt{7}\right)\right)\right\rceil, \qquad (3.31)$$

$\forall\ k$ при $\#E(\mathrm{GF}(2)) = 2$ и $\#E(\mathrm{GF}(2)) = 4$, соответственно.

Доказательство

Эллиптическая кривая $y^2 + xy = x^3$ (a=0, b=0) в поле GF(2) имеет только одно решение, а именно (0, 0), с учетом точки в бесконечности получаем $\#E(\mathrm{GF}(2)) = 2$. Поскольку $\#E(\mathrm{GF}(2)) = 2 + 1 - 2d$, следовательно $-2d = -1$. Решая систему уравнений $\begin{cases}\alpha + \beta = -1 \\ \alpha\beta = 2\end{cases}$, получим $\alpha = \frac{-1 \pm \sqrt{-7}}{2}$, $\beta = \frac{-1 \mp \sqrt{-7}}{2}$. Подставим значения α и β в равенство (3.30), откуда $\#E\left(\mathrm{GF}(2^k)\right) = 2^k + 1 - \frac{\left(1-\sqrt{-7}\right)^k + \left(1+\sqrt{-7}\right)^k}{2^k}$. Применяя тригонометрическое представление комплексного числа и его свойства при воз-

ведении в степень, приходим к итоговому равенству $\#E\left(\mathrm{GF}(2^k)\right) = 2^k + 1 - 2^{\frac{k}{2}+1}\left\lceil \cos\left(k \arctan\sqrt{7}\right)\right\rceil$. Аналогично, ЭК $y^2 + xy = x^3 + x + 1$ (a=1, b=1) имеет решение (0, 1) плюс точка в бесконечности и полученное выше равенство применимо к данной кривой для поиска порядка группы в поле GF(2^k).

Кривые $y^2 + xy = x^3 + 1$ (a=0, b=1) и $y^2 + xy = x^3 + x^2$ (a=1, b=0) в поле GF(2) имеют решения (0, 1), (1, 0), (1, 1) плюс точка в бесконечности, т.е. $\#E(\mathrm{GF}(2)) = 4$. Тогда $-2d = 1$ и решая систему уравнений $\begin{cases}\alpha+\beta=1\\ \alpha\beta=2\end{cases}$, получим $\alpha = \frac{1\pm\sqrt{-7}}{2}$, $\beta = \frac{1\mp\sqrt{-7}}{2}$. Подставив полученные значения в (3.30), получим $\#E\left(\mathrm{GF}(2^k)\right) = 2^k + 1 + \frac{\left(1+\sqrt{-7}\right)^k + \left(1-\sqrt{-7}\right)^k}{2^k}$. После окончательного преобразования

$$\#E\left(\mathrm{GF}(2^k)\right) = 2^k + 1 - 2^{\frac{k}{2}+1}\left\lceil \cos\left(k \cdot \arctan\left(-\sqrt{7}\right)\right)\right\rceil.$$

Утверждение 3.3

Если над полем GF(3) ЭК с коэффициентами $a, b, c \in \mathrm{GF}(3)$ имеет инвариант $j \neq 0$, тогда порядок группы точек $\#E(\mathrm{GF}(3))=\{1, 2, 3, 4, 5, 6, 7\}$, а порядки группы точек $\#E(\mathrm{GF}(3^k))$ над расширениями поля GF(3) степени k определяются из выражений:

$$\#E\left(\mathrm{GF}(3^k)\right) = 3^k + 1 - 2\cdot 3^{\frac{k}{2}}\left\lceil \cos\left(k \arctan\left(\pm\sqrt{11}\right)\right)\right\rceil \tag{3.32}$$

для любого k при $\#E(\mathrm{GF}(3))=5$ и $\#E(\mathrm{GF}(3))=3$, соответственно;

$$\#E\left(\mathrm{GF}(3^k)\right) = 3^k + 1 - 2\cdot 3^{\frac{k}{2}}\left\lceil \cos\left(k \arctan\left(\pm\sqrt{2}\right)\right)\right\rceil \tag{3.33}$$

для любого k при $\#E(\mathrm{GF}(3))=6$ и $\#E(\mathrm{GF}(3))=2$, соответственно;

$$\#E\left(\mathrm{GF}(3^k)\right) = 3^k + 1 - 2\cdot 3^{\frac{k}{2}}\left\lceil \cos\left(k \arctan\left(\pm 3^{-\frac{1}{2}}\right)\right)\right\rceil \tag{3.34}$$

для любого k при $\#E(\mathrm{GF}(3))=7$ и $\#E(\mathrm{GF}(3))=1$, соответственно;

$$\#E\left(\mathrm{GF}(3^k)\right) = \begin{cases} 3^k + 1 - 2\cdot(-3)^{k/2} \\ 3^k + 1 \end{cases} \tag{3.35}$$

для $\#E(\mathrm{GF}(3))=4$ при k - четном и k – нечетном, соответственно.

Доказательство

Над конечным полем GF(3) невырожденная ЭК $y^2 = x^3 + ax^2 + bx + c$ может иметь следующее количество решений, с учетом точки в бесконечности: {1, 2, 3, 4, 5, 6, 7} в зависимости от значения коэффициентов a, b, c. В табл. 3.10 представлены все возможные порядки групп точек на ЭК в зависимости от коэффициентов уравнения кривой. Поскольку ЭК с $\#E(\mathrm{GF}(3))=4$ встречаются наиболее часто, докажем выражение для порядка группы на этих кривых над полями $\mathrm{GF}(3^k)$.

Таблица 3.10

Порядок группы $\#E(\mathrm{GF}(3))$	**Коэффициенты уравнения ЭК**		
	a	***b***	***c***
1	0	2	2
2	2 2 2	0 1 2	2 2 0
3	1	1	2
4	0 0 0 0 0 0 0 1 1 1 1 1 1	0 0 0 1 1 1 2 0 0 1 2 2 2	0 1 2 0 1 2 0 0 2 0 0 1 2
5	2 2 2 2 2 2	0 0 1 1 2 2	0 1 0 1 1 2
6	1	1	1
7	0 1	2 0	1 1

Так как $\#E(\mathrm{GF}(3))=4$, то $-2d=0$, следовательно, получаем систему уравнений $\begin{cases}\alpha+\beta=0\\ \alpha\beta=3\end{cases}$, решая которую, найдем $\alpha=\pm\sqrt{-3}$, $\beta=\mp\sqrt{-3}$. Затем, подставив значения α и β в равенство (30), получим $\#E\left(\mathrm{GF}(3^k)\right)=3^k+1-\left(\sqrt{-3}\right)^k+\left(\sqrt{-3}\right)^k$. Из полученного равенства видно, что в случае нечетного значения степени расширения поля k комплексные числа взаимно уничтожаются, а при k четном удваиваются. Таким образом, $\#E\left(\mathrm{GF}(3^k)\right)=\begin{cases}3^k+1-2\cdot(-3)^{k/2} & \text{для } k-\text{четного},\\ 3^k+1 & \text{для } k-\text{нечетного}.\end{cases}$. Для кривых с другими порядками групп доказательство осуществляется аналогично.

Алгоритм поиска порядка группы точек ЭК над конечными полями GF(p^n) на основе применения последовательностей Лукаса

В работе [76] рассмотрена возможность использования последовательностей Лукаса в криптографических целях. Например, свойства делимости U_k используются для проверки на простоту числа N при разложении числа $N+1$ (что также относится и к криптоанализу систем, основанных на возведении в степень больших чисел). Здесь предлагается вариант алгоритма поиска порядка группы точек ЭК на основе использования последовательностей Лукаса.

Пусть даны: P, Q – два рациональных числа, α – корень в неквадратичном поле, β – обратная к α величина в данном поле. Тогда последовательность Лукаса $\{U_k\}_{k\geq 0}$ и $\{V_k\}_{k\geq 0}$ с параметрами P и Q определяется следующим образом:

$$U_k(P,Q)=\frac{\alpha^k-\beta^k}{\alpha-\beta} \tag{3.36}$$

$$V_k(P,Q)=\alpha^k+\beta^k. \tag{3.37}$$

Каждый последующий член последовательности можно определить с помощью выражений

$$U_{i+j}=U_iV_j-Q^jU_{i-j}, \tag{3.38}$$

$$V_{i+j}=V_iV_j-Q^jV_{i-j}. \tag{3.39}$$

Пусть p – простое число (характеристика конечного поля) $q=p^r$ и дана ЭК $E(\mathrm{GF}(q))$ такая, что порядок группы точек $\#E(\mathrm{GF}(q))=q+1-r$. Согласно (3.30) имеем

$$\#E(\mathrm{GF}(q^l))=q^l+1-\alpha^l-\beta^l, \tag{3.40}$$

где α и β в общем случае определяются из разложения $1 - tT + qT^2 = (1 - \alpha T)(1 - \beta T)$.

Пусть p=2, тогда ЭК согласно табл. 3.1 имеет вид *E:* $y^2 + xy = x^3 + ax^2 + b$. Примем число r маленьким, тогда $t = 2^r + 1 - \#E(\mathrm{GF}(2^r))$. Теперь ЭК будем рассматривать над полем $\mathrm{GF}(2^m)$, где $m=rl$. Подставив (3.37) в (3.40), получим

$$\#E(\mathrm{GF}(2^m)) = 2^m + 1 - V_l(t, 2^r), \qquad (3.41)$$

где $V_l(t, 2^r)$ – l-й член последовательности Лукаса $\{V_k\}$ с параметрами $P=t$ и $Q=2^r$, определяемый из выражений (3.38), (3.39).

Таким образом, зная порядок группы точек ЭК с коэффициентами в поле $\mathrm{GF}(2^r)$, несложно определить порядок группы этой же кривой над полем $\mathrm{GF}(2^m) = \mathrm{GF}(2^{rl})$. Полученные выражения (3.31-3.35) для порядка группы являются частным случаем (3.41), где для определения порядка группы используются последовательности Лукаса.

Алгоритм генерации ЭК вида $y^2 \equiv x^3 + b \pmod p$ с нулевым инвариантом j

Изменение порядка группы точек может происходить за счет изменения коэффициентов уравнения ЭК и конечного поля (кольца целых чисел), над которым рассматривается ЭК. На основе полученных результатов предлагаются алгоритмы выбора ЭК с требуемыми характеристиками.

Входом первого алгоритма является требуемая величина стойкости, что равносильно требуемой длине n характеристики поля p.

1. Сгенерировать характеристику поля $p \equiv 1 \pmod 6$ с требуемой длиной n-бит.
2. Разложить простое число p на множители в кольце $\mathbf{Z}[\omega]$ в соответствии с (3.26).
3. Проверить для одного из значений $\#E(\mathrm{GF}(p))$, полученного в соответствии с (3.27), выполнение равенства $\#E(\mathrm{GF}(p))=m$ или выполнение одного из условий следствия 1. В случае невыполнения перейти к шагу 1.
4. Проверить невыполнимость условия делимости (3.17) вплоть до заданного значения k, в ином случае перейти к шагу 1.
5. Сгенерировать точку $P_0 = (x_0, y_0)$, где $x_0, y_0 \in \mathrm{GF}(p)$ и не равны 0. Вычислить значение коэффициента $b := y_0^2 - x_0^3 \pmod p$.
6. Проверить выполнение условий соответствия коэффициента b следствию 1 для полученного на шаге 3 значения $\#E(\mathrm{GF}(p))$, т.е. b – квадратический и кубический невычет для $\#E(\mathrm{GF}(p)) = m$, b – квадратический невычет и кубический вычет для $\#E(\mathrm{GF}(p)) = 2m$, b – квадратический вычет и кубический невычет для $\#E(\mathrm{GF}(p)) = 3m$, b – квадратический и кубический вычет для $\#E(\mathrm{GF}(p)) = 6m$. Если условия не выполняются, то перейти к шагу 5.

7. Проверить выполнение равенства $\#E(\mathrm{GF}(p))P_0 = P_\infty$, где $P_\infty = \mathbf{O}$ является точкой в бесконечности, т.е. знаменатель в формулах группового закона сложения (5) обращается в ноль. Если равенство не выполняется, перейти к шагу 5.
8. Присвоить значение генератору циклической подгруппы точек ЭК $G := \frac{\#E(\mathrm{GF}(p))}{m} P_0$.
9. Выход: получены значения характеристики поля p, коэффициента уравнения ЭК – b, порядка циклической подгруппы – m, генератора циклической подгруппы G.

Сложность данного алгоритма оценивается величиной $O(log^4\, p)$, так как сложность каждого из шагов не превышает значения $O(log^3\, p)$ (что в первую очередь определяется сложностью тестирования числа p на простоту), а шаги 1-3 повторяются $O(log\, p)$ раз.

Алгоритм генерации ЭК вида $y^2 \equiv x^3 + ax \pmod p$ с инвариантом $j = 1728$

Входом алгоритма, как и в предыдущем случае, являются требуемая длина n характеристики поля p.

1. Сгенерировать характеристику поля $p \equiv 1 (\mathrm{mod}\ 4)$ с требуемой длиной n-бит.
2. Разложить простое число p на множители в кольце $\mathbf{Z}[i]$ в соответствии с (3.28).
3. Проверить для одного из значений $\#E(\mathrm{GF}(p))$, полученного из (3.29), выполнение одного из условий следствия 2. В случае невыполнения перейти к шагу 1.
4. Проверить невыполнимость условия делимости (3.17) вплоть до заданного значения k, в ином случае перейти к шагу 1.
5. Сгенерировать точку $P_0 = (x_0, y_0)$, где $x_0, y_0 \in \mathrm{GF}(p)$ и не равны 0. Вычислить значение коэффициента $a := \frac{y_0^2 - x_0^3}{x_0} \pmod p$.
6. Проверить выполнение условий соответствия коэффициента a следствию 2 для полученного на шаге 3 значения $\#E(\mathrm{GF}(p))$, т.е. $(p–a)$ – квадратический невычет для $\#E(\mathrm{GF}(p)) = 2m$, $(p–a)$ – квадратический вычет для $\#E(\mathrm{GF}(p)) = 4m$. Если условия не выполняются, перейти к шагу 5.
7. Проверить выполнение равенства $\#E(\mathrm{GF}(p))P_0 = \mathbf{O}$, т.е. знаменатель в формулах группового закона сложения (3.5) обращается в ноль. Если равенство не выполняется, перейти к шагу 5.
8. Присвоить значение генератору циклической подгруппы точек ЭК $G := \frac{\#E(\mathrm{GF}(p))}{m} P_0$.

9. Выход: получены значения характеристики поля p, коэффициента уравнения ЭК – b, порядка циклической подгруппы – m, генератора циклической подгруппы G.

Сложность данного алгоритма, как и предыдущего, оценивается величиной $O(log^4 p)$.

На основе использования следствий 1, 2, утверждений 1-3, выражений (3.17, 3.20-3.41) и табл. 3.1 можно построить обобщенную методику выбора ЭК с требуемым порядком группы точек для разработки эффективных криптографических конструкций.

Исходными данными являются: требуемая длина n характеристики поля p, либо требуемая величина порядка группы $\#E(\mathrm{GF}(q))$.

Обобщенная методика выбора ЭК над полем GF(q):

1. Определить в соответствии с исходными данными с помощью выражения (3.21) характеристику p конечного поля $\mathrm{GF}(p^k)$ или кольца $\mathbf{Z}_p$ и степень расширения – k.
2. Выбрать уравнение ЭК из табл. 3.1 в соответствии с характеристикой p.
3. Сгенерировать коэффициенты уравнения ЭК $a, b \in \mathrm{GF}(p)$. Если p=2 или p=3, то перейти к шагу 4, если p>3, то перейти к шагу 5 .
4. Определить порядок группы $\#E(\mathrm{GF}(q))$ в соответствии с выражениями (3.31-3.41). Перейти к шагу 6 алгоритма.
5. Определить порядок группы $\#E(\mathrm{GF}(q))$ в соответствии с условиями следствий 3.1, 3.2, утверждениями 3.1-3.3, выражениями (3.27, 3.29). Если коэффициенты ЭК a (при $p \equiv 2$ (mod 6)) или b (при $p \equiv 3$ (mod 4) равны 0, то порядок группы $\#E(\mathrm{GF}(q)) = p+1$.
6. Проверить невыполнимость условия делимости для полученного порядка группы согласно выражению (3.17) – $\#E(\mathrm{GF}(q)) \mid \left(q^i - 1\right)$ для значений i = 1, 2, …, 30. Если свойство делимости выполняется для одного из значений i, тогда вернуться к шагу 3 алгоритма.
7. Выход: получены значения характеристики поля p, уравнение ЭК с порядком группы, удовлетворяющим показателям стойкости.

Таким образом, для синтеза криптографических конструкций необходимо использовать ЭК над полем большой простой характеристики p – $\mathrm{GF}(p)$ (кольцом целых чисел $\mathbf{Z}_p$), либо над расширениями полей с характеристиками два, три – $\mathrm{GF}(2^k)$, $\mathrm{GF}(3^k)$. Правильно выбранный порядок группы обеспечивает высокую стойкость криптографических конструкций к различным методам анализа.

3.4.7 Оптимизация параметров эллиптических криптографических конструкций

Так как сложность анализа криптоалгоритма на основе ЭК оценивается величиной $W = O\left(\sqrt{N}\right)$, где $N = \#E(\mathrm{GF}(q))$ – порядок циклической группы (или наибольший общий множитель циклической подгруппы) точек ЭК, предположим, что N можно представить в виде $N = 2^n$, где значение n – соизмеримо с алгоритмической длиной ключа криптографической конструкции. Пусть $W_1 = \sqrt{2^{n_1}}$ – сложность вскрытия при длине ключа n_1. Если уменьшить алгоритмическую длину ключа в k раз, то есть принять $n_2 = n_1 / k$, но при этом конструктивную длину ключа оставить равной n_1, тогда сложность вскрытия информации $W_2 = k\sqrt{2^{n_1/k}}$. Следовательно, можно определить функцию проигрыша в сложности анализа криптографической конструкции

$$K_{\Pi}\left(n_1, k\right) = \frac{2^{\frac{n_1(k-1)}{2k}}}{k}. \tag{3.42}$$

Сложность криптографического преобразования информации, для алгоритмов на основе ЭК, оценивается величиной $W_к = \log^2 q$, где q – это мощность конечного поля Галуа, над которым осуществляется криптопреобразование. Если поле является расширением двоичного поля, тогда $q = 2^l$, где l – степень расширения поля. Будем считать, что q соизмеримо с порядком группы N, в этом случае l=n. Пусть $W_{к1} = \log^2 2^{n_1} = n_1^2$ – сложность криптопреобразования информации на алгоритмической длине ключа n_1, $W_{к2} = k\log^2 2^{n_2} = \frac{n_1^2}{k}$ – сложность криптопреобразования информации на конструктивной длине ключа n_1 с алгоритмической длиной ключа n_2. Очевидно, что во втором случае сложность преобразования уменьшается в k раз, поэтому можно ввести функцию выигрыша в сложности преобразования

$$K_{\mathrm{B}} = \frac{W_{k2}}{W_{k1}} = \frac{1}{k}. \tag{3.43}$$

Таким образом, выбор конструктивной длины ключа n_1 и алгоритмической длины ключа n_2 будет оптимальным, если выполняются условия:

$$\begin{cases} K_{\Pi*}\left(n_1, k\right) = \min\limits_{\{n_1, k\}} \dfrac{2^{\frac{n_1(k-1)}{2k}}}{k} \\ K_{\mathrm{B}*} = \min\limits_{\{k\}} \dfrac{1}{k} \end{cases} \tag{3.44}$$

на всем множестве допустимых параметров $n_1 = \{128, \ldots, 2048\}$ и $k = \{2, \ldots, 16\}$ с учетом ограничений на скорость преобразования информации и сложность анализа криптографической конструкции.

3.4.8 Протоколы защищенного информационного обмена на основе свойств эллиптических кривых

Протоколы аутентификации можно классифицировать в соответствии со следующими параметрами: типу аутентификации, тип используемой криптосистемы, вид реализации криптосистемы, количеству обменов между абонентами. Дополнительно они могут различаться наличием диалога и доверия между абонентами, а также использованием в протоколах отметок времени. Обобщенная классификация протоколов аутентификации в соответствии с данными параметрами представлена на рис. 3.4.

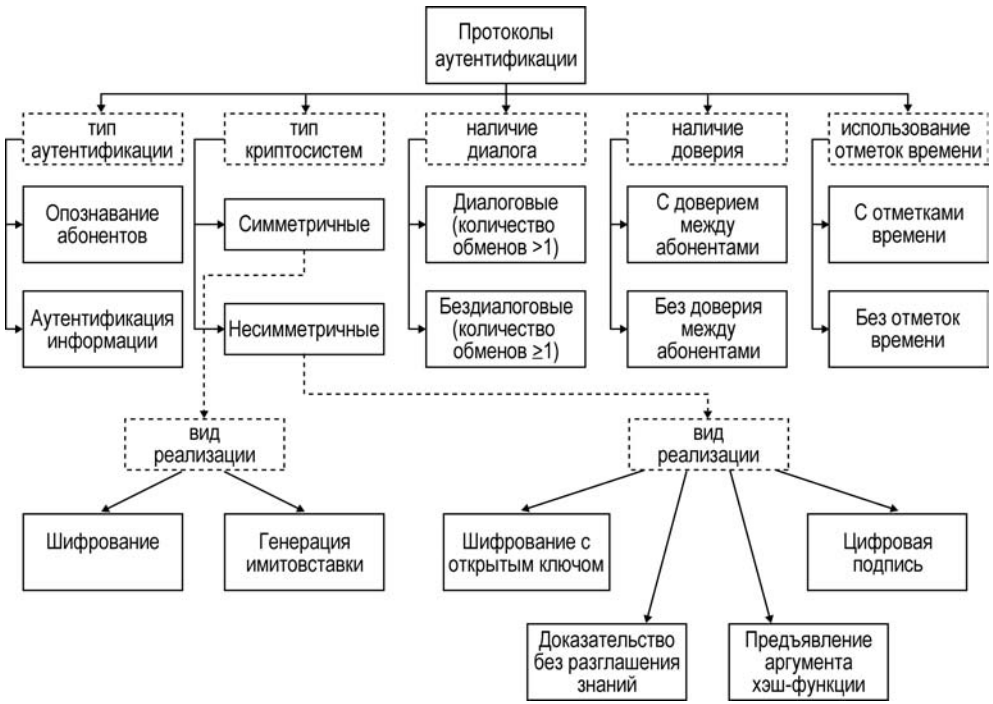

Рис. 3.4.

Наиболее простой случай соответствует аутентификации в условиях доверия абонентов. В этом случае весь набор защитных функций может быть реализован на основе стандартизированных криптографических функций, например, симметричного шифрования или ключевой хэш-функции по алгоритму ГОСТ 28147-89 [8]. Однократная аутентификация информации в условиях недоверия абонентов может осуще-

ствляться протоколами Меркля и Рабина [43, 113] с использованием симметричного шифрования по ГОСТ 28147-89. В основу однократного бездиалогового опознавания в условиях недоверия абонентов положена задача обращения вычислимой в одну сторону бесключевой хэш-функции [89]. В этом случае хэш-функция по ГОСТ Р 34.11-94 может использоваться, если в протоколе аутентификации допустимы коллизии.

Протоколы многократной аутентификации в условиях недоверия между абонентами, в большинстве случаев, базируются на задачах дискретного логарифмирования [47, 89, 113]. Они реализуются в виде шифрования с открытым ключом, цифровой подписи или протокола доказательства без разглашения знаний. Для защиты от навязывания ранее переданных или задержанных данных в бездиалоговых протоколах в ряде случаев включают код времени при этом необходима синхронность отсчета времени у всех абонентов.

Диалоговый протокол подтверждения подлинности абонентов на основе доказательства без разглашения знаний требует передачи объема служебной информации в сотни раз большего, чем протокол на основе использования шифрования с открытым ключом, но при этом обладает сравнимой безопасностью. Протокол на основе использования цифровой подписи является менее безопасным. Кроме того, в протоколе на основе цифровой подписи предъявляются более жесткие требования к генератору случайных чисел, так как появление хотя бы однажды предсказуемого случайного числа или появление двух одинаковых случайных чисел приводит к вскрытию секретного ключа абонента, проходящего аутентификацию. Следовательно, шифрование с открытым ключом является наилучшим видом реализации диалоговых протоколов аутентификации абонентов.

Протоколы распределения ключей на основе ЭК

На основе ЭК над конечными полями и кольцами целых чисел можно строить криптографические протоколы различного назначения.

Определим, что перед началом обмена информацией в сети выбраны общие криптографические параметры:

- эллиптическая кривая $E(GF_p)$;
- группа точек кривой с порядком $\#E(GF_p)$;
- циклическая подгруппа группы точек ЭК с примитивным элементом G.

Данные параметры также могут создаваться непосредственно в ходе сеанса связи между двумя абонентами сети. В этом случае абонент, который получает криптографические параметры, может проконтролировать их безопасность. Для этого он проверяет следующие условия:

- характеристика поля p является простым числом;
- порядок группы $\#E(GF_p)$ является большим простым числом и $\#E(GF_p)\neq p$;

- не выполняется условие (3.17).

Протокол распределения ключей заключается в следующем:

Абонент **A** генерирует случайное положительное число n_A, которое будет являться его секретным ключом, вычисляет $P_A=n_AG$ и отправляет его абоненту **B**. В свою очередь абонент **B** генерирует случайное n_B, вычисляет $P_B=n_BG$ и отправляет его абоненту **A**. Тогда оба абонента сети могут найти общий секретный ключ $P_{AB}=n_Bn_AG=n_AP_B=n_BP_A$. Задача криптоаналитика по нахождению n_A, n_B или n_Bn_AG по известным G, n_AG и n_BG имеет вычислительную сложность, равную сложности дискретного логарифмирования в группе точек ЭК. Как видно, данный протокол представляет собой протокол передачи сеансового ключа по незащищенному каналу (аналог протокола Диффи-Хеллмана для мультипликативных полей [69]). Поэтапное представление этого протокола представлено в табл. 3.11

Таблица 3.11

Этап	Действия
1	**A** генерирует свой секретный ключ n_A и вычисляет $P_A=n_AG$
2	**B** генерирует свой секретный ключ n_B и вычисляет $P_B=n_BG$
3	**A** передает абоненту **B** свой открытый ключ P_A
4	**B** передает абоненту **A** свой открытый ключ P_B
5	**A** вычисляет ключ $K_A=n_AP_B=n_An_BG$
6	**B** вычисляет ключ $K_B=n_BP_A=n_Bn_AG$
7	$K_A=K_B=K_C$ – сеансовый ключ сгенерирован

Если злоумышленник может подменять подлинные сообщения и формировать вместо них свои, то предлагаемый протокол допускает атаку следующего вида («атаку гроссмейстера») [43, 113, 122]. Криптоаналитик устанавливает свой сеансовый ключ с абонентом **A**, выдавая себя за **B**, и свой сеансовый ключ с абонентом **B**, выдавая себя за **A**. В ходе сеанса связи он расшифровывает информацию, отправленную от **A** к **B**, на ключе, установленном с **A**, зашифровывает полученную информацию на ключе, установленном с **B**, и отправляет зашифрованную информацию абоненту **B**. Для исключения такого рода атак дополнительно следует использовать цифровую подпись или систему сертификации ключей у верификатора. При этом в целях незначительного снижения скорости обмена информацией ключ цифровой подписи может иметь существенно меньшую стойкость, т.е. основываться на задаче меньшей размерности, чем сеансовый ключ.

Протоколы ЭЦП на эллиптических кривых

ЭЦП на основе использования операций группы точек ЭК, определенной над конечным полем, является новым стандартом на ЭЦП в РФ ГОСТ Р 34.10-2001 [26].

Криптографическая стойкость схемы ЭЦП основывается на сложности решения задачи дискретного логарифмирования в группе точек ЭК, а также на стойкости используемой хэш-функции (ГОСТ Р 34.11–94). В стандарте не определен процесс генерации параметров схемы цифровой подписи, который определяется субъектами схемы ЭЦП.

Параметры цифровой подписи:

- простое число $p>2^{255}$ – модуль ЭК;
- ЭК E, задаваемая коэффициентами $a, b \in GF(p)$ или инвариантом $J(E)$;
- целое число $m=\#E(GF(p))$ – порядок группы точек ЭК;
- простое число q – порядок циклической подгруппы группы точек ЭК, значение которого удовлетворяет условиям:

$$\begin{cases} m = nq, \ n \in \mathbf{Z}, n \geq 1 \\ 2^{254} < q < 2^{256} \end{cases};$$

- точка $P \in E(GF(p))$ с координатами (x_p, y_p): $P \neq O$, $qP=O$;
- хэш-функция в соответствии с ГОСТ Р 34.11 $h(\cdot)$: $V_\infty \to V_{256}$, отображающая сообщения, представленные в виде двоичных векторов произвольной конечной длины, в двоичные вектора длины 256 бит;
- ключ подписи – целое число d: $0 < d < q$;
- ключ проверки подписи – точка $Q \in E(GF(p))$ с координатами (x_q, y_q): $dP = Q$.

На параметры схемы ЭЦП накладываются ограничения:

- выполнение условия $p^t \neq 1 \pmod q$, для всех $t = 1, 2, \ldots B$, где B удовлетворяет неравенству $B \geq 31$;
- выполнение неравенства $m \neq p$;
- выполнение условия $J(E) \neq 0$ или $J(E) \neq 1728$.

Алгоритм формирования цифровой подписи:

Исходные данные: сообщение $M \in V_\infty$, ключ подписи d.

1. Вычислить значение хэш-функции: $\bar{h} = h(M)$.
2. По двоичному значению вектора $\bar{h}$ вычислить целое число α и определить $e \equiv \alpha \pmod{q}$. Если e =0, то определить e =1.
3. Сгенерировать случайное целое число k, удовлетворяющее неравенству $0 < k < q$.
4. Вычислить точку ЭК $C=kP$ и определить $r \equiv x_C \pmod{q}$, где x_C – координата точки C. Если r = 0, то вернуться к шагу 3.
5. Вычислить значение $s \equiv (rd + ke) \pmod{q}$. Если s=0, вернуться к шагу 3.
6. Вычислить двоичные векторы $\bar{r}$ и $\bar{s}$, соответствующие целым числам r и s, соответственно.

Выход: ЭЦП в виде конкатенации двух двоичных векторов $\zeta = (\bar{r} \| \bar{s})$.

Алгоритм проверки цифровой подписи:

1. По полученной подписи $\zeta = (\bar{r} \| \bar{s})$ вычислить целые числа r и s. Если $0 < r < q$, $0 < s < q$, то перейти к следующему шагу. В ином случае, подпись неверна.
2. Вычислить значение хэш-функции полученного сообщения M: $\bar{h} = h(M)$.
3. По двоичному значению вектора $\bar{h}$ вычислить целое число α и определить $e \equiv \alpha \pmod{q}$. Если e =0, то определить e =1.
4. Вычислить значение $\nu \equiv e^{-1} \pmod{q}$.
5. Вычислить значения $z_1 \equiv s\nu \pmod{q}$, $z_2 \equiv -r\nu \pmod{q}$.
6. Вычислить точку ЭК $C = z_1P + z_2Q$ и определить значение $R \equiv x_C \pmod{q}$.
7. Если выполняется равенство $R=r$, то подпись верна, в ином случае, подпись неверна.

Модифицированный вариант ЭЦП с использованием криптографических конструкций на ЭК организуется следующим образом. Общий для всей сети открытый ключ содержит уравнение ЭК $E(GF(q))$ и генератор группы G. Абонент **A** генерирует случайное число $n_{\mathbf{A}}$, вычисляет точку $n_{\mathbf{A}}G$ и отправляет ее в центр распределения ключей **S** в качестве своего персонального открытого ключа, откуда любой из других абонентов сети может по запросу получить этот ключ. Для выполнения цифровой подписи сообщения $\bar{\varphi}$ находится его хэш-образ, такой, что $0 < \eta(\bar{\varphi}) < \#E(GF(q))$, абонент **A** на этапе 2 (табл.3.12) генерирует случайное число $m_{\mathbf{A}}$, вычисляет точку $R=m_{\mathbf{A}}G$ и число $r = x_R + y_R \pmod{\#E(GF(q))}$. Далее находится число s как решение уравнения

$$\eta(\bar{\varphi}) = n_{\mathbf{A}} r + m_{\mathbf{A}} s \ \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right). \qquad (3.45)$$

Подписью будет число s и точка R, которые вместе с сообщением передаются абоненту **B**. При проверке подлинности информации абонент **B** проверяет равенство

$$\eta(\bar{\varphi})G = r\left(n_{\mathbf{A}}G\right) + s\left(m_{\mathbf{A}}G\right). \qquad (3.46)$$

Как видно из протокола, уравнения генерирования и проверки подписи имеют один и тот же вид, только первое работает с индексами.

Таблица 3.12

Этап	Действия
1	$\mathbf{A} \to \mathbf{S}$: $P_{\mathbf{A}} = n_{\mathbf{A}}G$;
2	$\mathbf{A} \to \mathbf{B}$: $(\bar{\varphi}, s, R)$;
3	$\mathbf{B}$: $\eta(\bar{\varphi})G = P_{\mathbf{A}} \circ sR$.

Преимущество применения ЭК для выполнения цифровой подписи сообщений наряду с более высокой криптостойкостью состоит в том, что длина подписи может быть сокращена (при использовании ЭК определенного типа над полем $GF(q)$). Данное утверждение основано на том, что координата y точки R однозначно определяется своим знаком и значением координаты x. Знак координаты y находится из условия: если $y > \dfrac{(p-1)}{2}$, то плюс, иначе – минус. Поэтому вместо передачи обеих координат точки R можно передать только координату x и знак координаты y. На приемном конце для восстановления значения координаты y необходимо извлечь квадратный корень из правой части уравнения ЭК $f(x)$ по модулю p и взять соответствующее значение корня в кольце целых чисел $\mathbf{Z}_p$.

Рассмотрим безопасность протокола. Составными частями безопасности протокола являются криптостойкость и имитостойкость. Криптостойкость определяется сложностью нахождения индекса. Если предположить, что данная проблема решена, т.е. задача нахождения индекса вычислительно сложна, тогда криптоаналитик может попытаться подделать ЭЦП без вскрытия ключа. Рассмотрим этот момент подробнее. Пусть $(\bar{\varphi}, R, s)$ сообщение с цифровой подписью, удовлетворяющей уравнению (3.45). Найдем гомоморфное преобразование (3.45), сохраняющее неподвижным ключ n_A. Для этого выберем произвольное число μ такое, что $0 < \mu < \#E(GF(q))$ и взаимно просто с $\#E(GF(q))$, т.е. НОД(μ, $\#E(GF(q))$) = 1, и предположим, что $m'_{\mathbf{A}} = \mu \ m_{\mathbf{A}} \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right)$.

Также предположим, что $r' = \gamma\ r \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right)$. На основании этих предположений получим гомоморфный образ (3.45) $\bar{\varphi}' = n_{\mathbf{A}} r' + m'_{\mathrm{A}} s' \ \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right)$ или $\bar{\varphi}' = n_{\mathbf{A}} \gamma\ r + \mu\ m_{\mathbf{A}} s' \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right)$. Отсюда находим условия пропорциональности

$$\begin{cases} \bar{\varphi}' = \gamma\ \bar{\varphi} \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right) \\ s' = \mu^{-1}\gamma\ s \left(\mathrm{mod}\ \#E\left(GF(q)\right)\right). \\ r' = \beta\ \ r \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right) \end{cases} \qquad (3.47)$$

Если криптоаналитик располагает несколькими подписанными сообщениями и в них используются различные значения $m_{\mathbf{A}}$, то вычисление ключа $n_{\mathbf{A}}$ из уравнения (3.45) невозможно, так как число неизвестных, включая все числа $m_{\mathbf{A}}$ и число $n_{\mathbf{A}}$, превышает число уравнений. Однако, если хотя бы одно число $m_{\mathbf{A}}$ повторяется дважды, то ключ $n_{\mathbf{A}}$ вычисляется решением системы из двух линейных уравнений с двумя неизвестными $n_{\mathbf{A}}$ и $m_{\mathbf{A}} = m_{\mathbf{A}_1} = m_{\mathbf{A}_2}$

$$\begin{cases} \bar{\varphi}_1 = n_{\mathbf{A}} r + m_{\mathbf{A}_1} s \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right) \\ \bar{\varphi}_2 = n_{\mathbf{A}} r + m_{\mathbf{A}_2} s \left(\mathrm{mod}\ \ \#E\left(GF(q)\right)\right) \end{cases}, \qquad (3.48)$$

что является тривиальной задачей.

При таких условиях знание одной подписи позволяет образовывать множество подписей и использовать их для подмены сообщений. Подмена сообщения может быть обнаружена проверкой на пропорциональность формул (3.47), однако для этого требуется, чтобы абоненту **B** были известны все подписанные абонентом **A** сообщения независимо от того, кому они были предназначены. Данное условие на практике не выполнимо. Если криптоаналитику удается вычислить хотя бы одно значение $m_{\mathbf{A}}$, тогда ключ $n_{\mathbf{A}}$ раскрывается сразу же решением равенства (3.45).

В качестве альтернативного протокола подтверждения подлинности предлагается использовать протокол Шнорра [48], предложенный для мультипликативных полей. Данный протокол можно обобщить для аддитивной циклической группы точек кривой. Открытый и секретный ключи те же, что и в предыдущем протоколе. При генерации цифровой подписи сообщения $\bar{\varphi}$ абонент **A** на этапе 2 (табл.3.13) генерирует случайное число $m_{\mathbf{A}}$, такое, что $0 < m_{\mathbf{A}} < \#E(GF(q))$, вычисляет значение функции $e=\eta(\bar{\varphi}, R)$, $\bar{\varphi}$ и R – это последовательно записанные сообщение $\bar{\varphi}$ и элемент R цик-

лической подгруппы точек ЭК $E(GF(q))$. Затем абонент **A** находит число s из равенства

$$s = m_{\mathbf{A}} + n_{\mathbf{A}} e \left(\mathrm{mod}\ \#E\left(GF(q)\right)\right). \qquad (3.49)$$

Цифровой подписью будет пара чисел (e, s). При проверке подлинности информации абонент **B** вычисляет элемент R' группы $E(GF(q))$: $R' = sG - eP_{\mathbf{A}}$, где $P_{\mathbf{A}}=n_{\mathbf{A}}G$, и затем проверяет истинность равенства $\eta\left(\bar{\varphi}, R'\right) = e$.

Преимущество рассмотренного протокола перед предыдущим заключается в следующем. Во-первых, при реализации протокола требуется два умножения элементов группы на число, а не три, как ранее. Данная операция является наиболее сложной по сравнению с другими и определяет общую производительность протоколов. Следовательно, при использовании данного протокола достигается в 1.5 раза большая производительность. Во-вторых, цифровая подпись содержит пару чисел, а не элементы группы и поэтому является в 1.5-2.5 раза короче (в зависимости от реализации), что также повышает производительность преобразования информации при использовании данного протокола.

Таблица 3.13

Этап	Действия
1	$\mathbf{A} \rightarrow \mathbf{S} : P_{\mathbf{A}}=n_{\mathbf{A}}G;$
2	$\mathbf{A} \rightarrow \mathbf{B} :(\bar{\varphi}, e, s);$
3	$\mathbf{B} : R' = sG - eP_{\mathbf{A}}, \eta(\bar{\varphi}, R') \stackrel{?}{=} e.$

Безопасность рассмотренного протокола, как и в предыдущем случае, основана на сложности нахождения индекса и невозможности изоморфного преобразования сравнения (3.49). Кроме того, числа $m_{\mathbf{A}}$ могут быть повторяющимися, так как криптоаналитик не может судить об их равенстве в силу того, что хэш-функция η вычислима в одну сторону. Сложность подмены информации без раскрытия ключа соответствует сложности вычисления прообраза функции η по ее значению.

Рассмотренные протоколы можно комбинировать – как, например, в протоколе, представленном в табл. 3.14, первые два шага представляют собой ни что иное, как протокол распределения ключей. Это позволяет в комплексе решать проблемы распределения сеансовых ключей, аутентификации абонентов сети и передаваемой ими информации, шифрования информации. Пример совместного протокола распределения ключей и подтверждения подлинности информации представлен в табл.3.14.

Здесь на 1 и 2 этапах абоненты сети обмениваются своими открытыми ключами и вырабатывают общий ключ $P_{\mathbf{AB}}$. Затем на 3 этапе абонент **B** вводит сообщение и

свой идентификатор в точку на кривой $P_{\bar{\varphi}_{\mathbf{B}}, \mathrm{I}_{\mathbf{B}}}$ и находит точку (шифрует сообщение) $P_{\mathbf{B}} = P_{\mathbf{AB}} + P_{\bar{\varphi}_{\mathbf{B}}, \mathrm{I}_{\mathbf{B}}}$ для исключения возможной подмены криптоаналитиком ответного сообщения.

Идентификатор $\mathrm{I}_{\mathbf{B}}$ и точка $P_{\mathbf{B}}$ передаются по сети абоненту **A**. Тот, в свою очередь, осуществив расшифрование на этапе 4, подтверждает абоненту **B**, что он понимает сообщение $\bar{\varphi}_{\mathbf{B}}$ и является пользователем **A**. Данный этап также содержит вызов $\bar{\varphi}_{\mathbf{A}}$ от **A** к **B** и сеансовый ключ, выбранный абонентом **A**. Этап 5 доказывает **A**, что другой пользователь сети может дешифровать сообщение. Итог: абонент **B** уверен, что другим абонентом является **A**, сообщение не подменено, потому что оно содержит ответ $\eta(\bar{\varphi}_{\mathbf{A}})$ на запрос от **A** к **B**, и сгенерирован сеансовый ключ $K_{\mathbf{AB}}$. Все предыдущие рассуждения по стойкости применимы и к данному протоколу.

Таблица 3.14

Этап	Действия
1	$\mathbf{A} \to \mathbf{B} : P_{\mathbf{B}}$;
2	$\mathbf{B} \to \mathbf{A} : P_{\mathbf{A}}$;
3	$\mathbf{B} : \left(\bar{\varphi}_{\mathbf{B}},\ \mathrm{I}_{\mathbf{B}}\right) \mapsto P_{\bar{\varphi}_{\mathbf{B}}, \mathrm{I}_{\mathbf{B}}}$;
4	$\mathbf{B} \to \mathbf{A} : \mathrm{I}_{\mathbf{B}}, P_{\mathbf{BA}} + P_{\bar{\varphi}_{\mathbf{B}}, \mathrm{I}_{\mathbf{B}}}$;
5	$\mathbf{A} : \left(\bar{\varphi}_{\mathbf{A}},\ \mathrm{I}_{\mathbf{A}}\right) \mapsto P_{\bar{\varphi}_{\mathbf{A}}, \mathrm{I}_{\mathbf{A}}}$, $\eta\left(\bar{\varphi}_{\mathbf{B}}\right) \mapsto P_{\eta\left(\bar{\varphi}_{\mathbf{B}}\right)}$, $K_{\mathbf{AB}} \mapsto P_{K_{\mathbf{AB}}}$;
6	$\mathbf{A} \to \mathbf{B} : P_{\mathbf{AB}} + P_{\bar{\varphi}_{\mathbf{A}}, \mathrm{I}_{\mathbf{A}}} + P_{\eta\left(\bar{\varphi}_{\mathbf{B}}\right)} + P_{K_{\mathbf{AB}}}$;
7	$\mathbf{B} : P_{K_{\mathbf{AB}}} \mapsto K_{\mathbf{AB}}$;
8	$\mathbf{B} \to \mathbf{A} : \mathrm{I}_{\mathbf{B}}, K_{\mathbf{AB}}G + P_{\eta\left(\bar{\varphi}_{\mathbf{A}}\right), \mathrm{I}_{\mathbf{B}}}$.

Таким образом, применение криптографических конструкций на основе ЭК для построения различных криптографических протоколов за счет более сложной алгебраической структуры абелевых групп, порождаемых ими, выглядит предпочтительнее по сравнению с использованием криптографических конструкций на основе мультипликативной группы элементов конечного поля.

3.5. Инфраструктура открытых ключей

Одним из наиболее совершенных механизмов подтверждения подлинности электронных документов и участников информационного обмена в распределенных сетях является ЭЦП. Практическое применение двухключевых криптографических алгоритмов (в первую очередь систем ЭЦП) требует наличия инфраструктуры открытых ключей (ИОК) (PKI – Public Key Infrastructure) – неотъемлемого сервиса для управления электронными сертификатами и ключами пользователей, прикладного обеспечения и систем.

Непосредственное использование открытых ключей требует дополнительной их защиты и подтверждения подлинности (аутентификации), т.е. корректной связи с секретным ключом. Без такой дополнительной защиты злоумышленник может представить себя как отправителем подписанных данных, так и получателем зашифрованных данных, заменив значение открытого ключа или нарушив его подлинность. Все это приводит к необходимости аутентификации открытого ключа, что является основным назначением ИОК. Для этих целей используется система организационно-нормативных мероприятий с привлечением программно-технических криптографических средств, обеспечивающих электронную сертификацию открытых ключей. Одним из важных понятий в данной системе является электронный сертификат.

Электронный сертификат представляет собой цифровой документ, который связывает открытый ключ с определенным пользователем или приложением, а также подтверждает его подлинность. Для заверения электронного сертификата используется ЭЦП доверенного центра – центра сертификации (ЦС). Исходя из функций, которые выполняет ЦС, он является основной компонентой всей ИОК. Используя открытый ключ ЦС, каждый пользователь может проверить достоверность электронного сертификата, выпущенного ЦС, и воспользоваться его содержимым.

3.5.1 Компоненты ИОК и их функции

В состав ИОК входят следующие компоненты:

- центр сертификации;
- центр регистрации (ЦР);
- конечные пользователи;
- сетевой справочник.

Центр сертификации или удостоверяющий центр (УЦ) – основная управляющая компонента ИОК, предназначенная для формирования электронных сертификатов подчиненных центров и конечных пользователей. Кроме сертификатов, ЦС формирует список отозванных сертификатов X.509 CRL (СОС) с регулярностью, определенной регламентом системы.

К основным функциям ЦС относятся:

- формирование собственного секретного ключа и сертификата ЦС;
- формирование сертификатов подчиненных центров;
- формирование сертификатов открытых ключей конечных пользователей;
- формирование списка отозванных сертификатов;
- ведение базы всех изготовленных сертификатов и списков отозванных сертификатов.

Центр регистрации – опциональная компонента ИОК, предназначенная для регистрации конечных пользователей. Основная задача ЦР – регистрация пользователей и обеспечение их взаимодействия с ЦС. В задачи ЦР может также входить публикация сертификатов и СОС в сетевом справочнике.

Пользователи – это абоненты, приложения и системы, являющиеся владельцами сертификата и использующие ИОК.

Сетевой справочник – компонента ИОК, содержащая сертификаты и списки отозванных сертификатов и служащая для целей распространения этих объектов среди пользователей с использованием протокола LDAP (HTTP, FTP).

3.5.2 Верификация цепочки сертификатов

Доверие любому сертификату пользователя определяется на основе цепочки сертификатов. Причем начальным элементом цепочки является сертификат центра сертификации, хранящийся в защищенном персональном справочнике пользователя.

Процедура верификации цепочки сертификатов описана в рекомендациях X.509 и RFC 2459 и проверяет связанность между именем владельца сертификата и его открытым ключом. Процедура верификации цепочки подразумевает, что все «правильные» цепочки начинаются с сертификатов, изданных одним доверенным ЦС. Под доверенным центром понимается главный ЦС, открытый ключ которого содержится в самоподписанном (корневом) сертификате. Такое ограничение упрощает процедуру верификации, хотя и не обеспечивает полной безопасности. Для обеспечения доверия к открытому ключу такого сертификата должны быть применены специальные способы его распространения и хранения, так как на данном открытом ключе проверяются все остальные сертификаты.

Алгоритм верификации цепочек использует следующие данные:

- имя издателя сертификата;
- имя владельца сертификата;
- открытый ключ издателя сертификата;
- срок действия открытого (секретного) ключа издателя и владельца сертификата;
- ограничивающие дополнения, используемые при верификации цепочек;
- список отозванных сертификатов (СОС) для каждого издателя сертификатов (даже если он не содержит отзываемых сертификатов).

Цепочка сертификатов представляет собой последовательность из n сертификатов, в которой:

- для всех $s \in \{1, (n-1)\}$, владелец сертификата s является издателем сертификата s+1;
- сертификат $s = 1$ есть самоподписанный сертификат;
- сертификат $s = n$ является сертификатом конечного пользователя;

Одновременно с цепочкой сертификатов используется цепочка СОС, представляющая собой последовательность из n СОС, в которой:

- для всех СОС $s \in \{1, n\}$, издатель сертификата s является издателем СОС s;
- СОС $s = 1$ есть СОС, изданный владельцем самоподписанного сертификата;
- СОС $s = n$ является СОС, изданным издателем сертификата конечного пользователя;

После построения двух цепочек (сертификатов и СОС) выполняется:

- криптографическая проверка сертификатов и СОС в цепочках;
- проверка сроков действия сертификатов и СОС;
- проверка соответствия имен издателя и владельца;
- проверка длины цепочки;
- проверка на отзыв сертификатов, причем, если сертификат промежуточного центра был отозван СОС вышестоящего центра, все сертификаты, изданные промежуточным центром, считаются недействительными;
- проверка приемлемых регламентов использования сертификата и приемлемых областей использования ключа.

3.5.3 Использование ИОК в приложениях

Инфраструктура открытых ключей применяется для управления ключами и электронными сертификатами в приложениях (таких как электронная почта, Web приложения, электронная коммерция), использующих криптографию для установления

защищенных сетевых соединений (S/MIME, SSL, IPSEC), или для формирования ЭЦП электронных документов, приложений и т.д. Кроме того, ИОК может быть использована для корпоративных приложений.

Электронная почта и документооборот

Защищенные электронная почта и документооборот используют криптографию для шифрования сообщений или файлов и формирования ЭЦП. Из наиболее известных и распространенных стандартов стоит отметить протокол S/MIME (Secure Multipurpose Internet Mail Extensions), который является расширением стандарта Internet почты MIME (Multipurpose Internet Mail Extensions).

Web – приложения

Web-броузеры и сервера используют ИОК для аутентификации и конфиденциальности сессии, а также для онлайновых банковских приложений и электронных магазинов. Наиболее распространенным протоколом в этой сфере является протокол SSL (Secure Sockets Layer). Протокол SSL не ограничивается применением только для защиты HTTP (Hypertext Transfer Protocol), а также может быть использован для FTP (File Transfer Protocol) и Telnet.

ЭЦП файлов и приложений

Использование ЭЦП для подписи приложений и файлов позволяет безопасно распространять их по сети Internet. При этом пользователь уверен в корректности полученного приложения от фирмы-разработчика.

3.5.4 Стандарты в области ИОК и основанные на ИОК

Стандарты в области ИОК делятся на две группы: первая описывает собственно реализацию ИОК, а вторая, которая относится к пользовательскому уровню, использует ИОК, не определяя ее. Стандартизация в области ИОК позволяет различным приложениям взаимодействовать между собой с использованием единой ИОК.

Стандартизация особенно важна в области:

- процедуры регистрации и выработки ключа;
- описания формата сертификата;
- описания формата СОС;
- описания формата криптографически защищенных данных;
- описания онлайновых протоколов.

Основным центром по выпуску согласованных стандартов в области ИОК является рабочая группа ИОК (PKI working group) организации IETF (Internet Engineering Task Force), известная как группа PKIX (от сокращения PKI for X.509 certificates).

Спецификации PKIX основаны на двух группах стандартов: X.509 ITU-T (Международный комитет по телекоммуникациям) и PKCS (Public Key Cryptography Standards) компании RSA Data Security.

Стандарт **X.509 ITU-T** является фундаментальным стандартом, лежащим в основе всех остальных, используемых в ИОК. Основное его назначение – определение формата электронного сертификата и списков отозванных сертификатов.

Из серии стандартов, изданных фирмой RSA Data Security, наиболее важными и используемыми в ИОК являются:

PKCS #7 Cryptographic Message Syntax Standard;
PKCS #10 Certificate Request Syntax Standard;
PKCS #12 Personal Information Exchange Syntax Standard.

Вместо устаревшего стандарта RSA PKCS #7, описывающего форматы криптографических сообщений, в 1999 году был принят стандарт RFC 2630 Cryptographic Message Syntax.

Стандарты, основанные на ИОК

Следует отметить, что большинство стандартов, использующих криптографию, разработано с учетом использования ИОК.

Стандарт **S/MIME** определен IETF для обеспечения защищенного обмена сообщениями. S/MIME использует ИОК для формирования ЭЦП и шифрования информации. В группе стандартов S/MIME наиболее важными являются следующие: Cryptographic Message Syntax, Message Specification, Certificate Handling и Certificate Request Syntax.

Протокол **SSL** (разработанный фирмой Netscape) и соответствующий ему стандарт IETF TLS (RFC 2246) являются наиболее часто используемыми стандартами для обеспечения защищенного доступа к Web. Вместе с этим, SSL и TLS широко используются для создания клиент-серверных приложений, не использующих Web. Оба эти протокола в своей основе используют ИОК.

Протокол SET **(Secure Electronic Transactions)** был разработан фирмами Visa и MasterCard и предназначен для обеспечения системы электронных банковских расчетов с использованием пластиковых карт. В данном протоколе ИОК является фундаментом, на котором базируется вся система аутентификации участников расчетов.

Протокол **IPSec** (Internet Protocol Security) разработан IETF как протокол для шифрования IP и является одним из основных протоколов, используемых для построения VPN. ИОК в протоколе IPSec используется для аутентификации и шифрования. В настоящее время протокол еще широко не распространен, но повсеместное развитие ИОК приводит к возрастания количества реализаций IPSec.

Таким образом, двухключевая криптография и электронные сертификаты позволяют реализовать по-настоящему защищенные системы и приложения, использующие современные технологии и сети передачи данных. Стандартизация в данной области позволяет различным приложением взаимодействовать между собой, используя единую ИОК.

ГЛАВА 4

Подстановочно-перестановочные сети с минимальным управляемым элементом

4.1. Управляемые битовые перестановки как криптографический примитив

Основное криптографическое применение операции управляемых перестановок (УП) связано с выполнением битовых перестановок в зависимости от преобразуемых данных. Впервые данный вариант применения этой операции предложен в работах [42, 92, 97]. Более ранние поиски по построению криптографических механизмов на основе УП [105] связаны с их применением в качестве операций, зависящих от ключа шифрования. Такой тип операции требует применения управляемых перестановочных сетей (УПС), которые хорошо известны в литературе по работам различных авторов [53, 54, 102, 124]. Хотя была показана возможность построения стойких криптосистем на основе применения УП, зависящих от ключа, предложенные шифры не могли конкурировать по быстродействию и простоте схемотехнической реализации с другими симметричными криптосистемами. Основной причиной этого является то, что битовая перестановка, зависящая от ключа, остается строго линейной операцией, поскольку она является фиксированной после ввода ключа. Принципиально ситуация изменяется, когда перестановка является переменной операцией, т. е. в случае ее выполнения в зависимости от значения преобразуемого блока данных, которое по своей природе является переменной величиной.

Для выполнения переменных перестановок хорошо подходят УПС со слоистой структурой, представленной на рис. 4.1, где основным строительным блоком является перестановочный элемент $\mathbf{P}_{2/1}$, который может быть назван элементарным блоком управляемых перестановок (БУП), поскольку он реализует две различные перестановки двух входных битов x_1 и x_2 в зависимости от одного управляющего бита v. Элементарный блок $\mathbf{P}_{2/1}$ управляется одним битом v и формирует двухбитовый выход (y_1,y_2), где $y_1 = x_{1+v}$ и $y_2 = x_{2-v}$. Поскольку имеются только две перестановки такого типа, то элементарный БУП реализует все возможные перестановки. С увеличением размера входа БУП реализация всех перестановок становится проблематичной, однако для всех практически значимых значений размера входа n достаточно легко могут быть построены БУП, реализующие все возможные $n!$ перестановок. При этом подобные операционные блоки являются достаточно быстродействующими для обеспечения высокой скорости шифрования на основе переменных перестановок. В слоистых БУП число активных слоев s связано с параметрами m и n следующим образом: $s = 2m/n$.

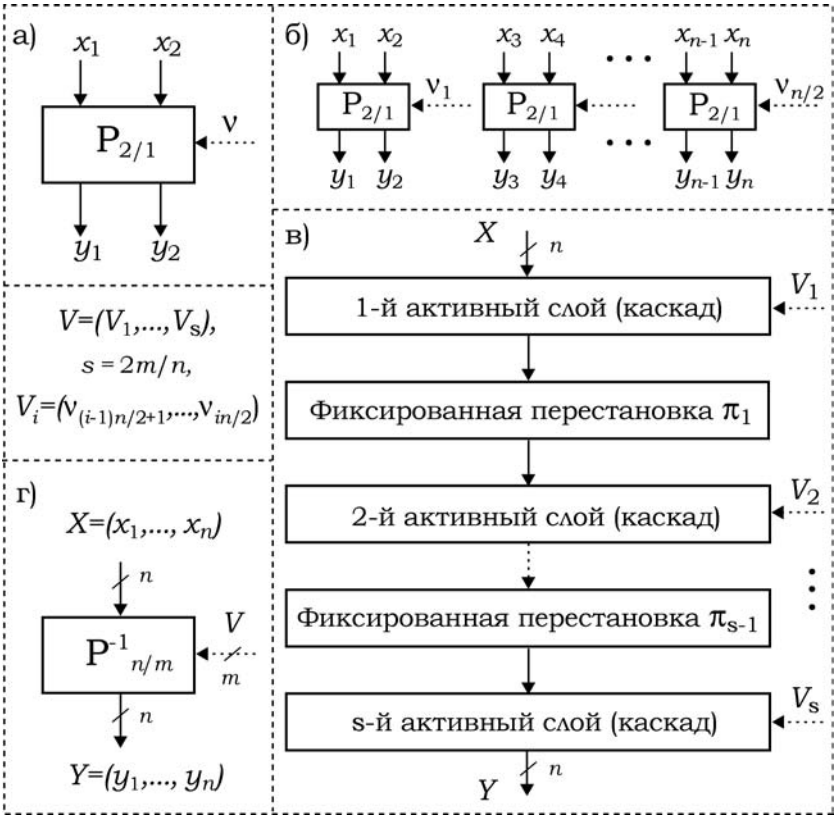

Рис. 4.1. Структура БУП:
а – элементарный блок $\mathbf{P}_{2/1}$, б – структура одного активного слоя,
в – общая структура многослойных БУП, г – обратный БУП $\mathbf{P}^{-1}{}_{n/m}$

Некоторый БУП реализует множество различных перестановок, соответствующих различным значениям управляющего вектора V. В соответствии с числом слоев вектор V может быть представлен как объединение s векторов $V_1,V_2,...,V_s \in \mathrm{GF}(2)^{n/2}$, то есть $V = (V_1, V_2, \ldots, V_s)$. При фиксировании значения управляющего вектора реализуется некоторая перестановка $\mathbf{\Pi}_V$. Блок управляемых перестановок $\mathbf{P}_{m/n}$ может быть описан с помощью упорядоченного множества модификаций $\{\Pi_0, \Pi_1, \ldots, \Pi_{2^m-1}\}$, где каждая модификация $\mathbf{\Pi}_V$, $V = 0, 1, \ldots, 2^m-1$, является фиксированной перестановкой n битов. Перестановки Π_V будем называть модификациями УП. Выполнение операции УП $\mathbf{P}_{m/n(V)}(X)$ заключается в осуществлении перестановки Π_V над $X : Y = \mathbf{P}_{m/n(V)}(X) = \Pi_V(X)$. Наиболее интересными для криптографических приложений являются значения n, являющиеся натуральной степенью числа два. При этом представляет интерес разработка БУП различного порядка, поскольку с уменьшением порядка уменьшается число активных слоев в БУП, что ведет к повышению их быстродействия.

Определение 4.1

Пусть для заданного $d \le n$, произвольных наборов индексов $\alpha_1, \alpha_2, \ldots, \alpha_h$ и $\beta_1, \beta_2, \ldots, \beta_h$ существует по крайней мере одно значение управляющего вектора V такое, что входные биты $x_{\alpha_1}, x_{\alpha_2}, \ldots, x_{\alpha_h}$ переходят, соответственно, в выходные биты $y_{\beta_1}, y_{\beta_2}, \ldots, y_{\beta_h}$. Максимальное из возможных значений d называется ***порядком*** БУП $\mathbf{P}_{m/n}$ и обозначается h.

Возможность конструирования достаточно однородных БУП различных порядков, представляющих наибольший интерес ($h = 1, 2, 4, \ldots, n/4$), составляет важный элемент арсенала разработчика шифров и позволяет найти необходимый компромисс между стоимостью реализации и производительностью проектируемого шифра. Другим важным моментом является простота конструирования обратных БУП.

Определение 4.2

Блоки управляемых перестановок $\mathbf{P}_{m/n}$ и $\mathbf{P}^{-1}{}_{m/n}$ называются ***взаимно-обратными***, если для всех возможных значений вектора V модификации битовых перестановок Π_V и $\Pi^{-1}{}_V$, реализуемых, соответственно, блоками $\mathbf{P}_{m/n}$ и $\mathbf{P}^{-1}{}_{m/n}$, являются взаимно-обратными.

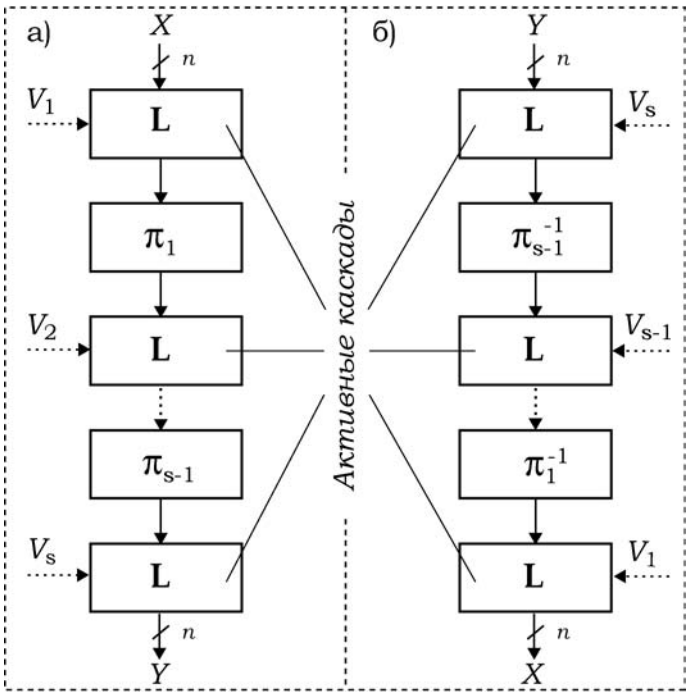

Рис. 4.2. Структура прямого (а) и обратного (б) БУП

На рис. 4.2 представлена схема построения прямого и обратного слоистых БУП в общем случае. Характерным моментом такого построения является то, что компоненты управляющего вектора $V = (V_1, V_2, \ldots, V_s)$ распределены по активным слоям в различном порядке. В прямом блоке они распределяются, начиная со входа к выходу (сверху вниз), а в обратном – с выхода ко входу (снизу вверх), причем нумерация активных каскадов в обоих блоках осуществляется одинаково – со входа к выходу. Таким образом, согласно принятому соглашению компонента V_l управляет l-ым активным слоем в прямом и $(s - l + 1)$-ым активным слоем в обратном БУП. Слоистый БУП может быть рассмотрен как матрица элементарных переключающих элементов, которые последовательно пронумерованы слева направо и сверху вниз в прямом БУП и слева направо и снизу вверх в обратном БУП, а i-ый бит вектора V управляет i-ым переключающим элементом $\mathbf{P}_{2/1}$.

Один активный каскад может рассматриваться как однослойный БУП $\mathbf{L}_n$. Очевидно, что $\mathbf{P}_{2/1} = \mathbf{P}^{-1}{}_{2/1}$, поэтому преобразование, осуществляемое с помощью активного каскада, является инволюцией, и мы имеем $\mathbf{L}_n = \mathbf{L}^{-1}{}_n$.Многослойный БУП $\mathbf{P}_{m/n}$ может быть представлен как суперпозиция

$$\mathbf{P}_{m/n} = \mathrm{L}_{V_1} \circ \pi_1 \circ \mathrm{L}_{V_2} \circ \pi_2 \circ \ldots \circ \pi_{s-1} \circ \mathrm{L}_{V_s} .$$

Соответствующий блок $\mathbf{P}^{-1}{}_{m/n}$ имеет следующую структуру

$$\mathbf{P}^{-1}{}_{m/n} = \mathrm{L}_{V_s} \circ \pi^{-1}{}_{s-1} \circ \mathrm{L}_{V_{s-1}} \circ \pi^{-1}{}_{s-2} \circ \ldots \circ \pi_1 \circ \mathrm{L}_{V_1} .$$

Таким образом, чтобы построить БУП, являющийся обратным по отношению к блоку $\mathbf{P}_{m/n}$, достаточно пронумеровать блоки $\mathbf{P}_{2/1}$ слева направо снизу вверх и заменить π_i на $\pi^{-1}{}_{s-i}$. Блоки УП использованы в ряде шифров, рассмотренных в восьмой главе и работах [14,72], для выполнения битовых перестановок, зависящих от преобразуемых данных. Несмотря на то, что переменные перестановки позволили существенно повысить скорость блочных шифров при уменьшении стоимости реализации, они как криптографический примитив не лишены недостатка. Пожалуй, наиболее существенным из них является то, что они являются линейным криптографическим примитивом, хотя единственная линейная комбинация выходов, являющаяся линейной булевой функцией, включает в себя все выходные биты $y_1, y_2, \ldots, y_n$. Такой линейной комбинацией является сумма $\Sigma = y_1 \oplus y_2 \oplus \ldots \oplus y_n$, которая равна сумме входных битов в силу природы перестановок.

Благодаря тому, что линейные комбинации с меньшим числом выходов являются нелинейными булевыми функциями, переменные битовые перестановки обеспечивают возможность эффективного построения блочных шифров при использовании дополнительных нелинейных криптографических примитивов, играющих вспомогательную роль, а именно, предотвращающих линейный криптоанализ с использованием масок с максимально возможным весом. Применение дополнительных примитивов маскирует вклад переменных перестановок в обеспечение стойкости разрабатываемых по указанной схеме шифров, поскольку высокая стойкость последних обеспечивается двумя основными примитивами. Демонстрация эффективности битовых перестановок, зависящих от преобразуемых данных, связана с их применением в со-

четании только с операцией XOR, фиксированными перестановками и блоками расширения, реализуемыми как простое разветвление проводников. Такая постановка вопроса фактически означает необходимость предварительного построения перестановочной операции, являющейся нелинейной как криптографический примитив.

Для решения такой задачи в шифре DDP-64, описываемом в следующем параграфе, использована идея урезания выходных битов БУП, используемых для реализации переменных перестановок. Это вполне может быть применено с сохранением возможности корректного выполнения процедуры расшифрования, если упомянутую перестановку использовать как элемент, включенный в функцию **F** сети Фейстеля, или непосредственно использовать ее в качестве функции **F**. Именно таким образом используется дополнительный примитив в известных шифрах на основе перестановок, зависящих от преобразуемых данных (например, нелинейная операция **G** в шифрах SPECTR-H64, SPECTR-128, CIKS-128 [14, 72]). Таким образом, в криптосистеме DDP-64 используются БУП, реализующие переменные перестановки двух типов: 1) обычные и 2) с урезанным выходом. Последние являются нелинейным примитивом в криптографическом определении.

4.2. Блочный шифр на основе переменных перестановок

При разработке блочной криптосистемы DDP-64 (название отражает аббревиатуру английского термина Data-Dependent Permutations – перестановки, зависящие от преобразуемых данных) основным замыслом было построение стойкого шифра, в котором единственным нелинейным примитивом являются переменные битовые перестановки. В качестве вспомогательных примитивов используются фиксированные перестановки, операции расширения и операция поразрядного суммирования по модулю два. В качестве критериев построения были использованы следующие:

1. Криптосистема должна быть блочным итеративным 64-битовым шифром, обеспечивающим высокую скорость преобразования данных при относительно недорогой аппаратной реализации.
2. Для выполнения процедур зашифрования и расшифрования должен использоваться один и тот же алгоритм. Смена режима преобразования должна обеспечиваться быстрой сменой расписания использования подключей.
3. Шифр должен обладать высокой производительность в приложениях, требующих частой смены ключа. Для этой цели расписание использования ключа должно быть простым и не требовать выполнения каких-либо предвычислений по построению расширенного ключа (совокупности раундовых подключей).
4. Процедура раундового шифрования должна характеризоваться относительно высоким параллелизмом вычислений, что направлено на обеспечение высокой скорости шифрования.
5. В качестве основного криптографического примитива должны использоваться только переменные перестановки. Кроме операций битовых перестановок и узлов расширения, реализуемых как простое разветвление проводников,

должна использоваться единственная дополнительная операция – поразрядное суммирование по модулю два (XOR).

В качестве прототипа раундового преобразования шифра DDP-64 использована процедура раундового преобразования криптосистемы SPECTR-H64, которая хорошо подходит для реализации принятых конструктивных критериев. Нелинейные переменные перестановки осуществляются операционным блоком **F**, синтез которого основан на использовании «урезанных» переменных битовых перестановок, выполняемых над левым подблоком данных. Другим типом использованных переменных перестановок являются перестановки, выполняемые с помощью БУП второго порядка $\mathbf{P}_{32/96(V)}$ и $\mathbf{P}^{-1}{}_{32/96(V')}$, представленных на рис. 4.3. Блоки $\mathbf{P}_{32/96(V)}$ и $\mathbf{P}^{-1}{}_{32/96(V')}$ построены на основе БУП $\mathbf{P}_{8/12}$ и $\mathbf{P}^{-1}{}_{8/12}$, показанных на рис. 4.3а и 4.3б. Каскад блоков $\mathbf{P}_{8/12}$ соединен с каскадом блоков $\mathbf{P}^{-1}{}_{8/12}$ с помощью коммутации, задающей следующую битовую перестановку, которая является инволюцией:

(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)(12,26)

(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32).

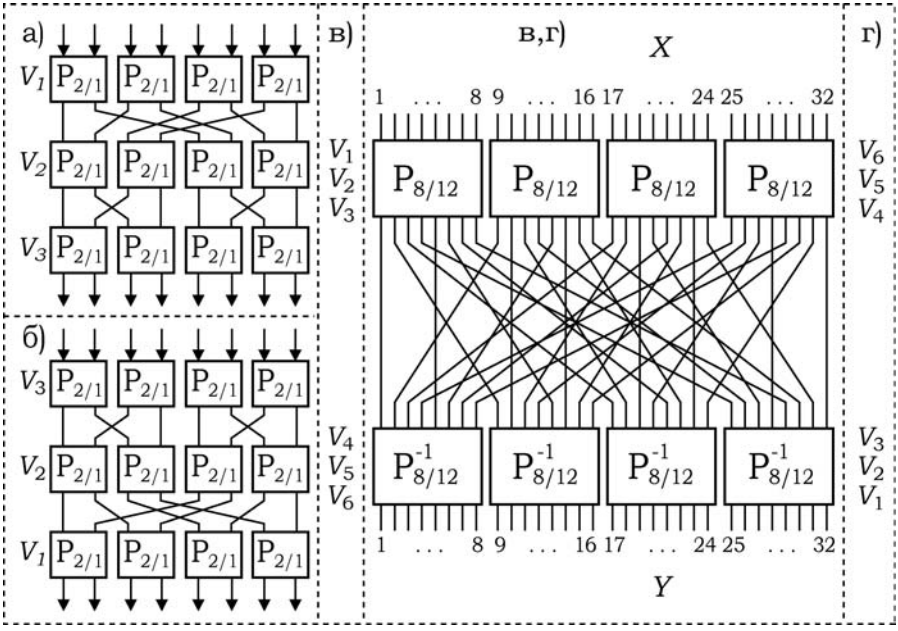

Рис. 4.3. Структура БУП $P_{8/12}$ (а), $P^{-1}{}_{8/12}$ (б), $P_{32/96}$ (в) и $P^{-1}{}_{8/12}$ (г)

Благодаря симметричной структуре блоков $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$ они отличаются между собой только распределением управляющих битов вектора *V*. Ввиду того, что в этих блоках используется 96-битовый управляющий вектор, а управляющий левый подблок данных имеет длину 32 бит, необходимо использовать блок расширения **E**, для синтеза которого мы использовали следующие критерии:

1. Для всех значений управляющего вектора перестановка каждого входного бита БУП должна быть определена шестью различными битами L.
2. От каждого бита управляющего подблока данных должны зависеть строго три бита управляющего вектора.

Пусть 96-битовый вектор $V = (V_1, V_2, V_3, V_4, V_5, V_6)$ является выходом блока **E**, а 32-битовый вектор $X = (X_l, X_h)$, где $X_l, X_h \in \mathrm{GF}(2)^{16}$, – его входом. В шифре DDP-64 применен блок расширения **E**, удовлетворяющий приведенным выше критериям и описываемый следующими соотношениями:

$$V_1 = X_l, \quad V_2 = (X_l)^{\lll 6}, \quad V_3 = (X_l)^{\lll 12},$$

$$V_4 = X_h, \quad V_5 = (X_h)^{\lll 6}, \quad V_6 = (X_h)^{\lll 12}.$$

Очевидно, что благодаря выполнению указанных критериев, каждый бит левого подблока данных L влияет точно на шесть битов правого подблока данных R (независимо от значения вектора, поступающего на вход БУП) в каждом из блоков $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$. Также достаточно очевидно, что произвольный заданный входной бит блоков в каждом из блоков $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$ перемещается в каждую выходную позицию с одинаковой вероятностью, если L является равномерно распределенной случайной величиной.

Операционные блоки **F** представляют собой специфический вариант задания переменных перестановок. Конструкция каждого из двух использованных **F**-блоков обеспечивает случайность изменения четности выходного значения. Следует заметить, что БУП $\mathrm{P}_{32/96(V)}$ и $\mathrm{P}^{-1}{}_{32/96(V)}$ не обладают таким свойством. Для формирования 80-битового управляющего вектора блоков **F** применен блок расширения **E′**, задаваемый следующим образом. Пусть 80-битовый вектор $W = (W_1, W_2, W_3, W_4, W_5)$ является выходом блока **E**, а 32-битовый вектор $X = (X_l, X_h)$, где $X_l, X_h \in \mathrm{GF}(2)^{16}$, – его входом. Тогда 16-битовые компоненты W_1, W_2, W_3, W_4, W_5 определяются следующими соотношениями:

$$V_1 = X_l, \quad V_2 = (X_l)^{\lll 5}, \quad V_3 = (X_l)^{\lll 10}, \quad V_4 = X_h, \quad V_5 = (X_h)^{\lll 5}.$$

Общая схема шифрующего преобразования в DDP-64 показана на рис. 4.4а, а структура его раундового преобразования $\mathrm{Crypt}^{(e)}$ – на рис. 4.4б. Бит e является битом, задающим режим преобразования: $e = 0$ соответствует зашифрованию, а $e = 1$ – расшифрованию.

Использование верхнего индекса у процедуры $\mathrm{Crypt}^{(e)}$ отражает то, что в данной процедуре использована переключаемая перестановка $\Pi^{(e')}$: при $e = 0$ выполняется прямая перестановка битов над текущим значением левого подблока, а при $e = 1$ соответствующая ей обратная перестановка. Два различных механизма определяют смену режима преобразования – изменение расписания использования ключа и переключение перестановки $\Pi^{(e')}$.

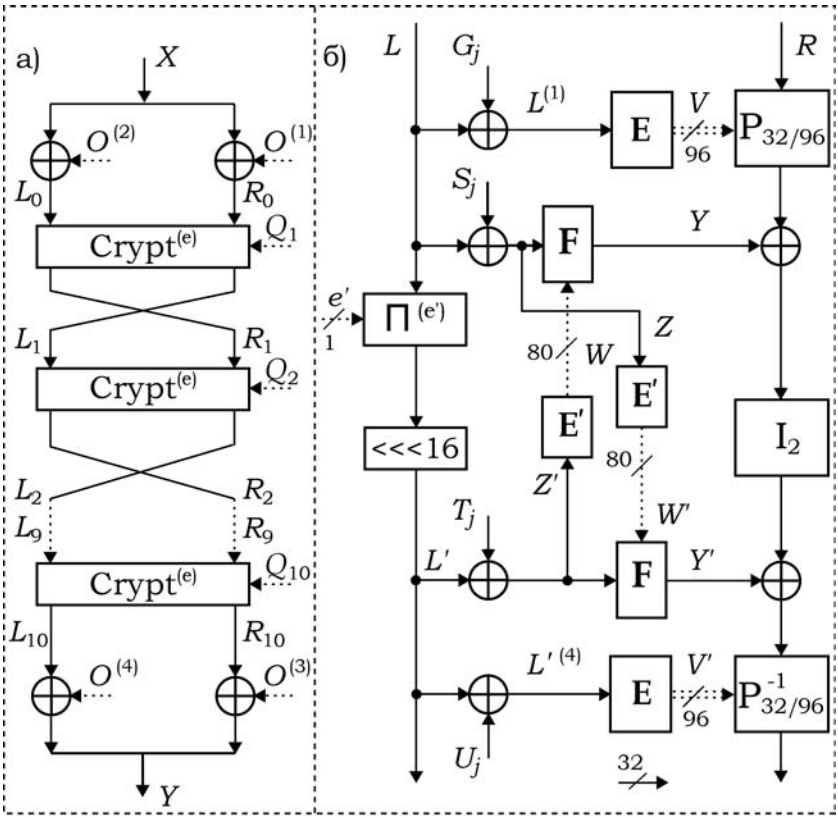

Рис. 4.4. Общая структура шифра DDP-64 (а) и процедура его раундового преобразования Crypt$^{(e)}$ (б)

Блочная криптосистема DDP-64 включает простое начальное преобразование 64-битового входного блока данных, десять раундов преобразования с помощью процедуры Crypt$^{(e)}$ и конечное простое преобразование. Преобразование блока данных можно представить в виде:

$$C = \mathrm{CRYPT}^{(e=0)}(M, K) \quad \text{и} \quad M = \mathrm{CRYPT}^{(e=1)}(C, K),$$

где M – открытый текст, C – шифртекст (M, $C \in \mathrm{GF}(2)^{64}$), K – секретный ключ ($K \in \mathrm{GF}(2)^{128}$). Шифр DDP-64 использует 128-битовый секретный ключ, рассматриваемый как совокупность четырех 32-битовых подключей K_i, $i = 1, 2, 3, 4$: $K = (K_1, K_2, K_3, K_4)$. Каждый раундовый ключ Q_j состоит из четырех независимых раундовых подключей G_j, S_j, T_j, $U_j \in \mathrm{GF}(2)^{32}$, то есть имеем $Q_j = (G_j, S_j, T_j, U_j)$. Таблица 4.1 описывает расписание ключа с использованием подключей O_1, O_2, O_3 и O_4, которые являются выходами блока перестановки подключей, показанного на рис. 4.5а и представляющего собой два однослойных БУП $\mathbf{P}^{(e)}_{2\times32/1}$. На первый блок $\mathbf{P}^{(e)}_{2\times32/1}$ по-

дается пара подключей K_1 и K_3, а на второй – пара подключей K_2 и K_4. Выходные подключи O_i зависят от значения e. При $e = 0$ имеем $O_i = K_i$ для $i = 1, 2, 3, 4$. При $e = 0$ имеем $O_1 = K_3$, $O_3 = K_1$ $O_2 = K_4$ и $O_4 = K_2$.

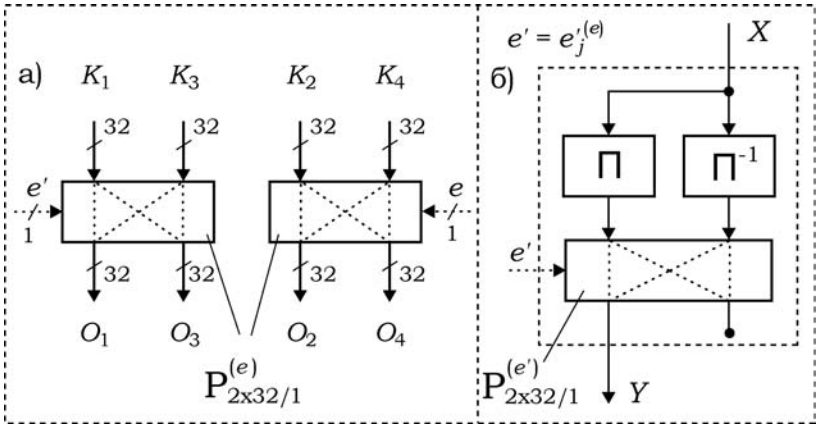

Рис. 4.5. Транспозиция подключей (а) и структура переключаемой перестановки (б)

Процедура шифрования выполняется следующим образом. Входной блок данных X делится на два 32-х битовых подблока L и R. Затем шифрование данных производится в соответствии со следующим алгоритмом:

1. Выполнить начальное преобразование.
2. Последовательно увеличивая на единицу значение j от $j = 1$ до $j = 9$, выполнить следующие преобразования:

   2.1. Преобразовать (L_j, R_j): $(L_j, R_j) :=$ **Crypt**(L_{j-1}, R_{j-1}, Q_j).

   2.2. Переставить подблоки данных: $(L_j, R_j) := (R_j, L_j)$.
3. Преобразовать (L_9, R_9): $(L_{10}, R_{10}) :=$ **Crypt**(L_9, R_9, Q_{10}).
4. Осуществить конечное преобразование.

Переключаемая перестановка $П^{(e')}$ реализована с использованием БУП $\mathbf{P}^{(e')}_{2\times32/1}$ и двух фиксированных взаимно-обратных перестановок П и $П^{-1}$, как показано на рис. 4.5б. Из приведенной схемы видно, что имеют место следующие соотношения: $П^{(0)} = П$ и $П^{(1)} = П^{-1}$. Используемая перестановка П описывается следующим образом:

(1,4,7,2,5,8,3,6)(9,12,15,10,13,16,11,14)

(17,20,23,18,21,24,19,22)(25,28,31,26,29,32,27,30).

Таблица 4.1

Расписание использования ключей и спецификация значения бита e' в режиме зашифрования ($e = 0$) и расшифрования ($e = 1$)

j =	1	2	3	4	5	6	7	8	9	10
G_j =	O_3	O_2	O_1	O_4	O_3	O_3	O_4	O_1	O_2	O_3
S_j =	O_2	O_1	O_4	O_3	O_4	O_4	O_3	O_4	O_1	O_2
T_j =	O_4	O_3	O_2	O_1	O_2	O_2	O_1	O_2	O_3	O_4
U_j =	O_1	O_4	O_3	O_2	O_1	O_1	O_2	O_3	O_4	O_1
$e'^{(e=0)}$	1	0	1	1	0	1	1	1	0	1
$e'^{(e=1)}$	0	1	0	0	0	1	0	0	1	0

Как видно из описания алгоритма, подключи K_i ($i = 1, 2, 3, 4$) непосредственно используются в каждом раунде без осуществления над ними каких-либо предвычислений. После сложения с левым подблоком данных подключи G и U используются для формирования управляющих векторов V и V', которые определяют реализацию текущих модификаций перестановок, реализуемых с помощью блоков $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$, соответственно. Подключи S и T также складываются с левым подблоком данных, но после этого результат преобразуется с помощью блока **F**. Отметим, что раундовое преобразование не является инволюцией по двум причинам. Первой из них является выполнение операции $\Pi^{(e')}$ над левым подблоком данных, а второй – использование взаимно-обратных БУП, управляемых различными по значению двоичными векторами, в правой ветви раундового преобразования. Отличие значений управляющих векторов V и V' обусловлено как преобразованием левого подблока, так и использованием различных подключей при формировании входного значения блоков расширения, соответствующих операциям $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$ (см. рис. 4.4б). Для того, чтобы обратить некоторый j-ый раунд зашифрования, соответствующие подключи должны быть переставлены, а бит e' инвертирован, т. е. на $(11-j)$-ом раунде расшифрования должно быть выполнено следующее преобразование:

$$X' = \mathbf{Crypt}^{(1)}(Y, Q_{11-j}),$$

где $Q_{11-j} = (U_j, T_j, S_j, G_j)$, а значение было получено с помощью преобразования $Y = \mathbf{Crypt}^{(0)}(X, Q_j)$, в котором $Q_j = (G_j, S_j, T_j, U_j)$. Таблица 4.1 задает значения бита e' при зашифровании ($e = 0$) и расшифровании ($e = 1$).

Характерным для раундового преобразования в шифре DDP-64 является использование фиксированных перестановок, являющихся инволюциями. Операция циклического сдвига «<<< 16», производимая над левым подблоком данных, применяется для задания “симметричного” использования старшей (L_h) и младшей (L_l) частей подблока данных L при выполнении двух переменных битовых перестановок, осуществляемых с помощью операционных блоков **F**. Перестановочная инволюция $\mathbf{I}_2$, выпол-

няемая над правым подблоком данных, используется для задания влияния каждого входного бита блока $\mathbf{P}_{32/96}$ на тридцать один выходной бит блока $\mathbf{P}^{-1}{}_{32/96}$ в случае $V = V'$. В указанном случае каждый i-ый входной бит не влияет только на i-ый выходной бит. Заметим, что если мы не используем перестановку $\mathbf{I}_2$, то в упомянутом случае каждый входной бит $\mathbf{P}_{32/96}$ влияет только на один выходной бит $\mathbf{P}^{-1}{}_{32/96}$. Перестановка $\mathbf{I}_2$ имеет очень простую структуру и может быть описана двумя циклическими сдвигами по 8 бит:

$$Y = \mathbf{I}_2(X_1,X_2) = (X_1{}^{<<<8}, X_2{}^{<<<8}).$$

Эта перестановка улучшает результирующую УП, соответствующую последовательному выполнению операций $\mathbf{P}_{32/96(V)}$ и $\mathbf{P}^{-1}{}_{32/96(V)}$. Действительно, даже в случае $V = V'$ суперпозиция $\mathbf{P}_{32/96(V)} \circ \mathbf{P}^{-1}{}_{32/96(V)}$ формирует эффективную УП, все модификации которой есть перестановочные инволюции. В общем случае мы имеем $V \neq V'$, так как данные объединяются с различными подключами при формировании управляющих векторов, соответствующих операциям $\mathbf{P}_{32/96(V)}$ и $\mathbf{P}^{-1}{}_{32/96(V)}$, поэтому влияние каждого входного бита блока $\mathbf{P}_{32/96}$ распространяется на все выходные биты блока $\mathbf{P}^{-1}{}_{32/96}$. Для изучения роли фиксированной перестановки между двумя взаимнообратными операциями УП ранее были проведены многочисленные статистические эксперименты [14, 99], которые показали, что использование такой перестановки существенно улучшает свойства преобразования, выполняемого двумя взаимнообратными БУП.

Структура блока **F** показана на рис. 4.6. Для обеспечения нелинейности всех линейных комбинаций выходных битов этого блока применены внутреннее расширяющее и сжимающее отображения. Блок **F** включает в себя два трехслойных БУП $\mathbf{P}_{32/48}$ и $\mathbf{P}^{-1}{}_{32/48}$, разделенных фиксированной перестановкой П, которая описывается следующим образом:

$$\Pi' = (1,33)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,34,29,40)$$

$$(10,35)(11,18)(12,26)(14)(15,36,22,38)$$

$$(16,30)(19,37)(20,27)(23)(24,31)(28,39)(32).$$

Строение трехслойных БУП $\mathbf{P}_{32/48}$ и $\mathbf{P}^{-1}{}_{32/48}$, состоящих соответственно из блоков $\mathbf{P}_{8/12}$ и $\mathbf{P}^{-1}{}_{8/12}$, показано на рис. 4.6б. Входное значение $Z = (Z_1, Z_2, Z_3, Z_4)$ блока **F** является одновременно входным значением блока $\mathbf{P}_{32/48}$. Двоичный вектор $D = (D_1, D_2, D_3, D_4)$ вырабатываемый на выходе блока $\mathbf{P}_{32/48}$, расширяется путем присоединения к нему постоянного двоичного вектора C = (10101010) до размера 40 бит. Полученный расширенный вектор (D_1, D_2, D_3, D_4, C) подается на вход перестановки Π', на выходе которой формируется вектор $(H_1, H_2, H_3, H_4, H_5)$, который разбивается на два вектора (H_1, H_2, H_3, H_4) и H_5. Первый из них поступает на вход второго внутреннего БУП $\mathbf{P}^{-1}{}_{32/48}$, а второй используется как часть управляющего вектора при выполнении операции $\mathbf{P}^{-1}{}_{32/48}$. Выходное значение блока $\mathbf{P}^{-1}{}_{32/48}$ является одновременно и выходом блока **F**.

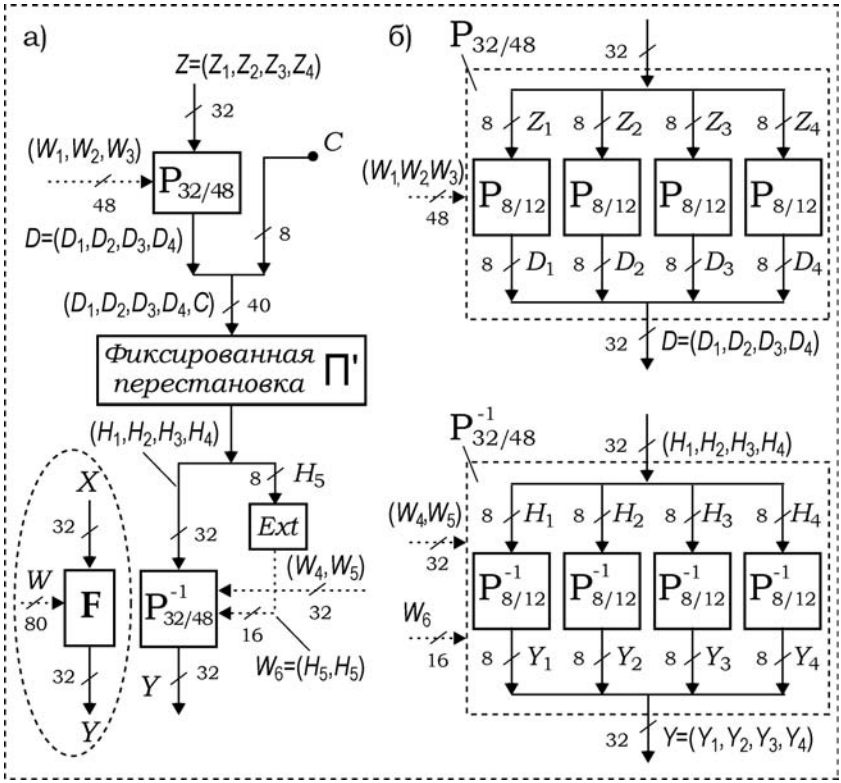

Рис. 4.6. Структура блока **F**, реализующего нелинейное криптографическое преобразование с помощью переменных перестановок

Управляющий вектор $W = (W_1, W_2, W_3, W_4, W_5)$, где $W_1, W_2, W_3, W_4, W_5 \in \mathrm{GF}(2)^{16}$, блока **F** используется следующим образом. Двоичные вектора W_1, W_2 и W_3 управляют первым, вторым и третьим активными слоями блока $\mathbf{P}_{32/48}$, соответственно, а вектора W_4 и W_3 – первым и вторым (начиная нумерацию со стороны входа, т. е. сверху) активными слоями блока $\mathbf{P}^{-1}{}_{32/48}$, соответственно. Вектор W_6, управляющий третьим слоем блока $\mathbf{P}^{-1}{}_{32/48}$, формируется с использованием 8-битового вектора H_5 в соответствии с выражением $W_6 = (H_5, H_5)$.

Легко видеть, что при фиксированном ключе левый подблок данных определяет значение вектора (D_1, D_2, D_3, D_4) и два бита со случайными номерами каждого из векторов D_1, D_2, D_3, D_4 перемещаются в вектор H_5, будучи заменены на один единичный и один нулевой биты константы C. Каждый из битов вектора (Z_1, Z_2, Z_3, Z_4) с вероятностью 1/4 может быть заменен, причем с вероятностью 1/8 он заменяется на нулевой бит и с такой же вероятностью – на единичный. В результате такого замещения перестановка, осуществляемая блоком **F**, случайным образом изменяет четность веса своего выходного значения.

4.3. Расширение класса управляемых операций с использованием элементарных управляемых инволюций

В предыдущих параграфах была показана эффективность использования переменных перестановок в качестве криптографического примитива. При этом наиболее важным моментом является управление выбором различных модификаций битовых перестановок в зависимости от преобразуемых данных. Это делает актуальным вопрос о поиске новых типов операций, зависящих от преобразуемых данных, которые могли бы быть легко реализованы в виде быстродействующих электронных схем. С этой целью в работах [18, 25] предложено, сохраняя общую топологию БУП, осуществить замену всех элементарных управляемых блоков $\mathbf{P}_{2/1}$ на некоторый другой управляемый элемент (УЭ) минимального размера, т. е. обладающий 2-битовыми входом и выходом и 1-битовым управляющим входом, используемый как типовой конструктивный узел. Для этого требуется сформулировать некоторые критерии выбора конкретных вариантов УЭ.

Поскольку блок $\mathbf{P}_{2/1}$ обеспечивает построение криптографически эффективных управляемых операций, он может служить в качестве прототипа для выбора УЭ, т. е. для формулирования критериев отбора УЭ из всех возможных вариантов. При наличии критериев эта задача может быть решена легко путем простого перебора, ввиду малого размера УЭ. Обозначим через $\mathbf{F}_{2/1}$ управляемый элемент общего вида. В общем случае УЭ может быть описан как пара булевых функций (см. рис. 4.7).

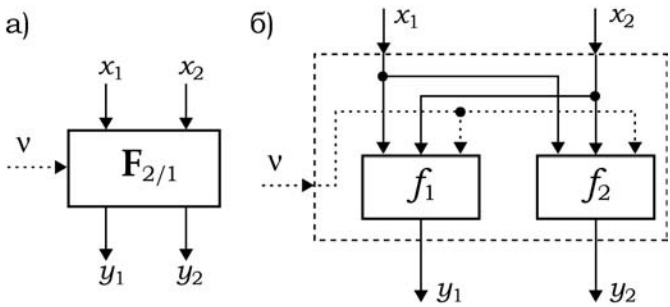

Рис. 4.7. Управляемый элемент (а) и его представление в виде пары булевых функций (б)

Другим представлением УЭ является пара подстановок размера 2×2, из которых одна подстановка выполняется над входным 2-битовым двоичным вектором (x_1, x_2) при $v = 0$, а вторая – при $v = 1$. Пару таких подстановок можно представить в виде пары таблиц или в виде пары элементарных схем преобразования. В частном случае блока $\mathbf{P}_{2/1}$ схемное представление имеет вид, представленный на рис. 4.8.

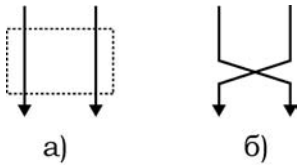

Рис. 4.8. Схемное представление блока $P_{2/1}$ в виде двух элементарных биективных преобразований, выполняемых над двухбитовым вектором (x_1, x_2) при $v = 0$ (а) и $v = 1$ (б)

Элементарный переключатель может быть охарактеризован следующим образом. Каждая из двух модификаций элементарных преобразований является инволюцией, т. е. $\mathbf{P}_{2/1}$ представляет собой элементарную управляемую инволюцию. Очевидно, обе элементарные модификации являются биективными, следовательно, каждая из пары булевых функций, описывающих УЭ $\mathbf{P}_{2/1}$, является сбалансированной. Булевы функции f_1 и f_2 являются нелинейными:

$$y_1 = f_1(x_1,x_2,v) = x_1v \oplus x_2v \oplus x_1; \qquad y_2 = f_2(x_1,x_2,v) = x_1v \oplus x_2v \oplus x_2.$$

Нелинейные булевы функции от трех переменных имеют одно и то же значение нелинейности в смысле минимального расстояния до множества афинных булевых функций от трех переменных. Учитывая изложенное, можно предложить следующие базовые критерии отбора УЭ:

К1. Булевы функции $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$ должны иметь максимальную нелинейность.

К2. Модификации элементарных преобразований, формируемых УЭ $\mathbf{F}^{(v)}{}_{2/1}$, а именно, $\mathbf{F}^{(0)}{}_{2/1}$ и $\mathbf{F}^{(1)}{}_{2/1}$, должны быть различными и представлять собой элементарное биективное преобразование вида $(x_1, x_2) \to (y_1, y_2)$.

К3. Каждая из двух модификаций УЭ $\mathbf{F}^{(v)}{}_{2/1}$ должна быть инволюцией.

Несмотря на то, что среди подстановок типа 2×2 имеются только линейные подстановки, нелинейность каждого из выходов УЭ реализуется ввиду зависимости элементарной модификации от управляющего бита. Могут быть использованы два варианта поиска УЭ. Первый состоит в переборе всех возможных пар булевых функций $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$, а второй путь – в переборе всех возможных пар модификаций $\mathbf{F}^{(0)}{}_{2/1}$ и $\mathbf{F}^{(1)}{}_{2/1}$, которые могут быть заданы как таблицы подстановок или схемно.

Последнее представление является более наглядным и простым, поскольку поиск ограничивается перебором 90 пар модификаций из 10 существующих элементарных инволюций, показанных на рис. 4.9.

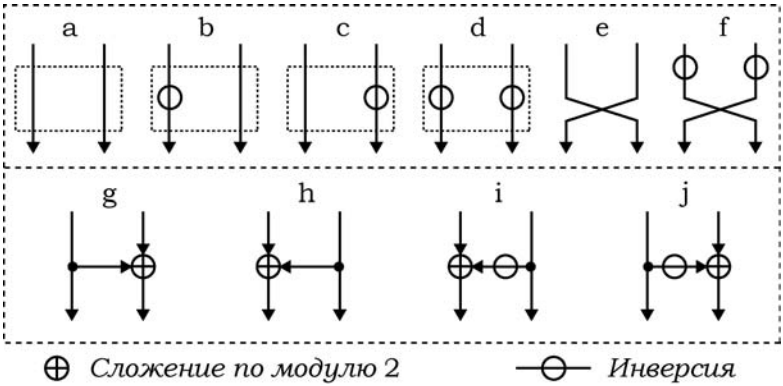

Рис. 4.9. Схемное представление всех различных существующих преобразований $(x_1, x_2) \to (y_1, y_2)$, являющихся инволюциями

Возможных различных булевых функций (БФ) от трех переменных существует 256. Чтобы ограничить число вариантов перебора, следует воспользоваться тем, что из критерия 2, требующего биективности каждой из модификаций $\mathbf{F}^{(0)}_{2/1}$ и $\mathbf{F}^{(1)}_{2/1}$, вытекает, что БФ должны быть сбалансированы. Это сразу существенно ограничивает варианты перебора. Таким образом, при переборе пар различных БФ первоначально следует отобрать полный набор сбалансированных БФ, число которых составляет 70, затем отобрать из них нелинейные, что ограничивает число интересующих нас БФ до 56. Все нелинейные сбалансированные функции показаны в табл. 4.2. После этого останется осуществить полный перебор из 56×55 вариантов пар нелинейных сбалансированных БФ от трех переменных. Это число вариантов, равное 3080, существенно больше числа вариантов перебора по парам элементарных инволюций. В первом подходе мы получаем представление УЭ в алгебраической форме, а во втором – в наглядной схемной форме, однако в итоге будут отобраны одни и те же УЭ, удовлетворяющие критериям К1 – К3.

Таблица 4.2

Множество сбалансированных БФ, удовлетворяющих показателю максимальной нелинейности ($NL(f(X))=2$)

№	АНФ	ТИ		№	АНФ	ТИ	
		Hex	Dec			Hex	Dec
1	$x_1v\oplus x_2v\oplus x_1x_2$	17	00010111	29	$x_2v\oplus x_1\oplus x_2\oplus v\oplus 1$	87	10000111
2	$x_1v\oplus x_2v\oplus x_1$	1B	00011011	30	$x_2v\oplus x_1x_2\oplus x_2\oplus v\oplus 1$	8B	10001011
3	$x_2v\oplus x_1x_2\oplus x_1$	1D	00011101	31	$x_1v\oplus x_2v\oplus x_2\oplus v\oplus 1$	8D	10001101
4	$x_2v\oplus x_1$	1E	00011110	32	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_2\oplus v\oplus 1$	8E	10001110
5	$x_1v\oplus x_2v\oplus x_2$	27	00100111	33	$x_1v\oplus x_1\oplus x_2\oplus v\oplus 1$	93	10010011
6	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_1\oplus x_2$	2B	00101011	34	$x_1x_2\oplus x_1\oplus x_2\oplus v\oplus 1$	95	10010101
7	$x_2v\oplus x_1\oplus x_2$	2D	00101101	35	$x_1x_2\oplus x_2\oplus v\oplus 1$	9A	10011010
8	$x_2v\oplus x_1x_2\oplus x_1\oplus x_2$	2E	00101110	36	$x_1v\oplus x_2\oplus v\oplus 1$	9C	10011100

№	АНФ	ТИ		№	АНФ	ТИ	
		Hex	Dec			Hex	Dec
9	$x_1v\oplus x_1x_2\oplus x_2$	35	00110101	37	$x_1v\oplus x_1x_2\oplus x_1\oplus v\oplus 1$	A3	10100011
10	$x_1v\oplus x_2$	36	00110110	38	$x_1x_2\oplus x_1\oplus v\oplus 1$	A6	10100110
11	$x_1v\oplus x_1\oplus x_2$	39	00111001	39	$x_1x_2\oplus v\oplus 1$	A9	10101001
12	$x_1v\oplus x_1x_2\oplus x_1\oplus x_2$	3A	00111010	40	$x_1v\oplus x_1x_2\oplus v\oplus 1$	AC	10101100
13	$x_2v\oplus x_1x_2\oplus v$	47	01000111	41	$x_1v\oplus x_2v\oplus x_1\oplus v\oplus 1$	B1	10110001
14	$x_2v\oplus x_1\oplus v$	4B	01001011	42	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_1\oplus v\oplus 1$	B2	10110010
15	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_1\oplus v$	4D	01001101	43	$x_2v\oplus x_1\oplus v\oplus 1$	B4	10110100
16	$x_1v\oplus x_2v\oplus x_1\oplus v$	4E	01001110	44	$x_2v\oplus x_1x_2\oplus v\oplus 1$	B8	10111000
17	$x_1v\oplus x_1x_2\oplus v$	53	01010011	45	$x_1v\oplus x_1x_2\oplus x_1\oplus x_2\oplus 1$	C5	11000101
18	$x_1x_2\oplus v$	56	01010110	46	$x_1v\oplus x_1\oplus x_2\oplus 1$	C6	11000110
19	$x_1x_2\oplus x_1\oplus v$	59	01011001	47	$x_1v\oplus x_2\oplus 1$	C9	11001001
20	$x_1v\oplus x_1x_2\oplus x_1\oplus v$	5C	01011100	48	$x_1v\oplus x_1x_2\oplus x_2\oplus 1$	CA	11001010
21	$x_1v\oplus x_2\oplus v$	63	01100011	49	$x_2v\oplus x_1x_2\oplus x_1\oplus x_2\oplus 1$	D1	11010001
22	$x_1x_2\oplus x_2\oplus v$	65	01100101	50	$x_2v\oplus x_1\oplus x_2\oplus 1$	D2	11010010
23	$x_1x_2\oplus x_1\oplus x_2\oplus v$	6A	01101010	51	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_1\oplus x_2\oplus 1$	D4	11010100
24	$x_1v\oplus x_1\oplus x_2\oplus v$	6C	01101100	52	$x_1v\oplus x_2v\oplus x_2\oplus 1$	D8	11011000
25	$x_1v\oplus x_2v\oplus x_1x_2\oplus x_2\oplus v$	71	01110001	53	$x_2v\oplus x_1\oplus 1$	E1	11100001
26	$x_1v\oplus x_2v\oplus x_2\oplus v$	72	01110010	54	$x_2v\oplus x_1x_2\oplus x_1\oplus 1$	E2	11100010
27	$x_2v\oplus x_1x_2\oplus x_2\oplus v$	74	01110100	55	$x_1v\oplus x_2v\oplus x_1\oplus 1$	E4	11100100
28	$x_2v\oplus x_1\oplus x_2\oplus v$	78	01111000	56	$x_1v\oplus x_2v\oplus x_1x_2\oplus 1$	E8	11101000

После отбора требуемых УЭ с использованием второго пути алгебраическое представление УЭ может быть легко получено следующим образом. Для каждой из двух элементарных модификаций заданного отобранного УЭ могут быть записаны БФ от двух переменных, описывающие выходы y_1 и y_2. Пусть, например, модификация $\mathbf{F}^{(0)}{}_{2/1}$ описывается парой БФ $y_1 = f'_1(x_1, x_2)$ и $y_2 = f'_2(x_1, x_2)$, а модификация $\mathbf{F}^{(1)}{}_{2/1}$ – парой $y_1 = f''_1(x_1, x_2)$ и $y_2 = f''_2(x_1, x_2)$. Тогда пара БФ от трех переменных, описывающая УЭ, легко записывается в виде следующих двух формул:

$$y_1 = (v \oplus 1)f'_1(x_1, x_2) \oplus vf''_1(x_1, x_2);$$

$$y_2 = (v \oplus 1)f'_2(x_1, x_2) \oplus vf''_2(x_1, x_2).$$

Используя второй подход, был найден полный набор УЭ, удовлетворяющих критериям К1 – К3. Данный набор представлен в табл. 4.3, строки и столбцы отмечены малыми латинскими буквами, обозначающими 10 элементарных инволюций, показанных на рис. 4.9. Знак “+” или “⊕” на пересечении строк и столбцов указывает, что модификации, соответствующие строке и столбцу, образуют пару, удовлетворяющую критериям К1 – К3, причем строка идентифицирует модификацию $\mathbf{F}^{(1)}{}_{2/1}$, а столбец – модификацию $\mathbf{F}^{(0)}{}_{2/1}$. Из указанного набора УЭ два варианта, отмеченные знаком “⊕”, соответствуют переключающим элементам, в частности пара e/a – элементарному блоку $\mathbf{P}_{2/1}$, изначально взятому в качестве прототипа. Обозначим через $\mathbf{P}'_{2/1}$ УЭ, опи-

сываемый как пара a/e. Эти два переключаемых элемента связаны следующими соотношениями: $\mathbf{P}^{(0)}{}_{2/1} = \mathbf{P}'^{(1)}{}_{2/1}$ и $\mathbf{P}^{(1)}{}_{2/1} = \mathbf{P}'^{(0)}{}_{2/1}$.

Таблица 4.3

Полный набор УЭ, удовлетворяющих критериям К1 – К3

$\mathbf{F^{(1)}\backslash F^{(0)}}$	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
a					⊕	+				
b					+	+				
c					+	+				
d					+	+				
e	⊕	+	+	+			+	+	+	+
f	+	+	+	+			+	+	+	+
g					+	+		+	+	
h					+	+	+			+
i					+	+	+			+
j					+	+		+	+	

Таким образом, мы получили 40 вариантов различных УЭ, которые могут быть применены для синтеза операций, зависящих от преобразуемых данных. Одними из важных характеристик криптографических примитивов являются дифференциальные.

Ввиду зависимости дифференциальных характеристик управляемых операций от размера входа и от топологии не представляется возможным описать такие характеристики для практически важных вариантов управляемых операций, построенных с использованием каждого из найденных УЭ. Однако это можно сделать для каждого отдельного УЭ. Это представляется достаточно интересным, поскольку дифференциальные характеристики типового конструктивного элемента определят для заданной топологии операционного блока дифференциальные характеристики последнего. На рис. 4.10 представлены варианты всех возможных разностей, относящихся к УЭ.

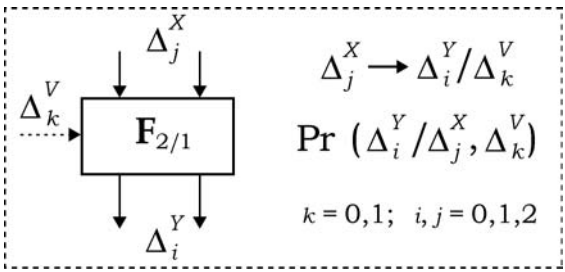

Рис. 4.10. Возможные варианты разностей, связанные с УЭ

Результаты исследования вероятностей всех нетривиальных дифференциальных характеристик всех 40 вариантов УЭ представлены в табл. 4.4. Примечательно, что

характеристики оказались идентичными для четырех подмножеств УЭ, из которых два подмножества выделяются существенно меньшей предсказуемостью значения выходной разности, что делает их предпочтительными для использования при построении управляемых операций. Рассматривая схемные представления модификаций УЭ из подмножества, относящегося к первому столбцу, можно заметить, что ни один из них не включает операции транспозиции. Обозначим УЭ данного подмножества через $\mathbf{S}_{2/1}$. Подмножество УЭ, относящихся ко второму столбцу, характеризуется тем, что его элементы включают транспозицию битов (с инверсией обоих битов или без нее) в качестве одной из элементарных модификаций, при этом второй элементарной модификацией является одна из модификаций, характерных для УЭ типа $\mathbf{S}_{2/1}$. Примем обозначение $\mathbf{R}_{2/1}$ для элементов второго подмножества. Элементы $\mathbf{R}_{2/1}$ представляются наиболее интересными для построения управляемых операционных блоков. Подмножества УЭ, относящихся к третьему и четвертому столбцам, отнесем к элементам типа $\mathbf{Z''}_{2/1}$ и $\mathbf{Z'}_{2/1}$, соответственно. Элемент $\mathbf{P}_{2/1}$ относится с УЭ типа $\mathbf{Z'}_{2/1}$.

Таблица 4.4

Дифференциальные характеристики 40 управляемых инволюций минимального размера (разбиение на 4 подмножества)

Типы УЭ *i j k*	g/h, g/i, h/g, h/j, i/g, i/j, j/h, j/i	e/g, e/h, e/i, e/j, fg, f/h, f/i, f/j, g/e, g/f, h/e, h/f, i/e, i/f, j/e, j/f	b/e, b/f, c/e, c/f, e/b, e/c, f/b, f/c	a/e, a/f, d/e, d/f, e/a, e/d, f/d, f/a
0 0 1	1/4	1/4	0	1/2
1 0 1	1/2	1/2	1	0
2 0 1	1/4	1/4	0	1/2
0 1 1	1/4	1/4	1/2	0
1 1 1	1/2	1/2	0	1
2 1 1	1/4	1/4	1/2	0
1 1 0	1/2	3/4	1	1
2 1 0	1/2	1/4	0	0
1 2 0	1	1/2	0	0
2 2 0	0	1/2	1	1
0 2 1	1/4	1/4	0	1/2
1 2 1	1/2	1/2	1	0
2 2 1	1/4	1/4	0	1/2

Рассматривая схемные представления УЭ в каждом из выделенных подмножеств, можно установить, что для подмножества $\{\mathbf{S}_{2/1}\}$ характерными являются УЭ, представленные на рис. 4.11, для подмножества $\{\mathbf{R}_{2/1}\}$ – УЭ, представленные на рис. 4.12, а для подмножества $\{\mathbf{Z}_{2/1}\} = \{\mathbf{Z'}_{2/1}, \mathbf{Z''}_{2/1}\}$ – УЭ, представленные на рис. 4.13.

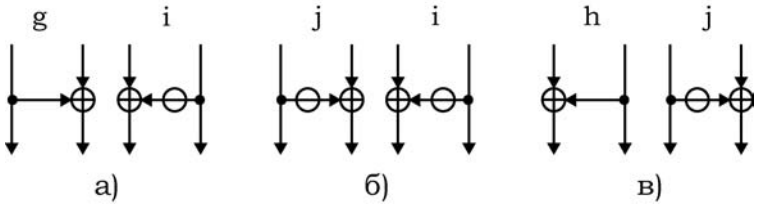

Рис. 4.11. Характерные УЭ типа $\mathbf{S}_{2/1}$, представленные парами элементарных модификаций g/i (а), j/i (б) и h/j (в)

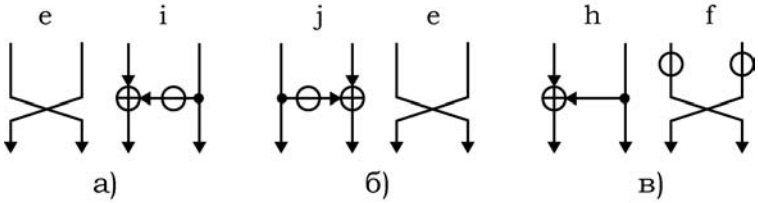

Рис. 4.12. Характерные УЭ типа $\mathbf{R}_{2/1}$, представленные парами элементарных модификаций e/i (а), j/e (б) и h/f (в)

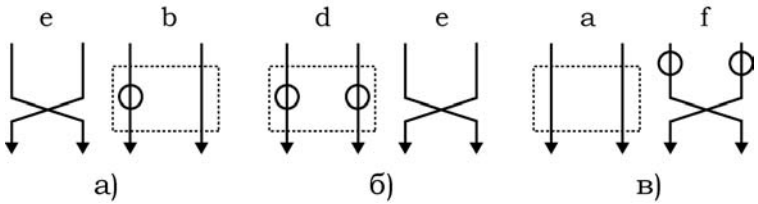

Рис. 4.13. Характерные УЭ типа $\mathbf{Z}_{2/1}$, представленные парами элементарных модификаций e/b (а), d/e (б) и a/f (в)

Для всех 40 управляемых элементарных инволюций, используя их схемные представления, были составлены пары БФ $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$. Из их анализа установлено, что каждый выход УЭ, принадлежащих подмножеству $\{\mathbf{Z}_{2/1}\}$, описывается БФ, содержащей члены x_1v и x_2v, поэтому их сумма является линейной БФ, а значит эти УЭ являются линейным криптографическим примитивом. Для УЭ, относящихся к подмножеству $\{\mathbf{S}_{2/1}\}$, БФ $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$ содержат только один квадратичный член, причем квадратичные члены в этих функциях являются различными. Благодаря этому сумма $y_1 \oplus y_2$ является нелинейной БФ, а сами УЭ – нелинейными примитивами. Для УЭ, относящихся к подмножеству $\{\mathbf{R}_{2/1}\}$, одна из БФ $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$ содержит только один квадратичный член, а вторая – два таких члена, один из которых совпадает с квадратичным членом первой БФ. Сумма $y_1 \oplus y_2$ является нелинейной БФ с одним квадратичным членом, т. е. УЭ типа $\mathbf{R}_{2/1}$ представляют собой нелинейные криптографические примитивы. Таким образом, УЭ, принадлежащие подмножествам $\{\mathbf{S}_{2/1}\}$ и $\{\mathbf{R}_{2/1}\}$, удовлетворяют еще одному важному критерию:

К4. Сумма БФ УЭ, $y_1 = f_1(x_1, x_2, v)$ и $y_2 = f_2(x_1, x_2, v)$ должна представлять собой нелинейную БФ с максимальной нелинейностью.

Во всех случаях квадратичные члены представляют собой произведение управляющего бита v и одного из входных битов. Это обстоятельство с очевидностью показывает, что использование только ключа для формирования управляющего вектора в БУП или других операционных блоках, построенных на основе УЭ типа $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$, приведет к тому, что управляемая операция будет осуществлять линейное преобразование. Только задание управления со стороны данных обеспечивает нелинейный режим функционирования управляемых операций, построенных на основе подстановочно-перестановочных сетей с УЭ минимального размера. Также видно, что УЭ типа $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$ являются существенно более предпочтительными по сравнению с УЭ типа $\mathbf{Z}_{2/1}$, включая $\mathbf{P}_{2/1}$, как по свойствам нелинейности, так и по дифференциальным характеристикам. Это позволяет сделать вывод, что на основе элементов $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$ могут быть построены перспективные управляемые операционные блоки для разработки скоростных аппаратно-ориентированных криптосистем. Полная характеристика УЭ, принадлежащих подмножествам $\{\mathbf{S}_{2/1}\}$ и $\{\mathbf{R}_{2/1}\}$, представлена в табл. 4.5. Оценка аппаратной реализации нелинейных УЭ с использованием стандартной технологии 0.33 мкм представлена в табл. 4.6.

Таблица 4.5

Полный набор УЭ, являющихся управляемыми нелинейными инволюциями

УЭ	Тип УЭ	$f_1(x_1,x_2,v)$		$f_2(x_1,x_2,v)$	
		АНФ	ТИ	АНФ	ТИ
g/e	**R**	$x_1v\oplus x_2v\oplus x_1$	00011011	$x_2v\oplus x_1\oplus x_2$	00101101
g/h	**S**	$x_2v\oplus x_1$	00011110	$x_1v\oplus x_1\oplus x_2$	00111001
e/g	**R**	$x_1v\oplus x_2v\oplus x_2$	00100111	$x_2v\oplus x_1$	00011110
e/h	**R**	$x_1v\oplus x_2$	00110110	$x_1v\oplus x_2v\oplus x_1$	00011011
h/g	**S**	$x_2v\oplus x_1\oplus x_2$	00101101	$x_1v\oplus x_2$	00110110
h/e	**R**	$x_1v\oplus x_1\oplus x_2$	00111001	$x_1v\oplus x_2v\oplus x_2$	00100111
g/i	**S**	$x_2v\oplus x_1\oplus v$	01001011	$x_1v\oplus x_1\oplus x_2$	00111001
g/f	**R**	$x_1v\oplus x_2v\oplus x_1\oplus v$	01001110	$x_2v\oplus x_2\oplus x_1\oplus v$	01111000
i/g	**S**	$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$	10000111	$x_1v\oplus x_2$	00110110
f/g	**R**	$x_1v\oplus x_2v\oplus x_2\oplus v\oplus 1$	10001101	$x_2v\oplus x_1\oplus v\oplus 1$	10110100
i/f	**R**	$x_1v\oplus x_2\oplus x_1\oplus 1$	11000110	$x_1v\oplus x_2v\oplus x_2\oplus v$	01110010
f/i	**R**	$x_1v\oplus x_2\oplus 1$	11001001	$x_1v\oplus x_2v\oplus x_1\oplus v\oplus 1$	10110001
h/j	**S**	$x_2v\oplus x_1\oplus x_2$	00101101	$x_1v\oplus x_2\oplus v$	01100011

УЭ	Тип УЭ	$f_1(x_1,x_2,v)$		$f_2(x_1,x_2,v)$	
		АНФ	ТИ	АНФ	ТИ
j/h	**S**	$x_2v\oplus x_1$	00011110	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$	10010011
j/f	**R**	$x_1v\oplus x_2v\oplus x_1\oplus v$	01001110	$x_2v\oplus x_2\oplus x_1\oplus 1$	11010010
f/h	**R**	$x_1v\oplus x_2\oplus v\oplus 1$	10011100	$x_1v\oplus x_2v\oplus x_1\oplus v\oplus 1$	10110001
f/j	**R**	$x_1v\oplus x_2v\oplus x_2\oplus v\oplus 1$	10001101	$x_2v\oplus x_1\oplus 1$	11100001
e/j	**R**	$x_1v\oplus x_2v\oplus x_2$	00100111	$x_2v\oplus x_1\oplus v$	01001011
j/e	**R**	$x_1v\oplus x_2v\oplus x_1$	00011011	$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$	10000111
j/i	**S**	$x_2v\oplus x_1\oplus v$	01001011	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$	10010011
i/e	**R**	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$	10010011	$x_1v\oplus x_2v\oplus x_2$	00100111
i/j	**S**	$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$	10000111	$x_1v\oplus x_2\oplus v$	01100011
h/f	**R**	$x_1v\oplus x_1\oplus x_2\oplus v$	01101100	$x_1v\oplus x_2v\oplus x_2\oplus u$	01110010
e/i	**R**	$x_1v\oplus x_2\oplus v$	01100011	$x_1v\oplus x_2v\oplus x_1$	00011011

Таблица 4.6

Основные характеристики аппаратной реализации УЭ типа $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$*

УЭ	Тип УЭ	ASIC 0.33 мкм		УЭ	Тип УЭ	ASIC 0.33 мкм	
		Площадь, sqmil	Частота, Ггц			Площадь, sqmil**	Частота, Ггц
g/e	**R**	3	0,95	h/j	**S**	3	1,35
g/h	**S**	3	1,37	j/h	**S**	3	0,59
e/g	**R**	2	1,92	j/f	**R**	3	1,28
e/h	**R**	2	1,92	f/h	**R**	3	0,89
h/g	**S**	3	1,35	f/j	**R**	2	1,72
h/e	**R**	3	0,95	e/j	**R**	4	0,95
g/i	**S**	3	1,37	j/e	**R**	4	0,60
g/f	**R**	4	0,83	j/i	**S**	4	0,74
i/g	**S**	4	0,74	i/e	**R**	4	0,60
f/g	**R**	3	0,89	i/j	**S**	4	0,63
i/f	**R**	3	0,89	h/f	**R**	4	0,83
f/i	**R**	2	1,72	e/i	**R**		

* Оценки получены в Патрасском университете (Греция)

** Площадь используемой поверхности полупроводникового кристалла приведена в технологических единицах sqmil; 1 sqmil = 7.45 10^{-4} мм^2

В указанной технологии УЭ $\mathbf{P}_{2/1}$ реализуется с использованием площади 3 sqmil и работает на частотах до 2.12 Ггц. Сравнение показывает, что среди нелинейных УЭ существуют более экономичные в реализации и близкие по быстродействию с переключающим элементом $\mathbf{P}_{2/1}$. Это позволяет сделать вывод, что, с учетом сравнения нелинейности и дифференциальных характеристик, правомерно ожидать, что характеристики аппаратной реализации шифров на основе переменных операций, построенных с использованием нелинейных УЭ, будут существенно превосходить характеристики реализации шифров DDP-64, CIKS-1 и SPECTR-H64.

4.4. Полная классификация нелинейных элементов $F_{2/1}$

В предыдущем параграфе был построен полный класс УЭ, являющихся элементарными управляемыми инволюциями. Но возникает вопрос о перечислении всех УЭ минимального размера, которые могут быть применены для построения управляемых операций, аналогичных БУП по своим криптографическим свойствам. То, что УЭ является инволюцией, является полезным свойством, которое упрощает построение управляемых операционных блоков. Однако наиболее важные топологии БУП могут быть реализованы также с использованием взаимно-обратных УЭ, например, в качестве прототипа могут быть взяты блоки $\mathbf{P}_{32/96}$. Поэтому представляет интерес поиск всех возможных вариантов УЭ, удовлетворяющих критериям К1 и К2, а также выделение в этом классе подмножества УЭ, удовлетворяющих дополнительному критерию нелинейности К4. В этом случае мы будем иметь полную картину о всех возможных вариантах и возможность выбора наиболее приемлемых элементов для решения задач синтеза управляемых операций.

Анализируя все возможные сочетания из двух нелинейных сбалансированных БФ, представленных в табл. 4.6, было найдено 288 вариантов УЭ, удовлетворяющих критериям К1 и К2 (см. табл. 4.7), и 192 варианта УЭ, удовлетворяющих критериям К1, К2 и К3 (см. табл. 4.8).

Таблица 4.7

Полный класс УЭ с двумя нелинейными выходами

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
1	00011011	1B	73	00100111	4B	145	10000111	87	217	11000110	C6
	00100111	27		00100111	27		00011011	1B		00011110	1E
2	00011011	1B	74	01001011	4B	146	10000111	87	218	11000110	C6
	00101101	2D		00101101	2D		00011110	1E		00100111	27
3	00011011	1B	75	01001011	4B	147	10000111	87	219	11000110	C6
	00110110	36		00111001	39		00110110	36		00110110	36
4	00011011	1B	76	01001011	4B	148	10000111	87	220	11000110	C6
	01100011	63		01101100	6C		01001011	4B		01001011	4B
5	00011011	1B	77	01001011	4B	149	10000111	87	221	11000110	C6
	01110010	72		01110010	72		01001110	4E		01100011	63
6	00011011	1B	78	01001011	4B	150	10000111	87	222	11000110	C6
	01111000	78		01111000	78		01100011	63		01110010	72
7	00011011	1B	79	01001011	4B	151	10000111	87	223	11000110	C6
	10000111	87		10000111	87		10011100	9C		10001101	8D

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
8	00011011	1B	80	01001011	4B	152	10000111	87	224	11000110	C6
	10001101	8D		10001101	8D		10110001	B1		10011100	9C
9	00011011	1B	81	01001011	4B	153	10000111	87	225	11000110	C6
	10011100	9C		10010011	93		10110100	B4		10110100	B4
10	00011011	1B	82	01001011	4B	154	10000111	87	226	11000110	C6
	11001001	C9		11000110	C6		1001001	C9		11001001	C9
11	00011011	1B	83	01001011	4B	155	10000111	87	227	11000110	C6
	11010010	D2		11010010	D2		11100001	E1		11011000	D8
12	00011011	1B	84	01001011	4B	156	10000111	87	228	11000110	C6
	11011000	D8		11011000	D8		11100100	E4		11100001	E1
13	00011110	1E	85	01001110	4E	157	10001101	8D	229	11001001	C9
	00100111	27		00100111	27		00011011	1B		00011011	1B
14	00011110	1E	86	01001110	4E	158	10001101	8D	230	11001001	C9
	00101101	2D		00101101	2D		00011110	1E		00101101	2D
15	00011110	1E	87	01001110	4E	159	10001101	8D	231	11001001	C9
	00111001	39		00110110	36		00111001	39		00111001	39
16	00011110	1E	88	01001110	4E	160	10001101	8D	232	11001001	C9
	01101100	6C		01100011	63		01001011	4B		01001110	4E
17	00011110	1E	89	01001110	4E	161	10001101	8D	233	11001001	C9
	01110010	72		01110010	72		01001110	4E		01101100	6C
18	00011110	1E	90	01001110	4E	162	10001101	8D	234	11001001	C9
	01111000	78		01111000	78		01101100	6C		01111000	78
19	00011110	1E	91	01001110	4E	163	10001101	8D	235	11001001	C9
	10000111	87		10000111	87		10010011	93		10000111	87
20	00011110	1E	92	01001110	4E	164	10001101	8D	236	11001001	C9
	10001101	8D		10001101	8D		10110001	B1		10010011	93
21	00011110	1E	93	01001110	4E	165	10001101	8D	237	11001001	C9
	10010011	93		10011100	9C		10110100	B4		10110001	B1
22	00011110	1E	94	01001110	4E	166	10001101	8D	238	11001001	C9
	11000110	C6		11001001	C9		11000110	C6		11000110	C6
23	00011110	1E	95	01001110	4E	167	10001101	8D	239	11001001	C9
	11010010	D2		11010010	D2		11100001	E1		11010010	D2
24	00011110	1E	96	01001110	4E	168	10001101	8D	240	11001001	C9
	11011000	D8		11011000	D8		11100100	E4		11100100	E4
25	00100111	27	97	01100011	63	169	10010011	93	241	11010010	D2
	00011011	1B		00011011	1B		00011110	1E		00011011	1B
26	00100111	27	98	01100011	63	170	10010011	93	242	11010010	D2
	00011110	1E		00101101	2D		00100111	27		00011110	1E
27	00100111	27	99	01100011	63	171	10010011	93	243	11010010	D2
	00111001	39		00111001	39		00110110	36		00110110	36
28	00100111	27	100	01100011	63	172	10010011	93	244	11010010	D2
	01001011	4B		01001110	4E		01001011	4B		01001011	4B
29	00100111	27	101	01100011	63	173	10010011	93	245	11010010	D2
	01001110	4E		01101100	6C		01100011	63		01001110	4E
30	00100111	27	102	01100011	63	174	10010011	93	246	11010010	D2
	01101100	6C		01111000	78		01110010	72		01100011	63
31	00100111	27	103	01100011	63	175	10010011	93	247	11010010	D2
	10010011	93		10000111	87		10001101	8D		10011100	9C
32	00100111	27	104	01100011	63	176	10010011	93	248	11010010	D2
	10110001	B1		10010011	93		10011100	9C		10110001	B1
33	00100111	27	105	01100011	63	177	10010011	93	249	11010010	D2
	10110100	B4		10110001	B1		10110100	B4		10110100	B4
34	00100111	27	106	01100011	63	178	10010011	93	250	11010010	D2
	11000110	C6		11000110	C6		11001001	C9		11001001	C9

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
35	00100111	27	107	01100011	63	179	10010011	93	251	11010010	D2
	11100001	E1		11010010	D2		11011000	D8		11100001	E1
36	00100111	27	108	01100011	63	180	10010011	93	252	11010010	D2
	11100100	E4		11100100	E4		11100001	E1		11100100	E4
37	00101101	2D	109	01101100	6C	181	10011100	9C	253	11011000	D8
	00011011	1B		00011110	1E		00011011	1B		00011011	1B
38	00101101	2D	110	01101100	6C	182	10011100	9C	254	11011000	D8
	00011110	1E		00100111	27		00101101	2D		00011110	1E
39	00101101	2D	111	01101100	6C	183	10011100	9C	255	11011000	D8
	00110110	36		00110110	36		00111001	39		00111001	39
40	00101101	2D	112	01101100	6C	184	10011100	9C	256	11011000	D8
	01001011	4B		01001011	4B		01001110	4E		01001011	4B
41	00101101	2D	113	01101100	6C	185	10011100	9C	257	11011000	D8
	01001110	4E		01100011	63		01101100	6C		01001110	4E
42	00101101	2D	114	01101100	6C	186	10011100	9C	258	11011000	D8
	01100011	63		01110010	72		01111000	78		01101100	6C
43	00101101	2D	115	01101100	6C	187	10011100	9C	259	11011000	D8
	10011100	9C		10001101	8D		10000111	87		10010011	93
44	00101101	2D	116	01101100	6C	188	10011100	9C	260	11011000	D8
	10110001	B1		10011100	9C		10010011	93		10110001	B1
45	00101101	2D	117	01101100	6C	189	10011100	9C	261	11011000	D8
	10110100	B4		10110100	B4		10110001	B1		10110100	B4
46	00101101	2D	118	01101100	6C	190	10011100	9C	262	11011000	D8
	11001001	C9		11001001	C9		11000110	C6		11000110	C6
47	00101101	2D	119	01101100	6C	191	10011100	9C	263	11011000	D8
	11100001	E1		11011000	D8		11010010	D2		11100001	E1
48	00101101	2D	120	01101100	6C	192	10011100	9C	264	11011000	D8
	11100100	E4		11100001	E1		11100100	E4		11100100	E4
49	00110110	36	121	01110010	72	193	10110001	B1	265	11100001	E1
	00011011	1B		00011011	1B		00100111	27		00100111	27
50	00110110	36	122	01110010	72	194	10110001	B1	266	11100001	E1
	00101101	2D		00011110	1E		00101101	2D		00101101	2D
51	00110110	36	123	01110010	72	195	10110001	B1	267	11100001	E1
	00111001	39		00111001	39		00110110	36		00111001	39
52	00110110	36	124	01110010	72	196	10110001	B1	268	11100001	E1
	01001110	4E		01001011	4B		01100011	63		01101100	6C
53	00110110	36	125	01110010	72	197	10110001	B1	269	11100001	E1
	01101100	6C		01001110	4E		01110010	72		01110010	72
54	00110110	36	126	01110010	72	198	10110001	B1	270	11100001	E1
	01111000	78		01101100	6C		01111000	78		01111000	78
55	00110110	36	127	01110010	72	199	10110001	B1	271	11100001	E1
	10000111	87		10010011	93		10000111	87		10000111	87
56	00110110	36	128	01110010	72	200	10110001	B1	272	11100001	E1
	10010011	93		10110001	B1		10001101	8D		10001101	8D
57	00110110	36	129	01110010	72	201	10110001	B1	273	11100001	E1
	10110001	B1		10110100	B4		10011100	9C		10010011	93
58	00110110	36	130	01110010	72	202	10110001	B1	274	11100001	E1
	11000110	C6		11000110	C6		11001001	C9		11000110	C6
59	00110110	36	131	01110010	72	203	10110001	B1	275	11100001	E1
	11010010	D2		11100001	E1		11010010	D2		11010010	D2
60	00110110	36	132	01110010	72	204	10110001	B1	276	11100001	E1
	11100100	E4		11100100	E4		11011000	D8		11011000	D8
61	00111001	39	133	01111000	78	205	10110100	B4	277	11100100	E4
	00011110	1E		00011011	1B		00100111	27		00100111	27

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
62	00111001	39	134	01111000	78	206	10110100	B4	278	11100100	E4
	00100111	27		00011110	1E		00101101	2D		00101101	2D
63	00111001	39	135	01111000	78	207	10110100	B4	279	11100100	E4
	00110110	36		00110110	36		00111001	39		00110110	36
64	00111001	39	136	01111000	78	208	10110100	B4	280	11100100	E4
	01001011	4B		01001011	4B		01101100	6C		01100011	63
65	00111001	39	137	01111000	78	209	10110100	B4	281	11100100	E4
	01100011	63		01001110	4E		01110010	72		01110010	72
66	00111001	39	138	01111000	78	210	10110100	B4	282	11100100	E4
	01110010	72		01100011	63		01111000	78		01111000	78
67	00111001	39	139	01111000	78	211	10110100	B4	283	11100100	E4
	10001101	8D		10011100	9C		10000111	87		10000111	87
68	00111001	39	140	01111000	78	212	10110100	B4	284	11100100	E4
	10011100	9C		10110001	B1		10001101	8D		10001101	8D
69	00111001	39	141	01111000	78	213	10110100	B4	285	11100100	E4
	10110100	B4		10110100	B4		10010011	93		10011100	9C
70	00111001	39	142	01111000	78	214	10110100	B4	286	11100100	E4
	11001001	C9		11001001	C9		11000110	C6		11001001	C9
71	00111001	39	143	01111000	78	215	10110100	B4	287	11100100	E4
	11011000	D8		11100001	E1		11010010	D2		11010010	D2
72	00111001	39	144	01111000	78	216	10110100	B4	288	11100100	E4
	11100001	E1		11100100	E4		11011000	D8		11011000	D8

Таблица 4.8

Полный класс нелинейных УЭ (подмножество $\{\mathbf{S}_{2/1}, \mathbf{R}_{2/1}\}$)

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
			73	00100111	4B	145	10000111	87	217	11000110	C6
				00100111	27		00011011	1B		00011110	1E
2	00011011	1B							218	11000110	C6
	00101101	2D								00100111	27
3	00011011	1B	75	01001011	4B	147	10000111	87			
	00110110	36		00111001	39		00110110	36			
4	00011011	1B	76	01001011	4B				220	11000110	C6
	01100011	63		01101100	6C					01001011	4B
			77	01001011	4B	149	10000111	87			
				01110010	72		01001110	4E			
6	00011011	1B				150	10000111	87	222	11000110	C6
	01111000	78					01100011	63		01110010	72
7	00011011	1B				151	10000111	87	223	11000110	C6
	10000111	87					10011100	9C		10001101	8D
			80	01001011	4B	152	10000111	87			
				10001101	8D		10110001	B1			
9	00011011	1B	81	01001011	4B				225	11000110	C6
	10011100	9C		10010011	93					10110100	B4
10	00011011	1B	82	01001011	4B	154	10000111	87			
	11001001	C9		11000110	C6		11001001	C9			
11	00011011	1B							227	11000110	C6
	11010010	D2								11011000	D8
			84	01001011	4B	156	10000111	87	228	11000110	C6
				11011000	D8		11100100	E4		11100001	E1

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
13	00011110	1E							229	11001001	C9
	00100111	27								00011011	1B
			86	01001110	4E	158	10001101	8D	230	11001001	C9
				00101101	2D		00011110	1E		00101101	2D
15	00011110	1E	87	01001110	4E	159	10001101	8D			
	00111001	39		00110110	36		00111001	39			
16	00011110	1E	88	01001110	4E	160	10001101	8D	232	11001001	C9
	01101100	6C		01100011	63		01001011	4B		01001110	4E
17	00011110	1E									
	01110010	72									
			90	01001110	4E	162	10001101	8D	234	11001001	C9
				01111000	78		01101100	6C		01111000	78
			91	01001110	4E	163	10001101	8D	235	11001001	C9
				10000111	87		10010011	93		10000111	87
20	00011110	1E									
	10001101	8D									
21	00011110	1E	93	01001110	4E	165	10001101	8D	237	11001001	C9
	10010011	93		10011100	9C		10110100	B4		10110001	B1
22	00011110	1E	94	01001110	4E	166	10001101	8D			
	11000110	C6		11001001	C9		11000110	C6			
			95	01001110	4E	167	10001101	8D	239	11001001	C9
				11010010	D2		11100001	E1		11010010	D2
24	00011110	1E							240	11001001	C9
	11011000	D8								11100100	E4
			97	01100011	63	169	10010011	93	241	11010010	D2
				00011011	1B		00011110	1E		00011011	1B
26	00100111	27	98	01100011	63	170	10010011	93			
	00011110	1E		00101101	2D		00100111	27			
27	00100111	27							243	11010010	D2
	00111001	39								00110110	36
28	00100111	27	100	01100011	63	172	10010011	93			
	01001011	4B		01001110	4E		01001011	4B			
									245	11010010	D2
										01001110	4E
30	00100111	27	102	01100011	63	174	10010011	93	246	11010010	D2
	01101100	6C		01111000	78		01110010	72		01100011	63
31	00100111	27	103	01100011	63	175	10010011	93	247	11010010	D2
	10010011	93		10000111	87		10001101	8D		10011100	9C
									248	11010010	D2
										10110001	B1
33	00100111	27	105	01100011	63	177	10010011	93			
	10110100	B4		10110001	B1		10110100	B4			
34	00100111	27							250	11010010	D2
	11000110	C6								11001001	C9
35	00100111	27	107	01100011	63	179	10010011	93			
	11100001	E1		11010010	D2		11011000	D8			
			108	01100011	63	180	10010011	93	252	11010010	D2
				11100100	E4		11100001	E1		11100100	E4
37	00101101	2D	109	01101100	6C	181	10011100	9C			
	00011011	1B		00011110	1E		00011011	1B			
			110	01101100	6C	182	10011100	9C	254	11011000	D8
				00100111	27		00101101	2D		00011110	1E
39	00101101	2D							255	11011000	D8
	00110110	36								00111001	39

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
			112	01101100	6C	184	10011100	9C	256	11011000	D8
				01001011	4B		01001110	4E		01001011	4B
41	00101101	2D									
	01001110	4E									
42	00101101	2D	114	01101100	6C	186	10011100	9C	258	11011000	D8
	01100011	63		01110010	72		01111000	78		01101100	6C
43	00101101	2D	115	01101100	6C	187	10011100	9C	259	11011000	D8
	10011100	9C		10001101	8D		10000111	87		10010011	93
44	00101101	2D									
	10110001	B1									
			117	01101100	6C	189	10011100	9C	261	11011000	D8
				10110100	B4		10110001	B1		10110100	B4
46	00101101	2D							262	11011000	D8
	11001001	C9								11000110	C6
			119	01101100	6C	191	10011100	9C	263	11011000	D8
				11011000	D8		11010010	D2		11100001	E1
48	00101101	2D	120	01101100	6C	192	10011100	9C			
	11100100	E4		11100001	E1		11100100	E4			
49	00110110	36							265	11100001	E1
	00011011	1B								00100111	27
50	00110110	36	122	01110010	72	194	10110001	B1			
	00101101	2D		00011110	1E		00101101	2D			
			123	01110010	72	195	10110001	B1	267	11100001	E1
				00111001	39		00110110	36		00111001	39
52	00110110	36	124	01110010	72	196	10110001	B1	268	11100001	E1
	01001110	4E		01001011	4B		01100011	63		01101100	6C
									269	11100001	E1
										01110010	72
54	00110110	36	126	01110010	72	198	10110001	B1			
	01111000	78		01101100	6C		01111000	78			
55	00110110	36	127	01110010	72	199	10110001	B1			
	10000111	87		10010011	93		10000111	87			
									272	11100001	E1
										10001101	8D
57	00110110	36	129	01110010	72	201	10110001	B1	273	11100001	E1
	10110001	B1		10110100	B4		10011100	9C		10010011	93
			130	01110010	72	202	10110001	B1	274	11100001	E1
				11000110	C6		11001001	C9		11000110	C6
59	00110110	36	131	01110010	72	203	10110001	B1			
	11010010	D2		11100001	E1		11010010	D2			
60	00110110	36							276	11100001	E1
	11100100	E4								11011000	D8
61	00111001	39	133	01111000	78	205	10110100	B4			
	00011110	1E		00011011	1B		00100111	27			
62	00111001	39							278	11100100	E4
	00100111	27								00101101	2D
			135	01111000	78	207	10110100	B4	279	11100100	E4
				00110110	36		00111001	39		00110110	36
64	00111001	39				208	10110100	B4	280	11100100	E4
	01001011	4B					01101100	6C		01100011	63
			137	01111000	78	209	10110100	B4			
				01001110	4E		01110010	72			
66	00111001	39	138	01111000	78				282	11100100	E4
	01110010	72		01100011	63					01111000	78

№	ТИ		№	ТИ		№	ТИ		№	ТИ	
	10	16		10	16		10	16		10	16
67	00111001	39	139	01111000	78				283	11100100	E4
	10001101	8D		10011100	9C					10000111	87
			140	01111000	78	212	10110100	B4			
				10110001	B1		10001101	8D			
69	00111001	39				213	10110100	B4	285	11100100	E4
	10110100	B4					10010011	93		10011100	9C
			142	01111000	78	214	10110100	B4	286	11100100	E4
				11001001	C9		11000110	C6		11001001	C9
71	00111001	39							287	11100100	E4
	11011000	D8								11010010	D2
72	00111001	39	144	01111000	78	216	10110100	B4			
	11100001	E1		11100100	E4		11011000	D8			

Для дальнейшей классификации и определения пар БФ, описывающих отобранные УЭ, последние удобно представить в схемном виде, т. е. в виде пар $\mathbf{F}^{(0)}{}_{2/1}$ и $\mathbf{F}^{(1)}{}_{2/1}$, являющихся элементарными биективными преобразованиями типа $(x_1, x_2) \to (y_1, y_2)$. Все возможные варианты таких преобразований представлены на рис. 4.13а.

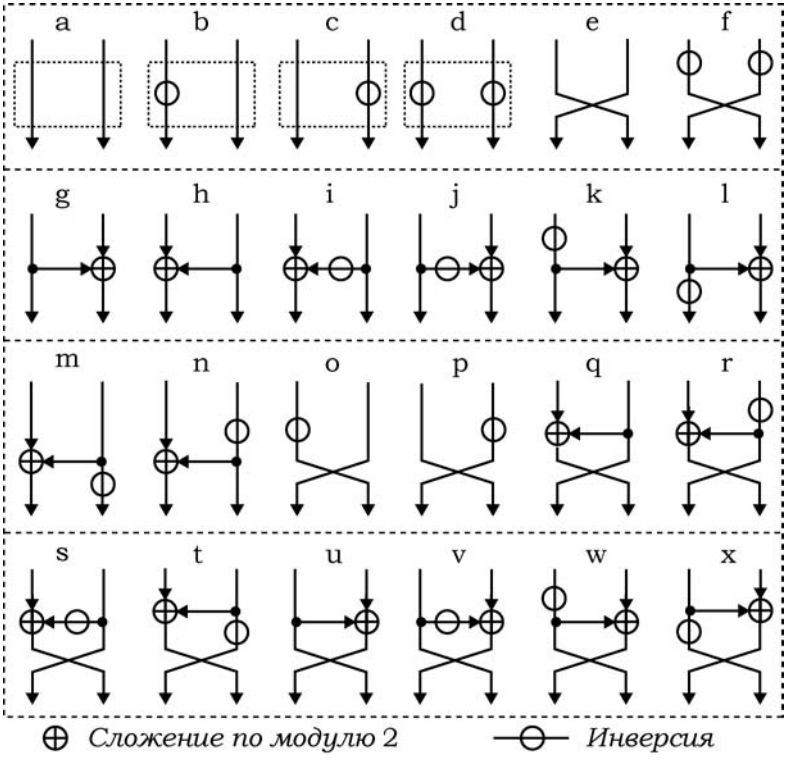

Рис. 4.13а. Все существующие варианты биективных преобразований типа $(x_1, x_2) \to (y_1, y_2)$

Исследование дифференциальных характеристик 248 вариантов УЭ, не являющихся инволюциями и обладающих нелинейными выходами, показало, что все они разбиваются на 4 подмножества, из которых два подмножества относятся к множеству 168 нелинейных УЭ и имеют дифференциальные характеристики идентичные дифференциальным характеристикам управляемых инволюций $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$ (см. табл. 4.9)

Изучая схемные представления УЭ в первом (и во втором) подмножествах, было установлено, что подмножество, которое сходно по дифференциальным характеристикам с элементами типа $\mathbf{S}_{2/1}$ (соответственно $\mathbf{R}_{2/1}$), является сходным с $\{\mathbf{S}_{2/1}\}$ (соответственно с $\{\mathbf{R}_{2/1}\}$) и по характерным реализуемым элементарным модификациям $\mathbf{F}^{(0)}{}_{2/1}$ и $\mathbf{F}^{(1)}{}_{2/1}$. Следовательно, данные подмножества УЭ разумно объединить с соответствующими ими подмножествами инволюций типа $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$ в два подкласса $\{\mathbf{S}_{2/1}\}$ и $\{\mathbf{R}_{2/1}\}$, соответственно (см. рис. 4.14 и 4.15).

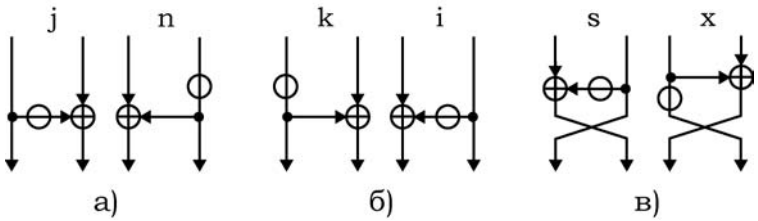

Рис. 4.14. Характерные УЭ типа $\mathbf{S}_{2/1}$, представленные парами элементарных модификаций j/n (а), k/i (б) и s/x (в)

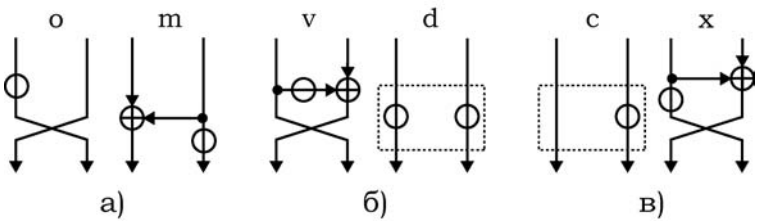

Рис. 4.15. Характерные УЭ типа $\mathbf{R}_{2/1}$, представленные парами элементарных модификаций o/m (а), v/d (б) и c/x (в)

Подмножества УЭ, относящихся к предпоследнему и последнему столбцам, представленные в табл. 4.9, отнесем, соответственно, к подмножествам $\{\mathbf{Z}^{*}{}_{2/1}\}$ и $\{\mathbf{Z}^{**}{}_{2/1}\}$ и расширим подкласс $\{\mathbf{Z}_{2/1}\}$: $\{\mathbf{Z}_{2/1}\} = \{\mathbf{Z}'_{2/1}, \mathbf{Z}''_{2/1}, \mathbf{Z}^{*}{}_{2/1}, \mathbf{Z}^{**}{}_{2/1}\}$. Последнее является естественным, поскольку для всех УЭ из $\{\mathbf{Z}_{2/1}\}$ сумма выходов является линейной БФ. Характерные схемные представления УЭ, относящихся к подмножествам $\{\mathbf{Z}^{*}{}_{2/1}\}$ и $\{\mathbf{Z}^{**}{}_{2/1}\}$, показаны на рис. 4.16., 4.17.

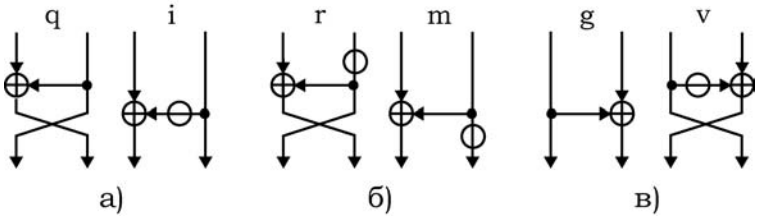

Рис. 4.16. Характерные УЭ типа $\mathbf{Z}^{*}_{2/1}$, представленные парами элементарных модификаций q/i (а), r/m (б) и g/v (в)

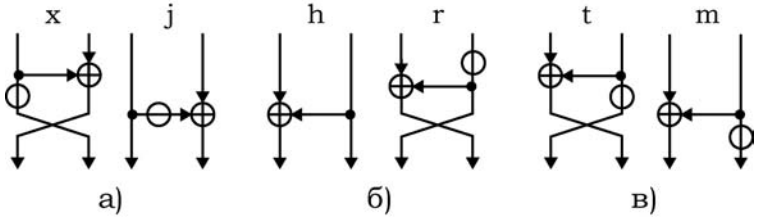

Рис. 4.17. Характерные УЭ типа $\mathbf{Z}^{**}_{2/1}$, представленные парами элементарных модификаций x/j (а), h/r (б) и t/m (в)

Таблица 4.9

Дифференциальные характеристики 248 УЭ, не являющихся инволюциями и отобранных из класса УЭ с двумя нелинейными выходами

Примеры и тип УЭ ***i j k***	l/h, l/i, l/m, l/n, m/g, m/j, m/k, u/q, v/s, w/t ($\mathbf{S}_{2/1}$)	g/o, h/p, o/l, o/m, p/k, m/p, a/r, b/t, a/x, d/w, t/c, u/b, w/d, p/n ($\mathbf{R}_{2/1}$)	q/i, q/m, r/i, r/m, t/h, t/n, k/v, k/x, h/t ($\mathbf{Z}^{*}_{2/1}$)	f/u, f/w, j/v, j/x, n/q, n/r, u/k, v/l, x/l ($\mathbf{Z}^{**}_{2/1}$)
0 0 1	1/4	1/4	0	1/2
1 0 1	1/2	1/2	1	0
2 0 1	1/4	1/4	0	1/2
0 1 1	1/4	1/4	1/4	1/4
1 1 1	1/2	1/2	1/2	1/2
2 1 1	1/4	1/4	1/4	1/4
1 1 0	1/2	3/4	1/2	1/2
2 1 0	1/2	1/4	1/2	1/2
1 2 0	1	1/2	1	1
2 2 0	0	1/2	0	0
0 2 1	1/4	1/4	1/2	0
1 2 1	1/2	1/2	0	1
2 2 1	1/4	1/4	1/2	0

В предложенной классификации имеют место следующие положения:

- Если $\mathbf{F}_{2/1} \in \{\mathbf{S}_{2/1}\}$, то $\mathbf{F}^{-1}{}_{2/1} \in \{\mathbf{S}_{2/1}\}$.
- Если $\mathbf{F}_{2/1} \in \{\mathbf{R}_{2/1}\}$, то $\mathbf{F}^{-1}{}_{2/1} \in \{\mathbf{R}_{2/1}\}$.
- Пусть УЭ $\mathbf{Z}'_{2/1}$ и $\mathbf{Z}''_{2/1}$ не являются инволюциями. Тогда, если $\mathbf{F}_{2/1} \in \{\mathbf{Z}'_{2/1}\} \cup \{\mathbf{Z}''_{2/1}\}$, то $\mathbf{F}^{-1}{}_{2/1} \in \{\mathbf{Z}'_{2/1}\} \cup \{\mathbf{Z}''_{2/1}\}$.
- Если $\mathbf{F}_{2/1} \in \{\mathbf{Z}^{*}{}_{2/1}\} \cup \{\mathbf{Z}^{**}{}_{2/1}\}$, то $\mathbf{F}^{-1}{}_{2/1} \notin \{\mathbf{Z}_{2/1}\} \cup \{\mathbf{R}_{2/1}\} \cup \{\mathbf{S}_{2/1}\}$, поскольку один из двух выходов таких $\mathbf{F}^{-1}{}_{2/1}$ является линейной БФ.

Первые два положения имеют конструктивное значение, поскольку показывают, что использование взаимно-обратных УЭ типов $\mathbf{S}_{2/1}$ и $\mathbf{R}_{2/1}$ относятся к одному и тому же подклассу. Поэтому при необходимости построения легко обращаемых операционных блоков можно использовать пары $\mathbf{S}_{2/1}$ и $\mathbf{S}^{-1}{}_{2/1}$ ($\mathbf{R}_{2/1}$ и $\mathbf{R}^{-1}{}_{2/1}$) без ущерба для нелинейности синтезируемого операционного блока.

4.5. Синтез управляемых операционных подстановок на основе элементов $\mathbf{F}_{2/1}$

4.5.1. Принципы построения управляемых операционных подстановок

С учетом целей использования управляемых операционных подстановок (УОП) в криптографических алгоритмах и показателей качества, введенных в п.2.3, они должны удовлетворять следующим принципам построения:

- преобразование, выполняемое УОП $Y=\mathbf{F}_{n/m}(X, V)$, где X, $Y \in \mathrm{GF}(2)^n$, $V \in \mathrm{GF}(2)^m$, биективно относительно X при фиксированном значении V;
- механизм управления УОП $\mathbf{F}_{n/m}$ обеспечивает формирование большого количества различных модификаций подстановочных операций;
- преобразование, выполняемое УОП $\mathbf{F}_{n/m}$, обладает высокой нелинейностью;
- булевы функции, реализующие УОП $\mathbf{F}_{n/m}$, имеют хорошие корреляционные показатели;
- аппаратная или аппаратно-программная реализация УОП $\mathbf{F}_{n/m}$ имеет относительно небольшую вычислительную сложность и обеспечивает высокое быстродействие преобразования информации.

Как показано в предыдущих параграфах, заменяя в БУП [11, 15] элементарный переключатель $\mathbf{P}_{2/1}$ на более общее элементарное преобразование $\mathbf{F}_{2/1}$ и сохраняя слоистую структуру БУП, легко синтезировать эффективные блоки управляемых операционных подстановок (БУОП) над векторами большой длины [18, 23, 25].

В качестве базового элемента БУОП используется элементарный подстановочный блок $\mathbf{F}_{2/1}$ (рис. 4.18), осуществляющий отображение вида $\mathrm{GF}(2)^3 \rightarrow \mathrm{GF}(2)^2$ на основе двух БФ от трех аргументов: $y_i = f(x_1, x_2, v)$, i=1, 2, где $y_i \in \mathrm{GF}(2)$, $x_1, x_2 \in \mathrm{GF}(2)$ - значения входных бит, $v \in \mathrm{GF}(2)$ - значение управляющего бита. Совокупность управляющих битов будем интерпретировать как управляющий вектор V.

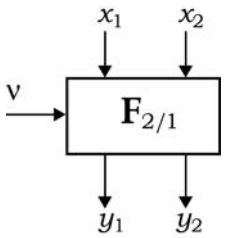

Рис. 4.18. Элементарный управляемый подстановочный элемент

Использование таких элементов приводит к увеличению количества возможных вариантов построения блоков управляемых операций, зависящих от значения управляющего вектора и включающих БУП как частный случай, поскольку совокупность БФ, реализующих управляемую перестановку двух бит является разновидностью элементарной управляемой подстановки $\mathbf{F}_{2/1}$.

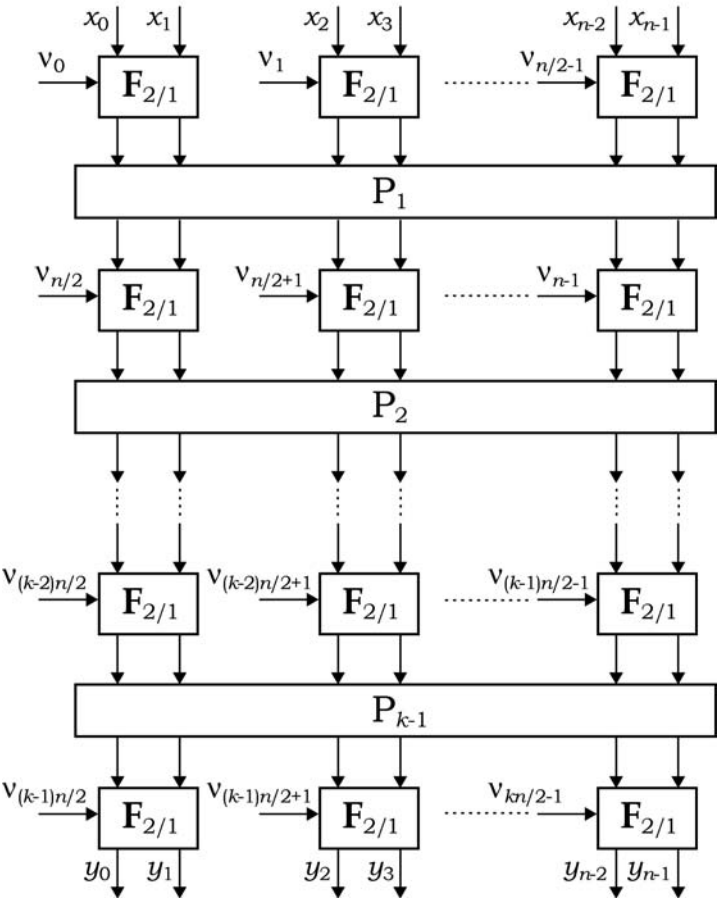

Рис. 4.19. Слоистая структура БУОП

Путем комбинирования базовых блоков подстановок $\mathbf{F}_{2/1}$ могут быть синтезированы БУОП $\mathbf{F}_{n/m}$, где n – число входных (выходных) бит, m – число управляющих бит. Такие блоки задают отображение вида $GF(2)^{n+m} \to GF(2)^n$. Представляет интерес использование слоистой схемы построения, каждый слой которой состоит из $\frac{n}{2}$ включенных параллельно элементов $\mathbf{F}_{2/1}$, а между слоями расположены фиксированные перестановки $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{k-1}\}$ (рис. 4.19). В этом случае БУОП $\mathbf{F}_{n/m}$ можно представить в виде суперпозиции k подстановочных слоев и k–1 фиксированных перестановок, а именно:

$$\mathbf{F}_{n/m} = \boldsymbol{\sigma}_1 \circ \mathbf{P}_1 \circ \boldsymbol{\sigma}_2 \circ \mathbf{P}_2 \circ \ldots \circ \mathbf{P}_{k-1} \circ \boldsymbol{\sigma}_k, \tag{4.1}$$

где $\boldsymbol{\sigma}_j$ – j-й слой $\frac{n}{2}$ блоков $\mathbf{F}_{2/1}$, $\mathbf{P}_j$ – j-й слой фиксированных перестановок.

На основе базового элемента $\mathbf{F}_{2/1}$ синтезированы БУОП, осуществляющие отображения $GF(2)^{112} \to GF(2)^{32}$, $GF(2)^{256} \to GF(2)^{64}$, $GF(2)^{576} \to GF(2)^{128}$ [18, 23, 25]. Слоистая структура БУОП с $n = 2^k$ - битовым входом для данных имеет управляющий вектор размера $m = \frac{kn}{2}$ бит, где k - количество слоев базовых подстановок.

4.5.2. Исследование основных свойств и оптимизация УОП

В синтезируемых УОП возможно использование таких фиксированных перестановок, которые отработаны в БУП [11, 14] и обеспечивают определенные свойства влияния входных и управляющих битов на выходные. В БУП это свойство описывается порядком перестановки, под которым понимается возможность (при определенной комбинации управляющих битов) перестановки l произвольно выбранных бит входного вектора в любые l разрядов на выходе. Очевидно, если для БУП набор фиксированных перестановок $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{k-1}\}$ обеспечивает влияние произвольных l входных битов на произвольные l выходных битов, то это будет выполняться и для любого БУОП с использованием различных наборов элементарных блоков подстановок $\mathbf{F}_{2/1}$, если последние имеют порядок 2. При этом базовые блоки $\mathbf{F}_{2/1}$ могут выбираться произвольными в пределах одного слоя или в пределах общей структуры, что не изменяет значения порядка БУОП, определяемого фиксированными перестановками.

Предлагается способ реализации БУОП первого порядка, при котором каждый слой фиксированных перестановок обеспечивает соединение входов и выходов двух слоев подстановок в соответствии со следующими правилами:

$$\begin{cases} \varsigma_{j,i} = \varsigma_{j+1,i} \\ \varsigma_{j,i+1} = \varsigma_{j+1,\gamma} \end{cases}, \tag{4.2}$$

где $\gamma = \left(i+1+2^{j+1}\right) \bmod n$, j – номер слоя подстановки, $j \in \{0, 1, \ldots, k-1\}$, $i = 2s$ - номер выходного (входного) разряда операционной подстановки, $s \in \left\{0, 1, \ldots, \frac{n-2}{2}\right\}$. Данный вариант структуры фиксированных перестановок обеспечивает влияние каждого входного бита на каждый выходной, кроме того, обеспечивается зависимость любого выходного бита от всех входных и части управляющих бит. Пример структуры БУОП $\mathbf{F}_{32/80}$ с предложенным вариантом фиксированных перестановок представлен на рис. 4.20.

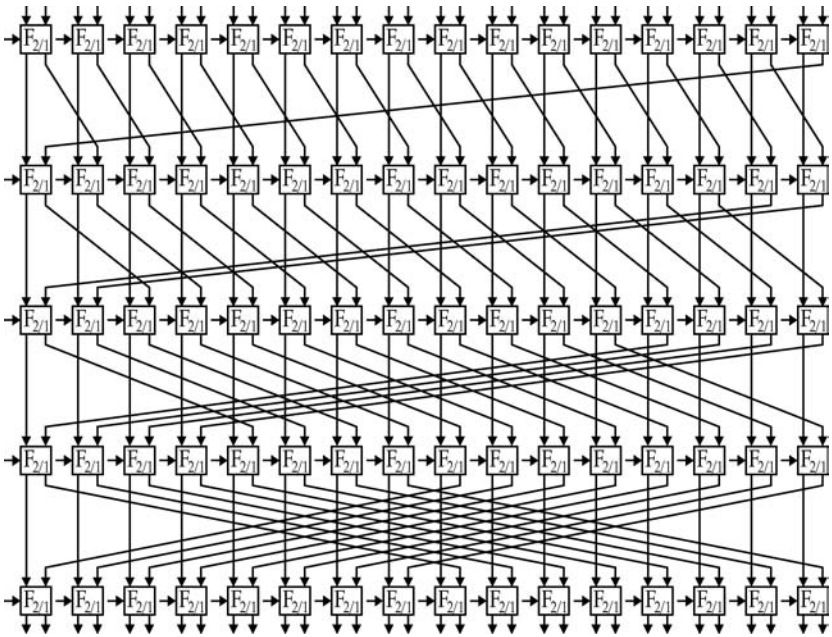

Рис. 4.20. Управляемый операционный блок $\mathbf{F}_{32/80}$

Для исследования свойств БУОП определим через $\mathcal{P}$ множество всех компонентных БФ, реализующих БУОП вида $\mathbf{F}_{n/m}$.

Утверждение 4.1

$\forall f_i \in \mathcal{P} : \mathrm{GF}(2)^{\mu} \to \mathrm{GF}(2)$, $i \in \{0, 1, \ldots, n-1\}$, количество аргументов функции определяется выражением $\mu = 2n - 1 = 2^{k+1} - 1$.

Доказательство

Каждое значение выхода БУОП f_i зависит от одного управляющего бита, соответствующего последнему k–му каскаду и двух бит данных от $k-1$ каскада. В свою оче-

редь два бита данных с $k-1$ каскада зависят от двух управляющих бит, соответствующих $k-1$ каскаду и четырех бит данных с $k-2$ каскада. Продвигаясь от выхода к входу БУОП, получим, что любой i–й выход f_i БУОП зависит от $2^k=n$ значений входных бит и $\sum_{j=1}^{k} 2^{(k-j)} = n-1$ значений управляющих бит.

При проектировании блоков $\mathbf{F}_{2/1}$ из всего множества образующих БФ от трех аргументов $\#\{f(x_1, x_2, x_3)\}= 2^{2^n} =256$, где $n=3$, а x_3 интерпретируется как управляющий бит v, выбирались только сбалансированные БФ, т.е. БФ, таблица истинности которых содержит одинаковое количество 0 и 1. Количество таких функций $\#\{f(x) \mid (\#f(x)=0) = (\#f(x)=1)\} = \begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix} = 70$. Основным критерием конструирования $\mathbf{F}_{n/m}$ являлся критерий нелинейности преобразования, осуществляемого подстановочным отображением.

Нелинейность подстановочного отображения зависит от нелинейности компонентных БФ, реализующих данное отображение. В п. 2.3 показано, что под нелинейностью БФ $\mathrm{NL}(f(x))$ понимается расстояние Хэмминга между ТИ исследуемой БФ и ТИ наиболее близкой БФ из всего множества аффинных БФ для того же количества аргументов. Следовательно, нелинейность связана с вероятностью совпадения функции с ее наилучшим линейным аналогом. Чем выше нелинейность БФ, реализующих подстановочное отображение и шифр в целом, тем сложнее осуществить атаку на шифр методом линейного криптоанализа.

Известно, что результат суперпозиции линейных БФ также является линейной БФ, что справедливо для выходного вектора блока БУОП, если БФ в базовом блоке $\mathbf{F}_{2/1}$ являются линейными. Поэтому для реализации блока $\mathbf{F}_{2/1}$ из полученного подмножества сбалансированных функций методом перебора выбирались БФ с максимальной нелинейностью $\mathrm{NL}(f(x))=2$. Сбалансированных функций, удовлетворяющих критерию максимальной нелинейности, найдено $\#\{f(x) \mid (\#f(x)=0) = (\#f(x)=1), \mathrm{NL}(f(x))=2\}=24$ (см. табл. 4.10). Нелинейность функции $f_i \in \mathcal{P}$, являющейся компонентом выходного вектора $Y \in \mathrm{GF}(2)^n$, для БУОП $\mathbf{F}_{n/m}$ может быть найдена из условия $\mathrm{NL}(f_i) = 2^{n-1} - \frac{1}{2} \max\limits_{\forall \alpha \in \mathrm{GF}(2)^n} |\hat{U}_\alpha(f_i)|$, где $\hat{U}_\alpha(f_i)$ — значение компоненты спектра ПУА функции $f_i \in \mathcal{P}$ для вектора $\alpha \in \mathrm{GF}(2)^n$.

Таблица 4.10

№	Таблица истинности	Дизъюнктивная нормальная форма	Алгебраическая нормальная форма
1	00011011	$x_1\overline{v} \vee x_2 v$	$x_1 v \oplus x_2 v \oplus x_1$
2	00011110	$x_1\overline{v} \vee x_1\overline{x}_2 v \vee \overline{x}_1 x_2 v$	$x_2 v \oplus x_1$
3	00100111	$x_1 v \vee x_2\overline{v}$	$x_1 v \oplus x_2 v \oplus x_2$

№	Таблица истинности	Дизъюнктивная нормальная форма	Алгебраическая нормальная форма
4	00101101	$x_1\nu \vee x_1\overline{x}_2\overline{\nu} \vee \overline{x}_1x_2\overline{\nu}$	$x_2\nu\oplus x_1\oplus x_2$
5	00110110	$x_2\overline{\nu} \vee \overline{x}_1x_2\nu \vee x_1\overline{x}_2\nu$	$x_1\nu\oplus x_2$
6	00111001	$x_2\nu \vee \overline{x}_1x_2\overline{\nu} \vee x_1\overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_1\oplus x_2$
7	01001011	$x_1\overline{\nu} \vee x_1x_2\nu \vee \overline{x}_1\overline{x}_2\nu$	$x_2\nu\oplus x_1\oplus\nu$
8	01001110	$x_1\overline{\nu} \vee \overline{x}_2\nu$	$x_1\nu\oplus x_2\nu\oplus x_1\oplus\nu$
9	01100011	$x_2\overline{\nu} \vee x_1x_2\nu \vee \overline{x}_1\overline{x}_2\nu$	$x_1\nu\oplus x_2\oplus\nu$
10	01101100	$\overline{x}_2\nu \vee \overline{x}_1x_2\overline{\nu} \vee x_1\overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_2\oplus x_1\oplus\nu$
11	01110010	$\overline{x}_1\nu \vee x_2\overline{\nu}$	$x_1\nu\oplus x_2\nu\oplus x_2\oplus\nu$
12	01111000	$\overline{x}_1\nu \vee \overline{x}_1x_2\overline{\nu} \vee x_1\overline{x}_2\overline{\nu}$	$x_2\nu\oplus x_2\oplus x_1\oplus\nu$
13	10000111	$x_1\nu \vee x_1x_2\overline{\nu} \vee \overline{x}_1\overline{x}_2\overline{\nu}$	$x_2\nu\oplus x_2\oplus x_1\oplus\nu\oplus 1$
14	10001101	$x_1\nu \vee \overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_2\nu\oplus x_2\oplus\nu\oplus 1$
15	10010011	$x_2\nu \vee x_1x_2\overline{\nu} \vee \overline{x}_1\overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_2\oplus x_1\oplus\nu\oplus 1$
16	10011100	$\overline{x}_2\overline{\nu} \vee x_1\overline{x}_2\nu \vee \overline{x}_1x_2\nu$	$x_1\nu\oplus x_2\oplus\nu\oplus 1$
17	10110001	$\overline{x}_1\overline{\nu} \vee x_2\nu$	$x_1\nu\oplus x_2\nu\oplus x_1\oplus\nu\oplus 1$
18	10110100	$\overline{x}_1\overline{\nu} \vee \overline{x}_1x_2\nu \vee x_1\overline{x}_2\nu$	$x_2\nu\oplus x_1\oplus\nu\oplus 1$
19	11000110	$\overline{x}_2\nu \vee x_1x_2\overline{\nu} \vee \overline{x}_1\overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_2\oplus x_1\oplus 1$
20	11001001	$\overline{x}_2\overline{\nu} \vee x_1x_2\nu \vee \overline{x}_1\overline{x}_2\nu$	$x_1\nu\oplus x_2\oplus 1$
21	11010010	$\overline{x}_1\nu \vee x_1x_2\overline{\nu} \vee \overline{x}_1\overline{x}_2\overline{\nu}$	$x_2\nu\oplus x_2\oplus x_1\oplus 1$
22	11011000	$\overline{x}_1\nu \vee \overline{x}_2\overline{\nu}$	$x_1\nu\oplus x_2\nu\oplus x_2\oplus 1$
23	11100001	$\overline{x}_1\overline{\nu} \vee x_1x_2\nu \vee \overline{x}_1\overline{x}_2\nu$	$x_2\nu\oplus x_1\oplus 1$
24	11100100	$\overline{x}_1\overline{\nu} \vee \overline{x}_2\nu$	$x_1\nu\oplus x_2\nu\oplus x_1\oplus 1$

Пары БФ в базовом элементе $\mathbf{F}_{2/1}$ целесообразно выбирать таким образом, чтобы элементарное преобразование, осуществляемое $\mathbf{F}_{2/1}$, было инволюцией. Экспериментально определено 24 нелинейных инволюций, образованных из найденного подмножества сбалансированных нелинейных БФ (табл. 4.11). На их основе просто синтезировать обратимые БУОП, осуществляющие преобразование векторов большого размера. Операционные подстановки $\mathbf{F}_{n/m}(X, V)$ являются обратимыми, если для лю-

бого $X \in \mathrm{GF}(2)^n$ существует обратная подстановка $\mathbf{F}^{-1}_{n/m}(X, V)$ и при данном значении управляющего вектора $V \in \mathrm{GF}(2)^m$ выполняется условие $\mathbf{F}^{-1}_{n/m}\left(\mathbf{F}_{n/m}(X,\ V), V\right) = X$. Обратное преобразование $\mathbf{F}^{-1}{}_{n/m}$ также можно представить в виде следующей суперпозиции:

$$\mathbf{F}^{-1}{}_{n/m} = \boldsymbol{\sigma}^{-1}{}_k \circ \mathbf{P}^{-1}{}_{k-1} \circ \boldsymbol{\sigma}^{-1}{}_{k-1} \circ \mathbf{P}^{-1}{}_{k-2} \circ \ldots \circ \mathbf{P}^{-1}{}_1 \circ \boldsymbol{\sigma}^{-1}{}_1, \qquad (4.3)$$

где $\boldsymbol{\sigma}^{-1}{}_i = \boldsymbol{\sigma}_i$, так как блоки $\mathbf{F}_{2/1}$ реализуют инволюции.

Таблица 4.11

$f_1(x_1,x_2,v)$	$f_2(x_1,x_2,v)$	$f_1(x_1,x_2,v)$	$f_2(x_1,x_2,v)$
$x_1v\oplus x_2v\oplus x_1$	$x_2v\oplus x_1\oplus x_2$	$x_2v\oplus x_1\oplus x_2$	$x_1v\oplus x_2\oplus v$
$x_2v\oplus x_1$	$x_1v\oplus x_1\oplus x_2$	$x_2v\oplus x_1$	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$
$x_1v\oplus x_2v\oplus x_2$	$x_2v\oplus x_1$	$x_1v\oplus x_2v\oplus x_1\oplus v$	$x_2v\oplus x_2\oplus x_1\oplus 1$
$x_1v\oplus x_2$	$x_1v\oplus x_2v\oplus x_1$	$x_1v\oplus x_2\oplus v\oplus 1$	$x_1v\oplus x_2v\oplus x_1\oplus v\oplus 1$
$x_2v\oplus x_1\oplus x_2$	$x_1v\oplus x_2$	$x_1v\oplus x_2v\oplus x_2\oplus v\oplus 1$	$x_2v\oplus x_1\oplus 1$
$x_1v\oplus x_1\oplus x_2$	$x_1v\oplus x_2v\oplus x_2$	$x_1v\oplus x_2v\oplus x_2$	$x_2v\oplus x_1\oplus v$
$x_2v\oplus x_1\oplus v$	$x_1v\oplus x_1\oplus x_2$	$x_1v\oplus x_2v\oplus x_1$	$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$
$x_1v\oplus x_2v\oplus x_1\oplus v$	$x_2v\oplus x_2\oplus x_1\oplus v$	$x_2v\oplus x_1\oplus v$	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$
$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$	$x_1v\oplus x_2$	$x_1v\oplus x_2\oplus x_1\oplus v\oplus 1$	$x_1v\oplus x_2v\oplus x_2$
$x_1v\oplus x_2v\oplus x_2\oplus v\oplus 1$	$x_2v\oplus x_1\oplus v\oplus 1$	$x_2v\oplus x_2\oplus x_1\oplus v\oplus 1$	$x_1v\oplus x_2\oplus v$
$x_1v\oplus x_2\oplus x_1\oplus 1$	$x_1v\oplus x_2v\oplus x_2\oplus v$	$x_1x_2\oplus x_1v\oplus v$	$x_1v\oplus x_2v\oplus x_2$
$x_1v\oplus x_2\oplus 1$	$x_1v\oplus x_2v\oplus x_1\oplus v\oplus 1$	$x_1x_2\oplus x_1v\oplus x_1\oplus v$	$x_1x_2\oplus x_1v\oplus x_2v\oplus x_2\oplus v$

Утверждение 4.2

Отображение, осуществляемое БУОП $Y = \mathbf{F}_{n/m}(X, V) : \mathrm{GF}(2)^{n+m} \to \mathrm{GF}(2)^n$, где $X, Y \in \mathrm{GF}(2)^n$, $V \in \mathrm{GF}(2)^m$, является биективным относительно X при фиксированном значении V.

Доказательство

Совокупность БФ, реализующих блок $\mathbf{F}_{2/1}$, осуществляет биективное отображение $X \in \mathrm{GF}(2)^2 \to Y \in \mathrm{GF}(2)^2$ при фиксированном значении управляющего вектора $V \in \mathrm{GF}(2)$. Следовательно, n/2 включенных параллельно блоков $\mathbf{F}_{2/1}$ в слое БУОП будут также выполнять биективное отображение вида $X \in \mathrm{GF}(2)^n \to Y \in \mathrm{GF}(2)^n$ при $V \in \mathrm{GF}(2)^{n/2}$, ввиду независимости аргументов БФ для каждого i-го блока $\mathbf{F}_{2/1}$, $i \in \{1, 2, \ldots, n/2\}$. В

свою очередь фиксированные перестановки между каскадами БУОП являются аффинными преобразованиями и не изменяют свойств биективности. Тогда каждый σ_j-й каскад БУОП, $j\in\{1, 2\ldots, k\}$, осуществляет биективное отображение входного вектора в выходной, и при фиксированном значении $V\in GF(2)^m$, где $m = kn/2$, отображение $X\in GF(2)^n \to Y\in GF(2)^n$, выполняемое БУОП, будет биективным.

Для определения нелинейности подстановки следует оценить нелинейности БФ, образующих подстановку. В общем случае для l-разрядной подстановки **S** ее нелинейность ограничена значением, определяемым выражением $\mathrm{NL}(\mathrm{S}) \le 2^{l-1} - 2^{\frac{l}{2}-1}$. Применительно к БУОП $\mathbf{F}_{n/m}$, для которых $l=\mu$, данное выражение преобразуется к виду

$$\mathrm{NL}(\mathbf{F}_{n/m}) \le 2^{2n-2} - 2^{n-\frac{3}{2}}. \qquad (4.4)$$

Например, для подстановки $\mathbf{F}_{4/4}$ максимально возможное значение нелинейности БФ каждого компонента выходного вектора оценивается величиной NL=56, тогда как действительное значение нелинейности для нашего случая согласно (2.26) $\mathrm{NL}(\mathbf{F}_{4/4})=48$.

В качестве дополнительного критерия проектирования БУОП использовалось следующее требование: все линейные комбинации БФ на выходе БУОП должны иметь значение нелинейности не ниже, чем минимальная нелинейность одной из БФ, реализующей компонент $f_i\in\mathcal{P}$ выходного вектора $Y\in GF(2)^n$. Для этого необходимо, чтобы линейная комбинация на выходе базового элемента $\mathbf{F}_{2/1}$ имела нелинейность $\mathrm{NL}(f_1(x_1, x_2, v)\oplus f_2(x_1, x_2, v))=2$. В данном случае линейная комбинация двух функций на выходе любого из n/2 блоков $\mathbf{F}_{2/1}$ заданного слоя будет иметь нелинейность, равную нелинейности одной из образующих БФ, а линейная комбинация БФ, относящихся к различным блокам $\mathbf{F}_{2/1}$, будет иметь нелинейность более высокого порядка, ввиду независимости аргументов БФ для различных блоков $\mathbf{F}_{2/1}$. Представляет интерес построение БУОП на основе УЭ S-типа, являющихся инволюциями. Последние формируются только парами из подмножества нелинейных сбалансированных БФ мощности 8. Данное подмножество представлено в табл. 4.12. Все инволюции $\mathbf{S}_{2/1}$ представлены парами БФ в табл. 4.13.

Таблица 4.12

№	ТИ БФ	АНФ БФ
1	00011110	$x_2x_3\oplus x_1$
2	00101101	$x_2x_3\oplus x_1\oplus x_2$
3	00110110	$x_1x_3\oplus x_2$
4	00111001	$x_1x_3\oplus x_1\oplus x_2$
5	01001011	$x_2x_3\oplus x_1\oplus x_3$

№	ТИ БФ	АНФ БФ
6	01100011	$x_1x_3 \oplus x_2 \oplus x_3$
7	10000111	$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$
8	10010011	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$

Таблица 4.13

$f_1(x_1, x_2, x_3)$	$f_2(x_1, x_2, x_3)$
$x_2x_3 \oplus x_1$	$x_1x_3 \oplus x_1 \oplus x_2$
$x_2x_3 \oplus x_1 \oplus x_2$	$x_1x_3 \oplus x_2$
$x_2x_3 \oplus x_1 \oplus x_3$	$x_1x_3 \oplus x_1 \oplus x_2$
$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	$x_1x_3 \oplus x_2$
$x_2x_3 \oplus x_1 \oplus x_2$	$x_1x_3 \oplus x_2 \oplus x_3$
$x_2x_3 \oplus x_1$	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$
$x_2x_3 \oplus x_1 \oplus x_3$	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$
$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	$x_1x_3 \oplus x_2 \oplus x_3$

Утверждение 4.3

$\forall f_i \in \mathcal{P} : \mathrm{GF}(2)^{\mu} \to \mathrm{GF}(2)$, $i \in \{0, 1, \ldots, n-1\}$, алгебраическая степень $A_S = deg(f)$ (наибольшее количество сомножителей в алгебраической нормальной форме БФ) определяется выражением $A_S = k+1$.

Доказательство

Алгебраическая степень любой БФ на выходе слоя σ_1 равна 2, данные БФ являются аргументами для БФ, реализующих слой σ_2. При этом алгебраическая степень увеличивается на единицу, так как дополнительным сомножителем для БФ-аргументов является только значение управляющего бита $v_i \in \mathrm{GF}(2)$. Методом индукции получаем, что любой из выходов $f_i \in \mathcal{P}$ последнего слоя σ_k является суперпозицией БФ предыдущих $k-1$ слоев, а алгебраическая степень БФ равна $k+1$.

Утверждение 4.4

$\forall f_i \in \mathcal{P} : \mathrm{GF}(2)^{\mu} \to \mathrm{GF}(2)$, $i \in \{0, 1, \ldots, n-1\}$, максимальное значение компонентов спектра УА:

$$\max |\hat{U}(f_i)| = \frac{2^{\mu}}{n} = \frac{2^{2n-1}}{n} = 2^{2^{k+1}-k-1}. \qquad (4.5)$$

Доказательство

На каждом слое БУОП выход каждого элементарного блока $\mathbf{F}_{2/1}$ реализуют сбалансированные БФ, принадлежащие одному классу. Используя данное свойство и представив вход второго слоя в виде кронекеровского произведения БФ f_1, f_2 с единичной строкой $\overline{1}$: $x_1 = f_1 \otimes \overline{1}^{\mathrm{T}}$, $x_2 = \overline{1} \otimes f_2^{\mathrm{T}}$, где количество элементов строки $\overline{1}$ равно количеству элементов таблицы истинности БФ, получим, что вход $\forall$ блока $\mathbf{F}_{2/1}$ на втором слое принимает $\forall$ значения $X=(x_1 \,\|\, x_2) \in \mathrm{GF}(2)^2$ одинаковое количество раз. Тогда, на основе свойства итерационного построения матриц Адамара [32], можно сделать вывод, что увеличение количества переменных БФ в БУОП $\mathbf{F}_{n/m}$ не меняет особенностей спектра УА для компонентных БФ. То есть, если для элементарного блока $\mu = 3$, $n = 2$, $\max|\hat{U}(f_i)| = \dfrac{2^{\mu}}{n}$, $i \in \{0, 1\}$, то данная зависимость сохраняется для всех компонентных БФ, реализующих БУОП $\mathbf{F}_{n/m}$.

Экспериментально проверена истинность выражения (4.5) для многочисленных вариантов БУОП с БФ из табл. 4.11 и входными векторами $X \in \mathrm{GF}(2)^n$, где $n \le 8$. При больших значениях n получение всех компонент спектра УА для БФ требует существенных вычислительных затрат.

Подставив (4.5) в (2.26), получим общее выражение для определения значения нелинейности $\forall f_i \in \mathcal{P} : \mathrm{GF}(2)^{\mu} \to \mathrm{GF}(2)$, $i \in \{0, 1, \ldots, n-1\}$:

$$\mathrm{NL}\left(f_i\right) = 2^{\mu-1} - \frac{2^{\mu-1}}{n} = \frac{2^{2n-2}\left(n-1\right)}{n} = \left(n-1\right)2^{2n-k-2} \ . \qquad (4.6)$$

В качестве положительного свойства БУОП следует отметить, что при увеличении размерности преобразуемых векторов отношение отклонения нелинейностей образующих БУОП БФ от максимальной для данного количества аргументов к максимальной нелинейности асимптотически уменьшается (рис.4.21).

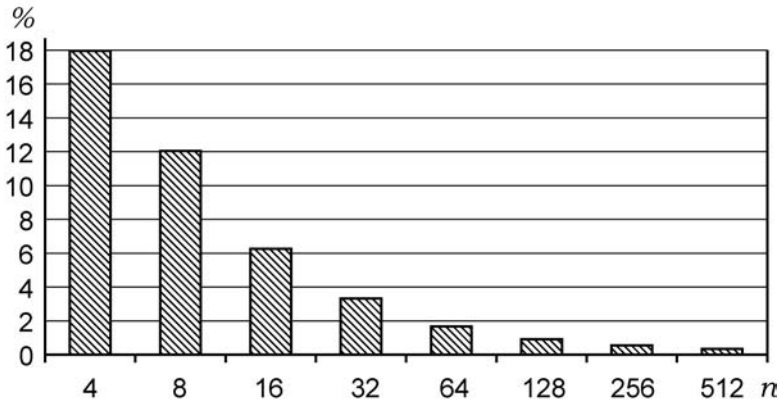

Рис. 4.21.

Произведена оценка корреляционных свойств БФ для БУОП над векторами небольшого размера. БФ является корреляционно-иммунной порядка q, если ее выход

статистически независим от любых q входных переменных. Как показано в п.п. 2.3, это свойство имеет место, если для всех векторов, удовлетворяющих условию $1 \le w \le q$, где w – вес Хэмминга, соответствующие компоненты спектра УА равны 0. Дополнительно, корреляционно эффективные функции – это такие функции, для которых не менее чем на половине векторов веса $1 \le w \le q$ значения компонентов спектра ПУА равны 0.

Исследования показали, что $\forall f_i \in \mathcal{P}$, количество нулевых компонент спектра УА:

$$\#\{\hat{U}(f_i) = 0\} = 2^{2^{k+1}-1} - \prod_{j=1}^{k}\left(3^{2^{k-1}} + 1\right), \qquad (4.7)$$

где k – количество подстановочных слоев в БУОП. Например, для БФ, реализующих блок $\mathbf{F}_{4/4}$, на 88 входных векторах (68% возможных значений) коэффициенты спектра УА имеют значение 0. Для блока $\mathbf{F}_{8/12}$ одна из компонентных БФ на выходе имеет вид:

$$f_1 = x_6x_{11}x_{14}x_{15} \oplus x_3x_{10}x_{15} \oplus x_5x_{14}x_{15} \oplus x_6x_{11}x_{15} \oplus x_7x_{12}x_{13} \oplus x_2x_9 \oplus$$
$$x_3x_{15} \oplus x_4x_{15} \oplus x_5x_{15} \oplus x_7x_{13} \oplus x_8x_{13} \oplus x_1,$$

а количество нулевых значений спектра $\#\{\hat{U}(f_1(X)) = 0\} = 29488$ (90% возможных значений), что соответствует (4.7), т.е. при больших значениях n процент нулевых значений компонентов спектра ПУА асимптотически увеличивается.

Аналогично определялся критерий распространения БФ для БУОП над векторами небольшого размера. В п.п.2.3.4. показано, что БФ удовлетворяет критерию распространения степени l, если ее АКФ, $r_\beta(f)$ определяемая выражением (2.10), равна 0 на всех векторах $\beta \in \mathrm{GF}(2)^n$ веса от 1 до l.

Утверждение 4.5

$\forall f_i \in \mathcal{P} : \mathrm{GF}(2)^\mu \to \mathrm{GF}(2)$, $i \in \{0,1,\ldots,n-1\}$, автокорреляционная функция $r_\beta(f_i) = 0$ на $2^n - 1$ векторах $\beta \in \mathrm{GF}(2)^n$.

Доказательство

Аналогично доказательству утверждения 4.4. Действительно, для всех БФ из табл. 4.8 получено, что их АКФ равна 0 для $2^n - 1$ значений векторов β, где n=3. Тогда ввиду того, что любой элементарный блок $\mathbf{F}_{2/1}$ каждого слоя реализуется идентичными совокупностями БФ и вход любого блока $\mathbf{F}_{2/1}$ на каждом слое принимает значения $X \in \mathrm{GF}(2)^2$ одинаковое количество раз, свойства автокорреляционной функции $\forall f_i \in \mathcal{P}$ аналогичны свойствам АКФ БФ, реализующих блок $\mathbf{F}_{2/1}$. Следовательно, $\forall f_i \in \mathcal{P}$, $r_\beta(f_i) = 0$ на $2^n - 1$ векторах $\beta \in \mathrm{GF}(2)^n$.

Проведение вычислительного эксперимента для оценки критерия распространения различных вариантов БУОП $\mathbf{F}_{4/4}$, $\mathbf{F}_{8/12}$, подтвердило, что значения АКФ БФ, реализующих БУОП, равны нулю для $2^n - 1$ векторов β.

4.5.3. Вероятностные характеристики УОП

При исследовании стойкости алгоритмов шифрования информации к разностному методу анализа используются вероятностные свойства алгоритмов и криптографических примитивов, реализующих данные алгоритмы. В частности, важным показателем стойкости к дифференциальному методу анализа является степень рассеивания (diffusion) криптографического примитива. Поэтому подробнее остановимся на исследовании вероятностных свойств УОП вида $\mathbf{S}_{n/m}$.

Для УОП различной размерности определим вероятности появления разностей с различным весом Хэмминга на выходе при единичной входной разности. В этих целях воспользуемся аппаратом производящих функций вероятности.

Вероятность того, что событие Ω в n- независимых опытах появится ровно m раз, равна коэффициенту при формальной переменной z^m в выражении производящей функции [5]:

$$\varphi_n(z) = \prod_{i=1}^{n} (q_i + p_i z), \tag{4.8}$$

где p_i – вероятность появления события Ω в i-м опыте, q_i=1–p_i – вероятность непоявления события Ω в i-м опыте. Выражение (4.8) равносильно равенству

$$\prod_{i=1}^{n} (q_i + p_i z) = \sum_{m=0}^{n} P_{m,n} z^m , \tag{4.9}$$

где левая и правая части представляют собой одну и ту же производящую функцию $\varphi_n(z)$, только слева она представлена в виде одночлена, а справа – в виде многочлена. Раскрывая скобки в левой части и выполняя приведение подобных членов, получим все вероятности $P_{0,n}, P_{1,n}, \ldots, P_{n,n}$ в виде коэффициентов, соответственно, при нулевой, первой и т.д. степенях формальной переменной z.

Для блока $\mathbf{S}_{2/1}$ получение производящей функции вероятностей появления разности веса $\boldsymbol{wt}(\Delta y)$=2 и $\boldsymbol{wt}(\Delta y)$=1 на выходе при единичной разности $\boldsymbol{wt}(\Delta x)$=1 на входе – тривиально. Если данные блоки соответствуют блокам **S**-типа (см., например, табл. 4.5), тогда с учетом дифференциальных характеристик, приведенных в табл. 4.5., производящая функция будет иметь вид

$$\varphi_2^{\mathbf{S}_{2/1}}(z) = \frac{1}{2} z + \frac{1}{2} z^2 , \tag{4.10}$$

т.е. вероятности появления разностей веса $\boldsymbol{wt}(\Delta y)$=1 и 2 равны $P_{1,2}$=$P_{2,2}$=0.5. Учитывая каскадную структуру УОП большей размерности, итерационно, путем подстановки выражений производящих функций с предыдущего слоя УОП в выражение производящей функции следующего слоя, можно получить производящие функции вероятностей вида $\varphi_n^{\mathbf{S}_{n/m}}(z)$. Так при n=4 $\varphi_4^{\mathbf{S}_{4/4}}(z)=\frac{1}{2}\varphi_2^{\mathbf{S}_{2/1}}(z)+\frac{1}{2}\left(\varphi_2^{\mathbf{S}_{2/1}}(z)\right)^2$ и, подста-

вив в данное выражение (4.10) и выполнив приведение подобных членов, получим выражение производящей функции в виде (4.9)

$$\varphi_4^{\mathbf{S}_{4/4}}(z)= 0.2z+0.4z^2+0.2z^3+0.2z^4, \qquad (4.11)$$

Аналогично можно получить:

$$\varphi_{64}^{\mathbf{S}_{64/192}}(z) = 0.016z+0.031z^2+0.04z^3+0.049z^4+0.054z^5+0.059z^6+0.061z^7+0.062z^8+$$
$$0.062z^9+0.061z^{10}+0.059z^{11}+0.056z^{12}+0.052z^{13}+0.048z^{14}+0.044z^{15}+0.04z^{16}+$$
$$0.035z^{17}+0.031z^{18}+0.027z^{19}+0.023z^{20}+0.019z^{21}+0.016z^{22}+0.013z^{23}+0.011z^{24}+$$
$$0.084z^{25}+0.065z^{26}+0.05z^{27}+0.038z^{28}+0.028z^{29}+0.02z^{30}+1.4\cdot10^{-3}z^{31}+9.8\cdot10^{-4}z^{32}+$$
$$6.6\cdot10^{-4}z^{33}+4.4\cdot10^{-4}z^{33}+2.8\cdot10^{-4}z^{35}+1.8\cdot10^{-4}z^{36}+1.1\cdot10^{-4}z^{37}+6.6\cdot10^{-5}z^{38}+3.8\cdot10^{-5}z^{39}+$$
$$2.1\cdot10^{-5}z^{40}+1.2\cdot10^{-5}z^{41}+6.2\cdot10^{-6}z^{42}+3.1\cdot10^{-6}z^{43}+1.5\cdot10^{-6}z^{44}+7.2\cdot10^{-7}z^{45}+3.3\cdot10^{-7}z^{46}+$$
$$1.4\cdot10^{-7}z^{47}+5.9\cdot10^{-8}z^{48}+2.3\cdot10^{-8}z^{49}+8.6\cdot10^{-9}z^{50}+3\cdot10^{-9}z^{51}+10^{-9}z^{52}+3.1\cdot10^{-10}z^{53}+$$
$$8.8\cdot10^{-11}z^{54}+2.3\cdot10^{-11}z^{55}+5.6\cdot10^{-12}z^{56}+1.2\cdot10^{-12}z^{57}+2.3\cdot10^{-13}z^{58}+3.9\cdot10^{-14}z^{59}+$$
$$5.6\cdot10^{-15}z^{60}+6.5\cdot10^{-16}z^{61}+5.7\cdot10^{-17}z^{62}+3.5\cdot10^{-18}z^{63}+1.1\cdot10^{-19}z^{64}. \qquad (4.12)$$

На рис. 4.22, 4.23 представлены дискретные распределения вероятностей и дискретные функции распределения вероятностей, соответственно, полученные на основе использования производящих функций вероятностей и имитационного моделирования для УОП $\mathbf{S}_{64/192}$ в случае, когда все биты управляющего вектора являются независимыми и равновероятными, а на вход УОП подается разность Δx, вес которой $\boldsymbol{wt}(\Delta x) = 1$.

Рис. 4.22.

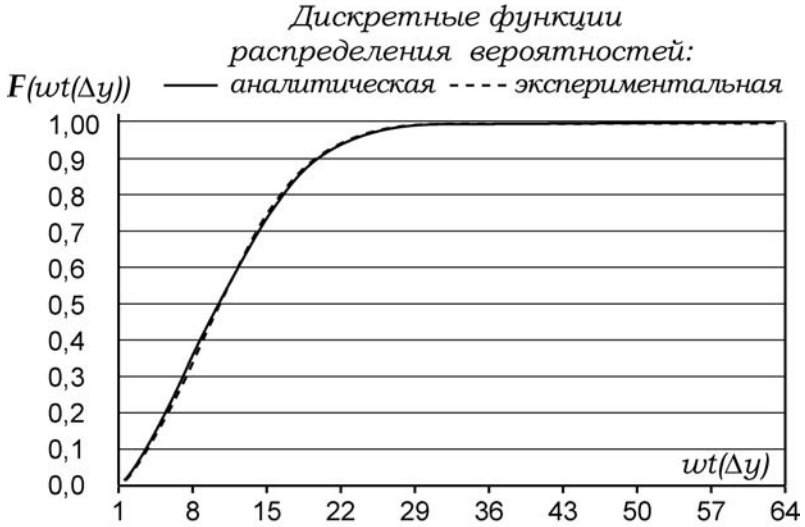

Рис. 4.23.

В приведенных таблицах (4.14 – 4.18) для различных размерностей блоков **S** представлены распределения вероятностей появления разности веса ***wt*** на выходе $\mathbf{S}_{n/m}$ в случае, когда на вход подается разность веса ***wt***(Δx)=1.

Таблица 4.14. $\mathbf{S}_{4/4}$

wt	***p***
1	0.25
2	0.375
3	0.25
4	0.125

Таблица 4.15. $\mathbf{S}_{8/12}$

wt	***p***	***wt***	***p***
1	0.125	5	0.125
2	0.219	6	0.078
3	0.219	7	0.038
4	0.195	8	0.008

Таблица 4.16. $\mathbf{S}_{16/32}$

wt	***p***	***wt***	***p***	***wt***	***p***	***wt***	***p***
1	0.063	5	0.13	9	0.049	13	$3.4\ 10^{-3}$
2	0.12	6	0.12	10	0.032	14	$1.1\ 10^{-3}$
3	0.14	7	0.095	11	0.018	15	$2.4\ 10^{-4}$
4	0.15	8	0.071	12	$8.5\ 10^{-3}$	16	$3.1\ 10^{-5}$

Таблица 4.17. $S_{32/80}$

wt	***p***	***wt***	***p***	***wt***	***p***	***wt***	***p***
1	0.031	9	0.077	17	0.011	25	$5.8\ 10^{-5}$
2	0.061	10	0.067	18	$6.8\ 10^{-3}$	26	$2.0\ 10^{-5}$
3	0.076	11	0.057	19	$4.2\ 10^{-3}$	27	$6.4\ 10^{-6}$
4	0.090	12	0.041	20	$2.5\ 10^{-5}$	28	$1.7\ 10^{-6}$
5	0.093	13	0.037	21	$1.4\ 10^{-3}$	29	$3.7\ 10^{-6}$
6	0.096	14	0.029	22	$7.0\ 10^{-4}$	30	$6.3\ 10^{-8}$
7	0.091	15	0.021	23	$3.3\ 10^{-4}$	31	$7.5\ 10^{-9}$
8	0.085	16	0.015	24	$1.5\ 10^{-4}$	32	$4.7\ 10^{-10}$

Таблица 4.18. $S_{64/192}$

wt	***p***	***wt***	***p***	***wt***	***p***	***wt***	***p***
1	0.016	17	0.035	33	$6.6\ 10^{-4}$	49	$2.3\ 10^{-8}$
2	0.031	18	0.031	34	$4.4\ 10^{-4}$	50	$8.6\ 10^{-9}$
3	0.040	19	0.027	35	$2.8\ 10^{-4}$	51	$3.0\ 10^{-9}$
4	0.049	20	0.023	36	$1.8\ 10^{-4}$	52	10^{-9}
5	0.054	21	0.019	37	$1.1\ 10^{-4}$	53	$3.1\ 10^{-10}$
6	0.059	22	0.016	38	$6.6\ 10^{-5}$	54	$8.8\ 10^{-11}$
7	0.061	23	0.013	39	$3.8\ 10^{-5}$	55	$2.3\ 10^{-11}$
8	0.062	24	0.011	40	$2.1\ 10^{-5}$	56	$5.6\ 10^{-12}$
9	0.062	25	0.084	41	$1.2\ 10^{-5}$	57	$1.2\ 10^{-12}$
10	0.061	26	0.065	42	$6.2\ 10^{-6}$	58	$2.3\ 10^{-13}$
11	0.059	27	0.05	43	$3.1\ 10^{-6}$	59	$3.9\ 10^{-14}$
12	0.056	28	0.038	44	$1.5\ 10^{-6}$	60	$5.6\ 10^{-15}$
13	0.052	29	0.028	45	$7.2\ 10^{-7}$	61	$6.5\ 10^{-16}$
14	0.048	30	0.02	46	$3.3\ 10^{-7}$	62	$5.7\ 10^{-17}$
15	0.044	31	$1.4\ 10^{-3}$	47	$1.4\ 10^{-7}$	63	$3.5\ 10^{-18}$
16	0.040	32	$9.8\ 10^{-4}$	48	$5.9\ 10^{-8}$	64	$1.1\ 10^{-19}$

Подобные характеристики можно получить и для УОП, относящихся к **R**-типу. Для этого найдем выражение производящей функции вероятностей

$$\varphi_2^{\mathbf{R}_{2/1}}(z) = \frac{3}{4}z + \frac{1}{4}z^2, \qquad (4.13)$$

т.е. вероятность появления разностей веса $\boldsymbol{wt}(\Delta y)=1$ $P_{1,2}=0.75$, а $\boldsymbol{wt}(\Delta y)=2$ $P_{2,2}=0.25$. На основе использования выражения (4.13), итерационно, получены распределения

вероятностей появления разностей различного веса на выходе R-блоков различной размерности при подаче на вход блока разности веса $wt(\Delta x)=1$ (табл. 4.19–4.23).

Таблица 4.19. $R_{4/4}$

wt	*p*
1	0.56
2	0.33
3	0.094
4	0.016

Таблица 4.20. $R_{8/12}$

wt	*p*	*wt*	*p*
1	0.42	5	0.02
2	0.33	6	$4.8\ 10^{-3}$
3	0.16	7	$7.3\ 10^{-4}$
4	0.065	8	$6.1\ 10^{-5}$

Таблица 4.21. $R_{16/32}$

wt	*p*	*wt*	*p*	*wt*	*p*	*wt*	*p*
1	0.32	5	0.055	9	$1.2\ 10^{-3}$	13	$2.3\ 10^{-6}$
2	0.29	6	0.025	10	$3.2\ 10^{-4}$	14	$2.8\ 10^{-7}$
3	0.19	7	0.01	11	$7.6\ 10^{-5}$	15	$2.2\ 10^{-8}$
4	0.11	8	$3.6\ 10^{-3}$	12	$1.5\ 10^{-5}$	16	$9.3\ 10^{-10}$

Таблица 4.22. $R_{32/80}$

wt	*p*	*wt*	*p*	*wt*	*p*	*wt*	*p*
1	0.24	9	$8.3\ 10^{-3}$	17	$5.2\ 10^{-6}$	25	$3.2\ 10^{-11}$
2	0.24	10	$4.0\ 10^{-3}$	18	$1.6\ 10^{-6}$	26	$4.5\ 10^{-12}$
3	0.19	11	$1.9\ 10^{-3}$	19	$4.4\ 10^{-7}$	27	$5.3\ 10^{-13}$
4	0.13	12	$8.1\ 10^{-4}$	20	$1.1\ 10^{-7}$	28	$5.3\ 10^{-14}$
5	0.086	13	$3.3\ 10^{-4}$	21	$2.1\ 10^{-8}$	29	$4.2\ 10^{-15}$
6	0.052	14	$1.3\ 10^{-4}$	22	$5.8\ 10^{-9}$	30	$2.6\ 10^{-16}$
7	0.03	15	$4.8\ 10^{-5}$	23	$1.2\ 10^{-9}$	31	$1.0\ 10^{-17}$
8	0.016	16	$1.6\ 10^{-5}$	24	$2.0\ 10^{-10}$	32	$2.2\ 10^{-19}$

Таблица 4.23. $R_{64/192}$

wt	*p*	*wt*	*p*	*wt*	*p*	*wt*	*p*
1	0.18	17	$2.6\ 10^{-4}$	33	$1.5\ 10^{-10}$	49	$8.4\ 10^{-21}$
2	0.2	18	$1.3\ 10^{-4}$	34	$4.6\ 10^{-11}$	50	$1.2\ 10^{-21}$
3	0.17	19	$6.4\ 10^{-5}$	35	$1.4\ 10^{-11}$	51	$1.7\ 10^{-22}$
4	0.14	20	$3.1\ 10^{-5}$	36	$4.0\ 10^{-12}$	52	$2.2\ 10^{-23}$
5	0.1	21	$1.4\ 10^{-5}$	37	$1.1\ 10^{-12}$	53	$2.6\ 10^{-24}$
6	0.074	22	$6.5\ 10^{-6}$	38	$3.0\ 10^{-13}$	54	$2.9\ 10^{-25}$
7	0.052	23	$2.8\ 10^{-6}$	39	$7.8\ 10^{-14}$	55	$2.9\ 10^{-26}$
8	0.034	24	$1.2\ 10^{-6}$	40	$2.0\ 10^{-14}$	56	$2.6\ 10^{-27}$
9	0.022	25	$5.0\ 10^{-7}$	41	$4.6\ 10^{-15}$	57	$2.1\ 10^{-28}$
10	0.014	26	$2.0\ 10^{-7}$	42	$1.1\ 10^{-15}$	58	$1.4\ 10^{-29}$
11	$8.6\ 10^{-3}$	27	$8.0\ 10^{-8}$	43	$2.3\ 10^{-16}$	59	$8.7\ 10^{-31}$
12	$5.1\ 10^{-3}$	28	$3.0\ 10^{-8}$	44	$4.8\ 10^{-17}$	60	$4.4\ 10^{-32}$
13	$3.0\ 10^{-3}$	29	$1.1\ 10^{-8}$	45	$9.5\ 10^{-18}$	61	$1.8\ 10^{-33}$
14	$1.7\ 10^{-3}$	30	$4.0\ 10^{-9}$	46	$1.8\ 10^{-18}$	62	$5.5\ 10^{-35}$
15	$9.2\ 10^{-4}$	31	$1.4\ 10^{-9}$	47	$3.2\ 10^{-19}$	63	$1.1\ 10^{-36}$
16	$4.9\ 10^{-4}$	32	$4.6\ 10^{-10}$	48	$5.3\ 10^{-20}$	64	$1.2\ 10^{-38}$

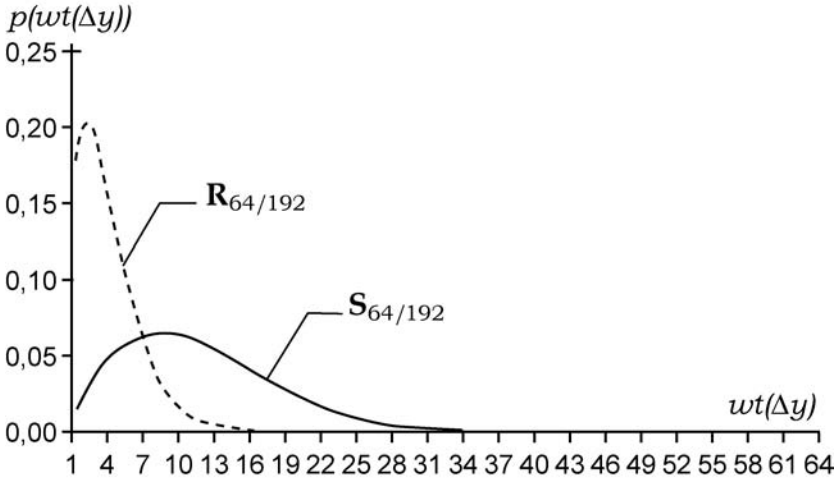

Рис. 4.24.

В качестве сравнения на рис. 4.24 представлены распределения вероятностей появления разностей различного веса на выходе УОП S- и R-типов при подаче на вход

блока разности веса $\boldsymbol{wt}(\Delta x)=1$. Анализ полученных гистограмм позволяет сделать вывод о предпочтительности использования УОП S-типа при проектировании блочных криптоалгоритмов, ввиду того что они имеют лучшие разностные свойства.

4.5.4. Оценка схемотехнической сложности реализации УОП

В отличие от табличных подстановок с длиной блока входных данных n ($\log_2 n \geq 5$) различные типы УП и УОП легко реализуются в виде несложных микроэлектронных схем с временем задержки, сравнимым с временем последовательного выполнения нескольких операций типа XOR.

Синтезированные БУОП имеют низкую сложность схемотехнической реализации $C_\Omega(\mathbf{F}_{n/m})$, под которой понимается количество вентилей, реализующих схему $\mathbf{F}_{n/m}$. Здесь $\Omega=\{\&, \vee, \bar{}\}$ – используемый полный базис простейших логических операций. Одноразрядные БФ в элементарном блоке $\mathbf{F}_{2/1}$ в данном базисе имеют сложность $C_\Omega(\mathbf{F}_{2/1})=9$. При оценке скоростных параметров БУОП рассмотрим время выполнения самой медленной базовой операции, именуемое тактом задержки τ_Ω. В приведенном полном базисе $\tau_\Omega=\tau_\&$. Для разработанных БУОП время выполнения преобразования $t(\mathbf{F}_{n/m})=k\tau_\&$, а сложность реализации $C_\Omega(\mathbf{F}_{n/m})=mC_\Omega(\mathbf{F}_{2/1})=6m$, что на основе применения современной микроэлектронной технологии позволяет изготовить крипточипы, реализующие блочные криптоалгоритмы, включающие БУОП с большим размером входа (n=32, 64, 128, 256). Благодаря этому можно достичь скорости шифрования информации существенно выше одного Гбит/с.

Несмотря на изначальную ориентацию на аппаратную реализацию, применение эффективных управляемых операций потенциально может привести к существенному скачку в росте скорости программных шифров. Это связано с тем, что некоторые типы управляемых операций, например УП, чрезвычайно эффективны как криптографический примитив и обладают очень низкой стоимостью схемотехнической реализации. Такое соотношение эффективности и стоимости реализации делает привлекательным для производителей массовых процессоров включение новых команд в состав стандартных команд процессора. Возможность обеспечения скорости программного шифрования до 800 – 2000 Мбит/с существенно повышает конкурентоспособность таких процессоров при минимальных схемотехнических затратах. Особый интерес представляет внедрение команды управляемой битовой перестановки, практическое значение которой выходит далеко за рамки криптографических применений (алгоритмы шифрования и хэширования данных). Варианты реализации такой команды рассмотрены в разделе 7.

ГЛАВА 5

Класс управляемых элементов $F_{2/2}$

5.1. Варианты представления и критерии отбора управляемых элементов $F_{2/2}$

В *главе 4* были проанализированы и исследованы УППС (УОП), использующие в качестве базового элемента управляемый элемент $\mathbf{F}_{2/1}$. Однако представляет интерес обоснование и применение в качестве элементарных УЭ для построения эффективных УППС также и других блоков, например, элементарных УЭ $\mathbf{F}_{2/2}$, схемное представление которых показано на рис. 5.1. Они также могут быть легко реализованы в виде быстродействующих электронных схем. В этом случае, сохраняя общую топологию УППС, можно осуществить замену всех элементарных управляемых блоков $\mathbf{F}_{2/1}$ на другой типовой УЭ, обладающий 2-битовыми входом и выходом и 2-битовым управляющим входом.

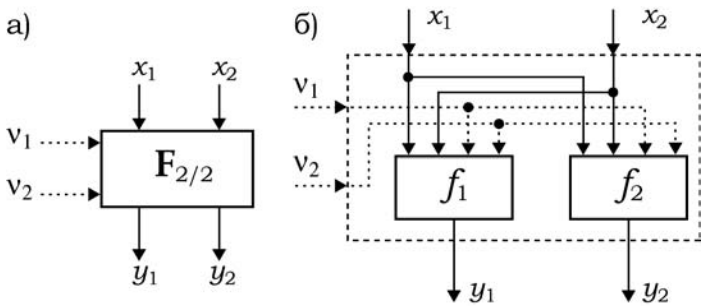

Рис. 5.1. Управляемый элемент $\mathbf{F}_{2/2}$ (а) и его реализация двумя булевыми функциями (б)

Практическая целесообразность перехода к УЭ с двухбитовым управляющим входом связана с их аппаратной реализацией на основе программируемых логических матриц, в которых типовые логические блоки обычно содержат две логические ячейки, представляющие собой шестнадцатибитовые элементы памяти (ячейки). Каждый такой элемент памяти позволяет реализовать произвольную булеву функцию от четырех переменных. При реализации любого УЭ типа $\mathbf{F}_{2/1}$ используются две ячейки, каждая из которых реализует ту или иную БФ от трех переменных. Это означает, что ресурс ячейки используется только наполовину, поскольку для реализации булевой функции от трех переменных достаточно использовать восьмибитовую па-

мять, т. е. 50% от имеющейся емкости ячейки. Оставшуюся часть памяти использовать эффективно не представляется возможным.

Переход к УЭ типа $\mathbf{F}_{2/2}$ обеспечивает полное использование потенциала ячеек, а кроме того, использование расширенного управляющего входа создает предпосылки для задания более существенного влияния управляющего подблока данных на преобразуемый. В частности, это может выражаться в повышении значения нелинейности каждого выхода УЭ, увеличении его алгебраической степени нелинейности и усилении лавинного эффекта при изменении одиночных битов управляющего подблока данных. Разумно предположить, что при соответствующем выборе УЭ типа $\mathbf{F}_{2/2}$ можно обеспечить существенное повышение криптографических показателей синтезируемого операционного блока. Это даст возможность сократить число раундов преобразования при сохранении высокой криптостойкости алгоритмов на основе управляемых операций, что обеспечивает снижение сложности аппаратной реализации (при конвейерной архитектуре) и увеличение скорости шифрования (при итеративной архитектуре реализации).

Для синтеза эффективных многослойных УППС требуется сформулировать некоторые критерии выбора конкретных вариантов УЭ вида $\mathbf{F}_{2/2}$.

В общем случае УЭ $\mathbf{F}_{2/2}$ может быть представлен в виде:

- двух булевых функций от четырех переменных;
- четырех подстановок размера 2×2, каждая из которых выполняется над входным 2-битовым двоичным вектором (x_1, x_2) при $v = (0, 0), (0, 1), (1, 0), (1, 1)$ соответственно.

Булевы функции от четырех переменных имеют различные значения нелинейности в смысле минимального расстояния до множества аффинных булевых функций от четырех переменных, различные значения алгебраической степени нелинейности и большее разнообразие дифференциальных характеристик по сравнению с булевыми функциями от трех переменных. Эти обстоятельства требуется учесть при формулировании критериев отбора УЭ типа $\mathbf{F}_{2/2}$ и их классификации.

Критерии построения блоков $\mathbf{F}_{2/2}$

На основании изложенного и результатов, полученных в *главе 4*, сформулируем следующие базовые критерии проектирования (отбора) УЭ $\mathbf{F}_{2/2}$:

1. Любой из двух выходов блока $\mathbf{F}_{2/2}$ должен представлять собой нелинейную БФ от четырех переменных: $y_1 = f_1(x_1, x_2, v_1, v_2)$ и $y_2 = f_2(x_1, x_2, v_1, v_2)$, каждая из которых удовлетворяет показателю нелинейности, близкой к максимальной.
2. Каждая из четырех элементарных модификаций блока $\mathbf{F}_{2/2}$, а именно $\mathbf{F}^{(0)}, \mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \mathbf{F}^{(3)}$, должна осуществлять биективное преобразование $(x_1, x_2) \to (y_1, y_2)$.
3. Каждая из четырех модификаций УЭ $\mathbf{F}^{(v)}$ должна быть инволюцией.

Можно использовать два варианта поиска эффективных УЭ $\mathbf{F}_{2/2}$, удовлетворяющих заданным критериям:

- перебор всех возможных пар булевых функций $y_1 = f_1(x_1, x_2, v_1, v_2)$ и $y_2 = f_2(x_1, x_2, v_1, v_2)$;
- перебор всех возможных наборов модификаций $\mathbf{F}^{(0)}$, $\mathbf{F}^{(1)}$, $\mathbf{F}^{(2)}$, $\mathbf{F}^{(3)}$, осуществляющих преобразование 2×2.

Для первого варианта объем вычислений, требуемых для выбора элементарных УЭ $\mathbf{F}_{2/2}$, удовлетворяющих заданным критериям, достаточно велик. Очевидно, что существует 2^{16} различных БФ от четырех переменных, следовательно, в общем случае необходимо перебрать $2^{16}\cdot(2^{16}-1) \approx 4{,}3\cdot10^9$ совокупностей различных БФ. Для ограничения количества вариантов перебора воспользуемся требованием сбалансированности БФ, которое является следствием второго критерия выбора. Количество сбалансированных БФ от n переменных $\#\{f(x_1, ..., x_n)\}_{Bal} = \begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix}$, следовательно, для n=4 $\#\{f(x_1, ..., x_4)\}_{Bal} = 12870$. В табл. 5.1 представлены данные о множестве всех БФ и сбалансированных БФ от четырех переменных с различными показателями нелинейности.

Учитывая первый критерий, требующий соответствия БФ показателю максимального значения нелинейности, и выражение (2.28), определяющее это значение для сбалансированных функций: $\mathrm{NL}(f) \le 2^{n-1} - 2^{\frac{n}{2}-1} - 2$, получим 10920 БФ, следовательно в этом случае необходимо перебрать $\approx 1{,}2\cdot10^8$ совокупностей БФ, что также требует больших вычислительных затрат.

Таблица 5.1

Множество БФ $f(x_1, x_2, x_3, x_4) : \mathrm{GF}(2)^4 \to \mathrm{GF}(2)$

Значение нелинейности БФ NL(*f*)	6	5	4	3	2	1	0
Количество БФ	896	14336	28000	17920	3840	512	32
Количество сбалансированных БФ	0	0	10920	0	1920	0	30

Несущественно сократить объем вычислений можно, если учесть требование сбалансированности линейной комбинации БФ, реализующих УЭ $\mathbf{F}_{2/2}$ (что также является следствием второго критерия). Очевидно, что распределение весов Хэмминга линейной комбинации БФ будет определяться выражением

$$\#\{\mathbf{F}_{2/m} : wt(f_1 \oplus f_2) = l\} = \#\{\mathbf{F}_{2/m} : wt(f_1 \oplus f_2) = 2^h - l\} =$$

$$\#\{f\}_{Bal}\begin{pmatrix} 2^{h-1} \\ 2^{h-1} - \dfrac{l}{2} \end{pmatrix}^2 = \begin{pmatrix} 2^h \\ 2^{h-1} \end{pmatrix}\begin{pmatrix} 2^{h-1} \\ 2^{h-1} - \dfrac{l}{2} \end{pmatrix}^2,$$

где $h = 2+m$; $m = 1,\ 2$; $l \in \{0, 2, \ldots, 2^h-2, 2^h\}$. Откуда для случая $h = 3$ получим $\#\{\mathbf{F}_{2/1} : wt(f_1 \oplus f_2) = 4\} = 2520$ сбалансированных линейных комбинаций БФ на выходе элемента $\mathbf{F}_{2/1}$, а для случая $h = 4$ $\#\{\mathbf{F}_{2/2} : wt(f_1 \oplus f_2) = 8\} = 63063000$ сбалансированных линейных комбинаций, определяющих количество вариантов перебора.

***Визуальное проектирование блоков* $\mathbf{F}_{2/2}$**

При использовании второго варианта построения эффективных УЭ $\mathbf{F}_{2/2}$ выбор пар БФ можно свести к формальному выбору УЭ с заданными свойствами. В этом случае блок $\mathbf{F}_{2/2}$ формируется четырьмя различными модификациями, зависящими от управляющего вектора $v=(v_1, v_2)$, $v_1, v_2 \in GF(2)$, и осуществляющими отображение $(x_1, x_2) \to (y_1, y_2)$:

- $\mathbf{F}^{(0)}$, если $v=(0, 0)$;
- $\mathbf{F}^{(1)}$, если $v=(0, 1)$;
- $\mathbf{F}^{(2)}$, если $v=(1, 0)$;
- $\mathbf{F}^{(3)}$, если $v=(1, 1)$.

Для осуществления биективного преобразования в целом, преобразование, выполняемое модификациями $\mathbf{F}^{(0)} \div \mathbf{F}^{(3)}$, должно быть биективным. Существует 24 варианта биективных модификаций (см. рис. 4.13а.). Визуальное проектирование сводится к выбору четверок модификаций из множества биективных вариантов. Следует заметить, что любая элементарная модификация осуществляет линейное преобразование. Нелинейные свойства блоков $\mathbf{F}_{2/2}$ определяются спецификой выбора элементарных модификаций в зависимости от значения $v = (v_1, v_2)$. Из всего множества 24^4=331776 вариантов совокупностей четверок подстановочных преобразований 2×2, осуществляющих биективное отображение, только 126720 определяют нелинейные $\mathbf{F}_{2/2}$ блоки, удовлетворяющие критериям 1-2. Учитывая третий критерий отбора, получим 10^4 вариантов построения УЭ $\mathbf{F}_{2/2}$, удовлетворяющих критериям 1–3. При этом важным является то, что такое значительное сокращение числа вариантов перебора обеспечивается на начальном этапе. В первом варианте отбора третий критерий не работает столь эффективно.

По заданным модификациям $\mathbf{F}^{(0)} \div \mathbf{F}^{(3)}$ легко получить алгебраическую нормальную форму БФ f_1 и f_2, реализующих УЭ $\mathbf{F}_{2/2}$. Пусть $\{f^1_1(x_1, x_2), f^1_2(x_1, x_2)\}$ – БФ, реализующие модификацию $\mathbf{F}^{(0)}$ при значении $v=(0, 0)$, а $\{f^2_1(x_1, x_2), f^2_2(x_1, x_2)\}$, $\{f^3_1(x_1, x_2), f^3_2(x_1, x_2)\}$, $\{f^4_1(x_1, x_2), f^4_2(x_1, x_2)\}$, – булевы функции, реализующие мо-

дификации $\mathbf{F}^{(1)}$, $\mathbf{F}^{(2)}$, $\mathbf{F}^{(3)}$ при значении $v = (0, 1)$, $v = (1, 0)$, $v = (1, 1)$ соответственно. Тогда конкретный вид двух БФ, реализующих блок $\mathbf{F}_{2/2}$, можно получить следующим образом:

$$y_1 = (v_1 \oplus 1)(v_2 \oplus 1) f_1^1(x_1,\ x_2) \oplus (v_1 \oplus 1) v_2 f_1^2(x_1,\ x_2) \oplus v_1 (v_2 \oplus 1) f_1^3(x_1,\ x_2) \oplus v_1 v_2 f_1^4(x_1,\ x_2),$$

$$y_2 = (v_1 \oplus 1)(v_2 \oplus 1) f_2^1(x_1,\ x_2) \oplus (v_1 \oplus 1) v_2 f_2^2(x_1,\ x_2) \oplus v_1 (v_2 \oplus 1) f_2^3(x_1,\ x_2) \oplus v_1 v_2 f_2^4(x_1,\ x_2).$$

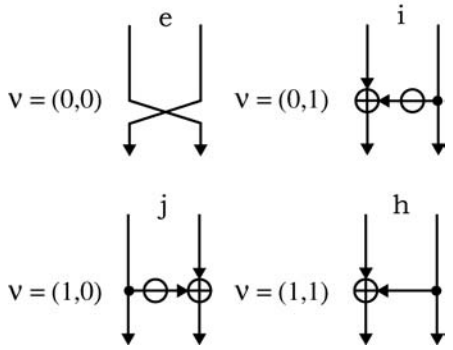

Рис. 5.2. Вариант построения блока $\mathbf{F}_{2/2}$

Например, для элементарного блока $\mathbf{F}_{2/2}$, представленного на рис. 5.2, имеем

$$f_1^1(x_1, x_2) = x_2; \quad f_2^1(x_1, x_2) = x_1;$$

$$f_1^2(x_1, x_2) = x_1 \oplus x_2 \oplus 1; \quad f_2^2(x_1, x_2) = x_2;$$

$$f_1^3(x_1, x_2) = x_1; \quad f_2^3(x_1, x_2) = x_1 \oplus x_2 \oplus 1;$$

$$f_1^4(x_1, x_2) = x_1 \oplus x_2; \quad f_2^4(x_1, x_2) = x_2.$$

Следовательно,

$$y_1 = v_1 v_2 x_1 \oplus v_1 v_2 x_2 \oplus v_1 x_1 \oplus v_1 x_2 \oplus v_1 v_2 \oplus v_2 x_1 \oplus v_2 \oplus x_2;$$

$$y_2 = v_1 v_2 x_2 \oplus v_1 x_2 \oplus v_2 x_1 \oplus v_2 x_2 \oplus v_1 v_2 \oplus v_1 \oplus x_1;$$

$$y_1 \oplus y_2 = v_1 v_2 x_1 \oplus v_1 x_1 \oplus v_2 x_2 \oplus v_2 \oplus v_1 \oplus x_2 \oplus x_1.$$

5.2. Классификация основных типов УЭ $F_{2/2}$ по нелинейным и дифференциальным свойствам

Представляет интерес исследование элементарных блоков $\mathbf{F}_{2/2}$ на предмет соответствия реализующих его БФ свойствам нелинейности, а преобразования в целом – дифференциальным свойствам. Сбалансированные БФ от четырех переменных имеют следующие значения нелинейности (см. табл. 5.1): $\mathrm{NL}(f) \in \{0, 2, 4)$. Исходя из этого, целесообразно классифицировать блоки $\mathbf{F}_{2/2}$ по трем параметрам:

- нелинейности БФ f_1;
- нелинейности БФ f_2;
- нелинейности линейной комбинации БФ $f_3 = f_1 \oplus f_2$.

Результаты исследования нелинейных свойств элементарных УЭ вида $\mathbf{F}_{2/2}$, осуществляющих биективное преобразование, представлены в табл. 5.2. Здесь обозначение типа блока, например, 4–2–2 означает, что одна из трех БФ (включая линейную комбинацию первых двух) имеет нелинейность $\mathrm{NL}(f) = 4$, а две другие БФ имеют нелинейность $\mathrm{NL}(f) = 2$.

Таблица 5.2

Тип УЭ (значения нелинейностей)	4–4–4	4–4–2	4–2–2	4–4–0	2–2–2	2–2–0	0–0–0
Количество УЭ	126720	110592	64512	8064	12288	9216	384

Аналогичные результаты представлены в табл. 5.3 для УЭ $\mathbf{F}_{2/2}$, которые осуществляют преобразование – инволюцию.

Таблица 5.3

Тип УЭ (значения нелинейностей)	4–4–4	4–4–2	4–2–2	4–4–0	2–2–2	2–2–0	0–0–0
Количество УЭ	2208	2304	3264	792	384	960	88

Таким образом, мы получили 2208 вариантов различных УЭ, реализующих преобразование-инволюцию и удовлетворяющих показателю максимально возможной нелинейности образующих БФ.

Рассмотрим первое подмножество УЭ $\mathbf{F}_{2/2}$ из табл. 5.3, которое характеризуется тем, что все модификации, реализуемые УЭ, удовлетворяющим максимально возможным значениям нелинейности, являются элементарными инволюциями. Возьмем, например, два набора модификаций: b/e/f/d и e/i/f/j (см. рис. 5.3а, 5.3б).

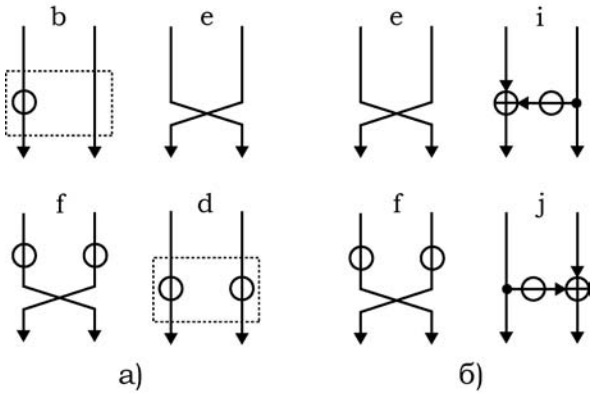

Рис. 5.3.

В первом случае элементарный блок $\mathbf{F}_{2/2}$ реализуется булевыми функциями:

$$y_1 = v_1x_1 \oplus v_1x_2 \oplus v_2x_1 \oplus v_2x_2 \oplus v_1v_2 \oplus v_2 \oplus x_1 \oplus 1;$$

$$y_2 = v_1x_1 \oplus v_1x_2 \oplus v_2x_1 \oplus v_2x_2 \oplus v_1 \oplus x_2;$$

$$y_1 \oplus y_2 = v_1v_2 \oplus v_1 \oplus v_2 \oplus x_2 \oplus x_1 \oplus 1.$$

Весовой спектр УА данных БФ имеет вид:

$$y_1 : \{0, 0, 0, 0, 0, –8, 8, 0, –8, 0, 0, –8, 0, 0, 0, 0\};$$

$$y_2 : \{0, 0, 0, 0, 0, 8, 8, 0, 0, –8, 8, 0, 0, 0, 0, 0\};$$

$$y_1 \oplus y_2 : \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, –8, 8, 8, 8\}.$$

Во втором случае элементарный блок $\mathbf{F}_{2/2}$ реализуется булевыми функциями:

$$y_1 = v_1v_2x_2 \oplus v_2x_1 \oplus v_2 \oplus v_1 \oplus x_2;$$

$$y_2 = v_1v_2x_1 \oplus v_2x_1 \oplus v_2x_2 \oplus v_1 \oplus x_1;$$

$$y_1 \oplus y_2 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_2x_2 \oplus v_2 \oplus x_2 \oplus x_1.$$

Весовой спектр УА данных БФ имеет вид:

$$y_1 : \{0, 0, 0, 0, 0, 0, 8, 8, 4, –4, –4, 4, –4, 4, –4, 4\};$$

$$y_2 : \{0, 0, 0, 0, 4, –4, 4, –4, 0, 0, 8, 8, –4, 4, 4, –4\};$$

$$y_1 \oplus y_2 : \{0, 0, 0, 0, –4, 4, 4, –4, –4, 4, –4, 4, 8, 8, 0, 0\}.$$

И в том и в другом случае нелинейность всех функций равна 4. Однако разностные свойства двух этих блоков существенно отличаются, что объясняется различием в компонентах весового спектра УА и различной степенью алгебраической нелинейности функций.

Наряду с нелинейностью другим показателем качества криптографического примитива являются его дифференциальные характеристики. Тем более что дифференциальные характеристики типового УЭ определяют для заданной топологии УППС ее дифференциальные характеристики.

Здесь, как и в случае УЭ $\mathbf{F}_{2/1}$, не представляется возможным описать все дифференциальные характеристики для различных вариантов УППС, построенных для каждого из найденных УЭ $\mathbf{F}_{2/2}$. Однако это можно сделать для отдельных УЭ $\mathbf{F}_{2/2}$ и использовать эти данные при проведении элементов дифференциального криптоанализа шифров, основанных на УППС, построенных с использованием УЭ данного типа.

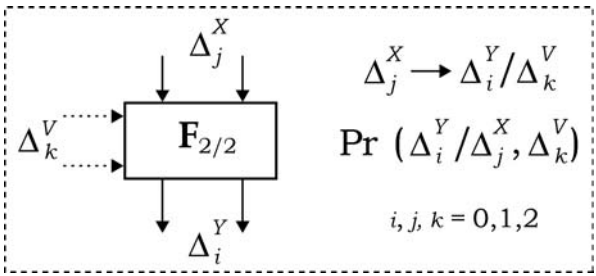

Рис. 5.4. Возможные варианты разностей для УЭ $\mathbf{F}_{2/2}$

На рис. 5.4 представлены варианты всех возможных разностей, относящихся к УЭ вида $\mathbf{F}_{2/2}$. Дифференциальная характеристика УЭ может быть представлена как четверка параметров $(\Delta^X_j, \Delta^V_k, \Delta^Y_i, p')$ или в виде равенства $p(\Delta^Y_i/\Delta^X_j, \Delta^V_k) = p'$, где p' – значение вероятности дифференциальной характеристики. Классификация УЭ по всем вариантам возможных разностей является достаточно трудоемкой. Поэтому в качестве первого этапа произведена классификация дифференциальных характеристик при фиксированном весе разности на управляющем входе, т. е. по значениям $p(\Delta^Y_i/\Delta^X_j, \Delta^V_k)$ при k = const. Результаты такой классификации, представляющей интерес с практической точки зрения, приведены в табл. 5.4-5.6. Полученные результаты показывают, что все множество УЭ $\mathbf{F}_{2/2}$ по своим дифференциальным свойствам делится на пять подмножеств: A, B, C, D, E, из которых подмножества A, B, D выделяются меньшей предсказуемостью значения выходной разности, что делает их более предпочтительными для применения при синтезе УППС большой размерности. Следует отметить, что подмножества A, B, D, E классифицируют все 2208 УЭ, БФ которых в соответствии с табл. 5.3 имеют максимальное значение нелинейности.

Таблица 5.4

Возможные типы блоков $\mathbf{F}_{2/2}$ относительно входных разностей при нулевых управляющих разностях

i	*j*	*k*	**A**	**B**	**C**	**D**	**E**
0	1	0	0	0	0	0	0
1	1	0	3/4	5/8	7/8	1/2	1
2	1	0	1/4	3/8	1/8	1/2	0
0	2	0	0	0	0	0	0
1	2	0	1/2	3/4	1/4	1	0
2	2	0	1/2	1/4	3/4	0	1

В табл. 5.5 представлены сводные данные по дифференциальным свойствам различных УЭ $\mathbf{F}_{2/2}$, осуществляющих преобразование-инволюцию. Здесь в целях уменьшения объема проанализированного материала введен новый, в достаточной степени условный, показатель $\overline{\mathrm{H}}$ – средняя энтропия всех дифференциальных характеристик. Он определяется следующим образом

$$\overline{\mathrm{H}} = \frac{\left(\sum_{j=0}^{2} \sum_{k=1}^{2} \mathrm{H}_{jk} + \sum_{j=1}^{2} \mathrm{H}_{j0} \right)}{8},$$

где каждое из значений энтропии характеризует отдельную группу дифференциальных характеристик: $\mathrm{H}_{jk} = -\sum_{i=0}^{2} p\left(\Delta_i^Y \big/ \Delta_j^X,\ \Delta_k^V\right)\ \log_3 p\left(\Delta_i^Y \big/ \Delta_j^X,\ \Delta_k^V\right)$.

Данные, представленные в последнем столбце табл. 5.5, позволяют связать вопросы проектирования эффективных УЭ $\mathbf{F}_{2/2}$ с результатами классификации УЭ вида $\mathbf{F}_{2/1}$. Полная классификация УЭ $\mathbf{F}_{2/1}$, осуществляющих преобразование-инволюцию, показана в табл. 5.6. Например, конфигурация УЭ $\mathbf{F}_{2/2}$ в базисе УЭ $\mathbf{F}_{2/1}$ вида RSRL-SR (см. рис. 5.5) означает, что совокупность пар модификаций $\mathbf{F}^{(0)}$–$\mathbf{F}^{(1)}$, $\mathbf{F}^{(1)}$–$\mathbf{F}^{(3)}$, $\mathbf{F}^{(3)}$–$\mathbf{F}^{(2)}$, $\mathbf{F}^{(2)}$–$\mathbf{F}^{(1)}$ по своим свойствам образует набор УЭ $\mathbf{F}_{2/1}$ R, S, R и L типов, а перекрестные пары $\mathbf{F}^{(0)}$–$\mathbf{F}^{(3)}$, $\mathbf{F}^{(1)}$–$\mathbf{F}^{(2)}$, образуют набор УЭ $\mathbf{F}_{2/1}$ R и S типов. При этом УЭ $\mathbf{F}_{2/2}$, различающиеся только порядком следования типов, например, RSRL–SR, LSRR–SR и LSRR-RS, относятся к одному и тому же подклассу. Причем любой представитель характеризует весь подкласс.

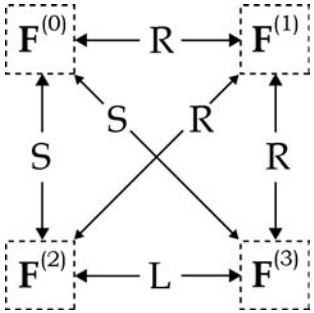

Рис. 5.5. Вариант представления блока $\mathbf{F}_{2/2}$ в базисе УЭ $\mathbf{F}_{2/1}$

Таблица 5.5

Средняя энтропия $\overline{\mathrm{H}}$	**Количество вариантов** ***N***	**Базовые модификации**	**Конфигурация УЭ** $\mathbf{F}_{2/2}$ **в базисе** $\mathbf{F}_{2/1}$
0.840	128	5 – 10	LRRS–RR
0.834	704	1 – 10	L $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ L– $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ L $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ L– $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ Z′ $\dot{\mathrm{L}}$ RS– $\ddot{\mathrm{L}}$ R Z′ $\ddot{\mathrm{L}}$ RS– $\dot{\mathrm{L}}$ R RSRL–SR LRRL–RR
0.815	128	1 – 4, 7 – 10	$\overline{\mathrm{L}}$ $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ L– $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ $\overline{\mathrm{L}}$ $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ L– $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$
0.813	192	1, 4, 5 – 10	$\dot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ RR–Z′S LRRL–RR (одинаковые пары)
0.812	256	2, 3, 5 – 10	$\overline{\mathrm{Z}}$ $\dot{\mathrm{L}}$ RS– $\ddot{\mathrm{L}}$ R $\overline{\mathrm{Z}}$ $\ddot{\mathrm{L}}$ RS– $\dot{\mathrm{L}}$ R
0.791	128	5 – 10	RRSS–RL
0.788	128	2, 3, 5 – 10	$\dot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ RR– $\overline{\mathrm{Z}}$ S
0.786	64	5 – 10	RRRR–LS
0.774	32	7 – 10	LSSL–SS
0.731	32	7 – 10	LSSL–SS (одинаковые пары)
0.719	96	1 – 10	$\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ $\dot{\mathrm{L}}$ –LL $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ $\ddot{\mathrm{L}}$ –LL RRRR–LL
0.710	16	7 – 10	SSSS–LL

Средняя энтропия $\overline{H}$	**Количество вариантов** N	**Базовые модификации**	**Конфигурация УЭ** $\mathbf{F}_{2/2}$ **в базисе** $\mathbf{F}_{2/1}$
0.695	192	1 – 10	$\overline{L}\,Z'\,\overline{Z}\,L–Z'\,\overline{Z}$ $\dot{L}\,\dot{L}\,\dot{L}\,\dot{L}–\overline{\dot{L}}\,L$ $\ddot{L}\,\ddot{L}\,\ddot{L}\,\ddot{L}–\overline{\ddot{L}}\,L$
0.641	32	5 – 10	RRRR–LL (одинаковые пары)
0.631	64	1 – 6	$Z'Z'\,\overline{Z}\,\overline{Z}–\overline{L}\,L$
0.513	16	7 – 10	SSSS–LL (одинаковые пары)

Таблица 5.6

УЭ $\mathbf{F}_{2/1}$ – инволюции

$\mathbf{F}_0/\mathbf{F}_1$	a	b	c	d	e	f	g	h	i	j
a	L	$\overline{L}$	$\overline{L}$	L	Z′	Z′	$\dot{L}$	$\ddot{L}$	$\ddot{L}$	$\dot{L}$
b	$\overline{L}$	L	L	$\overline{L}$	$\tilde{Z}$	$\tilde{Z}$	$\dot{L}$	$\ddot{L}$	$\ddot{L}$	$\dot{L}$
c	$\overline{L}$	L	L	$\overline{L}$	$\tilde{Z}$	$\tilde{Z}$	$\dot{L}$	$\ddot{L}$	$\ddot{L}$	$\dot{L}$
d	L	$\overline{L}$	$\overline{L}$	L	Z′	Z′	$\dot{L}$	$\ddot{L}$	$\ddot{L}$	$\dot{L}$
e	Z′	$\tilde{Z}$	$\tilde{Z}$	Z′	L	L	R	R	R	R
f	Z′	$\tilde{Z}$	$\tilde{Z}$	Z′	L	L	R	R	R	R
g	$\dot{L}$	$\dot{L}$	$\dot{L}$	$\dot{L}$	R	R	L	S	S	L
h	$\ddot{L}$	$\ddot{L}$	$\ddot{L}$	$\ddot{L}$	R	R	S	L	L	S
i	$\ddot{L}$	$\ddot{L}$	$\ddot{L}$	$\ddot{L}$	R	R	S	L	L	S
j	$\dot{L}$	$\dot{L}$	$\dot{L}$	$\dot{L}$	R	R	L	S	S	L

5.3. Вопросы построения управляемых операций

Путем комбинирования базовых УЭ $\mathbf{F}_{2/2}$ можно синтезировать УППС (УОП) $\mathbf{F}_{n/m}$, где n – число входных (выходных) бит, m – число управляющих бит. Такого рода управляемые операции задают отображение вида $GF(2)^{n+m} \to GF(2)^n$. Сохраняя топологию построения УППС на основе УЭ $\mathbf{F}_{2/1}$ и заменяя последний на УЭ $\mathbf{F}_{2/2}$, получим конкретный тип УППС (рис. 5.6).

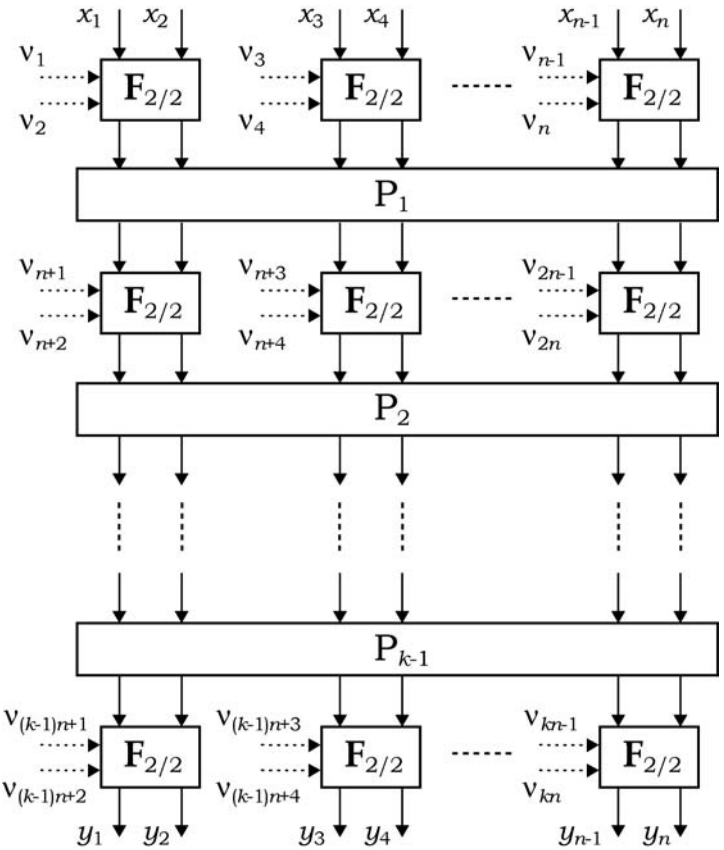

Рис. 5.6. Структура УППС $\mathbf{F}_{n/m}$ на основе элементарных управляемых элементов $\mathbf{F}_{2/2}$

Слоистая структура УППС с $n = 2^k$ - битовым входом для данных имеет управляющий вектор размера $m = kn$ бит, где k - количество подстановочных слоев.

На основе базового УЭ $\mathbf{F}_{2/2}$ достаточно просто синтезировать УППС, осуществляющие отображения вида $GF(2)^{n(k+1)} \to GF(2)^n$, например, $GF(2)^{32} \to GF(2)^8$, $GF(2)^{80} \to GF(2)^{16}$, $GF(2)^{182} \to GF(2)^{32}$, $GF(2)^{448} \to GF(2)^{64}$.

Можно отметить следующие основные свойства однородных блоков УППС $\mathbf{F}_{n/m}$ каскадного типа, построенных на основе элементарных подстановочных УЭ $\mathbf{F}_{2/2}$ и имеющих топологию, представленную на рис. 5.4:

- Для любого блока УППС $\mathbf{F}_{n/m}$, состоящего из k подстановочных слоев, где $k = \log_2 n$, и использующего в качестве основного конструктивного элемента биективный УЭ $\mathbf{F}_{2/2}$, управляющий вектор должен иметь разрядность $m = kn$.

- Если в каждом узле УППС $\mathbf{F}_{n/m}$ используется биективный УЭ $\mathbf{F}_{2/2}$, то преобразование $Y=\mathbf{F}_{n/m}(X, V) : \mathrm{GF}(2)^{n+m}\to\mathrm{GF}(2)^n$, где $X, Y \in \mathrm{GF}(2)^n$, $V \in \mathrm{GF}(2)^m$ будет биективным относительно X при фиксированных значениях управляющего вектора V и регулярным относительно множества входных и управляющих векторов;

- Если $f_i\in\Psi_n$, где Ψ_n – множество всех БФ, реализующих конкретный вид УППС, то для любого i=1, 2, …, n количество аргументов μ БФ f_i определяется выражением $\mu = 3\cdot 2^k - 2 = 3n - 2$. Действительно, выход любого подблока $\mathbf{F}_{2/2}$, принадлежащего первому слою блока $\mathbf{F}_{n/m}$, реализуется БФ от четырех переменных – двух битов x_i преобразуемого подблока данных и двух управляющих битов. Входные биты $x_i^{(2)}$ для следующего активного слоя сами являются БФ от четырех переменных (каждый из них является БФ от двух входных битов и двух управляющих битов). Следовательно, выходные биты второго слоя являются БФ от двух битов $x_i^{(2)}$ и двух новых управляющих битов, т. е. они являются БФ всего от десяти «первичных» переменных (последними являются биты преобразуемого подблока данных и управляющие биты). Если входные биты некоторого j-го слоя $x_i^{(j-1)}$ являются БФ от $\mu^{(j-1)}$ переменных, то выходные биты этого слоя являются БФ от $\mu^{(j)} = 2\mu^{(j-1)} + 2$ переменных. Эта рекуррентная формула позволяет легко рассчитать для выхода любого слоя число переменных, от которых зависят БФ, описывающие выходные биты. Таким образом, выход каждого следующего активного слоя УППС к числу переменных, от которых зависят два выходных бита предыдущего слоя, добавляет две переменных, которыми являются два новых управляющих бита соответствующего УЭ.
  Продвигаясь от входа к выходу, получим, что любая компонентная БФ зависит от $2^k = n$ битов данных и $\sum_{j=1}^{k} 2^j = 2^{k+1} - 2 = 2n - 2$ управляющих битов, следовательно, $\mu = 3n - 2 = 3\cdot 2^k - 2$.

- Если $f_i\in\Psi_n$, то при однородной структуре УППС алгебраическая степень нелинейности БФ $\deg(f_i)$ определяется выражениями

$$\deg(f_i) = k + 1, \text{ при } \deg(f^{\mathbf{F}_{2/2}}) = 2 \text{ для обоих выходов УЭ},$$

$$\deg(f_i) = 2k + 1, \text{ при } \deg(f^{\mathbf{F}_{2/2}}) = 3 \text{ для обоих выходов УЭ},$$

  где $\deg(f^{\mathbf{F}_{2/2}})$ – алгебраическая степень нелинейности функций, реализующих блок $\mathbf{F}_{2/2}$. Действительно, алгебраическая степень любой БФ на выходе первого слоя $\deg(f^{\mathbf{F}_{2/2}}) = 2$ либо $\deg(f^{\mathbf{F}_{2/2}}) = 3$. Данные БФ являются аргументами для БФ, реализующих второй слой. При этом алгебраическая степень увеличивается на единицу для $\deg(f^{\mathbf{F}_{2/2}}) = 2$ и на 2 для $\deg(f^{\mathbf{F}_{2/2}}) = 3$, так как в первом случае дополнительным сомножителем для БФ-аргументов является значение одного управляющего бита $v_i\in\mathrm{GF}(2)$, а во втором – произведение двух управляющих битов $v_i, v_{i+1} \in \mathrm{GF}(2)$. Методом индукции получаем, что любой из выходов $f_i\in\Psi_n$

последнего k-го слоя УППС $\mathbf{F}_{n/m}$ является суперпозицией БФ предыдущих k–1 слоев, а алгебраическая степень БФ равна $\deg(f_i) = k + 1$ либо $\deg(f_i) = 2k + 1$. Если БФ, описывающие выходы УЭ, имеют различную степень, то $k + 1 \le \deg(f_i) \le 2k + 1$.

- Применительно к УППС $\mathbf{F}_{n/m}$ верхнюю границу нелинейности образующих БФ можно найти, используя выражение (2.28) и учитывая количество аргументов БФ $\mu = 3n - 2$. В этом случае данное выражение преобразуется к виду

$$\mathrm{NL}(\mathbf{F}_{n/m}) \le 2^{3n-3} - 2^{\frac{3n-2}{2}-1} - 2.$$

Например, для УППС $\mathbf{F}_{4/8}$, $n = 4$, а верхняя граница значения нелинейности любой компонентной БФ оценивается величиной NL = 494, тогда как действительное значение нелинейности равно либо $\mathrm{NL}(\mathbf{F}_{4/8}) = 384$, либо $\mathrm{NL}(\mathbf{F}_{4/8}) = 448$, в зависимости от вида образующих элементарный УЭ $\mathbf{F}_{2/2}$ булевых функций.

В качестве дополнительного критерия проектирования УППС следует использовать требование максимально возможного значения нелинейности линейной комбинации БФ на выходе УППС $\mathbf{F}_{n/m}$. Для выполнения данного требования необходимо и достаточно, чтобы линейная комбинация на выходе базового УЭ $\mathbf{F}_{2/2}$ имела нелинейность $\mathrm{NL}(f_1(x_1, x_2, v_1, v_2) \oplus f_2(x_1, x_2, v_1, v_2)) = 4$. Очевидно, что различные линейные комбинации БФ между $\frac{n}{2}$ блоками $\mathbf{F}_{2/2}$ будут иметь нелинейность более высокого порядка, ввиду независимости аргументов БФ для различных блоков $\mathbf{F}_{2/2}$.

Выполненная оценка корреляционных свойств БФ, реализующих УППС с небольшим размером входа, показала, что они относятся к классу корреляционно-эффективных функций. Данное свойство напрямую следует из корреляционной эффективности БФ, реализующих элементарный УЭ $\mathbf{F}_{2/2}$.

Например, для БФ, реализующих УЭ $\mathbf{F}_{2/2}$, представленный на рис. 5.4а, на 12 (из 16) входных векторов коэффициенты спектра УА имеют значение 0. УППС $\mathbf{F}_{4/8}$, использующие данный УЭ, реализуются набором из четырех БФ от десяти переменных. Одна из БФ имеет вид:

$$f_1 = (v_5 \oplus v_6)((v_1 \oplus v_2)(x_1 \oplus x_2) \oplus v_1 v_2 \oplus v_2 \oplus x_1 \oplus 1 \oplus (v_3 \oplus v_4)(x_3 \oplus x_4) \oplus v_3 \oplus x_4) \oplus$$
$$v_5 v_6 \oplus v_6 \oplus x_1 \oplus (v_1 \oplus v_2)(x_1 \oplus x_2) \oplus v_1 v_2 \oplus v_2 \oplus x_1 \oplus 1,$$

а количество нулевых значений спектра $\#\{\hat{U}(f_1(x_1, \ldots, x_4, v_1, \ldots, v_6)) = 0\} = 1008$ (из 1024 возможных значений).

Вероятностные характеристики УППС

Блоки $\mathbf{F}_{n/m}$, построенные на основе УЭ более общего типа, по сравнению с переключающими элементами обладают следующим важным преимуществом: они вносят

существенный вклад в распространение лавинного эффекта при наличии измененных битов в преобразуемом подблоке данных. Этим свойством не обладают БУП, которые вносят вклад в лавинный эффект только за счет появления измененных битов на управляющем входе.

Последним свойством обладают также и блоки $\mathbf{F}_{n/m}$. Развитие лавинного эффекта тем более выражено, чем большее число активных каскадов содержится в управляемом операционном блоке. Представляет интерес рассмотрение особенностей развития лавинного эффекта в УППС $\mathbf{F}_{n/m}$ различного типа.

Для УППС $\mathbf{F}_{n/m}$ различной размерности определим условные вероятности $p(\boldsymbol{wt}(\Delta y)/\boldsymbol{wt}(\Delta x) = 1, \boldsymbol{wt}(\Delta v) = 0)$ появления разностей с различным весом Хэмминга на выходе при единичной входной разности и при нулевой разности на управляющем входе.

Получение производящей функции вероятностей появления разности веса $\boldsymbol{wt}(\Delta y)=2$ и $\boldsymbol{wt}(\Delta y)=1$ на выходе элементарного УЭ $\mathbf{F}_{2/2}$ при единичной разности $\boldsymbol{wt}(\Delta x)=1$ на входе тривиально. В этом случае будем использовать данные табл. 5.4. Для различных вариантов УЭ получим следующие виды производящих функций вероятностей:

$$\text{A}: \varphi_2^{\mathbf{F}_{2/2}}(z) = \frac{3}{4}z + \frac{1}{4}z^2,$$

$$\text{B}: \varphi_2^{\mathbf{F}_{2/2}}(z) = \frac{5}{8}z + \frac{3}{8}z^2,$$

$$\text{C}: \varphi_2^{\mathbf{F}_{2/2}}(z) = \frac{7}{8}z + \frac{1}{8}z^2,$$

$$\text{D}: \varphi_2^{\mathbf{F}_{2/2}}(z) = \frac{1}{2}z + \frac{1}{2}z^2,$$

$$\text{E}: \varphi_2^{\mathbf{F}_{2/2}}(z) = z.$$

Для УППС, имеющих однородную структуру, итерационно, путем подстановки выражений производящих функций с предыдущего слоя УППС в выражение производящей функции следующего слоя, получены производящие функции вероятностей вида $\varphi_n^{\mathbf{F}_{n/m}}(z)$.

В качестве примера на рис. 5.7 представлены дискретные распределения вероятностей $p(\boldsymbol{wt}(\Delta y) / \boldsymbol{wt}(\Delta x) = 1, \boldsymbol{wt}(\Delta v) = 0)$ появления разностей с различным весом Хэмминга на выходе УППС $\mathbf{F}_{n/m}$, относящихся к различному типу в соответствии с табл. 5.4, в случае, когда на вход данных подается разность веса $\boldsymbol{wt}(\Delta x)=1$, а на вход управления разность нулевого веса $\boldsymbol{wt}(\Delta v) = 0$.

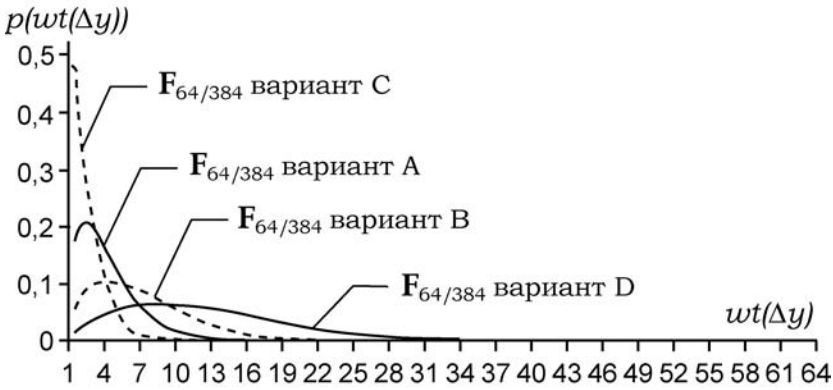

Рис. 5.7. Распределение вероятностей появления разностей различного веса $wt(\Delta y)$ на выходе УППС $\mathbf{F}_{64/384}$ при входной разности с весом $wt(\Delta x)=1$

Анализ полученных результатов позволяет сделать вывод о предпочтительности использования УППС, имеющих структуру в соответствии с вариантом D, разностные свойства которых аналогичны разностным свойствам УППС $\mathbf{S}_{n/m}$ типа, использующих УЭ $\mathbf{F}_{2/1}$, и имеют большую неопределенность.

Таким образом, результаты, полученные в *главах 4*, *5*, показывают существование достаточно большого числа различных УППС с минимальным размером УЭ, которые представляют интерес для использования в синтезе блочных шифров в качестве операций, зависящих от преобразуемых данных. Частным случаем таких операций являются управляемые перестановки, реализуемые с помощью БУП. Операционные блоки последнего типа были хорошо исследованы ранее и использованы в ряде стойких скоростных шифров. Ввиду того, что УЭ типа $\mathbf{S}_{2/1}$, $\mathbf{R}_{2/1}$ $\mathbf{F}_{2/2}$ по своим криптографическим свойствам значительно превосходят типовой узел БУП, которым является УЭ $\mathbf{P}_{2/1}$, существуют широкие возможности разработки новых более эффективных криптосистем по сравнению с перечисленными шифрами. Например, это может быть сделано заменой БУП в известных шифрах управляемыми блоками типов $\mathbf{F}_{n/m}$ различной топологической структуры. Кроме того, разнообразие разработанных примитивов позволяет использовать их для синтеза шифров, имеющих новые схемы построения.

Особенно предпочтительно использование новых типов операционных блоков при синтезе шифров, ориентированных на реализацию с использованием программируемых логических СБИС. В качестве новой команды универсального процессора более перспективно реализовать операцию управляемой перестановки, которая является не только эффективным криптографическим примитивом, но может быть использована в большом числе практически применяемых алгоритмов для существенного увеличения производительности последних. В случае микропрограммной реализации в контроллерах различного типа, включающих криптографические функции, перспективна реализация управляемой операции на основе блоков $\mathbf{F}_{n/m}$ различной топологической структуры.

ГЛАВА 6

Класс управляемых элементов $F_{3/1}$

6.1. Варианты представления и критерии отбора управляемых элементов $F_{3/1}$

Наряду с УППС, построенными на основе управляемых элементов $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$, представляет интерес разработка подстановочно-перестановочных сетей, имеющих топологию другого вида. Так как каждый элемент памяти программируемой логической матрицы позволяет реализовать произвольную БФ от четырех переменных, можно, используя, как и в предыдущих случаях две ячейки, построить УЭ вида $\mathbf{F}_{3/1}$ (рис. 6.1), осуществляющий преобразование $GF(2)^3 \rightarrow GF(2)^3$ под управлением однобитового управляющего кода. При этом ресурс ячейки памяти как и в случае УЭ $\mathbf{F}_{2/2}$ будет использоваться на 100 %. Кроме того, использование трехбитового входа данных позволяет осуществлять управление не только данными, но и непосредственно ключом. При этом, в отличие от УЭ с двухбитовым входом данных, возможно осуществление нелинейного преобразования. Следовательно, применение трехвходовых управляемых элементов для синтеза УППС позволяет обеспечить нелинейность УППС при фиксировании управляющего подблока данных (при этом фиксируются значения управляющих битов). Также очевидно, что при соответствующем выборе УЭ типа $\mathbf{F}_{3/1}$ можно обеспечить существенное повышение криптографических показателей синтезируемого операционного блока. В свою очередь это позволит сократить число раундов преобразования при сохранении высокой стойкости алгоритмов.

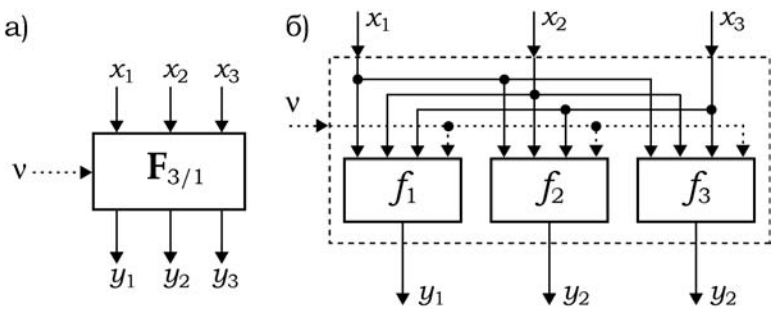

Рис. 6.1. Управляемый элемент $\mathbf{F}_{3/1}$ (а) и его реализация булевыми функциями (б)

Отмеченные свойства делают их интересными для построения УППС, несмотря на то, что они имеют нечетную разрядность входа (выхода). Если в некоторой криптосхеме предполагается разбиение блока данных на два подблока с одинаковой длиной, то последнее приводит к необходимости включения в каждый активный каскад УППС элементов $\mathbf{F}_{2/1}$ и/или $\mathbf{F}_{2/2}$ и некоторому усложнению анализа оценки стойкости шифров. Однако можно применять криптосхемы с несимметричным разбиением блока данных на подблоки. Например, в случае шифра с 64-битовым входом возможна разработка криптосхемы с разбиением на подблоки 16 бит и 48 бит, а в случае шифра со 128-битовым входом – криптосхемы с разбиением на подблоки 32 бит и 96 бит, где подблок данных меньшего размера следует использовать в качестве управляющего. В криптосхемах такого рода можно применять УППС, каждый активный каскад которых состоит из трехвходовых УЭ (шестнадцати в первом и тридцати двух во втором случае). Подобное разбиение обеспечивает определенный баланс: в каждом активном каскаде будет использоваться один раз каждый бит управляющего подблока.

Как будет показано далее в *главе 7*, элементы $\mathbf{F}_{3/1}$ могут найти самостоятельное применение в переключаемых операциях, используемых в криптосхемах с подблоками данных разной длины, или могут использоваться совместно с двухвходовыми управляемыми элементами в УППС, ориентированных на применение в криптосхемах с подблоками данных одинакового размера.

Большое количество возможных вариантов подстановок, осуществляющих отображение $GF(2)^3 \rightarrow GF(2)^3$, позволяет синтезировать элементарные УЭ $\mathbf{F}_{3/1}$, обладающие различными требуемыми свойствами. Следует отметить, что как и в случае элементов $\mathbf{F}_{2/2}$ полное перечисление, классификация и характеристика УЭ $\mathbf{F}_{3/1}$ является чрезмерно громоздким процессом.

Сформулируем некоторые критерии выбора конкретных вариантов УЭ $\mathbf{F}_{3/1}$. В общем случае УЭ $\mathbf{F}_{3/1}$ представим в виде:

- трех булевых функций от четырех переменных;
- двух подстановок, осуществляющих отображение $GF(2)^3 \rightarrow GF(2)^3$, каждая из которых выполняется над входным 3-битовым двоичным вектором (x_1, x_2, x_3) при $v = (0)$ или, $v = (1)$.

Критерии построения блоков $\mathbf{F}_{3/1}$

По аналогии с УЭ $\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$ сформулируем критерии проектирования УЭ $\mathbf{F}_{3/1}$:

1. Каждый из трех выходов блока $\mathbf{F}_{3/1}$ должен представлять собой нелинейную БФ от четырех переменных: $y_1 = f_1(x_1, x_2, x_3, v)$, $y_2 = f_2(x_1, x_2, x_3, v)$ и $y_3 = f_3(x_1, x_2, x_3, v)$, имеющих максимально возможное значение нелинейности относительно множества аффинных функций.
2. Каждая из двух элементарных модификаций блока $\mathbf{F}_{3/1}$: $\mathbf{F}^{(0)}$, $\mathbf{F}^{(1)}$ должна осуществлять биективное преобразование $GF(2)^3 \rightarrow GF(2)^3$.
3. Каждая из двух модификаций УЭ $\mathbf{F}^{(v)}$ должна быть инволюцией.

4. Все линейные комбинации компонентных БФ: $y_1 \oplus y_2$, $y_1 \oplus y_3$, $y_2 \oplus y_3$, $y_1 \oplus y_2 \oplus y_3$ должны иметь значение нелинейности не ниже, чем нелинейность компонентных булевых функций.

Оба варианта поиска эффективных УЭ $\mathbf{F}_{3/1}$, удовлетворяющих заданным критериям: перебор всех возможных троек булевых функций $y_i = f_i(x_1, x_2, x_3, v)$, $i \in \{1, 2, 3\}$ или перебор всех возможных наборов модификаций $\mathbf{F}^{(0)}$, $\mathbf{F}^{(1)}$ осуществляющих отображение $GF(2)^3 \to GF(2)^3$, требуют достаточно большого объема вычислений.

В первом случае следует проанализировать $\binom{2^4}{2^3}^3 \approx 2{,}1 \cdot 10^{12}$ различных совокупностей сбалансированных БФ от четырех переменных, имеющих показатели нелинейности, представленные в табл. 5.1.

Второй вариант является более предпочтительным, так как в этом случае проектирование эффективных УЭ $\mathbf{F}_{3/1}$ сводится к формальному выбору модификаций $\mathbf{F}^{(0)}$ и $\mathbf{F}^{(1)}$ с заданными свойствами, к одному из которых относится биективность выполняемого преобразования $GF(2)^3 \to GF(2)^3$.

Количество биективных модификаций определяется простейшим выражением: $\#\{\mathbf{F} : (GF(2)^3 \to GF(2)^3)_{biect}\} = (2^3)! = 40320$. Следовательно, визуальное проектирование сводится к выбору пар подобных модификаций и требует анализа $\approx 1{,}6 \cdot 10^9$ вариантов. Отметим, что элементарная модификация может осуществлять как линейное, так и нелинейное преобразование.

В целях проведения анализа свойств УЭ $\mathbf{F}_{3/1}$ по заданным модификациям $\mathbf{F}^{(0)}$ и $\mathbf{F}^{(1)}$ получим алгебраическую нормальную форму реализующих его БФ f_1, f_2 и f_3. Пусть $\{f^1_1(x_1, x_2, x_3), f^1_2(x_1, x_2, x_3), f^1_3(x_1, x_2, x_3)\}$ – БФ, реализующие модификацию $\mathbf{F}^{(0)}$ при значении v=(0), а $\{f^2_1(x_1, x_2, x_3), f^2_2(x_1, x_2, x_3), f^2_3(x_1, x_2, x_3)\}$ – БФ, реализующие модификацию $\mathbf{F}^{(1)}$ при значении v = (1).

Общее представление БФ от четырех переменных, описывающих элементы $\mathbf{F}_{3/1}$ через БФ от трех переменных, описывающие две его модификации, имеет вид:

$$y_1 = (v \oplus 1) f^1_1(x_1, x_2, x_3) \oplus v f^2_1(x_1, x_2, x_3);$$

$$y_2 = (v \oplus 1) f^1_2(x_1, x_2, x_3) \oplus v f^2_2(x_1, x_2, x_3);$$

$$y_3 = (v \oplus 1) f^1_3(x_1, x_2, x_3) \oplus v f^2_3(x_1, x_2, x_3).$$

Например, для элементарного блока $\mathbf{F}_{3/1}$, представленного на рис. 6.2, образующие БФ имеют вид:

$f^1_1(x_1, x_2, x_3) = x_1x_3 \oplus x_2x_3 \oplus x_1$; $\quad f^2_1(x_1, x_2, x_3) = x_1x_3 \oplus x_1x_2 \oplus x_3 \oplus 1$;

$f^1_2(x_1, x_2, x_3) = x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$; $\quad f^2_2(x_1, x_2, x_3) = x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_3 \oplus 1$;

$f^1_3(x_1, x_2, x_3) = x_1x_2 \oplus x_3$; $\quad f^2_3(x_1, x_2, x_3) = x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus 1$.

Следовательно,

$$y_1 = v(x_1x_2 \oplus x_2x_3 \oplus x_1 \oplus x_3 \oplus 1) \oplus x_1x_3 \oplus x_2x_3 \oplus x_1;$$
$$y_2 = v(x_1x_2 \oplus x_1x_3 \oplus x_2) \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1;$$
$$y_3 = v(x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus x_3 \oplus 1) \oplus x_1x_2 \oplus x_3;$$
$$y_1 \oplus y_2 = v(x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1) \oplus x_1x_3 \oplus x_2 \oplus x_3 \oplus 1;$$
$$y_1 \oplus y_3 = v(x_1x_3 \oplus x_1 \oplus x_2) \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_3;$$
$$y_2 \oplus y_3 = v(x_2x_3 \oplus x_3 \oplus 1) \oplus x_1x_2 \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus 1;$$
$$y_1 \oplus y_2 \oplus y_3 = v(x_1x_2 \oplus x_1) \oplus x_1x_2 \oplus x_1x_3 \oplus x_2 \oplus 1.$$

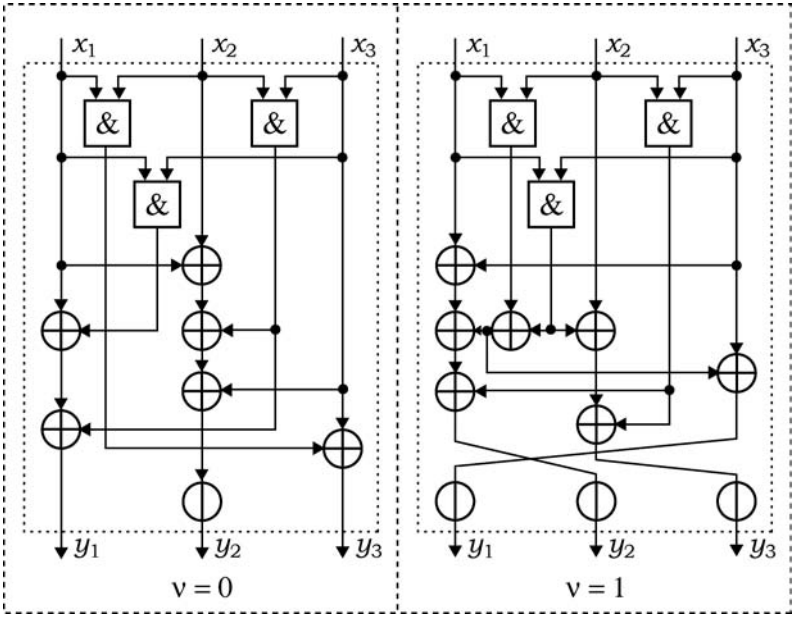

Рис. 6.2. Пример реализации управляемого элемента $\mathbf{F}_{3/1}$

6.2. Классификация УЭ $F_{3/1}$ по нелинейным и дифференциальным свойствам

Прежде чем проводить классификацию УЭ $\mathbf{F}_{3/1}$, определим множество модификаций $\mathbf{F}^{(v)}$, выполняющих инволюции. Число инволюций заданной степени определяется выражением

$$\#\{\mathbf{F} : (\mathrm{GF}(2)^n \to \mathrm{GF}(2)^n)_{inv}\} = \sum_{i=0}^{2^{n-1}} \frac{(2^n)_{2i}}{2^i i!},$$

где $(2^n)_{2i} = 2^n (2^n - 1) \ldots (2^n - 2i+1)$ есть число $2i$ размещений из 2^n элементов. Следовательно, для $n = 3$ получим $\#\{\mathbf{F} : (\mathrm{GF}(2)^3 \to \mathrm{GF}(2)^3)_{inv}\} = 764$. Ввиду того, что инволюции есть частный случай биективного преобразования, все реализующие их БФ $f_i(X) : \mathrm{GF}(2)^3 \to \mathrm{GF}(2)$, $X \in \mathrm{GF}(2)^3$, $i \in \{1, 2, 3\}$ являются сбалансированными.

Если пронумеровать в лексикографическом порядке все сбалансированные БФ от трех переменных (см. табл. 6.1), тогда модификацию $\mathbf{F}^{(v)}$, выполняющую инволюцию, можно представить в виде тройки номеров сбалансированных БФ.

В данной интерпретации в табл. 6.2 представлены все инволюции, осуществляющие отображение $\mathrm{GF}(2)^3 \to \mathrm{GF}(2)^3$. Инволюции, номера которых обозначены дополнительно одной звездочкой, реализуются тремя максимально нелинейными БФ $f_i(X) : \mathrm{GF}(2)^3 \to \mathrm{GF}(2)$, $X \in \mathrm{GF}(2)^3$, $i \in \{1, 2, 3\}$, а инволюции, номера которых помечены двумя звездочками, наряду с максимальной нелинейностью образующих БФ имеют максимальную нелинейность их линейных комбинаций.

Таблица 6.1

Множество сбалансированных БФ $f(x_1, x_2, x_3) : \mathrm{GF}(2)^3 \to \mathrm{GF}(2)$

№	АНФ	ТИ	№	АНФ	ТИ
1	x_1	0F	36	$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	87
2	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2$	17	37	$x_2x_3 \oplus x_1x_2 \oplus x_2 \oplus x_3 \oplus 1$	8B
3	$x_1x_3 \oplus x_2x_3 \oplus x_1$	1B	38	$x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus x_3 \oplus 1$	8D
4	$x_2x_3 \oplus x_1x_2 \oplus x_1$	1D	39	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_2 \oplus x_3 \oplus 1$	8E
5	$x_2x_3 \oplus x_1$	1E	40	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	93
6	$x_1x_3 \oplus x_2x_3 \oplus x_2$	27	41	$x_1x_2 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	95
7	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2$	2B	42	$x_3 \oplus x_2 \oplus x_1 \oplus 1$	96
8	$x_2x_3 \oplus x_1 \oplus x_2$	2D	43	$x_3 \oplus x_2 \oplus 1$	99
9	$x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2$	2E	44	$x_1x_2 \oplus x_2 \oplus x_3 \oplus 1$	9A
10	x_2	33	45	$x_1x_3 \oplus x_2 \oplus x_3 \oplus 1$	9C
11	$x_1x_3 \oplus x_1x_2 \oplus x_2$	35	46	$x_1x_3 \oplus x_1x_2 \oplus x_1 \oplus x_3 \oplus 1$	A3
12	$x_1x_3 \oplus x_2$	36	47	$x_3 \oplus x_1 \oplus 1$	A5
13	$x_1x_3 \oplus x_1 \oplus x_2$	39	48	$x_1x_2 \oplus x_1 \oplus x_3 \oplus 1$	A6
14	$x_1x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2$	3A	49	$x_1x_2 \oplus x_3 \oplus 1$	A9
15	$x_2 \oplus x_1$	3C	50	$x_3 \oplus 1$	AA
16	$x_2x_3 \oplus x_1x_2 \oplus x_3$	47	51	$x_1x_3 \oplus x_1x_2 \oplus x_3 \oplus 1$	AC
17	$x_2x_3 \oplus x_1 \oplus x_3$	4B	52	$x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_3 \oplus 1$	B1
18	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_3$	4D	53	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_3 \oplus 1$	B2
19	$x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_3$	4E	54	$x_2x_3 \oplus x_1 \oplus x_3 \oplus 1$	B4
20	$x_1x_3 \oplus x_1x_2 \oplus x_3$	53	55	$x_2x_3 \oplus x_1x_2 \oplus x_3 \oplus 1$	B8
21	x_3	55	56	$x_2 \oplus x_1 \oplus 1$	C3
22	$x_1x_2 \oplus x_3$	56	57	$x_1x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2 \oplus 1$	C5
23	$x_1x_2 \oplus x_1 \oplus x_3$	59	58	$x_1x_3 \oplus x_1 \oplus x_2 \oplus 1$	C6
24	$x_3 \oplus x_1$	5A	59	$x_1x_3 \oplus x_2 \oplus 1$	C9

№	АНФ	ТИ	№	АНФ	ТИ
25	$x_1x_3 \oplus x_1x_2 \oplus x_1 \oplus x_3$	5C	60	$x_1x_3 \oplus x_1x_2 \oplus x_2 \oplus 1$	CA
26	$x_1x_3 \oplus x_2 \oplus x_3$	63	61	$x_2 \oplus 1$	CC
27	$x_1x_2 \oplus x_2 \oplus x_3$	65	62	$x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2 \oplus 1$	D1
28	$x_3 \oplus x_2$	66	63	$x_2x_3 \oplus x_1 \oplus x_2 \oplus 1$	D2
29	$x_3 \oplus x_2 \oplus x_1$	69	64	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus x_2 \oplus 1$	D4
30	$x_1x_2 \oplus x_1 \oplus x_2 \oplus x_3$	6A	65	$x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus 1$	D8
31	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3$	6C	66	$x_2x_3 \oplus x_1 \oplus 1$	E1
32	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_2 \oplus x_3$	71	67	$x_2x_3 \oplus x_1x_2 \oplus x_1 \oplus 1$	E2
33	$x_1x_3 \oplus x_2x_3 \oplus x_2 \oplus x_3$	72	68	$x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus 1$	E4
34	$x_2x_3 \oplus x_1x_2 \oplus x_2 \oplus x_3$	74	69	$x_1x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus 1$	E8
35	$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3$	78	70	$x_1 \oplus 1$	F0

Результаты исследования нелинейных свойств элементарных УЭ вида $\mathbf{F}_{3/1}$, осуществляющих преобразование-инволюцию, показали, что существует 400 линейных УЭ, образующие БФ которых, а также все их линейные комбинации имеют значение нелинейности равное нулю и 131712 максимально нелинейных УЭ, у которых образующие БФ и их линейные комбинации имеют значение нелинейности равное четырем. Для остальных 451584 вариантов построения инволюций нелинейность образующих БФ $\mathrm{NL}(f_i) \in \{0, 2, 4\}$, $i=1, 2, \ldots, 7$.

Таблица 6.2

Множество инволюций $\mathbf{F}$: $\mathrm{GF}(2)^3 \to \mathrm{GF}(2)^3$

№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3
1	1	10	21	192**	7	19	35	383*	25	33	16	574**	46	2	33
2	1	10	22	193*	7	36	66	384**	25	33	17	575*	46	3	62
3	1	10	23	194**	7	36	67	385**	25	62	2	576**	46	3	63
4	1	10	24	195**	7	38	66	386*	25	62	3	577**	46	16	32
5	1	10	27	196*	7	39	69	387**	25	63	16	578*	46	16	33
6	1	10	28	197	8	10	21	388*	25	63	17	579**	46	17	62
7	1	10	29	198	8	10	23	389*	26	2	13	580*	46	17	63
8	1	10	30	199	8	10	41	390**	26	2	14	581	47	10	21
9	1	10	41	200	8	10	43	391	26	3	21	582	47	12	21
10	1	10	42	201	8	12	21	392**	26	3	22	583	47	26	21
11	1	10	43	202**	8	12	41	393**	26	36	49	584	47	28	21
12	1	10	44	203*	8	14	25	394	26	36	50	585	47	43	21
13	1	10	47	204**	8	14	45	395**	26	37	57	586	47	45	21
14	1	10	48	205	8	26	21	396*	26	37	58	587	47	59	21
15	1	10	49	206**	8	26	23	397	27	10	4	588	47	61	21
16	1	10	50	207	8	28	21	398**	27	12	4	589	48	10	34
17	1	11	20	208**	8	30	25	399*	27	23	2	590**	48	11	33
18	1	11	26	209*	8	46	57	400**	27	25	2	591**	48	26	34
19	1	11	40	210**	8	46	59	401**	27	46	38	592*	48	27	33
20	1	11	46	211**	8	48	57	402*	27	48	38	593*	48	44	64
21	1	12	21	212	8	50	61	403**	27	59	36	594**	48	45	63

№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3
22	1	12	27	213	9	10	34	404	27	61	36	595**	48	60	64
23	1	12	41	214	9	10	35	405	28	10	15	596	48	61	63
24	1	12	47	215	9	10	54	406	28	11	14	597	49	10	62
25	1	13	21	216	9	10	55	407	28	24	21	598**	49	14	65
26	1	13	27	217*	9	11	33	408	28	25	20	599**	49	26	62
27	1	13	41	218**	9	11	53	409	28	46	51	600*	49	30	65
28	1	13	47	219*	9	13	34	410	28	47	50	601*	49	41	32
29	1	14	25	220**	9	13	54	411	28	60	57	602**	49	45	35
30	1	14	31	221*	9	26	34	412	28	61	56	603**	49	57	32
31	1	14	45	222**	9	26	35	413	29	10	21	604	49	61	35
32	1	14	51	223**	9	27	33	414	29	14	25	605	50	10	70
33	1	15	21	224	9	29	34	415	29	21	10	606	50	13	70
34	1	15	24	225*	9	46	68	416	29	25	14	607	50	26	70
35	1	15	27	226**	9	46	69	417	29	46	57	608	50	29	70
36	1	15	30	227	9	47	67	418	29	50	61	609	50	42	70
37	1	15	41	228**	9	49	68	419	29	57	46	610	50	45	70
38	1	15	44	229	10	1	21	420	29	61	50	611	50	58	70
39	1	15	47	230	10	1	22	421	30	10	34	612	50	61	70
40	1	15	50	231	10	1	29	422**	30	13	34	613*	51	52	34
41	1	20	11	232	10	1	30	423*	30	22	32	614**	51	52	35
42	1	20	12	233	10	1	41	424**	30	25	32	615*	51	53	64
43	1	20	13	234	10	1	42	425**	30	46	68	616**	51	53	65
44	1	20	14	235	10	1	49	426*	30	49	68	617**	51	66	34
45	1	21	10	236	10	1	50	427**	30	58	66	618*	51	66	35
46	1	22	11	237*	11	3	16	428	30	61	66	619**	51	67	64
47	1	23	11	238**	11	3	36	429*	31	32	12	620*	51	67	65
48	1	24	15	239**	11	5	16	430**	31	32	14	621**	52	2	26
49	1	25	11	240*	11	5	36	431	31	33	21	622	52	3	56
50	1	25	14	241*	11	7	18	432**	31	33	23	623	52	6	21
51	1	26	21	242**	11	7	38	433**	31	66	48	624**	52	7	59
52	1	26	22	243**	11	9	18	434	31	66	50	625	52	38	21
53	1	26	23	244*	11	9	38	435**	31	67	57	626	52	38	29
54	1	26	24	245	12	3	21	436*	31	67	59	627*	53	2	32
55	1	27	20	246**	12	3	41	437*	32	2	7	628**	53	3	62
56	1	28	21	247**	12	4	20	438**	32	3	16	629**	53	6	34
57	1	29	21	248*	12	4	40	439**	32	6	4	630*	53	7	69
58	1	30	25	249*	12	7	31	440*	32	7	18	631*	53	39	64
59	1	31	21	250**	12	7	51	441*	32	18	2	632*	53	39	69
60	1	31	24	251**	12	8	30	442*	32	18	7	633	54	10	21
61	1	40	21	252	12	8	50	443**	33	2	13	634	54	10	28
62	1	40	22	253*	13	2	26	444	33	3	21	635	54	43	21
63	1	40	23	254**	13	2	46	445	33	6	15	636**	54	44	58
64	1	40	24	255**	13	5	30	446**	33	7	31	637**	54	49	31
65	1	41	20	256	13	5	50	447	33	19	21	638	54	50	61
66	1	42	21	257	13	6	21	448	33	19	29	639	55	10	62
67	1	43	21	258**	13	6	41	449	34	10	4	640	55	10	67
68	1	44	25	259**	13	9	25	450	34	10	9	641**	55	41	32
69	1	45	21	260*	13	9	45	451**	34	23	2	642	55	42	67
70	1	45	24	261*	14	2	32	452	34	24	16	643	55	47	34
71	1	46	57	262**	14	2	52	453	34	29	9	644**	55	48	64
72	1	46	58	263**	14	4	32	454**	34	30	18	645	56	10	21

№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3
73	1	46	59	264*	14	4	52	455	35	10	21	646	56	10	22
74	1	46	60	265*	14	6	34	456	35	10	28	647	56	10	27
75	1	47	56	266**	14	6	54	457	35	21	10	648	56	10	28
76	1	48	57	267**	14	8	34	458**	35	22	26	649	56	10	43
77	1	49	57	268*	14	8	54	459**	35	27	12	650	56	10	44
78	1	50	61	269	15	10	21	460	35	28	21	651	56	10	49
79	1	51	57	270	15	10	23	461	36	10	21	652	56	10	50
80	1	51	60	271	15	10	28	462	36	10	22	653*	57	32	2
81	1	56	21	272	15	10	30	463	36	10	27	654**	57	32	6
82	1	56	22	273	15	10	41	464	36	10	28	655**	57	34	2
83	1	56	23	274	15	10	43	465*	36	11	20	656*	57	34	6
84	1	56	24	275	15	10	48	466**	36	11	26	657*	57	52	4
85	1	56	47	276	15	10	50	467	36	12	21	658**	57	52	8
86	1	56	48	277	16	10	4	468**	36	12	27	659**	57	54	4
87	1	56	49	278	16	10	5	469*	36	20	11	660*	57	54	8
88	1	56	50	279	16	10	8	470**	36	20	12	661	58	33	21
89	1	57	20	280	16	10	9	471	36	21	10	662**	58	33	27
90	1	57	46	281*	16	11	3	472**	36	22	11	663**	58	34	20
91	1	58	21	282**	16	11	7	473	36	26	21	664*	58	34	26
92	1	58	47	283*	16	12	4	474**	36	26	22	665*	58	53	45
93	1	59	21	284**	16	12	8	475**	36	27	20	666**	58	53	51
94	1	59	47	285*	16	40	4	476	36	28	21	667**	58	54	44
95	1	60	25	286**	16	40	5	477	37	10	62	668	58	54	50
96	1	60	51	287**	16	41	3	478	37	10	63	669*	59	32	40
97	1	61	21	288	16	42	4	479	37	10	66	670**	59	32	46
98	1	61	24	289*	16	46	38	480	37	10	67	671**	59	35	44
99	1	61	47	290**	16	46	39	481*	37	13	62	672	59	35	50
100	1	61	50	291	16	47	37	482**	37	13	66	673	59	52	21
101*	2	7	18	292**	16	48	38	483*	37	14	65	674**	59	52	27
102*	2	7	19	293	17	10	21	484**	37	14	69	675**	59	55	25
103*	2	7	38	294	17	10	22	485*	37	20	52	676*	59	55	31
104*	2	7	39	295	17	10	27	486**	37	20	53	677*	60	33	62
105*	2	8	17	296	17	10	28	487**	37	23	52	678**	60	33	66
106**	2	8	37	297	17	13	21	488	37	24	55	679**	60	35	62
107*	2	9	18	298**	17	13	27	489*	37	26	62	680*	60	35	66
108**	2	9	38	299*	17	14	25	490**	37	26	63	681*	60	53	64
109*	2	17	8	300**	17	14	31	491	37	29	62	682**	60	53	68
110**	2	17	9	301	17	40	21	492**	37	30	65	683**	60	55	64
111*	2	18	7	302**	17	40	22	493	38	52	21	684*	60	55	68
112**	2	19	8	303	17	43	21	494*	38	52	23	685	61	70	21
113*	2	37	18	304**	17	44	25	495*	38	52	27	686	61	70	23
114**	2	37	19	305*	17	46	57	496	38	52	29	687	61	70	27
115**	2	38	17	306**	17	46	58	497	38	54	21	688	61	70	29
116*	2	39	18	307**	17	49	57	498**	38	54	27	689	61	70	42
117	3	6	21	308	17	50	61	499*	38	55	25	690	61	70	44
118*	3	6	22	309*	18	32	2	500**	38	55	31	691	61	70	48
119*	3	6	41	310*	18	32	3	501*	38	62	11	692	61	70	50
120	3	6	42	311*	18	32	6	502**	38	62	13	693	62	10	16
121	3	8	21	312*	18	32	7	503**	38	64	11	694	62	10	37
122**	3	8	41	313*	18	34	2	504	38	65	15	695**	62	27	2
123*	3	9	25	314**	18	34	6	505	38	66	21	696	62	29	37

№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3
124**	3	9	45	315*	18	35	5	506**	38	66	23	697	62	47	4
125*	3	16	11	316**	18	35	9	507	38	68	21	698**	62	49	18
126**	3	16	12	317*	18	62	2	508**	38	69	25	699	63	10	21
127**	3	18	11	318*	18	62	3	509*	39	53	64	700	63	10	43
128	3	19	15	319*	18	64	2	510*	39	53	65	701	63	28	21
129	3	36	21	320**	18	65	5	511*	39	53	68	702**	63	30	59
130**	3	36	22	321*	18	66	36	512*	39	53	69	703**	63	48	45
131	3	38	21	322**	18	66	37	513*	39	54	63	704	63	50	61
132**	3	39	25	323**	18	68	36	514**	39	54	67	705*	64	32	2
133	4	10	16	324*	18	69	39	515*	39	55	64	706**	64	33	16
134	4	10	17	325	19	33	21	516**	39	55	68	707**	64	52	4
135	4	10	36	326*	19	33	23	517*	39	63	54	708*	64	53	39
136	4	10	37	327*	19	33	27	518**	39	63	55	709*	64	69	18
137*	4	12	16	328	19	33	29	519*	39	64	53	710*	64	69	39
138**	4	12	36	329*	19	34	20	520**	39	65	54	711**	65	32	40
139*	4	14	19	330**	19	34	26	521*	39	67	64	712	65	33	56
140**	4	14	39	331	19	35	21	522**	39	67	65	713	65	52	21
141*	4	20	6	332**	19	35	27	523**	39	68	63	714**	65	53	58
142**	4	20	7	333	19	63	21	524*	39	69	64	715	65	68	21
143**	4	22	6	334**	19	63	23	525	40	6	21	716	65	68	42
144	4	24	9	335**	19	64	20	526**	40	6	22	717	66	10	21
145*	4	40	16	336	19	65	21	527*	40	7	59	718	66	10	43
146**	4	40	17	337*	19	67	57	528**	40	7	60	719	66	21	10
147	4	42	16	338**	19	67	59	529**	40	16	11	720**	66	23	40
148**	4	44	19	339	19	68	56	530*	40	16	12	721**	66	41	13
149	5	10	21	340**	19	69	57	531**	40	17	49	722	66	43	21
150	5	10	23	341*	20	6	4	532	40	17	50	723	67	10	34
151	5	10	41	342**	20	6	5	533	41	10	16	724	67	10	55
152	5	10	43	343*	20	7	18	534**	41	12	16	725**	67	22	32
153*	5	11	20	344**	20	7	19	535**	41	20	6	726	67	24	62
154**	5	11	40	345**	20	36	4	536*	41	22	6	727	67	42	55
155	5	13	21	346*	20	36	5	537*	41	49	18	728**	67	44	64
156**	5	13	41	347**	20	37	18	538**	41	51	18	729**	68	32	12
157*	5	20	11	348*	20	37	19	539**	41	59	8	730	68	33	21
158**	5	20	13	349	21	10	1	540	41	61	8	731	68	52	15
159	5	21	10	350	21	12	1	541	42	10	21	732**	68	53	45
160**	5	23	11	351	21	29	1	542	42	11	20	733	68	65	21
161	5	40	21	352	21	31	1	543	42	20	11	734	68	65	42
162**	5	40	23	353	21	40	1	544	42	21	10	735*	69	32	53
163**	5	41	20	354	21	42	1	545	42	50	61	736**	69	33	62
164	5	43	21	355	21	59	1	546	42	51	60	737**	69	52	34
165	6	3	21	356	21	61	1	547	42	60	51	738*	69	53	64
166*	6	3	22	357	22	10	4	548	42	61	50	739*	69	64	32
167*	6	3	41	358**	22	11	3	549	43	10	56	740*	69	64	53
168	6	3	42	359*	22	30	18	550	43	14	60	741	70	10	21
169*	6	4	20	360**	22	31	17	551	43	20	46	742	70	10	28
170**	6	4	40	361**	22	40	4	552	43	24	50	743	70	10	43
171	6	5	21	362*	22	41	3	553	43	47	21	744	70	10	50
172**	6	5	41	363**	22	60	18	554	43	51	25	745	70	21	10
173	6	17	21	364	22	61	17	555	43	57	11	746	70	22	26
174**	6	17	22	365	23	10	16	556	43	61	15	747	70	23	40

№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3	№	f_1	f_2	f_3
175**	6	18	20	366**	23	14	19	557	44	10	62	748	70	24	56
176	6	19	21	367*	23	27	2	558**	44	13	62	749	70	27	12
177*	6	37	57	368**	23	31	5	559**	44	20	52	750	70	28	21
178**	6	37	58	369**	23	40	16	560*	44	23	52	751	70	29	50
179	6	38	56	370*	23	44	19	561*	44	48	64	752	70	30	59
180**	6	39	57	371**	23	57	2	562**	44	51	64	753	70	41	13
181*	7	2	32	372	23	61	5	563**	44	58	54	754	70	42	50
182*	7	2	33	373	24	10	21	564	44	61	54	755	70	43	21
183*	7	2	52	374	24	13	21	565	45	52	21	756	70	44	58
184*	7	2	53	375	24	28	21	566**	45	52	23	757	70	47	15
185*	7	4	32	376	24	31	21	567*	45	53	58	758	70	48	45
186**	7	4	52	377	24	40	21	568**	45	53	60	759	70	49	31
187*	7	5	35	378	24	43	21	569**	45	62	11	760	70	50	61
188**	7	5	55	379	24	58	21	570*	45	62	13	761	70	61	21
189*	7	16	32	380	24	61	21	571**	45	63	48	762	70	61	29
190**	7	16	33	381*	25	32	2	572	45	63	50	763	70	61	42
191*	7	18	32	382**	25	32	3	573*	46	2	32	764	70	61	50

Спектр УА максимально нелинейных БФ, реализующих блок $\mathbf{F}_{3/1}$, по своей структуре не отличается от спектра УА максимально нелинейных БФ, реализующих блок $\mathbf{F}_{2/2}$, т.е. он содержит либо шесть, либо двенадцать нулевых значений из шестнадцати. При чем количество максимальных по модулю значений спектра $\#\{|\hat{U}_\alpha(f(x_1, x_2, x_3, v))| = 8\} = 2$ или 4, соответственно, $\alpha \in GF(2)^4$, что говорит об аффинной эквивалентности БФ, относящихся к первому или второму классу.

В примере, приведенном в п.п. 6.1, БФ имеют следующие таблицы истинности и вид спектра УА:

y_1: S_1={0100011011011010}, $\hat{U}_\alpha$ ={0,0,–4,4,0,8,–4,–4,4,4,0,8,4,–4,0,0};

y_2: S_2={1100010100101110}, $\hat{U}_\alpha$ ={0,0,–4,4,4,–4,0,0,0,8,–4,–4,–4,–4,–8,0};

y_3: S_3={0111001101101000}, $\hat{U}_\alpha$ ={0,0,4,4,–4,4,0,8,–4,4,8,0,0,0,–4,–4};

$y_1 \oplus y_2$ S_4={1000001111110100}, $\hat{U}_\alpha$ ={0,0,0,0,–4,–4,–4,–4,4,–4,4,–4,8,0,–8,0};

$y_1 \oplus y_3$ S_5={0011010110110010}, $\hat{U}_\alpha$ ={0,0,8,0,–4,–4,4,–4,0,8,0,0,4,–4,4,4};

$y_2 \oplus y_3$ S_6={1011011001000110}, $\hat{U}_\alpha$ ={0,0,0,8,0,0,0,–8,–4,–4,4,4,–4,–4,–4,4,–4};

$y_1 \oplus y_2 \oplus y_3$: S_7={1111000010011100}, $\hat{U}_\alpha$ ={0,0,–4,–4,–8,0,4,–4,0,0,4,4,–8,0,–4,4}.

Рассмотрим другой пример. Пусть УЭ $\mathbf{F}_{3/1}$ реализован двумя модификациями инволюций из табл. 6.2: $\mathbf{F}^0$=737, $\mathbf{F}^1$=439. Тогда получим

$y_1 = v(x_2+x_3+1)+x_1x_2+x_1x_3+x_2x_3+1$,

S_1={1011110110000001}, $\hat{U}_\alpha(y_1)$={0,0,0,–8,0,–8,0,0,–8,0,0,0,0,0,8,0};

$y_2 = v(x_1+x_2+x_3+1)+x_1x_3+x_2x_3+x_1+x_3+1$,

S_2={1000111000010111}, $\hat{U}_\alpha(y_2)$={0,0,0,0,8,0,0,–8,0,–8,–8,0,0,0,0,0};

$y_3=v(x_1+x_2+x_3)+x_2x_3+x_1x_2+x_2+x_3$,

S_3={0010101101110001}, $\hat{U}_\alpha(y_3)$={0,0,8,0,0,0,0,8,0,–8,0,0,8,0,0,0};

$y_1 \oplus y_2=vx_1+x_1x_2+x_1+x_3$,

S_4={0011001110010110}, $\hat{U}_\alpha(y_1 \oplus y_2)$={0,0,8,0,0,0,0,–8,0,0,8,0,0,0,0,8};

$y_1 \oplus y_3=v(x_1+1)+x_1x_3+x_2+x_3+1$,

S_5={1001011011110000}, $\hat{U}_\alpha(y_1 \oplus y_3)$={0,0,0,0,–8,0,0,–8,0,0,0,0,8,0,0,–8};

$y_2 \oplus y_3=v+x_1x_3+x_1x_2+x_1+x_2+1$,

S_6={1010010101100110}, $\hat{U}_\alpha(y_2 \oplus y_3)$={0,0,0,8,0,–8,0,0,0,0,0,–8,0,–8,0,0};

$y_1 \oplus y_2 \oplus y_3=v(x_2+x_3)+x_2x_3+x_1+x_2$,

S_7={0001100011100111}, $\hat{U}_\alpha(y_1 \oplus y_2 \oplus y_3)$={0,0,0,0,0,0,0,0,8,0,0,–8,0,8,8,0}.

Таким образом, можно сделать вывод о корреляционной эффективности БФ, реализующих УЭ $\mathbf{F}_{3/1}$, выполняющие преобразование - инволюцию. Откуда также следует корреляционная эффективность БФ, реализующих многослойные УППС $\mathbf{F}_{n/m} : \mathrm{GF}(2)^{n+m} \to \mathrm{GF}(2)^n$.

Рассмотренные два примера задают границы корреляционной эффективности. Другие варианты УЭ $\mathbf{F}_{3/1}$ реализованы БФ, спектр которых имеет структуру либо первого, либо второго вида. Следует отметить, что существует 4480 вариантов построения УЭ $\mathbf{F}_{3/1}$, значение компонент спектра УА которых $\hat{U}_\alpha = 0$ на 12 векторах $\alpha \in \mathrm{GF}(2)^4$.

Исследованы дифференциальные свойства УЭ $\mathbf{F}_{3/1}$ и проведена их классификация в соответствии с этими свойствами.

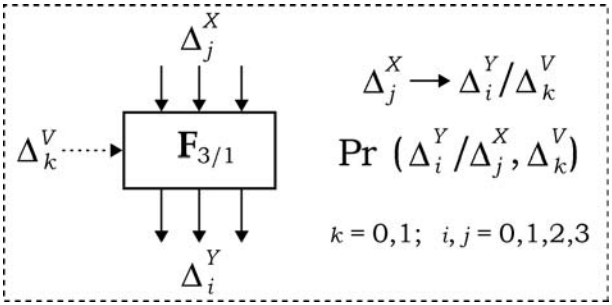

Рис. 6.3. Варианты разностей для УЭ $\mathbf{F}_{3/1}$

На рис. 6.3 представлены варианты всех возможных разностей, относящихся к УЭ вида $\mathbf{F}_{3/1}$. Наиболее полное представление о дифференциальных свойствах криптографических примитивов дает матрица переходных вероятностей входных разностей $\mathbf{P}_k=\|p(\Delta^Y_i/\Delta^X_j, \Delta^V_k)\|$. Однако в ряде практических случаев при проведении дифференциального анализа следует учитывать входные разности только одного веса Хэмминга. Причем наибольший практический интерес представляют результаты, которые получены при условии равенства нулю одной из разностей: на управляющем входе или входе данных соответственно. Классификация УЭ, осуществляющих преобразование - инволюцию, в соответствии с их дифференциальными характеристиками при фиксированном весе разности на управляющем входе и входе данных, т. е. по значениям $p(\Delta^Y_i/\Delta^X_j, \Delta^V_k) = p_{ijk}$ при j, k = const, представлена в табл. 6.3–6.6. Следует отметить, что здесь проанализированы только те элементы, образующие БФ которых и все их линейные комбинации имеют максимальное значение нелинейности NL(f)=4. Полученные результаты показывают, что в отличие от множества УЭ $\mathbf{F}_{2/2}$ и тем более $\mathbf{F}_{2/1}$ множество УЭ $\mathbf{F}_{3/1}$ по своим дифференциальным свойствам содержит гораздо большее количество подмножеств, среди которых ряд подмножеств (например, указанные в позициях 22, 23 табл. 6.3) выделяются большей неопределенностью значений выходных разностей, что делает их предпочтительными для применения в УППС, осуществляющих преобразование векторов большой размерности.

Таблица 6.3

Типы УЭ $\mathbf{F}_{3/1}$ относительно $p(\Delta^Y_i/\Delta^X_1, \Delta^V_0)$

№	p_{010}	p_{110}	p_{210}	p_{310}
1	0	7/12	1/3	1/12
2	0	5/8	1/3	1/24
3	0	11/24	5/12	1/8
4	0	7/12	1/4	1/6
5	0	5/8	1/4	1/8
6	0	13/24	1/3	1/8
7	0	1/2	5/12	1/12
8	0	5/12	5/12	1/6
9	0	17/24	1/6	1/8
10	0	7/12	5/12	0
11	0	2/3	1/6	1/6
12	0	5/12	1/2	1/12
13	0	3/8	1/2	1/8
14	0	11/24	1/2	1/24
15	0	1/3	1/2	1/6
16	0	13/24	1/4	5/24

№	p_{010}	p_{110}	p_{210}	p_{310}
17	0	7/24	7/12	1/8
18	0	1/2	1/3	1/6
19	0	13/24	5/12	1/24
20	0	1/2	1/2	0
21	0	7/24	1/2	5/24
22	0	3/8	5/12	5/24
23	0	5/12	1/3	1/4
24	0	11/24	1/3	5/24
25	0	1/2	1/4	1/4
26	0	1/4	1/2	1/4

Таблица 6.4

Типы УЭ $\mathbf{F}_{3/1}$ относительно $p(\Delta^Y_i/\Delta^X_2, \Delta^V_0)$

№	p_{020}	p_{120}	p_{220}	p_{320}
1	0	0,3333	0,5	0,1667
2	0	0,4167	0,5	0,0833
3	0	0,25	0,6667	0,0833
4	0	0,4167	0,4167	0,1667
5	0	0,1667	0,6667	0,1667
6	0	0,4167	0,3333	0,25
7	0	0,3333	0,4167	0,25
8	0	0,5	0,3333	0,1667
9	0	0,5	0,4167	0,0833
10	0	0,5833	0,3333	0,0833
11	0	0,5	0,5	0
12	0	0,3333	0,6667	0

Таблица 6.5

Типы УЭ $\mathbf{F}_{3/1}$ относительно $p(\Delta^Y_i/\Delta^X_3, \Delta^V_0)$

№	p_{030}	p_{130}	p_{230}	p_{330}
1	0	0,25	0,5	0,25
2	0	0,125	0,5	0,375

№	p_{030}	p_{130}	p_{230}	p_{330}
3	0	0,375	0,25	0,375
4	0	0,5	0,25	0,25
5	0	0,375	0,5	0,125
6	0	0,25	0,75	0
7	0	0	0,75	0,25
8	0	0,5	0,5	0
9	0	0,625	0,25	0,125
10	0	0,125	0,25	0,625
11	0	0,375	0	0,625
12	0	0	0	1
13	0	0,25	0,25	0,5
14	0	0,625	0	0,375
15	0	0,75	0	0,25
16	0	0,75	0,25	0

Таблица 6.6

Типы УЭ $\mathbf{F}_{3/1}$ относительно $p(\Delta^Y_i/\Delta^X_0, \Delta^V_1)$

№	p_{001}	p_{101}	p_{201}	p_{301}	№	p_{001}	p_{101}	p_{201}	p_{301}
1	0,25	0,25	0,5	0	32	0,375	0,125	0,125	0,375
2	0,25	0,25	0,25	0,25	33	0	0,125	0,5	0,375
3	0	0,5	0,5	0	34	0,375	0,375	0,125	0,125
4	0	0,75	0	0,25	35	0	0,875	0	0,125
5	0,25	0,375	0,25	0,125	36	0,375	0	0,625	0
6	0	0,25	0,5	0,25	37	0	0,625	0	0,375
7	0,25	0,125	0,5	0,125	38	0,5	0	0,5	0
8	0,25	0	0,75	0	39	0,5	0,25	0	0,25
9	0,25	0,5	0,25	0	40	0,25	0,5	0	0,25
10	0	0,5	0,25	0,25	41	0	0,375	0,25	0,375
11	0	0,375	0,5	0,125	42	0	0,75	0,25	0
12	0	0,625	0,25	0,125	43	0,5	0,5	0	0
13	0,125	0,375	0,375	0,125	44	0,25	0,75	0	0
14	0,125	0,25	0,375	0,25	45	0,125	0	0,625	0,25

№	p_{001}	p_{101}	p_{201}	p_{301}	№	p_{001}	p_{101}	p_{201}	p_{301}
15	0,5	0,125	0,25	0,125	46	1	0	0	0
16	0,5	0,25	0,25	0	47	0,625	0,25	0,125	0
17	0,5	0	0,25	0,25	48	0,625	0,125	0,125	0,125
18	0,25	0	0,5	0,25	49	0,375	0,25	0,125	0,25
19	0,125	0,5	0,375	0	50	0,25	0,375	0	0,375
20	0,125	0,5	0,125	0,25	51	0,25	0,625	0	0,125
21	0,125	0,125	0,625	0,125	52	0,5	0,375	0	0,125
22	0,125	0,25	0,625	0	53	0,25	0,125	0,25	0,375
23	0,375	0,25	0,375	0	54	0,375	0	0,375	0,25
24	0,375	0,125	0,375	0,125	55	0	0	1	0
25	0	0,25	0,75	0	56	0	0	0,75	0,25
26	0	0,125	0,75	0,125	57	0	0,25	0,25	0,5
27	0,125	0,625	0,125	0,125	58	0	0	0,5	0,5
28	0,125	0,75	0,125	0	59	0	1	0	0
29	0,125	0,375	0,125	0,375	60	0	0,5	0	0,5
30	0,125	0,125	0,375	0,375	61	0,625	0	0,375	0
31	0,375	0,5	0,125	0					

На основе результатов, представленных в табл. 6.3-6.5, была произведена классификация УЭ $\mathbf{F}_{3/1}$ относительно вероятностей появления разностей любого веса Хэмминга $i = 0, 1, 2, 3$ при нулевой разности на управляющем входе и любой разности $j = 0, 1, 2, 3$ на входе данных. Результаты показаны в табл. 6.7.

Таблица 6.7

Типы УЭ $\mathbf{F}_{3/1}$ относительно $p(\Delta^Y_i/\Delta^X_j, \Delta^V_0)$

№	p_{010}	p_{110}	p_{210}	p_{310}	p_{020}	p_{120}	p_{220}	p_{320}	p_{030}	p_{130}	p_{230}	p_{330}
1	0	0,5833	0,3333	0,0833	0	0,3333	0,5	0,1667	0	0,25	0,5	0,25
2	0	0,625	0,3333	0,0417	0	0,3333	0,5	0,1667	0	0,125	0,5	0,375
3	0	0,4583	0,4167	0,125	0	0,4167	0,5	0,0833	0	0,375	0,25	0,375
4	0	0,5833	0,25	0,1667	0	0,25	0,6667	0,0833	0	0,5	0,25	0,25
5	0	0,625	0,25	0,125	0	0,25	0,6667	0,0833	0	0,375	0,25	0,375
6	0	0,5417	0,3333	0,125	0	0,3333	0,5	0,1667	0	0,375	0,5	0,125
7	0	0,4583	0,4167	0,125	0	0,4167	0,4167	0,1667	0	0,375	0,5	0,125

№	p_{010}	p_{110}	p_{210}	p_{310}	p_{020}	p_{120}	p_{220}	p_{320}	p_{030}	p_{130}	p_{230}	p_{330}
8	0	0,5	0,4167	0,0833	0	0,4167	0,4167	0,1667	0	0,25	0,5	0,25
9	0	0,4167	0,4167	0,1667	0	0,4167	0,5	0,0833	0	0,5	0,25	0,25
10	0	0,7083	0,1667	0,125	0	0,1667	0,6667	0,1667	0	0,375	0,5	0,125
11	0	0,5	0,4167	0,0833	0	0,4167	0,3333	0,25	0	0,25	0,75	0
12	0	0,5833	0,4167	0	0	0,4167	0,3333	0,25	0	0	0,75	0,25
13	0	0,5833	0,3333	0,0833	0	0,3333	0,4167	0,25	0	0,25	0,75	0
14	0	0,6667	0,1667	0,1667	0	0,1667	0,6667	0,1667	0	0,5	0,5	0
15	0	0,4167	0,5	0,0833	0	0,5	0,3333	0,1667	0	0,25	0,5	0,25
16	0	0,375	0,5	0,125	0	0,5	0,3333	0,1667	0	0,375	0,5	0,125
17	0	0,4583	0,5	0,0417	0	0,5	0,3333	0,1667	0	0,125	0,5	0,375
18	0	0,3333	0,5	0,1667	0	0,5	0,4167	0,0833	0	0,5	0,25	0,25
19	0	0,5417	0,25	0,2083	0	0,25	0,6667	0,0833	0	0,625	0,25	0,125
20	0	0,2917	0,5833	0,125	0	0,5833	0,3333	0,0833	0	0,375	0,25	0,375
21	0	0,5	0,3333	0,1667	0	0,3333	0,5	0,1667	0	0,5	0,5	0
22	0	0,4167	0,4167	0,1667	0	0,4167	0,4167	0,1667	0	0,5	0,5	0
23	0	0,5417	0,4167	0,0417	0	0,4167	0,5	0,0833	0	0,125	0,25	0,625
24	0	0,375	0,5	0,125	0	0,5	0,5	0	0	0,375	0	0,625
25	0	0,5417	0,4167	0,0417	0	0,4167	0,4167	0,1667	0	0,125	0,5	0,375
26	0	0,375	0,5	0,125	0	0,5	0,4167	0,0833	0	0,375	0,25	0,375
27	0	0,5	0,5	0	0	0,5	0,5	0	0	0	0	1
28	0	0,4583	0,5	0,0417	0	0,5	0,4167	0,0833	0	0,125	0,25	0,625
29	0	0,5	0,4167	0,0833	0	0,4167	0,5	0,0833	0	0,25	0,25	0,5
30	0	0,4167	0,5	0,0833	0	0,5	0,4167	0,0833	0	0,25	0,25	0,5
31	0	0,2917	0,5	0,2083	0	0,5	0,5	0	0	0,625	0	0,375
32	0	0,375	0,4167	0,2083	0	0,4167	0,5	0,0833	0	0,625	0,25	0,125
33	0	0,4167	0,3333	0,25	0	0,3333	0,6667	0	0	0,75	0	0,25
34	0	0,4583	0,3333	0,2083	0	0,3333	0,6667	0	0	0,625	0	0,375
35	0	0,3333	0,5	0,1667	0	0,5	0,3333	0,1667	0	0,5	0,5	0
36	0	0,5	0,25	0,25	0	0,25	0,6667	0,0833	0	0,75	0,25	0
37	0	0,2917	0,5	0,2083	0	0,5	0,4167	0,0833	0	0,625	0,25	0,125
38	0	0,25	0,5	0,25	0	0,5	0,5	0	0	0,75	0	0,25

Используя данные табл. 6.7, легко получить матрицу переходных вероятностей входных разностей $\mathbf{P}_0=\|p(\Delta^Y_i/\Delta^X_j, \Delta^V_0)\|$. Например, для третьего и тридцать восьмого типов УЭ матрицы переходных вероятностей входных разностей будут иметь, соответственно, следующий вид:

$$\mathbf{P}_0 = \begin{Vmatrix} 0 & 0 & 0 \\ 0,4583 & 0,4167 & 0,375 \\ 0,4167 & 0,5 & 0,25 \\ 0,125 & 0,0833 & 0,375 \end{Vmatrix}, \quad \mathbf{P}_0 = \begin{Vmatrix} 0 & 0 & 0 \\ 0,25 & 0,5 & 0,75 \\ 0,5 & 0,5 & 0 \\ 0,25 & 0 & 0,25 \end{Vmatrix}.$$

Очевидно, что на практике при разработке УППС, осуществляющих преобразование векторов большой разрядности, следует использовать УЭ, имеющие матрицы переходных вероятностей с их более равномерным распределением.

6.3. Построение управляемых операционных блоков

Осуществляя комбинирование только УЭ $\mathbf{F}_{3/1}$ или УЭ $\mathbf{F}_{3/1}$ с двухвходовыми УЭ $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$, можно синтезировать УППС $\mathbf{F}_{n/m} : \mathrm{GF}(2)^{n+m} \to \mathrm{GF}(2)^n$, выполняющие преобразование входных векторов различной разрядности, где n – число входных (выходных) бит, m – число управляющих бит. При этом разрядность входного вектора может быть как кратна, так и не кратна двум. Топология построения УППС может быть самой различной. При использовании только УЭ $\mathbf{F}_{3/1}$ основным принципом коммутации УЭ, принадлежащих двум соседним слоям УППС, является принцип «каждый с каждым» (рис. 6.4).

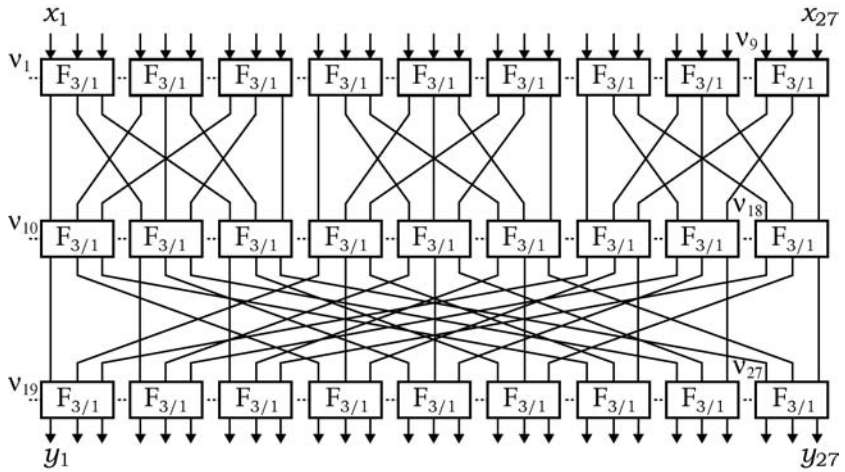

Рис. 6.4. Структура УППС $\mathbf{F}_{27/27}$

В этом случае УППС с $n = 3^k$ - битовым входом для данных имеет управляющий вектор с разрядностью $m = \frac{kn}{3}$, где k - количество подстановочных слоев. Следовательно, можно легко построить УППС, осуществляющие отображения вида $GF(2)^{n(k+3)/3} \to GF(2)^n$, например, $GF(2)^{15} \to GF(2)^9$, $GF(2)^{54} \to GF(2)^{27}$, $GF(2)^{189} \to GF(2)^{81}$.

Однородные блоки УППС $\mathbf{F}_{n/m}$ каскадного типа, построенные на основе элементарных подстановочных УЭ $\mathbf{F}_{3/1}$ и имеющие топологию, подобную той, что представлена на рис. 6.4, обладают следующими свойствами:

- Любая УППС $\mathbf{F}_{n/m}$, состоящая из k подстановочных слоев, где $k = \log_3 n$, управляется $m = \frac{kn}{3}$ разрядным вектором.

- При использовании биективной модели УЭ $\mathbf{F}_{3/1}$ в каждом узле УППС $\mathbf{F}_{n/m}$ преобразование $Y=\mathbf{F}_{n/m}(X, V) : GF(2)^{n+m} \to GF(2)^n$, где $X, Y \in GF(2)^n$, $V \in GF(2)^m$, будет биективным относительно X при фиксированных значениях управляющего вектора V и регулярным относительно множества входных и управляющих векторов;

- Если $f_i \in \Psi_n$, где Ψ_n – множество всех БФ, реализующих конкретный вид УППС, то для любого i=1, 2, …, n количество аргументов μ БФ f_i определяется выражением $\mu = n + \sum_{j=1}^{k} \frac{n}{3^j} = \frac{3n-1}{2}$.

- Если $f_i \in \Psi_n$, то алгебраическая степень нелинейности БФ $\deg(f_i)$ определяется выражениями

  $\deg(f_i) = 2^k + 1$, при $\deg(f^{\mathbf{F}_{3/1}}) = 2$ для всех выходов УЭ,

  $\deg(f_i) = 2^{k+1} - 1$, при $\deg(f^{\mathbf{F}_{3/1}}) = 3$ для всех выходов УЭ,

  где $\deg(f^{\mathbf{F}_{3/1}})$ – алгебраическая степень нелинейности функций, реализующих блок $\mathbf{F}_{3/1}$. В случае, если БФ, описывающие выходы УЭ, имеют различную алгебраическую степень нелинейности, получим: $2^k + 1 \le \deg(f_i) \le 2^{k+1} - 1$.

- Верхняя граница нелинейности БФ, реализующих УППС $\mathbf{F}_{n/m}$, определяется, в соответствии с (2.28), следующими выражениями

$$\mathrm{NL}(\mathbf{F}_{n/m}) \le \begin{cases} 2^{\frac{3}{2}(n-1)} - 2^{\frac{3n-5}{4}} - 2 \\ \left\lfloor 2^{\frac{3}{2}(n-1)} - 2^{\frac{3n-5}{4}} \right\rfloor \end{cases},$$

при $\mu = \dfrac{3n-1}{2}$ – четном и нечетном, соответственно.

Рассмотрим особенности развития лавинного эффекта в однородных УППС $\mathbf{F}_{n/m}$, использующих в качестве базового –УЭ $\mathbf{F}_{3/1}$. Здесь под однородностью понимается то обстоятельство, что в каждом узле УППС используется управляемый элемент только одного типа, имеющий одинаковые нелинейные и дифференциальные свойства.

Для получения производящей функции вероятностей появления разности веса $\boldsymbol{wt}(\Delta y)=k$, где $k\in\{1, 2, 3\}$, на выходе элементарного УЭ $\mathbf{F}_{3/1}$ при единичной разности $\boldsymbol{wt}(\Delta x)=1$ на входе, следует использовать данные табл. 6.3. Например, для УЭ, относящихся к 9 и 26 позициям табл. 6.3, производящие функции вероятностей имеют вид:

$$\varphi_2^{\mathbf{F}_{3/1}}(z) = \frac{z}{24}(17 + 4z + 3z^2),$$

$$\varphi_2^{\mathbf{F}_{3/1}}(z) = \frac{z}{4}(1 + 2z + z^2).$$

Производящие функции вероятностей появления разностей с различным весом Хэмминга $\varphi_n^{\mathbf{F}_{n/m}}(z)$ на выходе УППС $\mathbf{F}_{n/m}$ для случая $n = 27$, $m = 27$, получены итеративно от $\varphi_2^{\mathbf{F}_{3/1}}(z)$, а соответствующие им распределения вероятностей $p(\boldsymbol{wt}(\Delta y) / \boldsymbol{wt}(\Delta x) = 1, \boldsymbol{wt}(\Delta v) = 0)$ показаны на рис. 6.5. Все остальные типы УЭ, относящиеся к табл. 6.3, имеют распределения вероятностей и, соответственно вероятностные свойства, находящиеся между этими двумя позициями. Следовательно, указанные УЭ характеризуют наихудший и наилучший варианты построения УППС большой размерности с точки зрения разностных свойств.

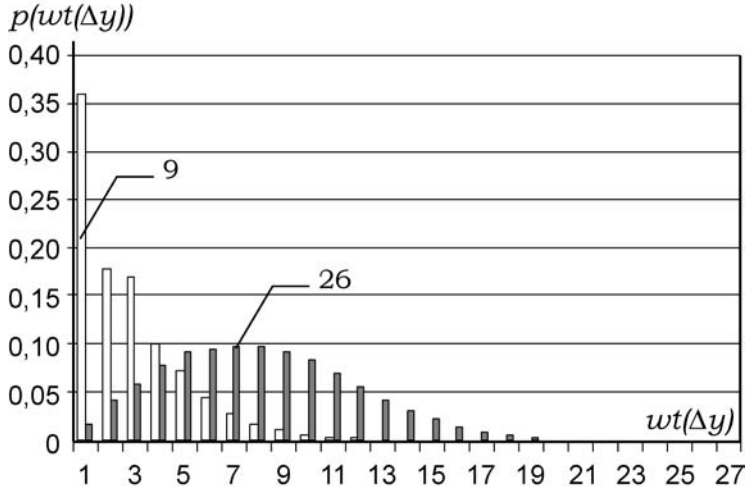

Рис. 6.5. Гистограммы распределения вероятностей появления разностей различного веса на выходе УППС $\mathbf{F}_{27/27}$

6.4. Особенности использования УОБ на основе УЭ $F_{3/1}$ в синтезе криптографических функций

Как было отмечено выше, определенные ограничения на использование УЭ типа $\mathbf{F}_{3/1}$ в синтезе шифров обусловливаются размером входа этого элемента. Тем не менее, элементы $\mathbf{F}_{3/1}$ обладают тем достоинством, что выходы УОБ, формируемых на их основе, являются булевыми функциями с нелинейными членами АНФ, представляющими собой произведение битов управляющего подблока данных и битов преобразуемого подблока. Кроме того, алгебраическая степень нелинейности выходов при заданном числе активных слоев существенно больше по сравнению с УОБ на основе УЭ типа $\mathbf{F}_{2/2}$ и типа $\mathbf{F}_{2/1}$. Эти моменты делают интересным рассмотрение криптографических механизмов, в которых трехбитовый вход УЭ не вносит существенных неудобств. Из таких механизмов можно назвать следующие:

- преобразование подключей в зависимости от одного из преобразуемых подблоков данных – подключи могут быть выбраны размером, превышающим размер подблоков данных и кратным трем;
- совместное преобразование подблоков данных и подключей;
- построение раундовой функции в схеме Фейстеля;
- разбиение преобразуемого блока данных на четыре подблока и преобразование на текущем шаге трех подблоков с помощью одного УОБ в зависимости от четвертого (управляющего) подблока;
- построение раундовой функции хэширования сжимающего типа.

Размеры и топологическая структура УОБ на основе УЭ $\mathbf{F}_{3/1}$, а также конкретный вариант последних могут быть выбраны в зависимости от конкретного механизма, где предполагается использовать УОБ, и от других механизмов и операций, входящих в криптографический алгоритм. Не вдаваясь в конкретные детали, поясним примерами варианты использования УОБ, построенных на основе элементов $\mathbf{F}_{3/1}$.

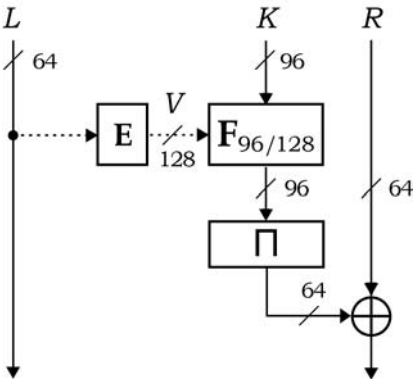

Рис. 6.6. Механизм преобразования подключей с помощью переменной операции (вариант использования УОБ, построенных из УЭ $\mathbf{F}_{3/1}$)

На рис. 6.6 показан механизм преобразования 96-битового подключа в зависимости от 64-битового подблока данных. Здесь предполагается, что блок $\mathbf{F}_{96/128}$ содержит четыре активных слоя, составленных из элементов $\mathbf{F}_{3/1}$, а П есть сжимающая операция, реализующая выбор 64 из 96 битов (схемно выполняется как простое соединение проводников). В четырехслойном блоке $\mathbf{F}_{96/128}$ может быть обеспечено влияние каждого входного бита на 81 выходных. Каждый бит подблока данных может быть использован для управления двумя УЭ блока $\mathbf{F}_{96/128}$. Распределение управляющих битов осуществляется блоком расширения **E**.

На рис. 6.7 показан механизм совместного преобразования 32-битового подблока данных и 64-битового подключа в зависимости от 32-битового подблока данных. В этом механизме также предполагается использовать упомянутый выше четырехслойный блок $\mathbf{F}_{96/128}$ и сжимающую операцию П. Каждый бит 32-битового управляющего подблока может быть использован для управления четырьмя элементами $\mathbf{F}_{3/1}$. Для равномерности перемешивания битов подключа с битами подблока данных L' на каждый из тридцати двух УЭ первого активного слоя УОБ можно подать два бита подключа с одним битом подблока L'.

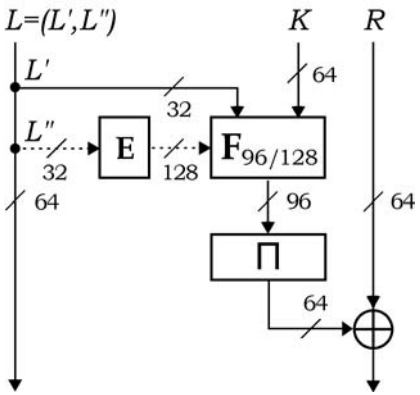

Рис. 6.7. Механизм совмещенного преобразования подключа и подблока данных с помощью переменной операции

Преобразование подключей в зависимости от данных можно рассматривать как наиболее простой частный случай построения раундовой функции криптосхемы Фейстеля. На рис. 6.8 представлен явный вариант построения раундовой функции. Особенностью данной раундовой функции является предварительное расширение подблока данных L путем присоединения к нему суммы $L' \oplus L''$ младшей и старшей частей L. Для формирования управляющего вектора V можно использовать конкатенацию левого подблока и подключа, что дает разновидность рассматриваемой схемы. Совместное использование подблока данных и подключа для формирования управляющих векторов разумно использовать при сравнительно большом числе активных слоев.

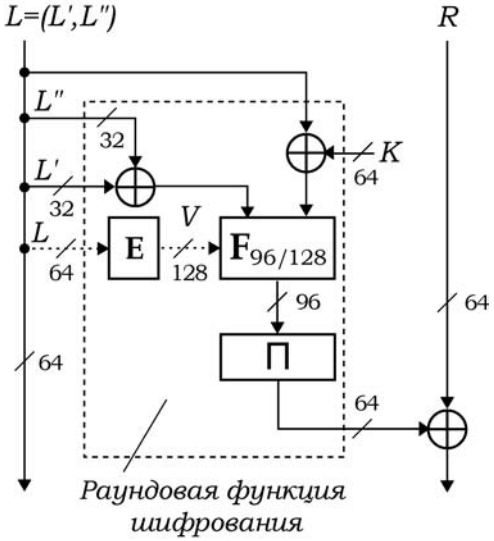

Рис. 6.8. Построение раундовой функции в криптосхеме Фейстеля

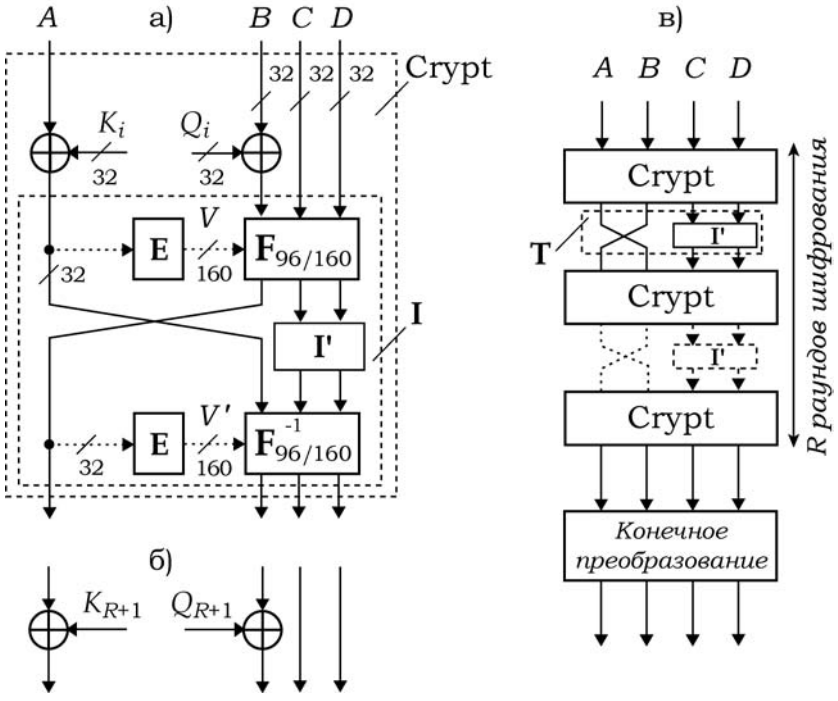

Рис. 6.9. Итеративный шифр на основе УОБ $\mathbf{F}_{96/160}$ и $\mathbf{F}^{-1}_{96/160}$:
а – структура раунда шифрования **Crypt,** б – конечное преобразование,
в – общая схема шифрования

Представляют интерес криптосхемы с разбиением входного блока данных на четыре подблока *A*, *B*, *C* и *D*, которое позволяет из трех подблоков сформировать один с размером входа кратным трем. Объединенный подблок данных (*B*, *C*, *D*) может быть преобразован с помощью УОБ, синтезированного на основе УЭ $\mathbf{F}_{3/1}$, в зависимости от подблока *A* меньшего размера. Обобщенная схема такого способа шифрования иллюстрируется на рис. 6.9, где в качестве УОБ используются два взаимно обратных пятислойных блока $\mathbf{F}_{96/160}$ и $\mathbf{F}^{-1}{}_{96/160}$, в которых число слоев является достаточным, чтобы обеспечить влияние каждого входного бита на каждый выходной бит. Подключи накладываются в начале каждого раунда только на два подблока: *A* и *B*. Особенностью данной универсальной криптосхемы является отсутствие начального преобразования при наличии конечного. В качестве перестаночной инволюции $\mathbf{I}'$ можно взять перестановку, соединяющую каскад из восьми блоков $\mathbf{P}_{8/12}$ с каскадом из восьми блоков $\mathbf{P}^{-1}{}_{8/12}$ в БУП $\mathbf{P}_{64/192}$ (см. рис. 8.17, раздел 8.3).

Необходимость использования конечного преобразования обусловлена тем, что один раунд шифрования не является инволюцией. Конечное преобразование обеспечивает свойство универсальности криптосхемы (т. е. возможность осуществлять как зашифрование, так и расшифрование). Заметим, что после выполнения преобразования объединенного подблока (*B*, *C*, *D*) достаточно переставить только третью его часть, а именно подблок *B*, с управляющим подблоком *A*, поскольку каждый выходной бит УОБ $\mathbf{F}_{96/160}$ является функцией от всех входных битов. По этой же причине достаточно наложить в начале раунда подключи только на подблоки *A* и *B* (наложение подключа именно на подблок *A* является важным, поскольку это делает неизвестным значение управляющего вектора при выполнении самой первой управляемой операции). Следует отметить, что выполнение обратной управляемой операции непосредственно после осуществления прямой операции и транспозиции подблоков *A* и *B* вносит «полновесный криптографический» эффект, поскольку операции $\mathbf{F}_{96/160}$ и $\mathbf{F}^{-1}{}_{96/160}$ контролируются независимыми управляющими векторами.

Легко показать, что корректное расшифрование в данной универсальной *R*-раундовой криптосхеме обеспечивается таким изменением расписания использования ключа, при котором выполняются следующие соотношения:

$$K'_1 = Q_{R+1}; \quad K'_j = Q_{R-j+2} \text{ для } j = 2, 3, \ldots, R; \quad K'_{R+1} = K_1;$$

$$Q'_1 = K_{R+1}; \quad Q'_j = K_{R-j+2} \text{ для } j = 2, 3, \ldots, R; \quad Q'_{R+1} = Q_1.$$

Действительно, обозначим преобразование блока данных (*A*, *B*, *C*, *D*), заключающееся в наложение на подблок *A* (*B*) подключа K_j (Q_j) с помощью операции поразрядного суммирования по модулю два, как функцию преобразования $\mathbf{A}_j$ ($\mathbf{B}_j$), т. е. имеем: $\mathbf{A}_j(A, B, C, D) = (A \oplus K_j, B, C, D)$ и $\mathbf{B}_j(A, B, C, D) = (A \oplus Q_j, B, C, D)$. Заметим, что раунд шифрования **Crypt** представляет собой следующую суперпозицию преобразований: $\mathbf{Crypt} = \mathbf{A}_j \circ \mathbf{B}_j \circ \mathbf{I}$, где **I** – преобразование, являющееся инволюцией. Процедура зашифрования может быть записана в следующем виде:

$$\mathbf{Crypt}^R = (\mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I}) \circ \mathbf{T} \circ (\mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I}) \circ \mathbf{T} \circ \ldots \circ (\mathbf{A}_R \circ \mathbf{B}_R \circ \mathbf{I}) \circ (\mathbf{A}_{R+1} \circ \mathbf{B}_{R+1}).$$

Процедура расшифрования (процедуры, отличающиеся от соответствующих процедур в режиме шифрования, отмечены штрихом) может быть записана в следующем виде:

$$\mathbf{Crypt}'^{R} = (\mathbf{A}'_1 \circ \mathbf{B}'_1 \circ \mathbf{I}) \circ \mathbf{T} \circ (\mathbf{A}'_2 \circ \mathbf{B}'_2 \circ \mathbf{I}) \circ \mathbf{T} \circ \ldots \circ (\mathbf{A}'_R \circ \mathbf{B}'_R \circ \mathbf{I}) \circ (\mathbf{A}'_{R+1} \circ \mathbf{B}'_{R+1}).$$

Процедура, включающая зашифрование и последующее расшифрование, записывается в следующем виде:

$$\mathbf{Crypt}^{R} \circ \mathbf{Crypt}'^{R} = \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots$$

$$\ldots \circ \mathbf{A}_R \circ \mathbf{B}_R \circ \mathbf{I} \circ (\mathbf{A}_{R+1} \circ \mathbf{B}_{R+1} \circ \mathbf{A}'_1 \circ \mathbf{B}'_1) \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}'_2 \circ \mathbf{B}'_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}'_R \circ \mathbf{B}'_R \circ \mathbf{I} \circ \mathbf{A}'_{R+1} \circ \mathbf{B}'_{R+1}.$$

Учитывая связь между подключами в режимах зашифрования и расшифрования, можно показать, что суперпозиция $\mathbf{A}_{R+1} \circ \mathbf{B}_{R+1} \circ \mathbf{A}'_1 \circ \mathbf{B}'_1$ представляет собой тождественное преобразование **O**, а также для $j = 2, 3, \ldots, R$ выполняется условие $\mathbf{A}_{R-j+2} \circ \mathbf{B}_{R-j+2} \circ \mathbf{T} \circ \mathbf{A}'_j \circ \mathbf{B}'_j = \mathbf{T}$. Теперь можно выполнить преобразования:

$$\mathbf{Crypt}^{R} \circ \mathbf{Crypt}'^{R} = \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots$$

$$\ldots \circ \mathbf{A}_R \circ \mathbf{B}_R \circ (\mathbf{I} \circ \mathbf{I}) \circ \mathbf{T} \circ \mathbf{A}'_2 \circ \mathbf{B}'_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}'_R \circ \mathbf{B}'_R \circ \mathbf{I} \circ \mathbf{A}'_{R+1} \circ \mathbf{B}'_{R+1} =$$

$$= \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ (\mathbf{A}_R \circ \mathbf{B}_R \circ \mathbf{T} \circ \mathbf{A}'_2 \circ \mathbf{B}'_2) \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}'_R \circ \mathbf{B}'_R \circ \mathbf{I} \circ \mathbf{A}'_{R+1} \circ \mathbf{B}'_{R+1} =$$

$$= \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}_{R-1} \circ \mathbf{B}_{R-1} \circ (\mathbf{I} \circ (\mathbf{T} \circ \mathbf{T}) \circ \mathbf{I}) \circ \mathbf{T} \circ \mathbf{A}'_3 \circ \mathbf{B}'_3) \circ \mathbf{I} \circ \mathbf{T} \circ \ldots =$$

$$= \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}_{R-2} \circ \mathbf{B}_{R-2} \circ \mathbf{I} \circ \mathbf{T} \circ (\mathbf{A}'_{R-1} \circ \mathbf{B}'_{R-1} \circ \mathbf{T} \circ \mathbf{A}'_4 \circ \mathbf{B}'_4) \circ \mathbf{I} \circ \mathbf{T} \circ \ldots =$$

$$= \mathbf{A}_1 \circ \mathbf{B}_1 \circ \mathbf{I} \circ \mathbf{T} \circ \mathbf{A}_2 \circ \mathbf{B}_2 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots \circ \mathbf{A}_{R-2} \circ \mathbf{B}_{R-2} \circ (\mathbf{I} \circ (\mathbf{T} \circ \mathbf{T}^{\circ}) \circ \mathbf{I}) \circ \mathbf{T} \circ \mathbf{A}'_4 \circ \mathbf{B}'_4 \circ \mathbf{I} \circ \mathbf{T} \circ \ldots = \ldots = \mathbf{O}.$$

Таким образом, при заданном секретном ключе расшифрование выполняется корректно.

В *главе 7* рассмотрен вопрос проектирования переключаемых управляемых операций различного типа, реализуемых с использованием УППС различного типа. Переключаемыми называются операционные блоки, реализующие управляемые операции и имеющие дополнительный управляющий одноразрядный вход, на который подается бит режима выполнения операции e (при $e = 0$ выполняется прямая управляемая операция, а при $e = 1$ – соответствующая ей обратная управляемая операция). Не представляет сложности разработать переключаемый операционный блок $\mathbf{F}^{(e)}_{96/160}$, используя который можно упростить рассмотренную выше итеративную схему шифрования как это показано на рис. 6.10. При этом при шифровании можно использовать «зеркально симметричное» расписание использования ключа, которое удовлетворяет следующим соотношениям между раундовыми подключами K_j и Q_j:

$$Q_1 = Q_{R+1}; \quad K_1 = K_{R+1}; \quad K_j = Q_{R-j+2} \text{ для } j = 2, 3, \ldots, R.$$

В этом случае корректность шифрования обеспечивается только за счет инвертирования бита e. Расписание использования подключей остается одинаковым при смене режима шифрования. При этом проблема возникновения слабых и полуслабых ключей не возникает, благодаря тому что алгоритмы зашифрования и расшифрования являются различными, т. е. сама процедура шифрования не является инволюцией

(обычно в универсальных криптосхемах это достигается сменой очередности расписания ключа). Сохранение одинакового расписания использования ключа вносит дополнительное упрощение аппаратной реализации шифров на основе переменных операций.

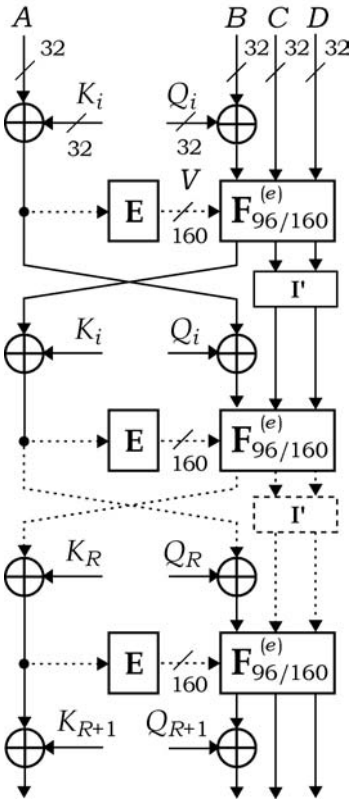

Рис. 6.10. Итеративный шифр на основе переключаемого УОБ $\mathbf{F}^{(e)}_{96/160}$

6.5. Сравнительная характеристика УЭ $F_{2/2}$ и $F_{3/1}$

Переход от минимальных УЭ типа $\mathbf{F}_{2/1}$ к элементам $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$ связан с практической потребностью наиболее полного использования потенциала типовых логических модулей программируемых логических матриц с сохранением высокой эффективности реализации на заказных СБИС. Использование УЭ $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$ с целью синтеза УППС с одинаковой полнотой решают данную задачу, однако различная длина двоичного вектора, преобразуемого этими элементами, приводит к ряду особенностей, которые вводят определенную специфику в построение УППС и проектирование шифров. Представленная ниже табл. 6.8 дает обобщенное сопоставление УЭ $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$.

Таблица 6.8

Характеристика	Элемент $F_{2/2}$	Элемент $F_{3/1}$
Размер преобразуемого двоичного вектора входа	Двухбитовый вход определяет: • реализацию линейных подстановок размера 2×2, реализуемых как модификации УЭ • легкость построения однородных УППС с размером входа равным натуральной степенью числа 2, что гармонично сочетается со стандартным разбиением блока данных в блочных шифрах на два 64-битовых или 128-битовых подблока	Трехбитовый вход определяет: • возможность реализации нелинейных подстановок минимального размера 3×3, реализуемых как модификации УЭ • однородные УППС имеют размер входа не равный натуральной степени числа 2, что вносит некоторые ограничения в синтез шифров со стандартным размером входа (64 и 128 бит)
Управляющий вектор	Нелинейные операции реализуются только в том случае, если управляющий вектор зависит от одного из подблоков данных	Нелинейные операции могут быть построены как при задании зависимости управляющего вектора от данных, так и в случае, когда он зависит только от ключа
Размер управляющего входа	Двухбитовый управляющий вход задает на текущем шаге выбор одной из четырех потенциально реализуемых модификаций линейных подстановок размера 2×2 Длина управляющего вектора УППС равна $m = kn$, где s – число активных слоев и n – размер входа УППС. Сравнительно большое значение m позволяет задать более эффективное распространение лавинного эффекта через управляющий вход	Наличие только одного управляющего бита обеспечивает реализацию только двух различных модификаций для некоторого фиксированного варианта УЭ, представляющих собой в общем случае подстановки размера 3×3 Длина управляющего вектора УППС равна $kn/3$, где s – число активных слоев и n – размер входа УППС
Структура старшего члена АНФ БФ, реализующей один из выходов	Старший член АНФ представляет собой произведение одного входного бита и двух управляющих битов	Старший член АНФ представляет собой произведение одного управляющего бита и двух входных битов
Число различных вариантов УЭ	Существует $(4!)^4 - 4! = 331752$ формально различных линейных и нелинейных УЭ, реализующих, по крайней мере, две различные модификации подстановок 2×2	Существует $8!^2 - 8! = 1\,625\,662\,080$ формально различных линейных и нелинейных УЭ, реализующих, по крайней мере, две различные модификации подстановок 3×3
Число БФ, описывающих УЭ	Две	Три
Старший член АНФ БФ, реализующей выход k-слойногоУППС первого порядка	Старший член АНФ представляет собой произведение одного входного бита и $2k$ управляющих битов	Старший член АНФ представляет собой произведение $2^k - 1$ управляющих битов и 2^k входных битов
БФ, реализующая выход k-слойной УППС первого порядка	Зависит от $3 \cdot 2^k - 2$ переменных (в случае независимых управляющих битов)	Зависит от $\frac{3 \cdot 3^k - 1}{2}$ переменных (в случае независимых управляющих битов)
Построение переключаемых УОБ	Существует сравнительно ограниченное число нелинейных УЭ, реализующих пары взаимно обратных модификаций	Существует сравнительно большое число нелинейных УЭ, реализующих пару взаимно обратных модификаций

ГЛАВА 7

Переключаемые управляемые операции

7.1. Построение управляемых подстановочно-перестановочных сетей различного порядка

Свойства управляемых операций, построенных на основе УППС, могут быть охарактеризованы как с привязкой к применению в качестве криптографических примитивов, так и без таковой. В первом случае, представляющем для нас наибольший интерес, они зависят от следующих факторов:

- свойств управляемых элементов, используемых для построения сети;
- однородности сети;
- топологии сети, описывающей связи управляемых элементов между собой и с управляющим входом;
- размеров информационного и управляющего входа операционного блока;
- размера управляющего подблока данных.

Синтез операционных блоков, предназначенных для использования в алгоритмах шифрования, целесообразно осуществлять с учетом перечисленных факторов. При их реализации в качестве исходного критерия можно использовать задание влияния каждого входного бита на каждый выходной. В случае управляемых перестановок это реализуется для БУП первого порядка. Как показано в *главе 4*, замена каждого элементарного переключателя в БУП первого порядка $\mathbf{P}_{n/m}$ на любой управляемый элемент (УЭ) из множества $\{\mathbf{R}_{2/1}, \mathbf{S}_{2/1}, \mathbf{Z}_{2/1}\}$ обеспечивает построение (УППС) $\mathbf{R}_{n/m}$, $\mathbf{S}_{n/m}$ и $\mathbf{Z}_{n/m}$, удовлетворяющих исходному критерию. Этот пример показывает, что топологические структуры БУП различного типа могут быть взяты в качестве прототипа при построении УППС $\mathbf{F}_{n/m}$ различных видов.

При построении БУП для криптографических применений представляет интерес понятие порядка управляемой перестановки. Поэтому целесообразно расширение этого понятия на УППС типа $\mathbf{F}_{n/m}$, построенные на основе УЭ с двухразрядным входом $\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$. В случае перестановки битов понятие порядка определено с учетом вариантов размещения заданного количества входных битов в заданное количество выходных разрядов БУП. В случае построения сетей на основе УЭ, отличающихся от элементарного переключателя, такая непосредственная физическая трактовка понятия порядка смазывается присутствием операций поразрядного суммирования по модулю два и инверсии внутри УЭ. Поэтому для блоков типа $\mathbf{F}_{n/m}$ понятие порядка следует дать в более общем виде, используя понятие не физического перемещения

битов, а распространения влияния. В свою очередь распространение влияния одного входного бита для некоторого заданного фиксированного значения управляющего вектора V можно интерпретировать как прохождение однобитовой разности через УППС, причем может иметь место размножение разностей. В последнем случае можно сказать, что при заданном значении V рассматриваемый входной разряд (или просто вход) влияет на такие-то выходные разряды или выходы. Влияние может быть показано как распространение однобитовых разностей Δ_1 и δ_1. На рис. 7.1 показан характерный тип схемы распространения влияния одного (правого и левого) входа в элементарных блоках $\mathbf{R}_{2/1}$ и $\mathbf{S}_{2/1}$. Из данных схем видны типичные особенности: в УЭ типа $\mathbf{R}_{2/1}$ влияние правого (распространение разности Δ_1) и левого входа (распространение разности δ_1) несимметрично, а в $\mathbf{S}_{2/1}$ – симметрично.

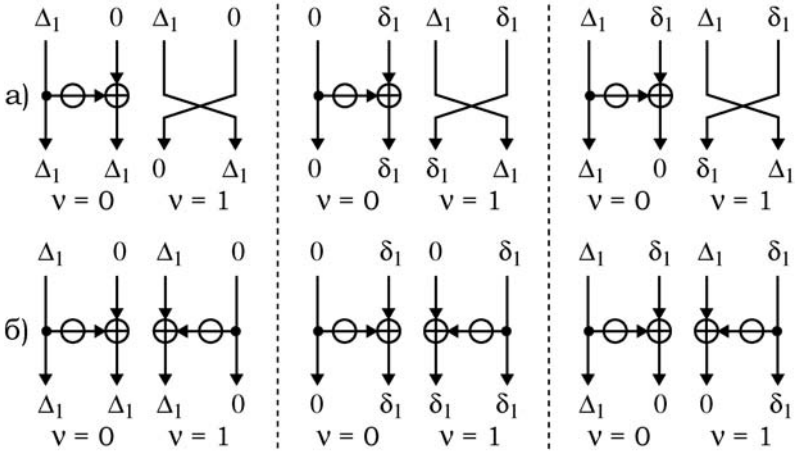

Рис. 7.1. Характерная схема распространения влияния входных битов в УЭ типа $\mathbf{R}_{2/1}$ (а) и $\mathbf{S}_{2/1}$ (б)

Из схемы распространения влияния двух выходов в элементарных блоках $\mathbf{R}_{2/1}$ и $\mathbf{S}_{2/1}$ (см. рис. 7.1) видно, что проявляется еще одна особенность, которая заключается в том, что в блоке $\mathbf{R}_{2/1}$ влияние каждого входа может распространяться в перекрестных направлениях в зависимости от значения управляющего бита, тогда как в УЭ $\mathbf{S}_{2/1}$ левый (правый) вход влияет только на левый (правый) выход, если правый (левый) вход влияет на левый (правый) выход. Это показывает, что УЭ $\mathbf{R}_{2/1}$ более сходны с $\mathbf{P}_{2/1}$, чем с $\mathbf{S}_{2/1}$. В случае УЭ $\mathbf{R}_{2/1}$ влияние левого и правого входов могут переставляться в зависимости от значения управляющего бита, также как и в $\mathbf{P}_{2/1}$, что позволяет сделать предположение о целесообразности введения понятия порядка для элементарных блоков $\mathbf{R}_{2/1}$.

Для УППС $\mathbf{R}_{n/m}$ порядок имеет следующий смысл. Пусть задано произвольное соответствие k входов $x_{\alpha_1}, x_{\alpha_2}, \ldots, x_{\alpha_k}$ и k выходов $y_{\alpha_1}, y_{\alpha_2}, \ldots, y_{\alpha_k}$:

$$x_{\alpha_1} \leftrightarrow y_{\beta_1}, \ldots, x_{\alpha_i} \leftrightarrow y_{\beta_i}, \ldots, x_{\alpha_k} \leftrightarrow y_{\beta_k}$$

(знак "↔" обозначает коммутацию (соответствие) пары входов и выходов). Пусть существует значение управляющего вектора такое, что для всех $i = 1, 2, \ldots, k$, при фиксировании всех входов, кроме x_{α_i}, инвертирование x_{α_i} приводит к инвертированию y_{β_i}. При этом может происходить или не происходить инвертирование других выходов, включая выходы из набора $y_{\alpha_1}, y_{\alpha_2}, \ldots, y_{\alpha_k}$. Максимальное значение k, для которого это условие выполняется, может быть принято за значение порядка h. Если принять такую трактовку порядка УППС **R**-типа, то замена всех УЭ $\mathbf{P}_{2/1}$ в некотором БУП $\mathbf{P}_{n/m}$ порядка h на элементы $\mathbf{R}_{2/1}$ приводит к формированию УППС $\mathbf{R}_{n/m}$ порядка h. В случае указанной замены мы будем говорить, что БУП и УППС **R**-типа имеют одинаковую топологическую структуру (или просто: одинаковую топологию). Аналогия УППС типа $\mathbf{S}_{n/m}$ с БУП менее очевидна, но с целью однообразия рассмотрения различных типов УППС мы примем следующее определение порядка УППС типа $\mathbf{F}_{n/m}$:

Определение 7.1

Пусть дана УППС типа $\mathbf{F}_{n/m}$. Блок $\mathbf{F}_{n/m}$ имеет порядок h, если он имеет одинаковую топологию с некоторым БУП порядка h.

При распространении влияния входов на выходы важным свойством является существование значения V, обеспечивающего влияние данного входа на заданное число произвольно выбираемых выходов. Чем больше таких выходов, тем более выраженным является лавинный эффект.

Во многих криптосхемах на основе управляемых операций используются взаимно-обратные операционные блоки. Также как и в случае построения БУП для любой УППС типа $\mathbf{F}_{n/m}$ легко может быть синтезирован соответствующий ему обратный блок $\mathbf{F}^{-1}{}_{n/m}$. В отличие от построения обратных БУП, где требовалось только применение обратных фиксированных перестановок, в общем случае построение блоков $\mathbf{F}^{-1}{}_{n/m}$ требует также использования обратных активных слоев, т. е. активных слоев составленных из обратных УЭ. На рис. 7.2 показана общая схема перехода от прямого к обратному блоку.

Наиболее интересные подклассы УЭ, а именно $\{\mathbf{R}_{2/1}\}$ и $\{\mathbf{S}_{2/1}\}$, для каждого конкретного вида УЭ включают и обратный ему УЭ. Это означает, что прямой и обратный блоки будут в определенном смысле равноценны. В случае УППС, построенной на основе УЭ $\mathbf{Z}_{2/1}$, переход к обратным блокам связан в общем случае с изменением дифференциальных и нелинейных свойств, причем в зависимости от варианта УЭ $\mathbf{Z}_{2/1}$ свойства элементов $\mathbf{Z}^{-1}{}_{2/1}$ оказываются различными даже в пределах подклассов $\{\mathbf{Z}'_{2/1}\}$, $\{\mathbf{Z}''_{2/1}\}$, $\{\mathbf{Z}^{*}{}_{2/1}\}$, $\{\mathbf{Z}^{**}{}_{2/1}\}$. В дальнейшем нас будет интересовать главным образом построение УППС на основе нелинейных УЭ.

Сказанное выше показывает, что топологические структуры, разработанные для БУП, могут быть использованы и для разработки УППС типов $\mathbf{R}_{n/m}$ и $\mathbf{S}_{n/m}$. В связи с этим интерес представляет построение УППС первого порядка на основе рекурсивной схемы построения первого типа [10, 16], показанной на рис. 7.3а, и соответствующих им обратных УППС первого порядка на основе рекурсивной схемы построения второго типа.

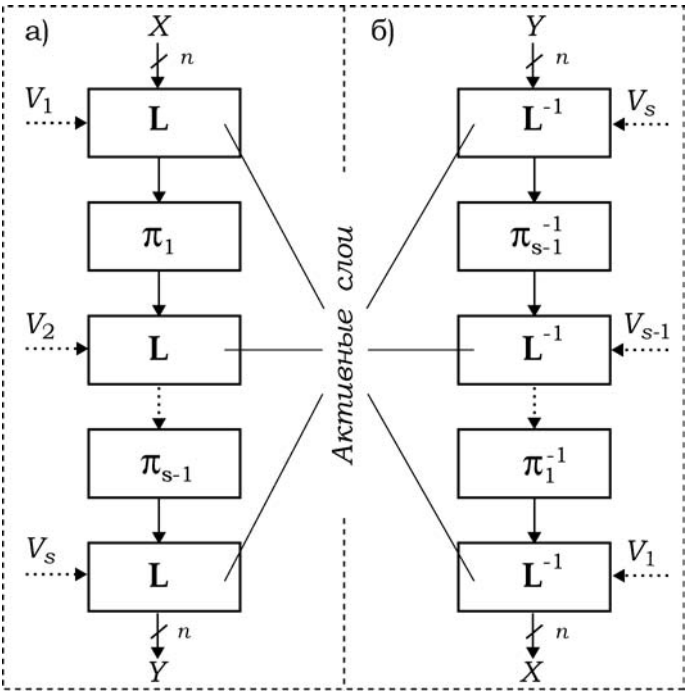

Рис. 7.2. Структура взаимно-обратных блоков $\mathbf{F}_{n/m}$ (а) и $\mathbf{F}^{-1}_{n/m}$ (б)

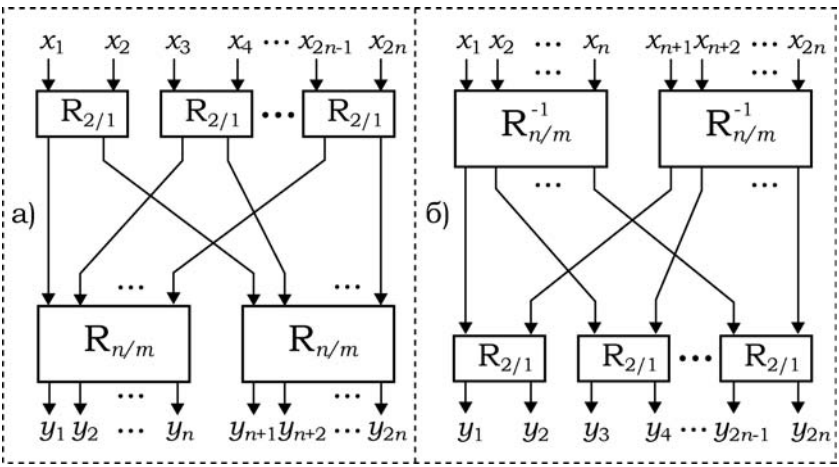

Рис. 7.3. Построение УППС первого порядка $\mathbf{R}_{2n/2m+n}$ (а) и $\mathbf{R}^{-1}_{2n/2m+n}$ (б) с использованием блоков первого порядка $\mathbf{R}_{n/m}$ (а) и $\mathbf{R}^{-1}_{n/m}$ (б)

Рассматривая данную пару рекурсивных конструктивных схем, легко показать, что минимальное число слоев, необходимых для реализации УППС $\mathbf{F}_{n/m}$ первого порядка, составляет

$$s' = \log_2 n.$$

Возможно также построение УППС $\mathbf{F}_{n/m}$ порядков 2, 4, …, $n/4$ и n на основе схемы рекурсивного построения с удвоением порядка (рекурсивная схема третьего типа) [16], представленной на рис. 7.4 для случая блоков **R**-типа. Последняя схема рекурсивного построения применена в указанной работе для построения БУП различных порядков для случая $n = 32$. По аналогии эта схема может быть применена для произвольного значения $n = 2^k$, где k – натуральное число. Однако формально это не было показано. Покажем формально построение УППС **R**-типа для случая произвольного k (это построение с очевидностью распространяется и на построение соответствующих БУП).

Пусть надо построить блок с размером входа $n = 2^k$ и порядком $h = 2^q$, где $q < k$. Возьмем блок первого порядка $\mathbf{R}_{2^{k-q}/m_0/1}$, где принято обозначение $\mathbf{R}_{n/m/h}$, в котором индекс h обозначает значение порядка УППС. Выполнив q последовательных шагов рекурсивного построения с удвоением порядка (по схеме, показанной на рис. 7.4), получим

$$\mathbf{R}_{2^{k-q}/m_0/1} \rightarrow \mathbf{R}_{2^{k-q+1}/m_1/2} \rightarrow \mathbf{R}_{2^{k-q+2}/m_2/2^2} \rightarrow \ldots$$

$$\rightarrow \mathbf{R}_{2^{k-q+i}/m_i/2^i} \rightarrow \ldots \rightarrow \mathbf{R}_{2^k/m_q/2^q}.$$

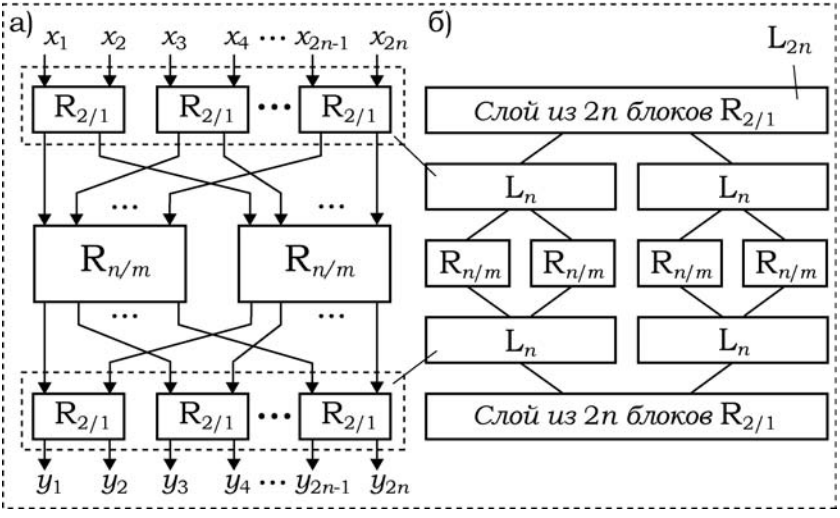

Рис. 7.4. Структура УППС $\mathbf{R}_{2n/2m+2n}$ (а) и $\mathbf{R}_{4n/4m+8n}$ (б)

Таким образом, мы выполнили построение требуемой УППС. Значения m_i после каждого шага выполненного рекурсивного построения легко определяются. Число слоев в построенной УППС равно числу слоев в исходной УППС первого порядка $\mathbf{R}_{2^{k-q}/m/1}$ ($s' = k - q$) плюс удвоенное число шагов рекурсивного построения ($2q$), что составляет

$$s = k + q = \log_2 n + \log_2 h = \log_2 nh.$$

Последняя формула позволяет определить число слоев для произвольных значений n и $h \le n/4$. Случай $h = n/2$ не представляет практического интереса, поскольку он требует использования такого же числа активных слоев, что и в случае УППС порядка $h = n$, для которого имеем $S = 2\log_2 n - 1$. Эта особенность легко объяснима, поскольку в случае порядка $h = n/2$ в рассмотренном способе формального построения требуется использование начальной УППС с двухбитовым входом, который имеет порядок $h = 2$, поскольку он по определению управляемости должен реализовывать хотя бы две различные перестановки входных битов, а их существует всего две. С использованием такого начального блока (например, УЭ $\mathbf{F}_{2/1}$) на каждом шаге рекурсии реализуется блок максимального порядка.

Примеры УППС $\mathbf{F}_{n/m}$ порядков $h = 1, 2, \ldots, n/4, n$ показаны на рис. 7.5 для случая $n = 32$. Начальный блок рекурсивного построения с удвоением порядка выделен пунктирной рамкой. Рассмотренные схемы построения являются универсальными для размера входа равного натуральной степени числа два. Однако в ряде частных случаев предпочтительно использовать другие варианты топологий УППС, которые обладают зеркальной симметрией и более выраженной структурированностью, что упрощает их схемную реализацию. Кроме того, для симметричных блоков упрощается построение механизма формирования управляющих векторов, удовлетворяющих некоторым заданным критериям. Использование симметричной структуры приводит к наиболее значимому упрощению в случае построения переключаемой УППС.

Пояснение. Переключаемой называется УППС, которая может реализовать как прямую управляемую операцию, так и обратную ей — в зависимости от значения некоторого дополнительного управляющего бита.

Ниже будут раскрыты причины, обусловливающие перспективность применения переключаемых управляемых операций для синтеза блочных шифров. Также будут описаны различные варианты их построения на основе УЭ различных типоразмеров. При этом прослеживается важная роль симметричных топологий УППС и некоторые отличия этого понятия для случаев использования различных механизмов задания свойства переключаемости (которое может быть также названо свойством обращаемости). Например, при использовании механизма обращаемости за счет инвертирования управляющих битов операционные блоки строятся с использованием УЭ, множество элементарных модификаций которых разбивается на пары взаимно-обратных модификаций, а под симметричными (или зеркально симметричными) УППС понимаются следующие.

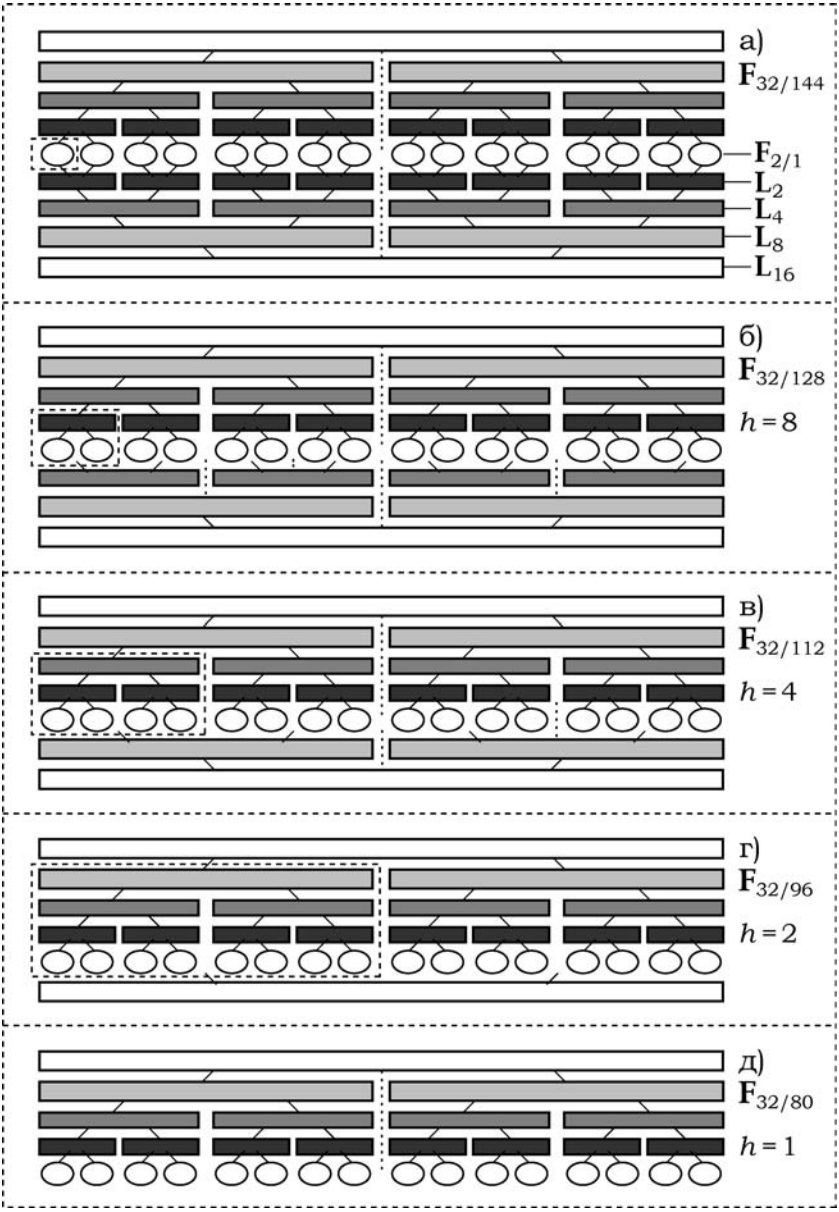

Рис 7.5. Управляемые операционные блоки $\mathbf{F}_{32/m}$ различных порядков
а) $h = 32$, б) $h = 8$, в) $h = 4$, г) $h = 2$, д) $h = 1$

Определение 7.2

Симметричной называется УППС, для которой при всех $i = 1, 2, \ldots, s-1$ выполняются следующие соотношения: $\mathbf{L}_i = \mathbf{L}_{s-i+1}$ и $\pi_i = (\pi_{s-i})^{-1}$.

В другом варианте построения обращаемая УППС переключение обеспечивается за счет перестановки управляющих векторов, соответствующих определенным парам активных каскадов. В этом случае под симметричной понимаются УППС, структура которых удовлетворяет следующему определению.

Определение 7.3

Симметричной называется УППС, для которой при всех $i = 1, 2, \ldots, s-1$ выполняются следующие соотношения: $\mathbf{L}_i = \mathbf{L}^{-1}_{s-i+1}$ и $\pi_i = (\pi_{s-i})^{-1}$.

Будут рассмотрены и другие (несимметричные топологии) переключаемые УППС, но наиболее экономичные с точки зрения аппаратных затрат решения обеспечиваются с применением УППС с симметричной топологией. Достаточно интересным представляется тот факт, что некоторые УППС могут быть реализованы как в симметричной, так и несимметричной структуре, причем оба варианта построения требуют использования одинакового количества активных слоев и обеспечивают сходные дифференциальные и нелинейные свойства, в частности, одинаковую алгебраическую степень нелинейности. Это показывает, что реализация дополнительного свойства симметричности осуществляется не в ущерб криптографическим качествам УППС в тех случаях, когда свойство симметричности органично сочетается с входным размером синтезируемого блока.

В работе [14] показано, что при произвольном k БУП первого порядка, синтезированные по первой схеме рекурсивного построения, реализуют такое множество модификаций перестановок, которое разбивается на пары взаимно-обратных модификаций. Это означает, что имеется принципиальная возможность построения переключаемых БУП первого порядка для произвольных размеров входа $n=2^k$. Если УППС первого порядка построена по первой рекурсивной схеме с использованием типового управляемого элемент $\mathbf{F}_{2/1}$, являющегося инволюцией, то при произвольном k данная УППС реализует множество модификаций управляемой операции, которое разбивается на пары взаимно-обратных модификаций. Это утверждение легко доказывается с помощью топологических преобразований, примененных в [16] для случая $\mathbf{F}_{2/1} = \mathbf{P}_{2/1}$.

Утверждение 7.1

Один шаг рекурсивного построения третьего типа сохраняет свойство разбиения модификаций исходной УППС на пары взаимно-обратных модификаций, если исходная УППС обладает таким свойством.

Доказательство

Рассмотрим блок $\mathbf{F}_{2n/2m+2n}$, построенный по рекурсивной схеме третьего типа с использованием исходного блока $\mathbf{F}_{n/m}$ (см. рис. 7.6). Построенный блок может быть представлен как суперпозиция однослойной УППС $\mathbf{L}_1$, УППС $\mathbf{F}_{2n/2m}$, представляющей

собой каскад из двух блоков $\mathbf{F}_{n/m}$ и однослойной УППС $\mathbf{L}_S$. Блок $\mathbf{F}_{2n/2m}$ обозначим в виде $(\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m})$, где W_1 и W_2 – управляющие вектора, соответствующие левому и правому блокам $\mathbf{F}_{n/m}$. Управляющие вектора блоков $\mathbf{L}_1$ и $\mathbf{L}_S$ обозначим V_1 и V_S, соответственно. Пусть начальные блоки $\mathbf{F}_{n/m}$ обладают рассматриваемым свойством разбиения. Тогда для произвольных W_1 и W_2 существуют значения W'_1 и W'_2, такие, что выполняется условие

$$(\mathbf{F}^{(W'_1)}{}_{n/m}|\mathbf{F}^{(W'_2)}{}_{n/m}) = ((\mathbf{F}^{(W_1)}{}_{n/m})^{-1}|(\mathbf{F}^{(W_2)}{}_{n/m})^{-1}) = (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m})^{-1}.$$

Теперь легко показать, что УППС $\mathbf{F}_{2n/2m+2n}$ при управляющих векторах $V'_1 = V_S$, $V'_S = V_1$, W'_1 и W'_2 реализует модификацию обратную по отношению к той, которая реализуется им при управляющих векторах V_1, V_S, W_1 и W_2. Действительно, с учетом того, что УЭ $\mathbf{F}_{2/1}$ являются инволюциями, а, следовательно, одинаковые УППС $\mathbf{L}_1$ и $\mathbf{L}_S$ также являются управляемыми инволюциями, мы имеем при преобразовании входного вектора X:

$$Y = (X)\,\mathbf{L}_1{}^{(V_1)} \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_S)};$$

$$Y' = (Y)\,\mathbf{L}_1{}^{(V'_1)} \bullet (\mathbf{F}^{(W'_1)}{}_{n/m}|\mathbf{F}^{(W'_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V'_S)} =$$

$$((X)\,\mathbf{L}_1{}^{(V_1)} \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_S)})\,\mathbf{L}_1{}^{(V_S)} \bullet (\mathbf{F}^{(W'_1)}{}_{n/m}|\mathbf{F}^{(W'_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_1)} =$$

$$= (X)\,\mathbf{L}_1{}^{(V_1)} \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_S)} \bullet \mathbf{L}_1{}^{(V_S)} \bullet (\mathbf{F}^{(W'_1)}{}_{n/m}|\mathbf{F}^{(W'_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_1)} =$$

$$= (X)\,\mathbf{L}_1{}^{(V_1)} \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m}) \bullet (\mathbf{F}^{(W'_1)}{}_{n/m}|\mathbf{F}^{(W'_2)}{}_{n/m}) \bullet \mathbf{L}_S{}^{(V_1)} =$$

$$= (X)\,\mathbf{L}_1{}^{(V_1)} \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m}) \bullet (\mathbf{F}^{(W_1)}{}_{n/m}|\mathbf{F}^{(W_2)}{}_{n/m})^{-1} \bullet \mathbf{L}_S{}^{(V_1)} =$$

$$= (X)\,\mathbf{L}_1{}^{(V_1)} \bullet \mathbf{L}_S{}^{(V_1)} = X.$$

Что и требовалось доказать.

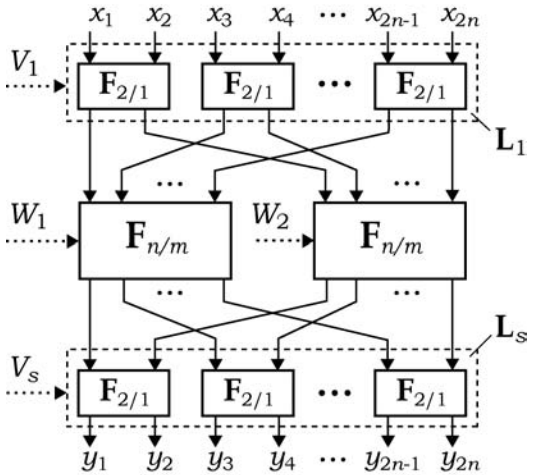

Рис. 7.6. Структура блока $\mathbf{F}_{2n/2m+2n}$

Используя утверждение 7.1, легко доказать следующую теорему, имеющую прикладное значение, поскольку она показывает принципиальную возможность построения переключаемых БУП и УППС порядков $h = 1, 2, \ldots, n/4, n$.

Теорема 7.1

Для порядков $h = 1, 2, \ldots, n/4, n$ и $n = 2^k$, где k – натуральное число, УППС, построенные по рекурсивной схеме третьего типа с использованием типового УЭ $\mathbf{F}_{2/1}$, являющегося инволюцией, реализуют множество модификаций управляемой операции $\mathbf{F}_{n/m}$, которое разбивается на пары взаимно-обратных модификаций.

Доказательство

В случае $h = n$ утверждение вытекает непосредственно из зеркальной симметрии УППС максимального порядка. В случае $h = 1$ доказательство легко осуществляется с помощью топологического преобразования, изложенного в [14] для случая $\mathbf{F}_{2/1} = \mathbf{P}_{2/1}$. Поскольку в УППС порядков $h = 2, \ldots, n/4$ в качестве начального блока используется УОП первого порядка, для которого имеет место свойство разбиения, то согласно утверждению 7.1 это свойство имеет место для $h = 2$. Применяя последовательно утверждение 7.1, легко доказывается, что это свойство имеет место для $h = 4, \ldots, n/4$.

Доказанный факт позволяет осуществлять построение переключаемой УППС различных порядков, однако для получения более наглядной структурированности такого построения требуется использование симметричной УППС в частных случаях или другого конструктивного механизма — в общем случае.

При оценке аппаратных ресурсов, необходимых для реализации УППС $\mathbf{F}_{n/m}$ различного порядка, можно воспользоваться формулой $N_{\text{УЭ}} = (n \log_2 nh)/2$ для вычисления числа УЭ $N_{\text{УЭ}}$ в УППС. Таблицы 7.1 и 7.2 представляют оценки необходимых схемотехнических ресурсов в случае заказных СБИС, изготавливаемых с использованием технологии с разрешением 0.33 мкм.

Таблица 7.1

Площадь кристалла, необходимая для реализации УППС $\mathbf{F}_{32/m}$ различных порядков (данные приведены в единицах измерения sqmil)

УЭ	$h = 1$	$h = 2$	$h = 4$	$h = 8$	$h = 32$
e/g, e/h, f/i, f/j	160	192	224	256	288
g/e, g/h, h/g, h/e, g/i, f/g, i/f, h/j, j/h, j/f, f/h	240	288	336	384	432
g/f, i/g, e/j, j/e, j/i, i/e, i/j, h/f	320	384	448	512	576

Таблица 7.2

Площадь кристалла, необходимая для реализации УППС $\mathbf{F}_{64/m}$ различных порядков (данные приведены в единицах измерения sqmil)

УЭ	$h = 1$	$h = 2$	$h = 4$	$h = 8$	$h = 16$	$h = 64$
e/g, e/h, f/i, f/j	384	448	512	576	640	704
g/e, g/h, h/g, h/e, g/i, f/g, i/f, h/j, j/h, j/f, f/h	576	672	768	864	960	1056
g/f, i/g, e/j, j/e, j/i, i/e, i/j, h/f	768	896	1024	1152	1280	1408

Из приведенных данных видно, что все рассмотренные УППС требуют вполне приемлемых аппаратных затрат, поэтому они могут быть применены для построения шифров различных типов. Наиболее привлекательными для использования представляются 32-битовые УППС $\mathbf{F}_{32/96}$ и 64-битовые УППС $\mathbf{F}_{64/192}$, обладающие весьма низкой сложностью схемотехнической реализации. Эти оценки показывают, что шифры на основе переменных операций (управляемых подстановочно-перестановочных сетей) могут быть реализованы достаточно экономично, а именно — сложность их реализации примерно равна, а в некоторых случаях (для определенных типов УЭ) существенно меньше сложности реализации шифров на основе перестановок, зависящих от преобразуемых данных, для которых экономичность построения показана в различных работах [118-120].

В данной книге рассматриваются достаточно подробно УЭ типа $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$ (*см. главы 4, 5, 6*), но аналогичные построения различной УППС могут быть осуществлены с использованием УЭ, имеющих другой размер, например, $\mathbf{F}_{3/2}$, $\mathbf{F}_{4/1}$, $\mathbf{F}_{3/3}$ и $\mathbf{F}_{4/2}$. В последнем случае мы фактически имеем блоки подстановок типа 6×4. В качестве таких блоков можно использовать блоки подстановок, хорошо апробированные в шифре DES. Они детально описаны во многих книгах по криптографии, поэтому мы их не будем обсуждать. Заметим только, что критерии отбора управляемых элементов $\mathbf{F}_{4/2}$ не обязательно должны совпадать с критериями выбора подстановок 6×4, обсуждавшимися ранее при обосновании их выбора для криптографических применений.

Это обусловлено тем, что элементы $\mathbf{F}_{4/2}$ предполагается использовать в криптосхемах другого типа, а именно в случае, когда при выполнении управляемой операции, построенной на их основе, управляющий подблок данных и преобразуемый являются независимыми. В случае алгоритма DES подстановки 6×4 используются несколько в другом плане: каскад из 8 таких подстановок фактически реализует фиксированную подстановочную операцию, выполняемую над 32-битовым подблоком

данных посредством выполнения управляемых операций над 4-битовыми подблоками данных (выбор одной из четырех подстановок типа 4×4 в зависимости от значений двух битов, принадлежащих другим подблокам). Отмечая резкое возрастание числа УЭ, удовлетворяющих критериям применения для синтеза УППС, при переходе от элементов $\mathbf{F}_{2/1}$ к $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$ можно предположить, что УЭ $\mathbf{F}_{4/2}$, пригодных для синтеза УППС, окажется чрезвычайно много и их число будет существенно превосходить число подстановок типа 6×4, считающихся удовлетворяющими критериям применения в классических подстановочно-перестановочных шифрах.

Таким образом, криптосхемы, основанные на использовании переменных операций, реализуемых с помощью УППС, позволяют:

- эффективно использовать УЭ сравнительно малого размера, включая минимальные УЭ;
- значительно расширить число вариантов элементов $\mathbf{F}_{4/2}$ и УЭ большего размера, представляющих интерес для криптографических применений.

Управляемые элементы с минимальным размером входа ($\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$) представляют собой наименьшие строительные блоки в синтезе криптографических операций. Их применение является целесообразным по следующим причинам:

- они прошли апробацию в многочисленных шифрах, в частности, в шифрах на основе переменных битовых перестановок,
- дана полная классификация УЭ $\mathbf{F}_{2/1}$ и найдены два лучших их подкласса,
- сформулированы критерии отбора элементов $\mathbf{F}_{2/2}$ и дана достаточно полная их классификация,
- их применение обеспечивает существенно более экономичную аппаратную реализацию по сравнению с построением УППС с размером входа 4 бита и более.

Применение УЭ $\mathbf{F}_{3/1}$ также представляет интерес и обеспечивает экономичность построения шифров. Несмотря на то, что трехразрядный вход вносит некоторые ограничения их применения, они могут быть использованы, например, для синтеза УППС, ориентированной на применения:

- в криптосхемах с разбиением на неравные подблоки данных (например, 128-битовый блок данных разбивается на 32-битовый и 96-битовый подблоки, последний преобразуется с помощью УППС, а первый служит для задания управляющего вектора),
- в хэш-функциях, где не требуется обеспечение обратимости преобразований и поэтому имеется возможность использовать операцию расширения преобразуемого двоичного вектора,
- в раундовой функции обобщенной криптосхемы Фейстеля.

7.2. Проблемы построения блочных шифров с простым расписанием использования ключа

Использование в качестве базового криптографического примитива операций, зависящих от шифруемых данных, т. е. переменных операций преобразования, создает предпосылки для построения скоростных шифров, обладающих малой сложностью схемотехнической реализации. На основе примитивов подобного типа возможно построение таких криптосхем преобразования, в которых усиливается параллелизм вычислений и одновременно с преобразованием подблоков данных возможно преобразование раундовых подключей. Последнее является предпосылкой для применения простого расписания использования ключа (ПРИК), которое заключается в том, что в качестве раундовых ключей используются фрагменты секретного ключа (подключи) или некоторые их комбинации.

Применение ПРИК обеспечивает

1) существенную дополнительную экономию используемых схемотехнических ресурсов при аппаратной реализации шифров;

2) повышение производительности устройств шифрования при частой смене ключей, что является важным для многих сетевых приложений.

Примерами шифров, базирующихся на операциях, зависящих от преобразуемых данных, и использующих ПРИК, являются SPECTR-H64, SPECTR-128, CIKS-128, COBRA-F64a, COBRA-F64b и др. [14]. Однако какие бы сложные преобразования ни использовались в одном раунде итеративных криптосхем, в которых режим зашифрования переключается на режим расшифрования путем смены очередности использования раундовых ключей на обратную, ПРИК приводит к тому, что для некоторых классов ключей (пусть даже их доля в полном пространстве ключей будет пренебрежимо малой) процедура зашифрования будет совпадать с процедурой расшифрования. Такие ключи считаются слабыми. В случае упомянутых криптосхем ПРИК приводит к существованию слабых ключей такого типа за счет того, что в обоих режимах шифрования используются одинаковые процедуры преобразования. Более того, для некоторых подклассов ключей применение ПРИК приводит к тому, что все раундовые ключи оказываются одинаковыми, а следовательно все раунды шифрования будут представлять собой одно и тоже преобразование (это имеет место в некоторых практически используемых шифрах, например, в криптосистеме ГОСТ 28147–89). Последнее является предпосылкой для осуществления «скользящей атаки», предложенной в работе [58]. Несмотря на то, что вероятность выбора слабого ключа является чрезвычайно малой ($2^{-192} – 2^{-128}$) для упомянутых шифров, желательно устранить данную особенность. При построении раундовых хэш-функций, основанных на использовании блочных шифров, наличие даже малой доли слабых ключей является нежелательным. Например, для итеративных хэш-функций, в которых блоки хэшируемых данных входят в преобразования вместо ключа, наличие слабых ключей дает возможность легко формировать большое число различных сообщений, хэш-функция от которых равна одному и тому же значению, что является недопустимым.

Для устранения слабых ключей в итеративных блочных шифрах, использующих ПРИК, можно предложить следующие подходы:

1. Применение различных процедур преобразования для выполнения зашифрования и расшифрования. Существенным недостатком этого подхода является то, что при аппаратной реализации он фактически требует реализации двух алгоритмов. Кроме того, сохраняется класс ключей, для которых все раунды зашифрования будут представлять собой одно и то же преобразование, хотя и отличающееся от раунда расшифрования. Это сохраняет предпосылки уязвимости к «скользящей атаке».
2. Включение в раундовое преобразование констант, значения которых зависят от номера раунда и режима шифрования. Этот способ позволяет устранить как слабые ключи, так и одинаковость всех раундов шифрования. Некоторым недостатком является то, что в каждый раунд требуется включать, по крайней мере, по одной дополнительной операции и требуется реализовать механизм соответствующего изменения очередности использования констант при изменении режима шифрования.
3. Применение переключаемых операций, управляемых битом, задающим прямую или обратную очередность использования раундовых ключей (благодаря чему задается выбор режима зашифрования или расшифрования). Этот подход требует минимальных дополнительных затрат схемотехнических ресурсов и устраняет необходимость переключения значений констант в каждом раунде при изменении режима шифрования. Более того, он дает предпосылки для обоснования шифров, в которых не требуется осуществлять изменение очередности использования раундовых ключей. За счет последнего может быть достигнуто даже общее снижение стоимости аппаратной реализации.

Сравнение перечисленных выше подходов к построению шифров с ПРИК показывает, что наибольший интерес представляет использование переключаемых управляемых операций. Вопрос состоит в разработке переключаемых операций, которые удовлетворяют следующим требованиям:

- невысокая сложность схемотехнической реализации,
- эффективность при использовании в качестве криптографического примитива,
- высокое быстродействие.

Свойство переключаемости операции по своей сути является специальным вариантом реализации свойства управляемости. Это наводит мысль на идею реализовать переключаемые управляемые операции (ПУО) путем соответствующего модифицирования схем построения управляемых операций. Кроме того, имеются предпосылки эффективной реализации ПУО, заключающиеся в существовании управляемых операций, в которых все множество потенциально реализуемых модификаций разбивается на пары взаимно-обратных модификаций [14]. Пусть в некоторой гипотетической управляемой операции имеется дополнительный одноразрядный управляющий вход, на который подается бит e, задающий режим выполнения операции, и при $e = 0$

выполняется прямая управляемая операция, а при $e = 1$ выполняется соответствующая ей обратная управляемая операция. Это означает, что при любом заданном фиксированном значении управляющего вектора будет реализовываться прямая модификация при зашифровании ($e = 0$) и соответствующая ей обратная при расшифровании $e = 1$, т. е. реализуется некоторая ПУО. Идея построения ПУО находит эффективную реализацию на основе симметричных топологий управляемых перестановочных сетей и УППС. В следующих разделах этой главы рассматривается построение ПУО различного типа.

7.3. Понятие переключаемой операции

В общем случае понятие управляемой операции можно задать следующим образом.

Определение 7.4

Пусть $\{F_1, F_2,\ldots, F_{2^m}\}$ есть множество операций, определенных формулой $Y = F_i = F_i(X_1, X_2,\ldots, X_q)$, где $i = 1, 2,\ldots, 2^m$, $X_1, X_2,\ldots, X_q$ являются входными n-мерными двоичными векторами (операндами), а Y – выходной n-мерный вектор. Тогда зависимую от V операцию $F^{(V)}$, определяемую формулой $Y = F^{(V)}(X_1, X_2,\ldots, X_q) = F_V(X_1, X_2,\ldots, X_q)$, где V – m-мерный управляющий вектор, будем называть управляемой q-местной операцией. Операции $F_1, F_2,\ldots, F_{2^m}$ будем называть ***модификациями управляемой операции*** $F^{(V)}$.

В качестве примеров управляемых операций можно назвать управляемые перестановки [14, 97] и управляемые двуместные операции [9, 14, 27, 74]. Для дальнейшего рассмотрения представляет интерес понятие обратной управляемой операции.

Определение 7.5

Пусть $\{F_1, F_2,\ldots, F_{2^m}\}$ есть множество модификаций управляемой операции $F^{(V)}$. Операция $(F^{(V)})^{-1}$, содержащая модификации $F_1^{-1}, F_2^{-1},..., F_{2^m}^{-1}$, называется ***обратной*** по отношению к управляемой операции $F^{(V)}$, если для всех V модификации F_V^{-1} и F_V являются взаимно-обратными.

В общем случае переключаемую операцию можно определить следующим образом.

Определение 7.6

Пусть $F'^{(e)}$, где $e \in \{0,1\}$, – некоторая зависящая от e операция, содержащая две модификации $F'^{(0)} = F'_1$ и $F'^{(1)} = F'_2$, где $F'_2 = F'^{-1}_1$. Тогда операция $F'^{(e)}$ называется ***переключаемой операцией***.

В дальнейшем будут рассмотрены различные варианты управляемых операций, но важнейшим и наиболее интересным для криптографических применений является случай ПУО.

Определение 7.7

Пусть две модификации переключаемой операции $F'^{(e)}$ представляют собой пару взаимно-обратных управляемых операций $F'^{(0)} = F^{(V)}$ и $F'^{(1)} = (F^{(V)})^{-1}$. Тогда $F'^{(e)}$ называется ***переключаемой управляемой операцией*** $F^{(V,e)}$.

Впервые частный вариант ПУО, а именно переключаемые управляемые перестановки, были построены на основе перестановочных сетей с симметричной структурой [14]. Построение управляемых операций на основе УППС значительно расширяет класс ПУО и дает новые возможности конструирования скоростных блочных шифров, ориентированных на эффективную аппаратную реализацию. Можно предложить несколько подходов к реализации ПУО на основе УППС. Главными из них являются подходы, основанные на использовании:

- элементов симметрии топологии сетей;
- симметрии распределения битов управляющего вектора.

В первом случае в качестве внутреннего узла ПУО используется некоторый переключающий блок, изменяющий распределение управляющих битов таким образом, что при заданном управляющем векторе реализуемая модификация переключается с прямой на обратную. Этот случай обеспечивает возможность использования управляющего вектора, в котором каждый бит является независимым. Во втором случае в качестве внутренних узлов ПУО используются 1) переключающий блок (имеющий меньший размер и реализуемый с меньшими дополнительными затратами схемотехнических ресурсов) и 2) блок расширения, реализуемый как простое разветвление проводников и практически не вносящий дополнительных аппаратных затрат. В первом варианте блок расширения не является необходимым элементом, хотя в частных случаях использования ПУО он может применяться, например, для формирования управляющего вектора большого размера по подблоку данных малого размера. Симметрия топологической структуры некоторой ПУО предполагает, что управляемые элементы, расположенные в симметричных позициях, являются либо инволюциями, либо взаимно-обратными управляемыми элементами общего типа.

При использовании специальных типов управляемых элементов механизм переключения состоит не в скачкообразном перераспределении управляющих битов по УЭ УППС, а в инвертировании всех битов управляющего вектора. Для реализации этого механизма переключения управляющий элемент проектируется таким образом, чтобы все его модификации разбивались на пары взаимно-обратных модификаций. Если такие управляемые элементы входят в симметричную топологию, переключение прямой управляемой операции на обратную осуществляется путем инвертирования каждого бита управляющего вектора.

7.4. Управляемые операционные подстановки как класс попарно взаимно-обратных модификаций

Рекурсивные схемы построения перестановочных сетей [14] имеют практическую значимость и могут быть применены для проектирования УОП. В плане по-

строения переключаемых УОП представляет интерес доказательство того, что рекурсивное построение формирует управляемые перестановочные и подстановочно-перестановочные сети, множество реализуемых модификаций которых разбивается на пары взаимно-обратных модификаций. Данный результат доказан в [10, 14] для рекурсивных блоков управляемых перестановок (БУП) первого порядка. Схема рекурсивного построения БУП первого порядка может быть распространена на УОП путем замены элементарного управляемого переключателя на управляемый подстановочный элемент $\mathbf{F}_{2/1}$. Это приводит к рекурсивной схеме построения переключаемых УОП (ПУОП), показанной на рис. 7.7 и относящейся к первому типу. Данное построение заключается в использовании двух одинаковых параллельных блоков $\mathbf{F}_{n/m}$, входы которых соединены с выходами активного каскада $\mathbf{L}_n$, состоящего из n параллельных элементов $\mathbf{F}_{2/1}$, в соответствии со следующим правилом. Левый (правый) одноразрядный выход каждого i-го элемента $\mathbf{F}_{2/1}$ активного каскада $\mathbf{L}_n$ соединен с i-м одноразрядным входом левого (правого) блока $\mathbf{F}_{n/m}$. В результате формируется блок $\mathbf{F}_{2n/2m+n}$. На первом шаге рекурсивного построения первого типа в качестве блоков $\mathbf{F}_{n/m}$ используются управляемые элементы $\mathbf{F}_{2/1}$, а в качестве $\mathbf{L}_n$ – активный каскад $\mathbf{L}_2$, состоящий из двух элементов $\mathbf{F}_{2/1}$ (см. рис. 7.7в).

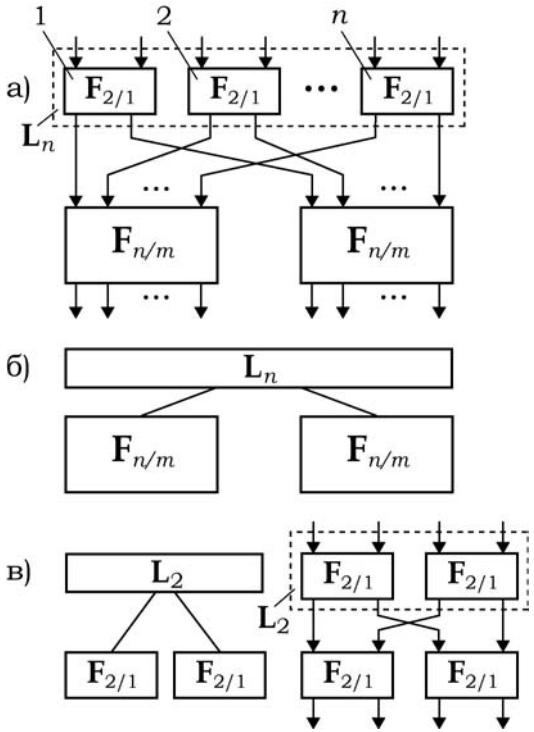

Рис 7.7. Первый вариант рекурсивного построения УППС:
а) шаг удвоения размера блока, б) схематическое
представление шага удвоения, в) первый шаг построения

Доказательство разбиения всех модификаций БУП на пары взаимно-обратных модификаций опирается на использование специфики топологии рекурсивного построения первого типа и на то, что переключаемый элемент представляет собой управляемую элементарную инволюцию. С учетом этого легко установить, что это доказательство распространяется на УОП с аналогичной топологией, если элементарная управляемая операционная подстановка (т. е. УЭ $\mathbf{F}_{2/1}$) представляет собой элементарную управляемую инволюцию. Ввиду наличия большого числа различных вариантов элементов $\mathbf{F}_{2/1}$, обладающих нелинейными свойствами, имеются предпосылки построения большого числа различных ПУОП.

Указанный тип разбиения имеет место также и в случае ПУОП, полученных в соответствии с рекурсивным построением второго типа, которое является аналогичным рекурсивному построению первого типа и состоит в следующем. Выходы двух параллельных блоков $\mathbf{F}'_{n/m}$ соединяются со входами активного каскада $\mathbf{L}_n$ в соответствии со следующим правилом. Левый (правый) одноразрядный вход каждого i-го элемента $\mathbf{F}_{2/1}$ активного каскада $\mathbf{L}_n$ соединен с i-м одноразрядным выходом левого (правого) блока $\mathbf{F}'_{n/m}$. В результате формируется блок $\mathbf{F}'_{2n/2m+n}$. На первом шаге рекурсивного построения второго типа в качестве блоков $\mathbf{F}'_{n/m}$ используются управляемые элементы $\mathbf{F}_{2/1}$. Блок $\mathbf{F}'_{2n/2m+n}$. является зеркально симметричным по отношению к блоку $\mathbf{F}_{2n/2m+n}$, полученному по рекурсивной схеме построения первого типа (см. рис. 7.8).

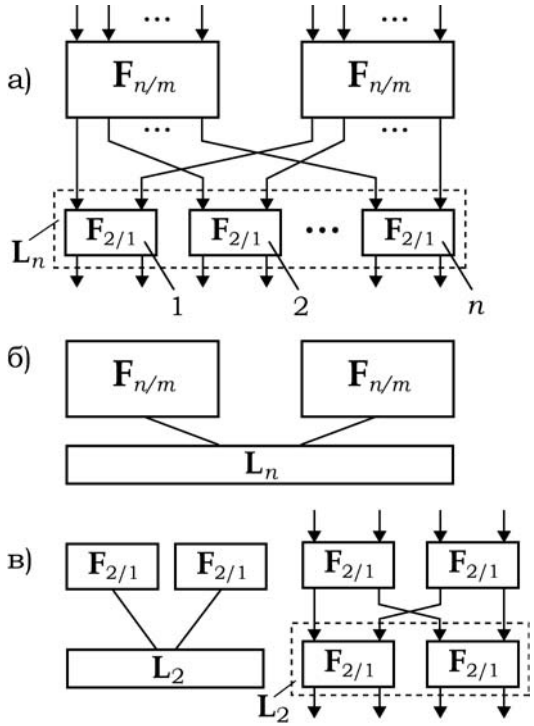

Рис 7.8. 2-ой вариант рекурсивного построения УППС: а) шаг удвоения размера блока, б) схематическое представление шага удвоения, в) первый шаг построения

Это означает, что если в блоках $\mathbf{F}'_{2n/2m+n}$ биты управляющего вектора распределить по управляемым элементам по правилу соответствующему обратным УППС, то мы будем иметь $\mathbf{F}'_{2n/2m+n} = (\mathbf{F}'_{2n/2m+n})^{-1}$. Поскольку модификации блоков $\mathbf{F}_{2n/2m+n}$ разбиваются на пары взаимно-обратных модификаций, то из последнего соотношения вытекает, что это свойство имеет место также и для УППС $\mathbf{F}'_{2n/2m+n}$. Первый и второй типы рекурсивного построения приводят на каждом шаге к синтезу ППС первого порядка.

Рассмотренные два варианта построения могут быть использованы для построения взаимно-обратных блоков при использовании управляемых элементов $\mathbf{F}_{2/1}$ общего типа. В этом случае рекурсивное построение первого типа выполняется с использованием прямых элементов $\mathbf{F}_{2/1}$, а рекурсивное построение второго типа выполняется с использованием соответствующих обратных элементов $(\mathbf{F}_{2/1})^{-1}$, как это показано на рис. 7.9.

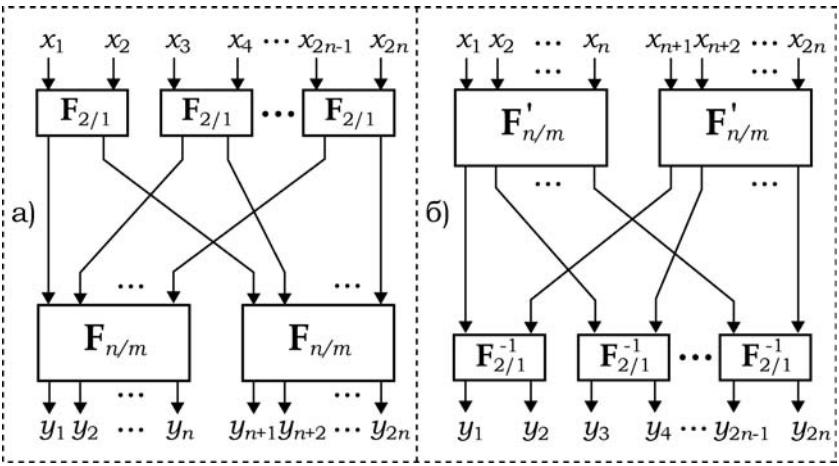

Рис. 7.9. Схемы рекурсивного построения УППС с взаимно симметричной топологией

Для синтеза УППС представляет интерес также и третий тип рекурсивного построения БУП, который обеспечивает синтез сетей максимального порядка. Этот вариант построения может быть представлен как объединение двух первых. На одном шаге из двух блоков $\mathbf{F}_{n/m}$ и двух активных каскадов $\mathbf{L}_n$ формируется блок $\mathbf{F}_{2n/2m+2n}$ (рис. 7.10). При этом удваивается как размер входа УППС, так и его порядок. На первом шаге построения используются два управляемых элемента $\mathbf{F}_{2/1}$ и два активных каскада типа $\mathbf{L}_2$. Поскольку элементы $\mathbf{F}_{2/1}$ имеют максимальный порядок, то каждый шаг рекурсии приводит к синтезу УППС максимального порядка с удвоенным размером входа. Во всех трех типах рекурсивного построения предполагается использование управляемых элементарных инволюций $\mathbf{F}_{2/1}$, для того чтобы реализуемые модификации разбивались на пары взаимно-обратных модификаций (в случае общего построения это не является обязательным). Подобное разбиение свидетельствует о

принципиальной возможности построения на основе УППС переключаемых ПУО, хотя вопрос сложности такого построения требует отдельного рассмотрения.

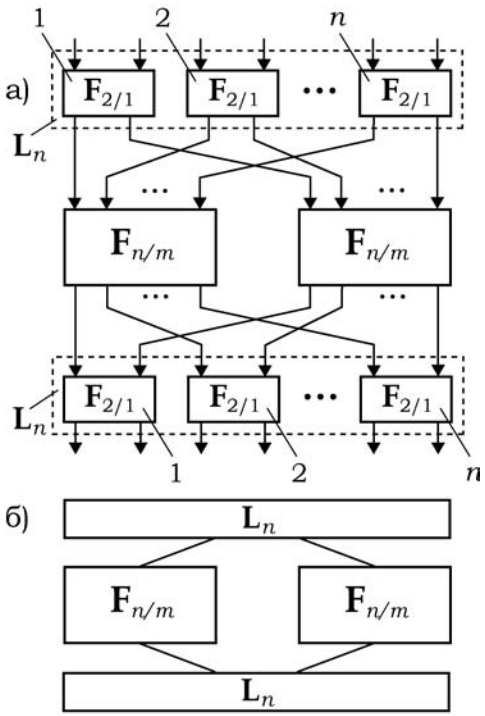

Рис 7.10. Третий вариант рекурсивного построения УППС:
а) шаг удвоения размера блока, б) схематическое
представление шага удвоения

Ввиду внутренней зеркальной симметрии блоков $\mathbf{F}_{2n/2m+2n}$ существование для них указанного разбиения является достаточно очевидным. Действительно, для любого значения управляющего вектора $V = (V_1, V_2, ..., V_s)$ реализуется модификация, которая является обратной по отношению к модификации, соответствующей управляющему вектору $V' = (V_s, V_{s-1}, ..., V_1)$, полученному из V путем обратной записи компонентов $V_1, V_2, ..., V_s$. Аналогичным образом можно показать, что для любой симметричной УППС реализуется указанное разбиение. Симметричные УППС (и БУП как частный случай) являются удобными для синтеза ПУО различного типа. При этом в симметричных УППС можно использовать не только элементарные управляемые инволюции, но и пары взаимно-обратных элементов $\mathbf{F}_{2/1}$ общего типа, расположенные в симметричных позициях. Например, если на первом шаге рекурсивного построения третьего типа использовать две элементарные управляемые инволюции $\mathbf{F}^*_{2/1}$ и на каждом шаге рекурсии в качестве верхнего $\mathbf{L}_n$ и нижнего $\mathbf{L}^{-1}{}_n$ активных каскадов использовать каскады, построенные из прямых $\mathbf{F}_{2/1}$ и соответствующих обратных эле-

ментов $(\mathbf{F}_{2/1})^{-1}$, то будем иметь новый широкий класс симметричных УППС. Эта схема построения представлена на рис. 7.11.

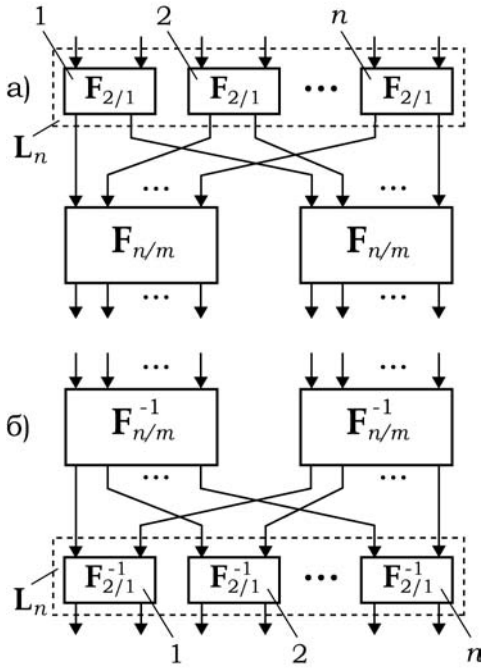

Рис. 7.11. Рекурсивное построение взаимно-обратных УППС на основе взаимно-обратных управляемых элементов
а) прямой блок, б) обратный блок

Используя свойство удвоения порядка, реализуемое в третьем типе рекурсивного построения, можно построить УППС порядков 2, 4, …, $n/4$. Это построение осуществляется по аналогии с синтезом БУП указанных порядков, описанным ранее в [10, 14]. Минимальное число активных каскадов, необходимое для реализации УППС первого порядка, составляет $s_{\min} = \log_2 n$. Это легко установить, рассматривая первый или второй варианты рекурсивного построения. Осуществляя шаг рекурсии третьего типа при использовании двух блоков $\mathbf{F}_{n/m}$ первого порядка, получаем блок $\mathbf{F}_{2n/2m+2n}$ второго порядка с числом слоев

$$s_{\min} = 2 + \log_2 n = 1 + \log_2 2n,$$

где $2n$ есть размер синтезированного блока второго порядка, т. е. для данного размера входа минимально необходимое число активных каскадов для реализации УППС второго порядка на единицу превосходит значение $s_{\min}$ для УППС первого порядка. Если исходный блок $\mathbf{F}_{n/m}$ имеет порядок h, шаг рекурсии третьего типа обеспечивает построение блока $\mathbf{F}_{2n/2m+2n}$ порядка $2h$. Легко показать, что для данных значений n

и $h \le n/4$ значение $s_{\min}$ составляет $s_{\min} = \log_2 nh$. Случай $h = n/2$ может быть реализован, но он не представляет практического интереса, поскольку реализация требует использования числа активных каскадов $s_{\min} = 2\log_2 n - 1$, которое равно числу активных каскадов в УППС n-го порядка.

Покажем, что УППС порядков 2, 4, …, $n/4$ обладают свойством разбиения модификаций на два подмножества взаимно-обратных модификаций. На рис. 7.12 показана структура блока **F** порядка $2h$, в которой внутренние блоки $\mathbf{F}'$ и $\mathbf{F}''$ имеют порядок h и обладают указанным свойством. На первом шаге рекурсии можно, например, взять взаимно-обратные УППС первого порядка, построенные с использованием элементарных управляемых инволюций по первой и второй рекурсивным схемам построения. В частном случае, в качестве начальных блоков $\mathbf{F}'$ и $\mathbf{F}''$ можно взять пару УЭ $\mathbf{F}'_{2/1}$ и $\mathbf{F}''_{2/1}$, в которой для элементарных модификаций, реализуемых УЭ, выполняются соотношения $\mathbf{F}'^{(0)}_{2/1} = (\mathbf{F}'^{(1)}_{2/1})^{-1}$ и $\mathbf{F}''^{(0)}_{2/1} = (\mathbf{F}''^{(1)}_{2/1})^{-1}$, т. е. используются УЭ с множеством реализуемых модификаций, разбивающимся на два подмножества взаимно-обратных модификаций. В качестве таких УЭ можно взять УЭ, описываемые как следующие пары элементарных модификаций: q/u, v/s, u/q, x/r, w/t, r/x, s/v, t/v.

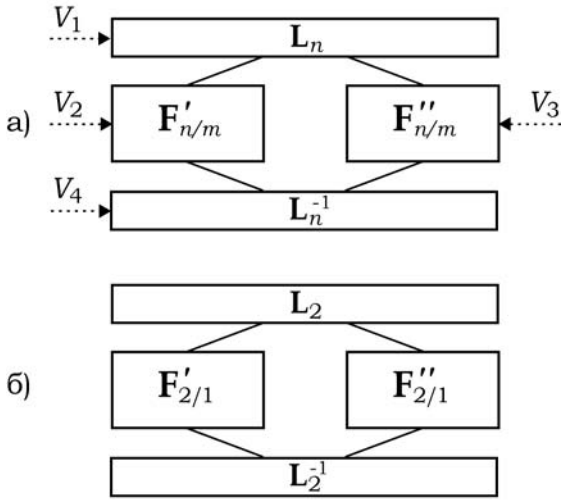

Рис 7.12. Третий вариант рекурсивного построения симметричных УППС с использованием различных блоков:
а) общая схема, б) первый шаг построения

Управляющий вектор V, соответствующий блоку **F**, может быть представлен в виде конкатенации компонентов V_1, V_2, V_3, V_4, являющихся управляющими векторами, соответственно, верхнего каскада $\mathbf{L}_n$, блока $\mathbf{F}'$, блока $\mathbf{F}''$ и нижнего каскада $\mathbf{L}_n$: $V = (V_1, V_2, V_3, V_4)$. Для произвольного управляющего вектора V в силу предполагаемых свойств блоков $\mathbf{F}'$ и $\mathbf{F}''$ можно указать другой управляющий вектор $V' = (V_4, V'_2, V'_3, V_1)$, где V'_2 есть такой управляющий вектор блока $\mathbf{F}'$, при котором

последний реализует модификацию, обратную модификации, реализуемой при V_2; V'_3 есть такой управляющий вектор блока $\mathbf{F}''$, при котором последний реализует модификацию, обратную модификации, реализуемой при V_3. В силу зеркальной симметричности включения верхнего и нижнего каскадов $\mathbf{L}_n$ блок $\mathbf{F}$ при $V' = (V_4, V'_2, V'_3, V_1)$ реализует модификацию, которая является обратной модификации, реализуемой им при $V = (V_1, V_2, V_3, V_4)$. Действительно, представим УППС $\mathbf{F}_{2n/2m+2n}$ в виде суперпозиции однослойного УППС $\mathbf{L}_n$, УППС $\mathbf{F}'_{2n/2m}$, представляющего собой каскад из двух блоков $\mathbf{F}'_{n/m}$ и $\mathbf{F}''_{n/m}$, и однослойного УППС $\mathbf{L}_n^{-1}$. Блок $\mathbf{F}'_{2n/2m}$ обозначим в виде $(\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m})$, где V_2 и V_3 – управляющие вектора, соответствующие левому и правому внутренним блокам.

Теперь осуществим преобразование входного вектора X по следующей схеме:

$$Y = (X)\,\mathbf{L}_n^{(V_1)} \bullet (\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_4)};$$

$$Y' = (Y)\,\mathbf{L}_n^{(V_4)} \bullet (\mathbf{F}'^{(V'_2)}_{n/m}|\mathbf{F}''^{(V'_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_1)} =$$

$$= ((X)\,\mathbf{L}_n^{(V_1)} \bullet (\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_4)})\,\mathbf{L}_n^{(V_4)} \bullet (\mathbf{F}'^{(V'_2)}_{n/m}|\mathbf{F}''^{(V'_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_1)} =$$

$$= (X)\,\mathbf{L}_n^{(V_1)} \bullet (\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_4)} \bullet \mathbf{L}_n^{(V_4)} \bullet (\mathbf{F}'^{(V'_2)}_{n/m}|\mathbf{F}''^{(V'_3)}_{n/m}) \bullet (\mathbf{L}_n^{-1})^{(V_1)} =$$

$$= (X)\,\mathbf{L}_n^{(V_1)} \bullet (\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m}) \bullet (\mathbf{F}'^{(V_2)}_{n/m}|\mathbf{F}''^{(V_3)}_{n/m})^{-1} \bullet (\mathbf{L}_n^{-1})^{(V_1)} =$$

$$= (X)\,\mathbf{L}_n^{(V_1)} \bullet (\mathbf{L}_n^{-1})^{(V_1)} = X.$$

Таким образом, для произвольного значения управляющего вектора V можно указать значение V', такое, что выполняется следующее соотношение $\mathbf{F}^{(V)}_{2n/2m+2n} = (\mathbf{F}^{(V)}_{2n/2m+2n})^{-1}$, что и требовалось показать.

Если внутренние блоки $\mathbf{F}'$ и $\mathbf{F}''$ построены по рекурсивной схеме первого или второго типов, то они обладают рассматриваемым свойством разбиения, следовательно, оно доказано для блоков $\mathbf{F}$ второго порядка. Используя такие блоки второго порядка, приходим, в соответствии с рассматриваемой схемой, к построению блоков четвертого порядка, обладающих данным свойством разбиения. Далее переходим к блокам 8-го, 16-го и т. д. порядков.

Таким образом, на основе трех типов рекурсивного построения блоков $\mathbf{F}_{n/m}$ в принципе могут быть построены ПУОП порядков 1, 2, …, n, однако только в случае максимального порядка реализуется симметричная топология, предоставляющая удобную реализацию механизма обращения управляемой операции. В случае порядков 2, 4, …, $n/4$ разработка механизма переключения распределения управляющих битов требует детализированной проработки с рассмотрением каждого шага рекурсивных построений, использованных при синтезе блока $\mathbf{F}_{n/m}$. В следующих разделах при построении ПУОП будут использованы частные варианты построения симметричных УППС первого и второго порядков, а также другая схема удвоения порядка, которая позволяет упростить синтез ПУОП порядков 1, 2, 4, …, $n/4$ для произвольного значения n, являющегося натуральной степенью числа 2.

7.5. Переключаемые управляемые операционные подстановки с симметричной топологической структурой

При разработке шифров с использованием управляемых перестановок нашли применение симметричные БУП $\mathbf{P}_{32/96}$ и $\mathbf{P}_{64/192}$, обладающие, соответственно, вторым и первым порядком [14]. Используя топологию этих блоков легко построить ПУОП $\mathbf{F}^{(V,e)}{}_{32/96}$ и $\mathbf{F}^{(V,e)}{}_{64/192}$, обладающих различными свойствами, которые определяются выбором управляемых элементов того или иного типа. В отличие от синтеза БУП, осуществляемого с использованием единственного варианта управляемого элемента, а именно элементарной управляемой перестановки $\mathbf{P}_{2/1}$, являющейся инволюцией, в случае УППС мы имеем большое число различных типов управляемых элементов. Некоторые из последних являются инволюциями, а некоторые (большинство) не являются таковыми. Рассмотрим построение УППС с симметричной топологией. Вначале построим блоки первого порядка $\mathbf{F}_{8/12}$ и $\mathbf{F}^{-1}{}_{8/12}$, содержащие три активных каскада (рис. 7.13). При этом построение блока $\mathbf{F}_{8/12}$ выполним, используя элементы $\mathbf{F}_{2/1}$ произвольного типа, а построение блоков $\mathbf{F}^{-1}{}_{8/12}$ выполним, используя элементы $\mathbf{F}^{-1}{}_{2/1}$, являющиеся обратными элементам $\mathbf{F}_{2/1}$.

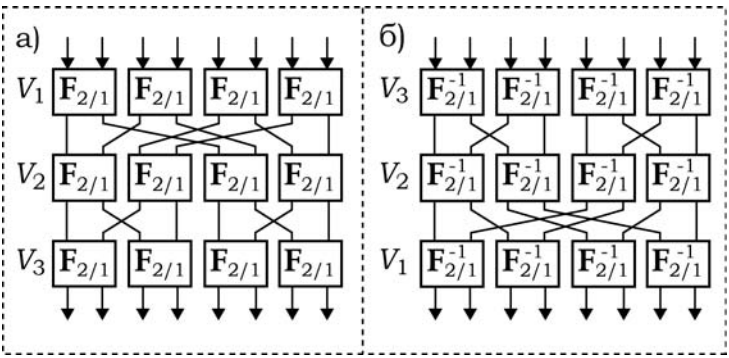

Рис. 7.13. Структура операционных блоков $\mathbf{F}_{8/12}$ и $\mathbf{F}^{-1}{}_{8/12}$

Если типовой элемент представляет собой элементарную управляемую инволюцию, то мы имеем соотношение $\mathbf{F}^{-1}{}_{2/1} = \mathbf{F}_{2/1}$. В этом случае активные каскады в блоках $\mathbf{F}_{8/12}$ и $\mathbf{F}^{-1}{}_{8/12}$ являются идентичными и представляют собой однослойные УППС, являющиеся инволюцией.

Построив блоки $\mathbf{F}_{8/12}$ и $\mathbf{F}^{-1}{}_{8/12}$, переходим к построению шестислойных блоков $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$. Блок $\mathbf{F}_{32/96}$ реализуется с использованием четырех блоков $\mathbf{F}_{8/12}$, образующих входной каскад, и четырех блоков $\mathbf{F}^{-1}{}_{8/12}$, образующих выходной каскад (рис. 7.14а).

Выходы первого каскада соединены с входами второго каскада в соответствии с фиксированной перестановкой $\mathbf{I}_1$, являющейся инволюцией и описываемой следующей цикловой структурой:

(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)

(11,18)(12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32).

Управляемый операционный блок $\mathbf{F}^{-1}_{32/96}$ (рис. 7.14б), являющийся обратным блоку $\mathbf{F}_{32/96}$, имеет такую же структуру, но отличается тем, что компоненты управляющего вектора V_1, V_2, V_3, V_4, V_5, V_6 распределены по активным слоям снизу вверх, тогда как в прямом блоке они распределены сверху вниз.

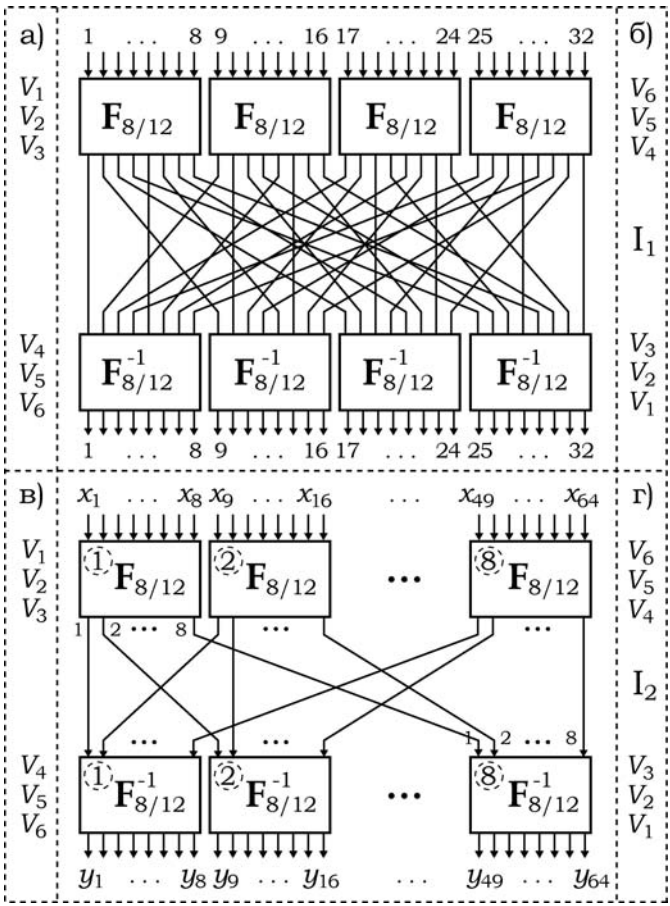

Рис. 7.14. Структура операционных блоков $\mathbf{F}_{32/96}$ (а), $\mathbf{F}^{-1}_{32/96}$ (б), $\mathbf{F}_{64/192}$ (в), $\mathbf{F}^{-1}_{64/192}$ (г)

Блок $\mathbf{F}_{64/192}$ реализуется с использованием восьми блоков $\mathbf{F}_{8/12}$, образующих входной каскад, и восьми блоков $\mathbf{F}^{-1}_{8/12}$, образующих выходной каскад (рис. 7.14в). Выходы первого каскада соединены с входами второго каскада в соответствии с фик-

сированной перестановкой $\mathbf{I}_2$, являющейся инволюцией и описываемой следующей цикловой структурой:

(1)(2,9,3,17,4,25,5,33,6,41,7,49,8,57)(10)

(11,18,12,26,13,34,14,42,15,50,16,58)(19)

(20,27,21,35,22,43,23,51,24,59)(28)(29,36,30,44,31,52,32,60)(37)

(38,45,39,53,40,61)(46)(47,54,48,62)(55)(56,63)(64).

В силу зеркальной симметрии блоков $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$ соответствующие им обратные блоки $\mathbf{F}^{-1}{}_{32/96}$ и $\mathbf{F}^{-1}{}_{64/192}$ отличаются только тем, что компоненты управляющего вектора $V = (V_1, V_2, ..., V_6)$ распределены по активным каскадам в обратном порядке. В случае прямых УППС они распределены сверху вниз, а в случае обратных УППС снизу вверх. Благодаря симметричности строения блоков $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$ модификации, реализуемые ими при значении управляющего вектора $V = (V_1, V_2, ..., V_6)$, являются обратными по отношению к модификациям, реализуемым при управляющем векторе $V' = (V_6, V_5, ..., V_1)$, где для $i = 1, 2, \ldots, 6$ компоненты V_i имеют длину 16 бит для $\mathbf{F}_{32/96}$ и 32 бит для $\mathbf{F}_{64/192}$. Данное свойство является верным и для пары блоков $\mathbf{F}^{-1}{}_{32/96}$ и $\mathbf{F}^{-1}{}_{64/192}$. Это связано с тем, что компонента V_i управляет i-ым активным слоем в случае $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$ и $(7 - i)$-ым активным слоем в случае $\mathbf{F}^{-1}{}_{32/96}$ и $\mathbf{F}^{-1}{}_{64/192}$. В свою очередь, это означает, что изменением порядка использования компонентов управляющего вектора можно задать переключение прямой УППС на обратную УППС. Очевидно, что такой способ построения ПУОП пригоден для произвольной симметричной структуры УППС, т. е. проблема построения ПУОП может быть решена предварительным построением УППС с симметричной топологией.

Если каждое значение V_i, где $i = 1, 2,\ldots, s$, формируется до того, как биты данных пройдут через i-ый слой, то время задержки УППС будет определяться числом активных слоев. Время задержки одного активного слоя находится в пределах от τ до 2τ в зависимости от варианта схемотехнической реализации управляемых элементов, где τ – время задержки поразрядной операции исключающего ИЛИ (операция XOR, обозначаемая как $\oplus$), что приблизительно соответствует времени задержки при прохождении сигнала через один вентиль. Время задержки ПУОП практически равно времени задержки УППС обычного типа.

Для построения переключаемого блока $\mathbf{F}^{(V,e)}{}_{32/96}$ компоненты управляющего вектора можно подавать на активные каскады блока $\mathbf{F}^{(V)}{}_{32/96}$ через блок перестановки 16-битовых компонентов V_i $\mathbf{P}^{(e)}{}_{96/1}$, как это показано на рис. 7.15. Блок $\mathbf{P}^{(e)}{}_{96/1}$ реализуется в виде однокаскадного БУП, состоящего из трех параллельных однослойных блоков $\mathbf{P}^{(e)}{}_{2\times16/1}$ (рис. 7.15а). Каждый из блоков $\mathbf{P}^{(e)}{}_{2\times16/1}$ имеет 16-битовый левый и 16-битовый правый входы и выходы. Блок $\mathbf{P}^{(e)}{}_{2\times16/1}$ представляет собой шестнадцать параллельных блоков $\mathbf{P}^{(e)}{}_{2/1}$, управляемых одним и тем же битом e. Правый (левый) входной (выходной) бит каждого из шестнадцати параллельных блоков $\mathbf{P}^{(e)}{}_{2/1}$ формирует правый (левый) 16-битовый вход (выход) блока $\mathbf{P}^{(e)}{}_{2\times16/1}$. На вход блока $\mathbf{P}^{(e)}{}_{96/1}$ подается управляющий вектор $V = (V_1, V_2, ..., V_6)$. Каждый из трех блоков $\mathbf{P}^{(e)}{}_{2\times16/1}$ выполняет зависимую от e перестановку некоторой пары 16-битовых компонентов

управляющего вектора V таким образом, что при $e = 0$ компоненты $V_1, V_2, ..., V_6$ распределяются сверху вниз, а при $e = 1$ – снизу вверх. Этот механизм обеспечивает реализацию прямой операции $\mathbf{F}_{32/96}$ при $e = 0$ и обратной $\mathbf{F}^{-1}{}_{32/96}$ при $e = 1$. Структура переключаемого блока $\mathbf{F}^{(e)}{}_{32/96}$ показана на рис. 7.15б.

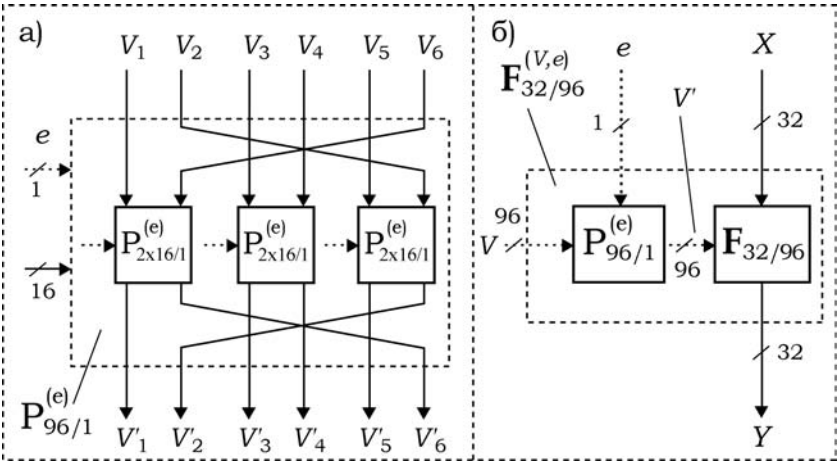

Рис. 7.15. Построение переключаемого блока $\mathbf{F}^{(V,e)}{}_{32/96}$: а) механизм перераспределения компонентов управляющего вектора, б) структура ПУОП $\mathbf{F}^{(V,e)}{}_{32/96}$

Аналогичным образом переключаемую УППС $\mathbf{F}^{(V,e)}{}_{64/192}$ можно построить с помощью однокаскадного перестановочного блока $\mathbf{P}^{(e)}{}_{192/1}$, представляющего собой три параллельных однослойных блока $\mathbf{P}^{(e)}{}_{2\times32/1}$ (рис. 7.16а). Каждый блок $\mathbf{P}^{(e)}{}_{2\times32/1}$ является множеством из 32 параллельных блоков $\mathbf{P}^{(e)}{}_{2/1}$, каждый из которых управляется битом e. Структура блока $\mathbf{F}^{(e)}{}_{64/192}$ показана на рис. 7.16б.

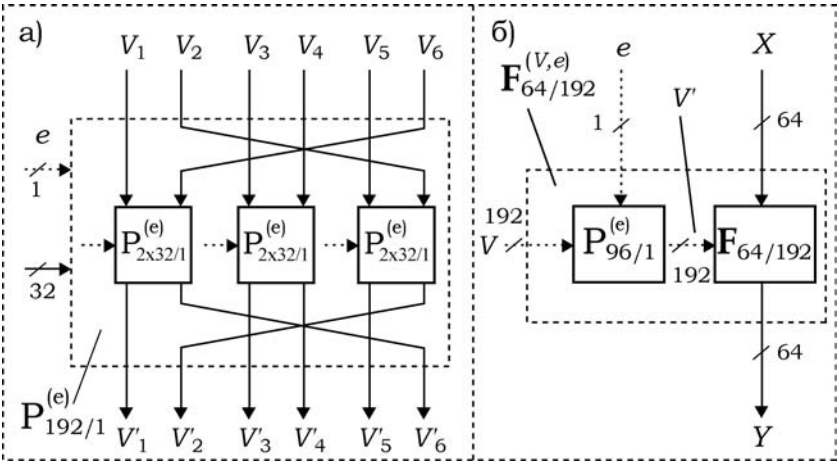

Рис. 7.16. Построение переключаемого блока $\mathbf{F}^{(V,e)}{}_{64/192}$:
а) механизм перераспределения компонентов управляющего вектора,
б) структура ПУОП $\mathbf{F}^{(V,e)}{}_{64/192}$

7.6. Переключаемые УППС различных порядков

Использование симметричной топологии при проектировании переключаемых УППС является частным случаем. В общем же случае достаточно трудно построить БУП с симметричной структурой, которые бы обладали заданными значениями порядка и размера входа. В связи с этим представляют интерес методы синтеза ПУОП различных порядков. Наибольший интерес представляет синтез блоков порядков 2, 4, 8, ..., $n/4$ (переключаемые блоки n-го порядка с симметричной топологией легко синтезируются с использованием рекурсивного построения третьего типа, рассмотренного в разделе 7.4). Универсальный метод построения переключаемых УППС указанных порядков может быть получен на основе использования двух взаимно-обратных блоков $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ половинного порядка и двух взаимно-обратных каскадов $\mathbf{L}_n$ (входной) и $\mathbf{L}^{-1}{}_n$ (выходной), соединенных с блоками $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ в соответствии со схемой, показанной на рис. 7.17. При этом реализуется типовое правило рекурсивного построения: каждый управляемый элемент верхнего (входного) и нижнего (выходного) каскадов соединен с каждым из блоков $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$.

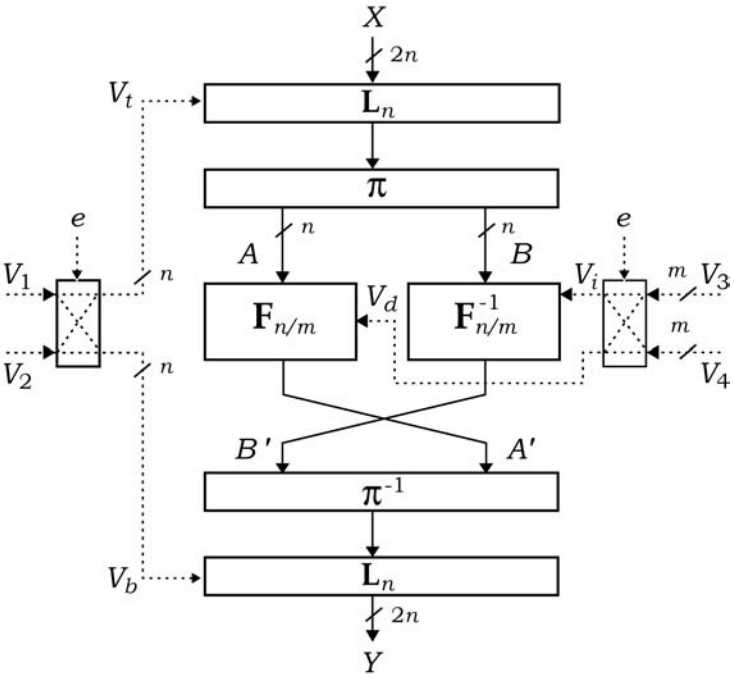

Рис. 7.17. Построение ПУО на основе двух взаимно-обратных УППС

Эта схема построения напоминает один шаг рекурсивного построения третьего типа, однако в данном случае использование двух взаимно-обратных внутренних

УППС обеспечивает реализацию попарно взаимно-обратных модификаций при любой фиксированной битовой перестановки $\boldsymbol{\pi}$ (перестановка $\boldsymbol{\pi}^{-1}$ определяется выбором $\boldsymbol{\pi}$). Но для того чтобы эта схема приводила к удвоению порядка синтезируемого УППС, необходимо, чтобы перестановка $\boldsymbol{\pi}$ отвечала типовому правилу рекурсивного построения (очевидно, что при этом перестановка $\boldsymbol{\pi}^{-1}$ также будет отвечать этому правилу). Число перестановок $\boldsymbol{\pi}$, удовлетворяющих данному правилу, является очень большим, однако представляется целесообразным выбрать перестановку, которая используется при рекурсивном построении, что упрощает описание структуры операционного блока.

Используя механизмы рекурсивного построения УППС, легко построить УППС $\mathbf{F}_{n/m}$ порядков 1, 2, 4, ..., $n/4$. Если такое построение выполнить с использованием элементарных управляемых инволюций $\mathbf{F}^{*}_{2/1}$ или элементов $\mathbf{F}_{2/1}$ произвольного типа, то в обоих случаях не представляет труда построить УППС $\mathbf{F}^{-1}{}_{n/m}$, являющуюся обратной по отношению к заданному s-каскадному блоку

$$\mathbf{F}_{n/m} = \mathbf{L}\circ\boldsymbol{\pi}_1\circ\mathbf{L}\circ\boldsymbol{\pi}_2\circ\ldots\circ\boldsymbol{\pi}_{s-1}\circ\mathbf{L}:$$

$$\mathbf{F}^{-1}{}_{n/m} = \mathbf{L}^{-1}\circ\boldsymbol{\pi}^{-1}{}_{s-1}\circ\mathbf{L}^{-1}\circ\boldsymbol{\pi}^{-1}{}_{s-2}\circ\ldots\circ\boldsymbol{\pi}^{-1}{}_{1}\circ\mathbf{L}^{-1}.$$

Если блок $\mathbf{F}_{n/m}$ имеет порядок h, то и блок $\mathbf{F}^{-1}{}_{n/m}$ имеет порядок h, причем реализация БУП $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ одного порядка требует использования одинакового минимального числа активных слоев. Таким образом, метод, представленный на рис. 7.17, позволяет, используя две взаимно-обратные УППС $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$, имеющие одинаковый порядок h, создать переключаемую УППС $\mathbf{F}^{(V,e)}{}_{2n/2(m+n)}$ порядка $2h$, где $V = (V_1, V_2, V_3, V_4)$ – управляющий вектор построенной УППС. Легко показать, что для всех значений V имеем

$$X = \mathbf{F}^{(V,1)}{}_{2n/2(m+n)}(Y), \text{ если } Y = \mathbf{F}^{(V,0)}{}_{2n/2(m+n)}(X),$$

$$X = \mathbf{F}^{(V,0)}{}_{2n/2(m+n)}(Y), \text{ если } Y = \mathbf{F}^{(V,1)}{}_{2n/2(m+n)}(X).$$

В случае, когда $n = 2^k$, где k – некоторое натуральное число, минимальное число активных слоев в блоке $\mathbf{F}_{n/m}$ (или $\mathbf{F}^{-1}{}_{n/m}$) порядка $h = 1, 2, \ldots, n/4$ равно

$$s_{min} = \log_2 hn.$$

Для случая максимального порядка $h = n$ [14] имеем

$$s_{min} = \log_2 hn - 1.$$

Чтобы с помощью описанного метода построения создать переключаемый блок $\mathbf{F}^{(V,e)}{}_{n'/m'}$ порядка $h' = 2, 4, \ldots, n'/4$, где $n' = 2n$ и $m' = 2(n+m)$, следует использовать блоки $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ порядка $h = h'/2$ и добавить к ним два дополнительных активных слоя.

Таким образом, минимальное число необходимых активных слоев равно

$$s'_{min} = \log_2 hn + 2 = \log_2 4hn = \log_2 h'n'.$$

Случай $h' = n'$ соответствует переключаемым УППС максимального порядка, которые можно построить при помощи блоков $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ порядка $h = n$. Можно показать, что в этом случае имеем

$$s'_{min} = \log_2 h' n' - 1.$$

Если $n = 2^k$, то легко создать блок максимального порядка $\mathbf{F}^{(V)}{}_{n/m}$ с симметричной структурой. Следовательно, переключаемый блок максимального порядка $\mathbf{F}^{(V,e)}{}_{n/m}$ может быть также построен и с помощью метода, описанного в разделе 7.4.

7.7. Упрощение аппаратной реализации ПУОП

Рассмотренные ранее варианты синтеза ПУОП используют скачкообразное изменение распределения всех управляющих битов m-битового вектора V. Свойство переключаемости обеспечивается в них для произвольного распределения зависимых или независимых управляющих битов. Однако для реализации данных схем построения ПУОП требуется использование $m/2$ переключающих элементов $\mathbf{P}_{2/1}$ (каждый элемент изменяет распределение двух управляющих битов). При разработке шифров в большинстве случаев размер управляющего подблока данных, в зависимости от которого формируется управляющий вектор V, существенно меньше значения m и равен значению n (т. е. размеру входа УППС). Удобно использовать такие УППС, в которых отношение m/n равно целому натуральному числу, что позволяет обеспечить определенную однородность влияния всех битов управляющего подблока данных на выбор текущей модификации УОП. Для значения $n = 32$ указанное отношение в блоках второго порядка равно 3. Такое же значение этого отношения имеет место в УППС первого порядка с размером входа равным 64 бит. В этих случаях задается зависимость «непересекающихся» троек битов управляющего вектора V от каждого бита управляющего подблока данных. Это осуществляется путем расширения управляющего подблока данных (например, утроения его длины). Расширенный подблок данных может использоваться непосредственно в качестве управляющего вектора. Последний может также формироваться путем наложения (обычно с помощью операции XOR) на расширенный управляющий подблок данных одного или нескольких подключей. В этом случае обеспечивается зависимость текущей реализуемой модификации УППС не только от подблока данных, но и от секретного ключа.

Если управляющий вектор является равновероятной случайной величиной, то можно использовать произвольное распределение управляющих битов по управляемым элементам УППС. В случае, когда имеется зависимость между управляющими битами, распределение выбирается исходя из определенных критериев, обеспечивающих равномерность влияния всех битов управляющего подблока на выбор текущей модификации управляемой операции. Построение распределения управляющих битов, удовлетворяющего тем или иным критериям, более просто выполняется для некоторых специальных топологических структур УППС. Такими, например, являются симметричные структуры блоков $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$, описанные в предыдущем разделе. В случае 64- и 128-битовых шифров получение независимости битов в управляющем векторе связано с разбиением входного блока данных на подблоки различ-

ных размеров, причем управляющий подблок должен иметь размер в два – три раза больший размера преобразуемого подблока данных. Такое разбиение является нетипичным и требует разработки специальных криптосхем, а кроме того, в этом случае требуется большее число раундов для обеспечения влияния каждого входного бита данных на каждый выходной бит.

Поскольку от зависимости управляющих битов в векторе V избавиться разумным способом не удается, то можно попытаться найти более экономичные механизмы переключения распределения управляющих битов, приводящие к обращению УППС, т. е. к смене прямой операции на обратную. Действительно, можно предположить, что достаточно осуществить некоторую перестановку n битов управляющего подблока данных, которая приведет к зависящему от бита режима операции e перераспределению m битов вектора V. Можно также предположить, что реализация такого способа обращения управляемой операции должна опираться на использование не только симметрии топологической структуры УППС, но и соответствующей симметрии распределения битов управляющего подблока данных. Ниже рассматривается способ обращения УППС, показывающий практическую реализуемость такого подхода. Существенное достоинство излагаемых далее механизмов переключения распределения битов управляющего подблока данных состоит в том, что для своей реализации они требуют использования только $n/2$ переключающих элементов $\mathbf{P}_{2/1}$ вместо $m/2$ элементов в случае переключения произвольного распределения битов управляющего вектора V. В случае реализации переключаемых УППС $\mathbf{F}^{(e)}{}_{32/96}$ и $\mathbf{F}^{(e)}{}_{64/192}$ потребуется на 32 и 64 блоков $\mathbf{P}_{2/1}$ меньше, что уменьшит сложность схемотехнической реализации данных переключаемых УППС до значения 1.03 – 1.17 от сложности реализации УППС $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$, которые не обладают свойством переключаемости. Указанный разброс значений связан с различной сложностью реализации управляемых элементов $\mathbf{F}_{2/1}$ различного типа и возможностью различных вариантов их схемотехнической реализации (при этом схемотехнические ресурсы, затрачиваемые на создание механизма переключения, остаются фиксированными).

Переключаемые УППС $\mathbf{F}^{(e)}{}_{32/96}$ и $\mathbf{F}^{(e)}{}_{64/192}$ такого типа имеют одинаковый размер управляющего и информационного входов, а блок расширения является внутренней частью. Если на формируемый расширенный управляющий вектор предполагается накладывать ключи, то необходимо предусмотреть дополнительный вход для введения ключей и дополнительную схему для выполнения операции, с помощью которой будет осуществляться такое наложение. Во многих криптосхемах, основанных на управляемых операциях, не используется элемент наложения подключей на расширенный управляющий вектор и в них могут быть использованы более простые ПУОП.

На рис. 7.18, где УППС $\mathbf{F}_{n/m}$ с зеркально-симметричной структурой представлен в виде суперпозиции двух взаимно-обратных блоков $\mathbf{F}_{n/m'}$ и $\mathbf{F}^{-1}{}_{n/m}$, схематично представлено построение переключаемых операционных блоков в случае всех независимых управляющих битов (а) и в случае расширения управляющего подблока данных для формирования управляющего вектора (б). В последнем случае управляющие биты векторов V'_1 и V'_2 распределяются по блокам $\mathbf{F}_{n/m'}$ и $\mathbf{F}^{-1}{}_{n/m'}$ в соответствии с зеркальной симметрией, благодаря чему при перестановке векторов L_1 и L_2 обеспечива-

ется корректное переключение УППС с режима выполнения прямой управляемой операции в режим выполнения обратной операции.

Переключаемые УППС $\mathbf{F}^{(e)}_{32/96}$ и $\mathbf{F}^{(e)}_{64/192}$ могут быть легко построены в соответствии с конструктивной схемой, ориентированной на экономичную аппаратную реализацию (рис. 7.18б). По сравнению с УППС $\mathbf{F}_{32/96}$ и $\mathbf{F}_{64/192}$ обычного типа блоки $\mathbf{F}^{(e)}_{32/96}$ и $\mathbf{F}^{(e)}_{64/192}$ в указанном варианте реализации требуют дополнительного использования 16 и 32 элементарных переключателей $\mathbf{P}_{2/1}$, соответственно.

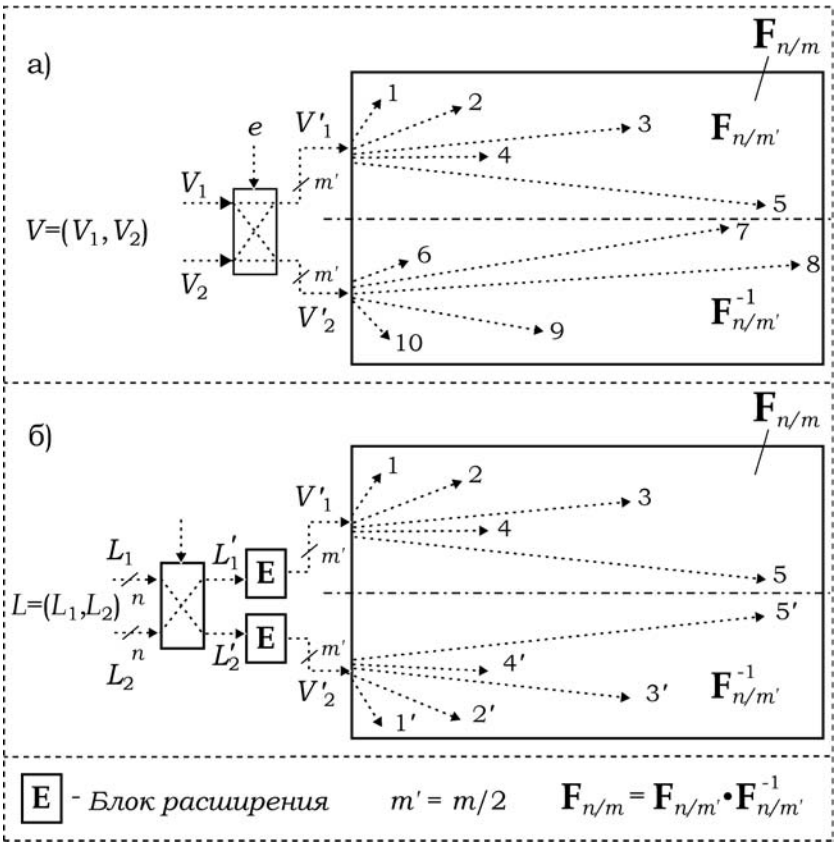

Рис. 7.18. Сопоставление двух вариантов реализации ПУОП:
а) с произвольным управляющим вектором, б) с использованием расширения управляющего подблока данных

7.8. Переключаемые УППС с управляемыми элементами, включающими попарно взаимно-обратные модификации

Представляет интерес реализация ПУОП, в которых механизм обращения выполняемой управляемой операции основан на том, что УППС построена на основе

управляемых элементов $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$ или $\mathbf{F}_{3/1}$, модификации которых являются попарно взаимно-обратными. Подобные управляемые элементы могут быть обращены путем инвертирования управляющих битов. В случае элемента $\mathbf{F}_{2/2}$ это реализуется, если каждая из двух пар модификаций, соответствующих паре управляющих векторов (00) и (11) и паре (01) и (10), включает две взаимно-обратные модификации.

Способ обращения УППС на основе инвертирования всех управляющих битов также требует использования симметричной топологии УППС и симметричного распределения управляющих битов. В связи с последним данный способ применим только в случае, когда не более половины управляющих битов являются независимыми (на управляющий вход двух управляемых элементов, расположенных в зеркально симметричных позициях, должен подаваться один и тот же бит). Если управляющий вектор блока $\mathbf{F}_{n/m}$ формируется по подблоку данных размером n бит, то, как правило, биты управляющего подблока данных легко распределить симметричным образом. Однако, если отношение m/n есть нечетное число, то биты из одной половины этого подблока придется использовать для управления большим числом элементарных подстановочных элементов, чем биты из второй половины. Эта неравномерность не является критической, однако ее следует иметь в виду при разработке ПУОП для применения в конкретных шифрах. Также определенным недостатком метода обращения УППС путем инвертирования управляющих битов является то, что указанное симметричное распределение управляющих битов делает невозможным реализацию одного из полезных критериев формирования управляющих векторов, который может быть сформулирован следующим образом: управляющие биты должны быть распределены так, чтобы ни один бит преобразуемого подблока данных не подвергался дважды влиянию одного и того же бита управляющего подблока данных.

Несмотря на указанные особенности, рассматриваемый метод обращения УППС имеет практическое значение, поскольку он предоставляет еще один способ экономичного построения механизма переключения прямой операции на обратную. Рассмотрим варианты построения ПУОП такого типа с использованием управляемых элементов различного вида.

7.8.1. Переключаемые УППС на основе элементов типа $\mathbf{F}_{2/1}$

Рассмотрев существующие типы нелинейных управляемых элементов $\mathbf{F}_{2/1}$, описанных в разделе 4, легко установить, что элементы, заданные парами модификаций q/u, r/x, s/v, t/w, , u/q, x/r, v/s и w/t (см. рис. 7.19), представляют собой управляемые элементарные подстановки, реализующие пару взаимно-обратных модификаций.

Использование других пар элементарных модификаций подстановок типа 2×2 не приводит к построению вариантов элементов $\mathbf{F}_{2/1}$, удовлетворяющих критерию нелинейности обоих выходов. Перечисленные восемь вариантов являются единственными (фактически мы имеем четыре пары модификаций подстановок типа 2×2, каждая из которых задает два варианта элементов $\mathbf{S}_{2/1}$, отличающиеся выбором модификации, реализуемой при нулевом (единичном) значении управляющего бита). Благоприятным оказывается то обстоятельство, что не только каждый выход этих элементов является нелинейной булевой функцией, аргументами которой являются два

входных бита и управляющий бит, но и сумма выходов также является нелинейной булевой функцией от указанных переменных, т. е. они относятся к элементам типа $\mathbf{S}_{2/1}$. Таблица 7.3 описывает указанные четыре варианта элементов, обладающих свойством $\mathbf{S}^{(1)} = (\mathbf{S}^{(0)})^{-1}$, как пару булевых функций от трех переменных.

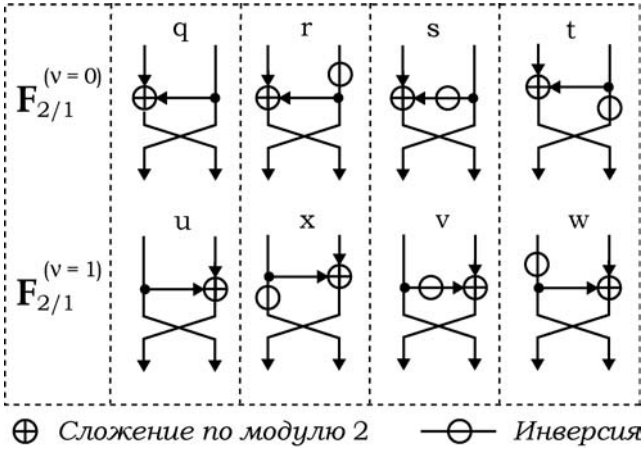

Рис. 7.19. Пары взаимно-обратных элементарных преобразований типа $(x_1,x_2) \to (y_1,y_2)$

Таблица 7.3

Булевы функции, описывающие управляемые элементы $\mathbf{S}_{2/1}$, реализующие пару взаимно-обратных модификаций.

	$y_1 = f_1(x_1, x_2, v)$	$y_2 = f_2(x_1, x_2, v)$	$y_1 \oplus y_2$
q/u	$vx_1 \oplus x_2$	$vx_2 \oplus x_1 \oplus x_2$	$vx_1 \oplus vx_2 \oplus x_1$
s/v	$vx_1 \oplus v \oplus x_2$	$vx_2 \oplus v \oplus x_1 \oplus x_2 \oplus 1$	$vx_1 \oplus vx_2 \oplus x_1 \oplus 1$
t/w	$vx_1 \oplus x_2 \oplus 1$	$vx_2 \oplus v \oplus x_1 \oplus x_2$	$vx_1 \oplus vx_2 \oplus v \oplus x_1 \oplus 1$
r/x	$vx_1 \oplus v \oplus x_2 \oplus 1$	$vx_2 \oplus x_1 \oplus x_2 \oplus 1$	$vx_1 \oplus vx_2 \oplus v \oplus x_1$
u/q	$vx_1 \oplus x_1 \oplus x_2$	$vx_2 \oplus x_1$	$vx_1 \oplus vx_2 \oplus x_2$
v/s	$vx_1 \oplus v \oplus x_1 \oplus x_2 \oplus 1$	$vx_2 \oplus v \oplus x_1$	$vx_1 \oplus vx_2 \oplus x_2 \oplus 1$
w/t	$vx_1 \oplus x_1 \oplus x_2 \oplus 1$	$vx_2 \oplus v \oplus x_1 \oplus 1$	$vx_1 \oplus vx_2 \oplus v \oplus x_2$
x/r	$vx_1 \oplus v \oplus x_1 \oplus x_2$	$vx_2 \oplus x_1 \oplus 1$	$vx_1 \oplus vx_2 \oplus v \oplus x_2 \oplus 1$

7.8.2. Переключаемые УППС на основе элементов типа $\mathbf{F}_{2/2}$

По сравнению с синтезом элементов $\mathbf{F}_{2/1}$ формирование элементарных управляемых подстановок типа $\mathbf{F}_{2/2}$ связано с рассмотрением гораздо большего числа вариан-

тов, однако требование взаимной обратимости двух пар модификаций элемента $\mathbf{F}_{2/2}$ очень сильно сужает выбор. Более того, если в случае элементов $\mathbf{F}_{2/1}$ данное требование не привело к отбрасыванию наиболее эффективных нелинейных элементов этого типа, то в случае элементов $\mathbf{F}_{2/2}$ наиболее эффективные варианты (по критериям нелинейности) не удовлетворяют условию разбиения четырех реализуемых модификаций на две пары взаимно-обратных модификаций. Тем не менее, остающиеся типы элементов $\mathbf{F}_{2/2}$ обладают достаточно хорошими нелинейными свойствами и свойствами распространения изменений, что позволяет построить на их основе ПУОП, пригодные для применения в криптосистемах. По крайней мере, булевы функции, описывающие каждый их двух выходов этого элемента и их сумму, имеют более высокое значение нелинейности и алгебраической степени нелинейности по сравнению с элементами $\mathbf{F}_{2/1}$.

При этом, несмотря на существенные ограничительные условия $\mathbf{F}^{(11)} = (\mathbf{F}^{(00)})^{-1}$ и $\mathbf{F}^{(10)} = (\mathbf{F}^{(01)})^{-1}$, число элементов $\mathbf{F}_{2/2}$ такого типа, реализующих четыре различных модификации, достаточно велико. Еще возможны варианты элементов $\mathbf{F}_{2/2}$, реализующие одну и ту же модификацию при двух различных значениях управляющего вектора (v_1, v_2). Например, оба выхода элемента m/e/e/n, где в последней записи перечислены модификации подстановок типа 2×2, реализуемые при значениях управляющего вектора (0, 0), (0, 1), (1, 0) и (1, 1), соответственно, и их сумма являются нелинейными булевыми функциями от четырех переменных:

$$y_1 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_1v_2 \oplus v_1x_2 \oplus v_2x_2;$$
$$y_2 = v_1x_1 \oplus v_1x_2 \oplus v_2x_2 \oplus v_1 \oplus v_2 \oplus x_2 \oplus 1;$$
$$y_1 \oplus y_2 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_1v_2 \oplus v_1x_1 \oplus v_2x_1 \oplus v_1 \oplus v_2 \oplus x_2 \oplus 1.$$

Следует отметить, что при фиксировании одного управляющего бита v_1 или v_2 в элементе m/e/e/n он превращается в элемент типа $\mathbf{S}_{2/1}$. Приведем примеры управляемых элементов с четырьмя различными модификациями. Элемент q/k/l/u описывается следующими булевыми функциями (см. рис. 7.20а):

$$y_1 = v_1v_2x_1 \oplus v_1x_2 \oplus v_2x_2 \oplus v_2x_1 \oplus v_1x_1 \oplus v_1 \oplus v_2 \oplus x_2;$$
$$y_2 = v_1v_2x_2 \oplus v_1v_2 \oplus x_1 \oplus x_2 \oplus v_2;$$
$$y_1 \oplus y_2 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_1v_2 \oplus v_1x_1 \oplus v_1x_2 \oplus v_2x_1 \oplus v_2x_2 \oplus v_1 \oplus x_1.$$

Элемент q/o/p/u описывается следующими булевыми функциями (см. рис. 7.20б):

$$y_1 = v_1v_2x_1 \oplus v_1v_2 \oplus v_1 \oplus x_2;$$
$$y_2 = v_1v_2x_2 \oplus v_1v_2 \oplus v_1x_2 \oplus v_2x_2 \oplus v_2 \oplus x_1 \oplus x_2;$$
$$y_1 \oplus y_2 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_1x_2 \oplus v_2x_2 \oplus v_1 \oplus v_2 \oplus x_1.$$

Приведенные варианты реализации управляемых элементов $\mathbf{F}_{2/2}$, обращаемых путем инвертирования управляющих битов, могут быть использованы для синтеза ПУОП.

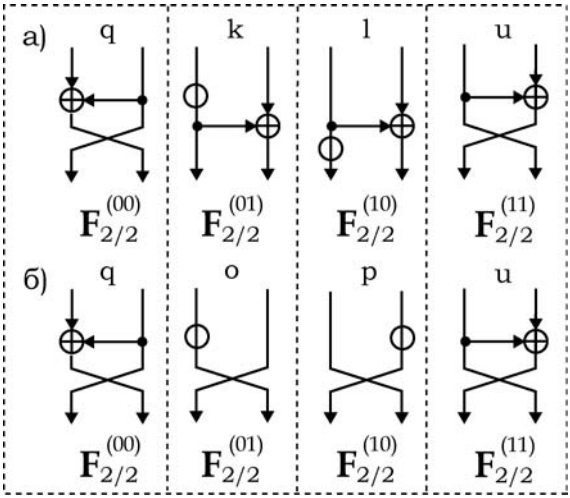

Рис. 7.20. Переключаемые управляемые элементы
а) q/k/l/u, б) q/o/p/u

Заметим, что в ПУОП на основе элементов $\mathbf{F}_{2/2}$ симметричное распределение управляющих битов предполагает, что на управляющий вход элементов, расположенных в симметричных позициях, подается одинаковый управляющий вектор (v_1v_2), т. е. этими элементами реализуется одна и та же модификация подстановки типа 2×2.

7.8.3. Переключаемые УОП на основе элементов типа $F_{3/1}$

Большое число возможных вариантов подстановок типа 3×3 позволяет синтезировать элементарные управляемые подстановки типа $\mathbf{F}_{3/1}$, обладающие свойством $\mathbf{F}^{(1)} = (\mathbf{F}^{(0)})^{-1}$ и требуемыми свойствами нелинейности. Интерес использования трехвходовых управляемых элементов для синтеза ПУОП состоит в том, что они обеспечивают нелинейность УОП при фиксировании управляющего подблока данных, т. е. каждый выходной бит УОП является булевой функцией, содержащей члены, в которые в качестве сомножителей входит большое число входных битов и большое число управляющих битов. Для сравнения укажем, что высокая алгебраическая степень нелинейности выходов УППС, построенных на основе элементов $\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$, связана с тем, что соответствующие им булевы функции содержат члены с большим числом управляющих битов в качестве сомножителей, но ни один из членов не содержит более одного сомножителя, являющегося входным битом. Можно говорить, что УППС на основе элементов $\mathbf{F}_{3/1}$ обеспечивают более сильное совместное перемешивание управляющего и преобразуемого подблока данных. В свою очередь УППС на основе элементов $\mathbf{F}_{2/2}$ дают «более высокую степень перемешивания управляющего подблока в преобразуемом подблоке данных».

Именно это свойство делает их интересными для построения ПУОП, несмотря на неудобство, связанное с наличием нечетного размера входа. Если в некоторой криптосхеме предполагается разбиение блока данных на два подблока с одинаковой длиной, то последнее приводит к необходимости включения в каждый активный каскад УППС элементов $\mathbf{F}_{2/1}$ и/или $\mathbf{F}_{2/2}$ и некоторому усложнению анализа оценки стойкости шифров. Однако могут быть использованы криптосхемы с несимметричным разбиением блока данных на подблоки. Например, в случае шифра с 64-битовым входом может разработана криптосхема с разбиением 16 бит + 48 бит, а в случае шифра со 128-битовым входом – криптосхема с разбиением 32 бит + 96 бит, где подблок данных меньшего размера предполагается использовать в качестве управляющего. В таких криптосхемах представляется возможным использовать блоки, активный каскад которых состоит из трехвходовых управляемых элементов (шестнадцати в первом и тридцати двух во втором случае). Указанное разбиение обеспечивает определенный баланс: в каждом активном каскаде будет использоваться один раз каждый бит управляющего подблока. Такой баланс характерен и для УППС на основе элементов $\mathbf{F}_{2/2}$ в случае равенства размеров управляющего и преобразуемого подблоков данных. Элементы $\mathbf{F}_{3/1}$ могут найти самостоятельное применение в переключаемых УППС, применяемых в криптосхемах с подблоками данных разной длины, или могут использоваться совместно с двухвходовыми управляемыми элементами в УППС, ориентированных на применение в криптосхемах с подблоками данных одинакового размера.

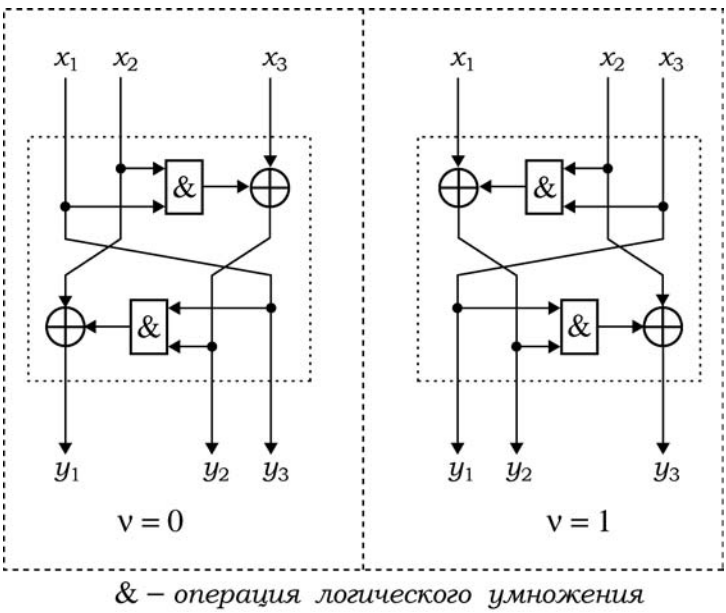

Рис. 7.21. Управляемый элемент, модификация $\mathbf{F}_{3/1}$, заданный в виде двух взаимно-обратных модификаций:
а) модификация $\mathbf{F}^{(1)}_{3/1}$, б) модификация $\mathbf{F}^{(0)}_{3/1}$

Так же как и в случае элементов $\mathbf{F}_{2/2}$ полное перечисление и характеризация элементов $\mathbf{F}_{3/1}$, которые могут быть обращены путем инвертирования управляющего бита, является чрезмерно громоздким. В связи с этим приведем только некоторые характерные примеры. На рис. 7.21 представлен элемент $\mathbf{F}_{3/1}$, включающий модификации, для которых выполняется условие $\mathbf{F}^{(1)}{}_{3/1} = (\mathbf{F}^{(0)}{}_{3/1})^{-1}$. Как показано в *главе 6*, общее представление булевых функций от четырех переменных, описывающих элементы $\mathbf{F}_{3/1}$ через булевы функции от трех переменных, описывающие две его модификации, имеет вид:

$$y_1 = (v \oplus 1) f'_1(x_1, x_2, x_3) \oplus v f''_1(x_1, x_2, x_3) x_2 ;$$

$$y_2 = (v \oplus 1) f'_2(x_1, x_2, x_3) \oplus v f''_2(x_1, x_2, x_3) x_2 ;$$

$$y_3 = (v \oplus 1) f'_3(x_1, x_2, x_3) \oplus v f''_3(x_1, x_2, x_3) x_2 ,$$

где функции f'_1, f'_2, f'_3 относятся к модификации, реализуемой при нулевом значении управляющего бита v, а f''_1, f''_2, f''_3 – к модификации, реализуемой при $v = 1$. По рис. 7.21 легко записать все указанные БФ от трех переменных и найти выражение для булевых функций от четырех переменных, описывающих каждый выход элемента $\mathbf{F}_{3/1}$:

$$y_1 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2;$$

$$y_2 = vx_1x_2 \oplus vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_3;$$

$$y_3 = vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus x_1.$$

Линейные комбинации выходов имеют вид нелинейных булевых функций:

$$y_1 \oplus y_2 = vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus x_1x_3 \oplus x_2 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_2;$$

$$y_2 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2 \oplus vx_3 \oplus x_1x_2 \oplus x_1 \oplus x_3;$$

$$y_1 \oplus y_2 \oplus y_3 = x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3.$$

Анализ булевых функций рассмотренного элемента $\mathbf{F}_{3/1}$ показывает, что он является нелинейным примитивом и пригоден для использования при построении переключаемых УППС.

Рассмотрим несколько других дополнительных примеров аналитического задания нелинейных УЭ типа $\mathbf{F}_{3/1}$, для которых выполняется условие $\mathbf{F}^{(1)}{}_{3/1} = (\mathbf{F}^{(0)}{}_{3/1})^{-1}$.

Пример 1. Управляемый элемент задан следующими тремя булевыми функциями от четырех переменных:

$$y_1 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2 \oplus vx_3 \oplus x_2;$$

$$y_2 = vx_2x_3 \oplus vx_3 \oplus x_2x_3 \oplus x_1 \oplus x_3;$$

$$y_3 = vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus vx_3 \oplus x_2x_3 \oplus x_1.$$

Линейные комбинации выходов имеют вид:

$$y_1 \oplus y_2 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_2 \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus x_2x_3 \oplus x_1 \oplus x_2;$$

$$y_2 \oplus y_3 = vx_1 \oplus vx_2 \oplus x_3;$$

$$y_1 \oplus y_2 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_1 \oplus vx_3 \oplus x_2 \oplus x_3.$$

В рассмотренном примере линейная комбинация второго и третьего выходов имеет алгебраическую степень нелинейности, равную двум, а остальные линейные комбинации и сами выходы описываются булевыми функциями с алгебраической степенью, равной трем.

Пример 2. Управляемый элемент задан нелинейными булевыми функциями с алгебраической степенью нелинейности, равной 3:

$$y_1 = vx_1x_3 \oplus vx_2 \oplus vx_3 \oplus x_2;$$

$$y_2 = vx_1x_2 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_3;$$

$$y_3 = vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1.$$

Линейные комбинации выходов описываются следующим образом:

$$y_1 \oplus y_2 = vx_1x_2 \oplus vx_1x_3 \oplus vx_1 \oplus vx_2 \oplus x_1x_2 \oplus x_2 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1 \oplus x_2;$$

$$y_2 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_2 \oplus vx_3 \oplus x_1 \oplus x_3;$$

$$y_1 \oplus y_2 \oplus y_3 = vx_1x_2 \oplus vx_2x_3 \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3.$$

В данном примере все булевы функции, описывающие значения выходных битов и их линейные комбинации, имеют алгебраическую степень 3.

Пример 3. Рассматриваемый набор булевых функций, соответствующий этому примеру, представлен следующими формулами:

$$y_1 = vx_1x_2 \oplus vx_2 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_2;$$

$$y_2 = vx_1x_2 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_3;$$

$$y_3 = vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1 \oplus x_2.$$

Линейные комбинации выходов имеют вид:

$$y_1 \oplus y_2 = vx_2 \oplus x_2 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3;$$

$$y_2 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3;$$

$$y_1 \oplus y_2 \oplus y_3 = vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus vx_3 \oplus x_1 \oplus x_3.$$

В примере 3 с целью усиления вклада управляемого элемента $\mathbf{F}_{3/1}$ в распространение лавинного эффекта при осуществлении преобразования с помощью УППС в каждую из двух взаимно-обратных модификаций элемента $\mathbf{F}_{3/1}$ введены дополнительные операции сложения по модулю два.

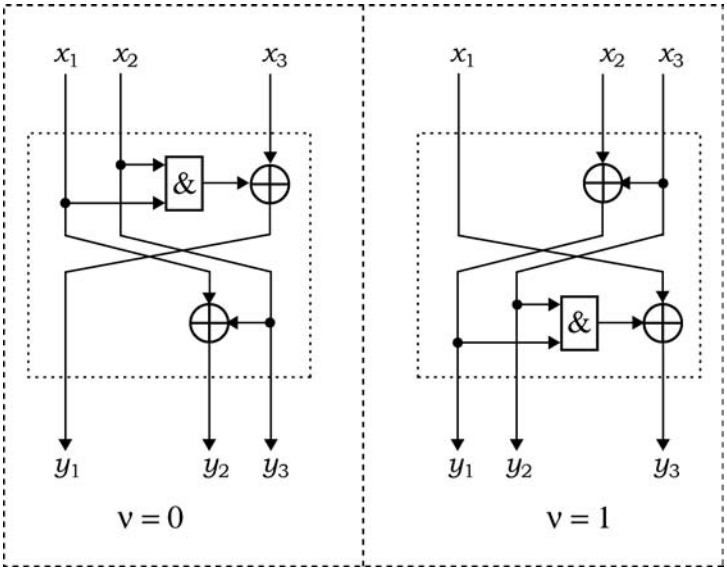

Рис. 7.22. Управляемый элемент $\mathbf{F}_{3/1}$, заданный в виде двух взаимно-обратных модификаций, описываемых БФ второй степени

Пример 4 поясняется рисунком 7.22, где имеем следующий набор булевых функций:

$$y_1 = vx_1x_2 \oplus vx_2 \oplus x_1x_2 \oplus x_3;$$

$$y_2 = vx_1 \oplus vx_2 \oplus vx_3 \oplus x_1 \oplus x_2;$$

$$y_3 = vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus vx_3 \oplus x_2.$$

Линейные комбинации выходов имеют вид нелинейных булевых функций:

$$y_1 \oplus y_2 = vx_1x_2 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_1 \oplus x_2 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_2x_3 \oplus vx_1 \oplus vx_3 \oplus x_1x_2 \oplus x_2 \oplus x_3;$$

$$y_2 \oplus y_3 = vx_2x_3 \oplus x_1;$$

$$y_1 \oplus y_2 \oplus y_3 = vx_2x_3 \oplus vx_1x_2 \oplus vx_2 \oplus x_1x_2 \oplus x_1 \oplus x_3.$$

Пример 5 поясняется рисунком 7.23, где имеем следующий набор булевых функций:

$$y_1 = vx_1x_2 \oplus x_2x_3 \oplus x_1;$$

$$y_2 = vx_1x_2 \oplus x_3;$$

$$y_3 = vx_1x_3 \oplus vx_2x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_2.$$

Линейные комбинации выходов имеют вид нелинейных булевых функций:

$$y_1 \oplus y_2 = x_2x_3 \oplus x_1 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_2x_3 \oplus vx_1x_3 \oplus x_1x_3 \oplus x_1 \oplus x_2;$$

$y_2 \oplus y_3 = vx_2x_3 \oplus vx_1x_3 \oplus vx_1x_2 \oplus x_2x_3 \oplus x_1x_3 \oplus x_2 \oplus x_3$;

$y_1 \oplus y_2 \oplus y_3 = vx_2x_3 \oplus vx_1x_3 \oplus x_1x_3 \oplus x_1 \oplus x_3$.

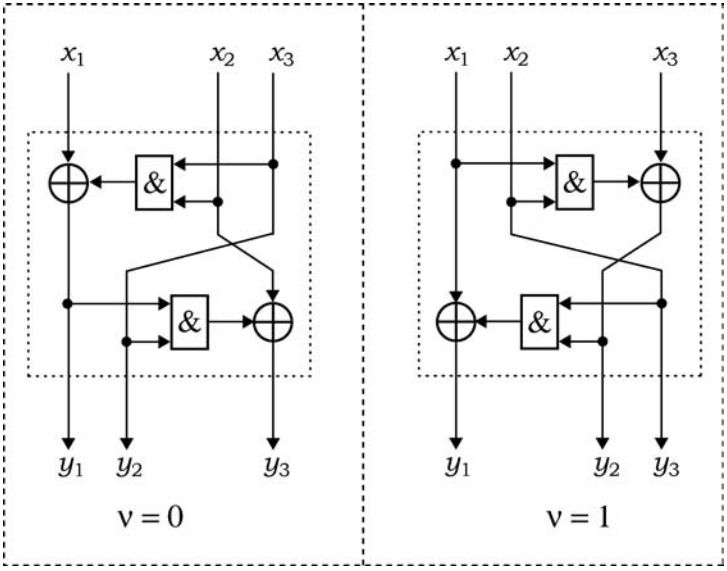

Рис. 7.23. Управляемый элемент $\mathbf{F_{3/1}}$ примера **5**

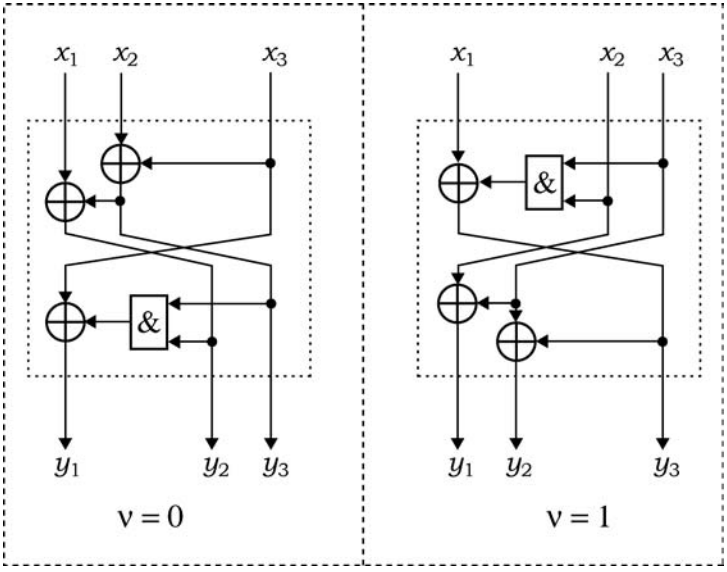

Рис. 7.24. Управляемый элемент $\mathbf{F_{3/1}}$ примера **6**

Пример 6 (см. рис. 7.24) описывается следующими булевыми функциями третьей степени:

$$y_1 = vx_1x_2 \oplus vx_1x_3 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2;$$

$$y_2 = vx_2x_3 \oplus vx_2 \oplus x_1 \oplus x_2 \oplus x_3;$$

$$y_3 = vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus vx_3 \oplus x_2 \oplus x_3.$$

Линейные комбинации выходов имеют вид нелинейных булевых функций:

$$y_1 \oplus y_2 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_2 \oplus vx_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_3;$$

$$y_1 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus vx_2x_3 \oplus vx_1 \oplus vx_2 \oplus x_1x_2 \oplus x_1x_3 \oplus x_3;$$

$$y_2 \oplus y_3 = vx_1 \oplus vx_3 \oplus x_1;$$

$$y_1 \oplus y_2 \oplus y_3 = vx_1x_2 \oplus vx_1x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_2.$$

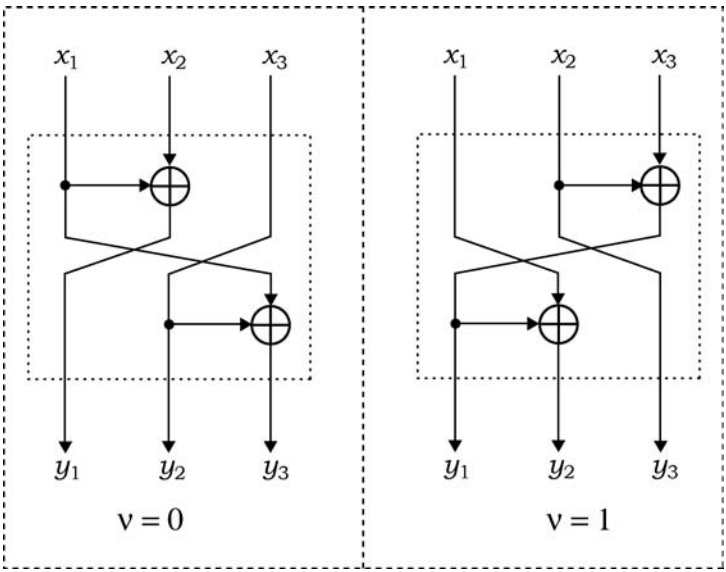

Рис. 7.25. Управляемый элемент $\mathbf{F_{3/1}}$ с двумя линейными модификациями

Пример 7 (см. рис. 7.25) отличается использованием только линейных операций при задании модификаций, реализуемых УЭ, и соответствует следующему набору булевых функций:

$$y_1 = vx_1 \oplus vx_3 \oplus x_1 \oplus x_2;$$

$$y_2 = vx_1 \oplus vx_2 \oplus x_3;$$

$$y_3 = vx_1 \oplus vx_2 \oplus vx_3 \oplus x_1 \oplus x_3.$$

Линейные комбинации выходов имеют вид нелинейных булевых функций второй степени:

$y_1 \oplus y_2 = vx_2 \oplus vx_3 \oplus x_1 \oplus x_2 \oplus x_3$;

$y_1 \oplus y_3 = vx_2 \oplus x_2 \oplus x_3$;

$y_2 \oplus y_3 = vx_3 \oplus x_1$;

$y_1 \oplus y_2 \oplus y_3 = vx_1 \oplus x_2$.

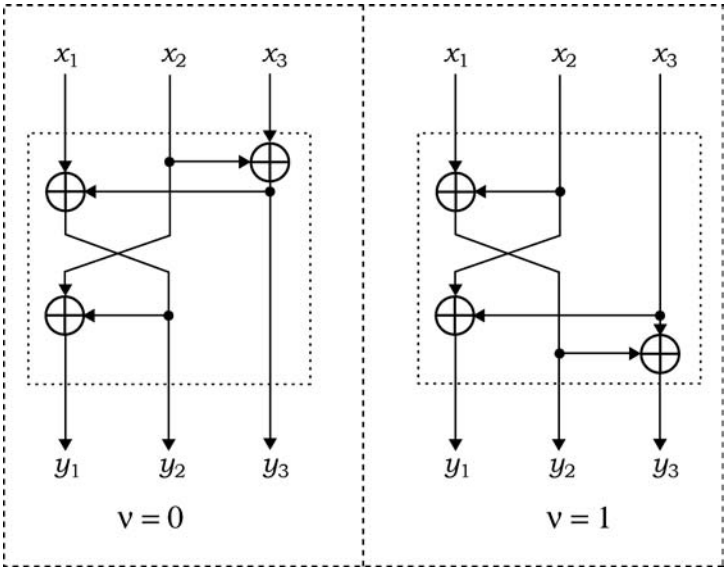

Рис. 7.26. Управляемый элемент $\mathbf{F_{3/1}}$ с двумя линейными модификациями

Пример 8 (см. рис. 7.26) задает такой набор булевых функций второй степени:

$y_1 = vx_1 \oplus vx_2 \oplus x_1 \oplus x_3$;

$y_2 = vx_3 \oplus x_1 \oplus x_2 \oplus x_3$;

$y_3 = vx_1 \oplus x_2 \oplus x_3$.

Линейные комбинации выходов имеют вид нелинейных булевых функций:

$y_1 \oplus y_2 = vx_1 \oplus vx_2 \oplus vx_3 \oplus x_2$;

$y_1 \oplus y_3 = vx_1 \oplus vx_3 \oplus x_1$;

$y_2 \oplus y_3 = vx_2 \oplus x_1 \oplus x_2$;

$y_1 \oplus y_2 \oplus y_3 = vx_2 \oplus vx_3 \oplus x_3$.

В таблицах 7.4 и 7.5 приведены значения нелинейности и алгебраической степени нелинейности для рассмотренных выше примеров. Сравнение показывает, что даже в случае, когда обе модификации, реализуемые УЭ, являются линейными (при-

меры 7 и 8), булевы функции от четырех переменных, описывающие УЭ, обладают максимально возможной нелинейностью. Это показывает, что при использовании элементов $\mathbf{F}_{3/1}$ имеются достаточно широкие возможности проектирования стойких ПУОБ, обращаемых путем ивертирования управляющих битов.

Таблица 7.4

Значение нелинейности NL булевых функций, описывающих выходы УЭ с взаимно-обратными модификациями и их линейные комбинации

Пример	y_1	y_2	y_3	$y_1 \oplus y_2$	$y_1 \oplus y_3$	$y_2 \oplus y_3$	$y_1 \oplus y_2 \oplus y_3$
1	2	2	4	4	4	4	4
2	4	4	4	4	4	4	4
3	4	4	4	2	4	4	4
4	2	4	4	4	2	4	4
5	4	2	2	4	4	4	4
6	4	2	4	4	4	4	2
7	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4

Таблица 7.5

Значение алгебраической степени нелинейности булевых функций, описывающих выходы УЭ с взаимно-обратными модификациями и их линейные комбинации

Пример	y_1	y_2	y_3	$y_1 \oplus y_2$	$y_1 \oplus y_3$	$y_2 \oplus y_3$	$y_1 \oplus y_2 \oplus y_3$
1	3	3	3	3	3	2	3
2	3	3	3	3	3	3	3
3	3	3	3	2	3	3	3
4	3	2	3	3	3	3	3
5	3	3	3	3	3	3	3
6	3	3	3	3	3	2	3
7	2	2	2	2	2	2	2
8	2	2	2	2	2	2	2

7.9. Расширение свойства переключаемости УОП

Одной из важных целей применения ПУОП является устранение гомогенности итеративного шифрования при использовании ключа сравнительно малого размера в случае ПРИК. Для этой цели при наличии одной переключаемой операции в процедуре одного раунда шифрования в последовательных раундах можно установить различные значения бита режима операции таким образом, чтобы они образовали непериодическую последовательность нулевых и единичных значений. Более эффективным вариантом является использование нескольких ПУОП в одном раунде. Это позволит обеспечить более существенное различие в процедурах соседних раундов. Повторяемость раундов шифрования имеет место, только когда повторятся биты на переключающем входе всех ПУОП. Можно было бы разработать криптосхему с большим числом переключаемых операций, выполняемых параллельно над подблоками данных уменьшенного размера. Это обеспечит возможность использования достаточно большого числа независимых переключающих битов, однако эффективность управляемой операции существенно возрастает, когда с их помощью преобразуются подблоки данных большого размера. В связи с этим не предполагается использование в одном раунде большого числа переключаемых операций.

Тем не менее, идея введения более существенных различий не только в соседние раунды, но и в раунды, отстоящие друг от друга на несколько шагов, заслуживает внимания. Эта задача может быть решена путем разработки ПУОП с расширенным диапазоном переключаемости, в которых переключение осуществляется не с помощью одного бита, а с помощью переключающего вектора E длиной, например, равной $k = 4 - 8$ бит. Можно представить, что для данного конкретного значения $E = (e_1, e_1, \ldots, e_k)$ будет реализовываться некоторая управляемая операция, а для $E = (e_1 \oplus 1,\ e_1 \oplus 1,\ \ldots,\ e_k \oplus 1)$ – обратная ей управляемая операция. Получаем $2^{k/2}$ пар взаимно-обратных управляемых операций.

Реализация ПУОП с расширенным диапазоном переключаемости основывается также на скачкообразных изменениях распределения управляющих битов по элементарным управляемым подстановочным блокам $\mathbf{F}_{2/1}$ в некоторой фиксированной топологии. Тип УППС определяется топологией взаимных связей управляемых элементов и распределением управляющих битов. Выше были рассмотрены ПУОП с двухвариантным распределением, которые переключались одним битом e. По аналогии могут быть построены ПУОП с многовариантным распределением управляющих битов. Легко оценить максимальную длину переключающего вектора E. Для этого нужно учесть следующие особенности, характерные ПУОП, основанным на перераспределении управляющих битов:

- исходная УППС $\mathbf{F}_{n/m}$ должна иметь симметричную топологию;
- два бита, подаваемые на вход одного и того же элементарного переключателя $\mathbf{P}_{2/1}$, входящего в состав некоторого блока переключения (перераспределения)

управляющих битов $\mathbf{P}^{(E)}_{m/g}$ (в случае всех независимых битов в управляющем векторе) или $\mathbf{P}^{(E)}_{n/g}$ (в случае экономичной реализации ПУОП при использовании перестановки битов управляющего подблока данных), где g есть длина вектора E, должны поступать на вход одной или нескольких пар симметрично расположенных управляемых элементов $\mathbf{F}_{2/1}$ в блоке $\mathbf{F}_{n/m}$.

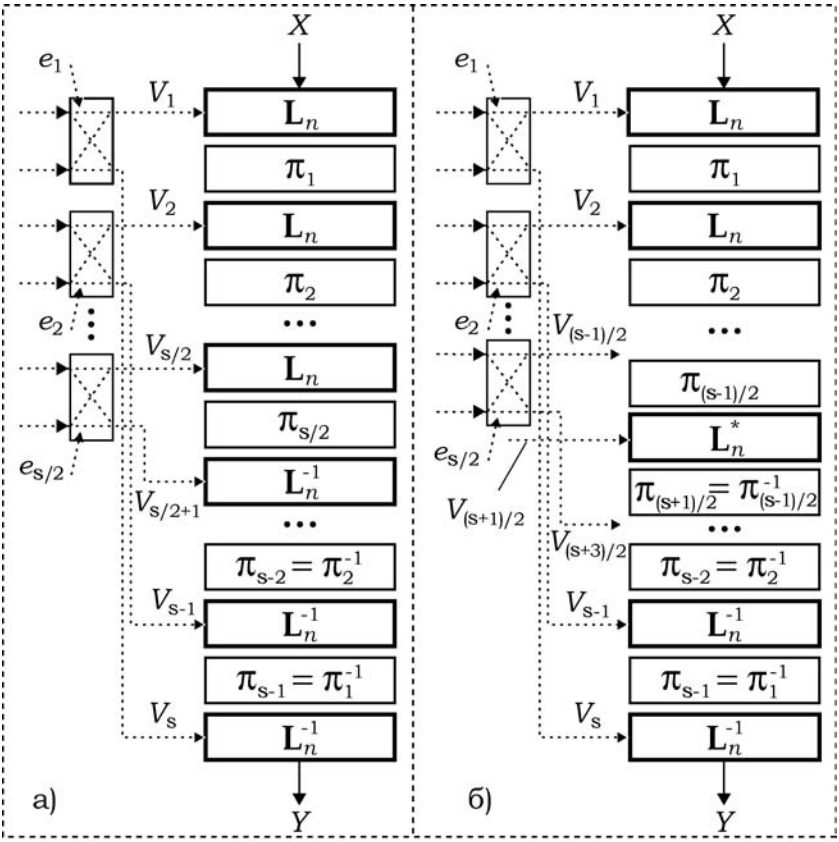

Рис. 7.27. Построение ПУОП на основе УППС с зеркально симметричной топологией: а) с четным числом активных каскадов, б) с нечетным числом активных каскадов ($\mathbf{L}^*$ – инволюция)

Если эти условия выполнены, то инвертирование всех разрядов произвольного вектора E приведет к формированию новой управляемой операции, которая является обратной по отношению к исходной, т. е. будет иметь место соотношение $\mathbf{F}^{(V,E)}_{n/m} = (\mathbf{F}^{(V,E')}_{n/m})^{-1}$. Это означает, что число пар взаимно-обратных модификаций управляемой операции $\mathbf{F}^{(V,E)}_{n/m}$ равно 2^{g-1}. Наибольшее значение g соответствует случаю

$g = m/2$ (реализация ПУОП с m независимыми управляющими битами) и $g = n/2$ (экономичная реализация ПУОП). Для блоков типа $\mathbf{F}^{(V,E)}_{32/96}$ мы можем иметь до 2^{47} или 2^{15} различных вариантов распределения управляющих битов в зависимости от типа реализации ПУОП (общий или экономичный).

Даже при экономичном типе реализации ПУОП обеспечивается достаточно большой диапазон переключаемости. На практике достаточно использовать векторы E длиной до 6 битов, что обеспечит получение 32 различных пар взаимно-обратных управляемых операций, реализуемых с помощью одного ПУОП. Это означает, что каждый бит вектора E будет управлять многими элементарными переключателями в блоках переключения $\mathbf{P}^{(E)}_{m/g}$ или $\mathbf{P}^{(E)}_{n/g}$.

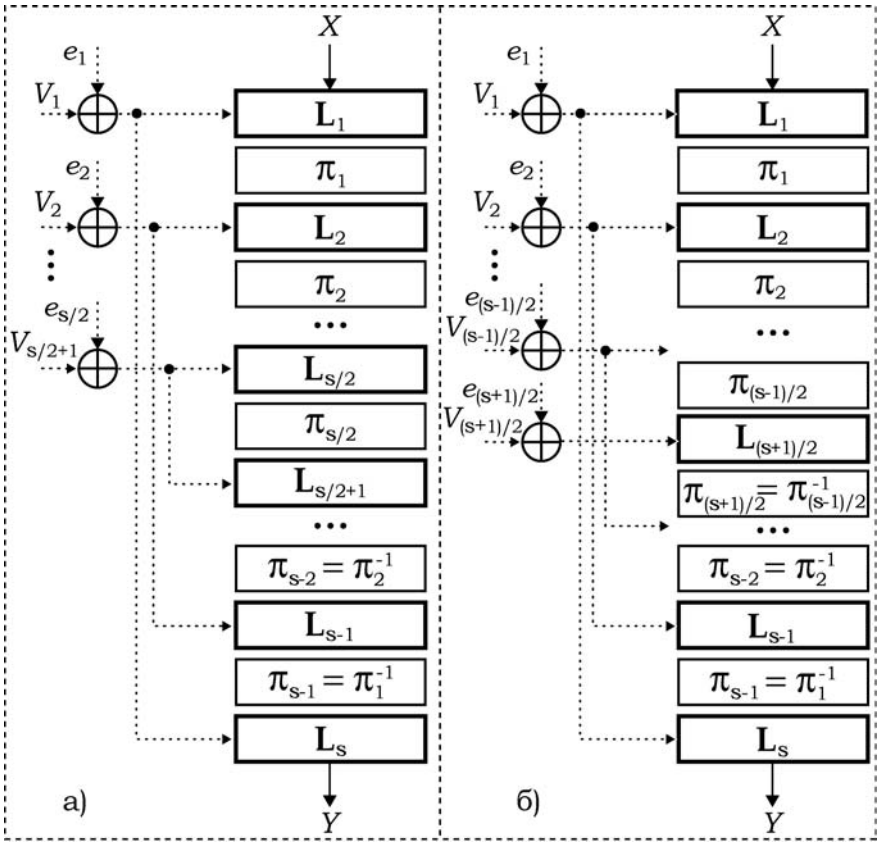

Рис. 7.28. Построение ПУОП с расширенным диапазоном переключаемости на основе УППС с зеркально симметричной топологией:
а) с четным числом активных каскадов,
б) с нечетным числом активных каскадов

При практическом построении ПУОП удобно структурировать управляемые элементы $\mathbf{F}_{2/1}$ таким образом, чтобы они формировали в исходной УППС $\mathbf{F}_{n/m}$ пары активных каскадов или пары некоторых внутренних УППС, которые переключаются одним из битов вектора E. Это упрощает проектирование и анализ ПУОП с расширенным диапазоном переключаемости. Важно то обстоятельство, что свойства исходной УППС (нелинейность, лавинный эффект и др.) сохраняются в определенном смысле неизменными для каждого варианта управляемой операционной подстановки из диапазона переключаемости. Поэтому обоснование УППС является одновременно и обоснованием ПУОП как криптографического примитива. На рис. 7.27 представлена схема построения ПУОП, обращаемой за счет перестановки управляющих векторов, соответствующих симметрично расположенным активным слоям в случае их четного (а) и нечетного (б) числа. Различные комбинации начальных значений битов $e_1, e_1, \ldots, e_{s/2}$ соответствуют различным прямым УОП. Одновременная инверсия всех переключающих битов преобразует каждый из них в соответствующую ему обратную УОП. Аналогичные схемы построения УОП с расширенным диапазоном переключаемости могут быть использованы и в случае УОП на основе управляемых элементов с взаимно-обратными модификациями (рис. 7.28).

Выводы

Одной из важных задач, связанных с проектированием шифров с ПРИК, является предупреждение появления слабых ключей. В данном разделе было показано, что использование операций, зависящих от преобразуемых данных, является предпосылкой к разработке дешевых аппаратно-ориентированных шифров, и был предложен для решения проблемы слабых ключей подход, основанный на применении ПУО. Также были представлены различные варианты методов построения переключаемых операций. Являясь частным случаем свойства управляемости, свойство переключаемости может быть легко и естественно встроено в различные типы управляемых операций.

Другая проблема связана с так называемыми «слайд» – атаками [59]. Данный тип атак основан на периодичности повторов использования подключей, ведущей к повторению одних и тех же раундовых преобразований. Потенциальная возможность проведения таких атак должна быть учтена в шифрах рассматриваемого типа. Использование секретного ключа большого размера (128…256 бит) позволяет легко построить расписание использования ключа, свободное от какой-либо периодичности, и тем самым устранить предпосылку проведения «слайд» – атак. Однако при использовании ключей небольшого размера (такая ситуация часто встречается в практических приложениях) простое расписание ключа фактически представляет собой использование одинаковых подключей в каждом раунде. Это наиболее благо-

приятный случай для успешного осуществления «слайд»–атак. Таким образом, очевидна необходимость обеспечения возможности устранения повторов раундовых преобразований путем использования некоторых других механизмов, не связанных с длиной ключа.

Свойство переключаемости позволяет предложить следующие три варианта таких механизмов:

1. Использование смешанного итеративного преобразования, включающего комбинацию из зашифровывающих ($e = 0$) и расшифровывающих ($e = 1$) раундов. Такое смешанное преобразование может быть представлено в виде некоторой последовательности значений e, например, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0 (10 раундов зашифрования) и 1, 1, 0, 1, 0, 0, 1, 1, 0, 1 (10 раундов расшифрования). Это простейший метод для внесения непериодичности, но следует избегать случаев, когда два последовательных раунда являются взаимнообратными (в некоторых криптосхемах такой случай возможен).

2. Разработка специального варианта переключаемых операций, которые могут быть названы расширенными переключаемыми операциями. Эти операции представляют собой некоторые зависимые от параметра E операции $\mathbf{F}^{(E,V)}$, где E – некоторый k-битовый вектор ($k \geq 2$), зависящий от e и от номера раунда шифрования, причем для всех V модификации $\mathbf{F}^{(E,V)}$ и $\mathbf{F}^{(E',V)}$ являются взаимнообратными при $E \oplus E' = \{1\}^k$. Таким образом, $\mathbf{F}^{(E,V)}$ содержит 2^{k-1} пар взаимнообратных модификаций управляемой операции $\mathbf{F}^{(V)}$, что позволяет легко построить 2^k уникальных раундовых преобразований для каждого раундового подключа. Использование большого числа разнотипных раундовых преобразований позволяет легко устранить периодичность при использовании одного и того же раундового ключа во всех раундах шифрования, т.е. в случае наиболее простого расписания использования ключа. Например, улучшенный переключаемый БУП $\mathbf{P}^{(E,V)}{}_{32/96}$, где $E = (e_1, e_2, e_3)$, может быть легко построен с помощью независимых переключающих битов e_1, e_2 и e_3 для каждого блока $\mathbf{P}^{(e)}{}_{2\times16/1}$ (см. рис. 7.15). Аналогично блок $\mathbf{P}^{(V,e)}{}_{64/192}$ (см. рис. 7.16) может быть преобразован в $\mathbf{P}^{(V,E)}{}_{64/192}$, где $E = (e_1, e_2, e_3)$.

3. Разработка раундового преобразования, содержащего несколько (например, k) ПУО, управляемых независимыми переключающими битами $e_1, e_2, \ldots, e_k$, каждому из которых присваивается значение, зависящее от номера раунда.

Таким образом:

1. Использование ПУО позволяет избежать появления слабых ключей в шифрах с наиболее простым расписанием ключа.

2. Использование расширенных ПУО позволяет предотвратить повторы одинаковых раундовых преобразований при использовании одного и того же раундового ключа во всех раундах шифрования.
3. Применение ПУО позволяет устранить в шифрах с простым расписанием использования ключа слабости к «слайд»-атакам, основанным на возможности выбора одинаковых раундовых ключей для всех раундов шифрования.
4. Использование управляемых операций и ПУО обеспечивает построение скоростных и недорогих при аппаратной реализации шифров.
5. ПУО совместно с управляемыми операциями позволяют эффективно применять новые типы криптосхем, которые комбинируют сеть Фейстеля с подстановочно-перестановочными сетями.
6. Механизмы задания свойства переключаемости УППС являются достаточно многообразными.
7. Существуют эффективные конструктивные криптосхемы для построения ПУО порядков 1, 2, 4,…, n/4, n, где n – разрядность входа ПУО.
8. С точки зрения разнообразия решений по построению переключаемых УППС, использование УЭ типа $\mathbf{F}_{3/1}$ предоставляет наибольшие возможности по сравнению с $\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$. Переход к ПУО на основе элементов $\mathbf{F}_{3/2}$ и $\mathbf{F}_{4/1}$ предоставит еще более широкие возможности конструирования эффективных ПУО, однако полная классификация УЭ последнего типа представляет собой существенно более сложную задачу по сравнению с классификацией УЭ $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$.
9. Переключаемые УППС по своим свойствам идентичны УППС обычного типа, поскольку обладают сходными топологиями. Оценки вероятностных свойств, нелинейности и других характеристик, полученные для последних, могут быть распространены и на первые.
10. При правильном выборе криптосхемы ПУО и УО при одинаковом числе активных слоев вносят одинаковую задержку в процедуру шифрования.
11. Применение ПУО позволяет применять криптосхемы, свободные от инвертирования расписания использования ключа при смене режима шифрования, и тем самым предоставляет возможность дальнейшего уменьшения сложности схемотехнической реализации блочных шифров.

Глава 8

Скоростные шифры с простым расписанием ключа

8.1. Криптосхемы и шифры на основе управляемых и переключаемых операций

8.1.1. Варианты реализации шифров на основе переменных перестановок

Одной из особенностей разработки блочных шифров на основе управляемых операций является то, что при выполнении одной операции используются все биты преобразуемого подблока данных, причем характер их использования зависит от того, к какому подблоку данных они относятся, а именно — к управляющему или преобразуемому. С этим моментом связана специфика проектирования блочных шифров при использовании такого типа примитивов. В настоящее время хорошо апробировано применение перестановок, зависящих от преобразуемых данных, в качестве основного примитива в ряде блочных шифров. Блочная криптосистема SPECTR-H64 [14, 72] является примером достаточно эффективного использования перестановочных сетей в качестве криптографического примитива. Структура раундового преобразования может быть рассмотрена как усовершенствованная криптосхема Фейстеля [14], где вводится преобразование правого подблока данных с помощью БУП типа $\mathbf{P}_{32/80}$, выполняемого одновременно с вычислением раундовой функции по левому подблоку. В этом случае мы имеем преобразование:

$$L \leftarrow L;$$

$$R \leftarrow \mathrm{F}(L) \oplus \mathbf{P}^{(V)}{}_{32/80}(R),$$

где "$\leftarrow$" обозначает операцию присваивания, R и L – левый и правый подблоки данных, F – раундовая функция, V – управляющий вектор, формируемый в зависимости от левого подблока.

Достоинством модернизированной криптосхемы является усиление параллелизма шифрующих преобразований. Для сохранения возможности расшифрования блока шифртекста с использованием того же алгоритма, который выполняет зашифрование (свойство универсальности), после наложения значения раундовой функции на преобразованный правый подблок выполняется обратная управляемая перестановка $\mathbf{P}^{-1}{}_{32/80}$. Обе упомянутые операции $\mathbf{P}_{32/80}$, $\mathbf{P}^{-1}{}_{32/80}$ выполняются в зависимости от

левого подблока данных, что определяет его участие при выполнении каждой операции, выполняемой в раунде шифрования. Несмотря на активное использование левого подблока данных при выполнении раундового преобразования сам он не подвергается преобразованию. С этим связана проблема формирования различных значений управляющих векторов для операций $\mathbf{P}_{32/80}$ и $\mathbf{P}^{-1}{}_{32/80}$. Эта проблема состоит в том, что в случае равенства управляющих векторов выполнение переменных двух операций $\mathbf{P}_{32/80}$ и $\mathbf{P}^{-1}{}_{32/80}$ сводится к выполнению одной операции $\mathbf{P}^{-1}{}_{32/80}$, осуществляемой над вычисленным значением раундовой функции. В SPECTR-H64 данная проблема решается тем, что при формировании управляющих векторов для $\mathbf{P}_{32/80}$ и $\mathbf{P}^{-1}{}_{32/80}$ дополнительно к L используются различные подключи. Полный раунд преобразования имеет следующий вид:

$$L \leftarrow L;$$

$$R \leftarrow \mathrm{F}(L) \oplus \mathbf{P}^{(V)}{}_{32/80}(R);$$

$$R \leftarrow (\mathbf{P}^{-1})^{(V')}{}_{32/80}(R).$$

Рассматриваемая криптосхема представляет общий интерес при проектировании шифров на основе переменных операций, в связи с чем необходимо найти, если имеются, недостатки и рассмотреть возможность их устранения. Определенным недостатком криптосистемы SPECTR-H64 является использование сравнительно большого числа подключей — шести 32-битовых подключей. Это накладывает требование использования процедур расширения ключа или использования секретных ключей сравнительно большого размера — в частности в SPECTR-H64 используется 256-битовый секретный ключ. Таким образом, при построении шифров на основе взаимно обратных управляемых перестановок разработчик сталкивается со специфической проблемой реализации эффективного механизма формирования различных управляющих векторов, соответствующих взаимно обратным БУП. При этом следует сохранить свойства универсальности и высокого параллелизма преобразований.

Для решения данной проблемы был предложен способ, который не требует использования дополнительных подключей и заключается в применении фиксированных перестановок, выполняемых над левым и/или правым подблоком данных. При этом сохранение свойства универсальности алгоритма шифрования достигается тем, что для такого преобразования используются две одинаковые раундовые функции и дополнительно следующие элементы:

- перестановочные инволюции (над левым и/или правым подблоком),
- переключаемые фиксированные перестановки (над левым и/или правым подблоком),
- преобразование левого подблока путем наложения на него подключа.

Рассмотрим некоторые варианты реализации криптосхем в соответствии с данным способом решения проблемы формирования управляющих векторов для выполнения взаимно обратных переменных операций. Применение переключаемой операции, выполняемой над правым подблоком, показано на рис. 8.1, где **G** представляет

собой некоторую операцию, построенную по типу аналогичной операции в шифре SPECTR-H64, а $\mathbf{Crypt}^{(e)}$ обозначает процедуру раундового преобразования в целом.

Данный механизм основан на том факте, что преобразование в правой ветви представляет собой суперпозицию $\mathbf{P}^{(V)}_{n/m} \bullet \mathbf{\Pi}^{(e)} \bullet (\mathbf{P}^{-1}_{n/m})^{(V)}$, которая фактически представляет собой операционный блок, управляемый вектором V и выполняющий переменные перестановки, обладающие цикловой структурой перестановки $\mathbf{\Pi}^{(e)}$.

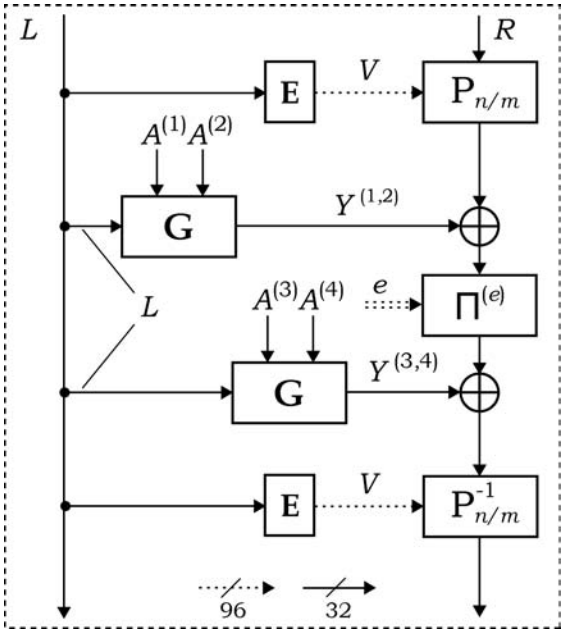

Рис. 8.1. Общий вид процедуры $\mathbf{Crypt}^{(e)}$ с переключаемой операцией

Очевидно, что в данной суперпозиции для реализации только перестановок, имеющих цикловую структуру перестановки $\mathbf{\Pi}^{(e)}$, естественным требованием является условие равенства управляющих векторов, соответствующих прямому $\mathbf{P}_{n/m}$ и обратному $\mathbf{P}^{-1}_{n/m}$ блокам. В общем случае для разных значений левого подблока (разных значений управляющего подблока данных L) реализуются различные модификации перестановок с указанной цикловой структурой. Достоинством такого варианта построения криптосхемы является возможность использования блока расширения, реализуемого в виде простого разветвления проводников, что экономит аппаратные ресурсы. Также не требуется использовать дополнительные подключи для формирования различных управляющих векторов. Возможность осуществления зашифрования и расшифрования с помощью одного и того же алгоритма обеспечивается тем, что обе операции **G** одинаковы, а для обращаемой фиксированной перестановки выполняется условие $\Pi^{(e=1)}=(\Pi^{(e=0)})^{-1}$. Как при зашифровании, так и при расшифровании указанная суперпозиция реализует перестановки с заданной цикловой структурой, например, одноцикловые перестановки.

В частном случае одноцикловой перестановки $\Pi^{(e)}$ механизм оптимизации заключается в том, что бит из j-го разряда на входе блока $\mathbf{P}_{n/m}$ попадает с примерно одинаковой вероятностью во все выходные разряды блока $\mathbf{P}^{-1}{}_{n/m}$, кроме j-го, в который он не попадает ни при каком значении управляющего вектора. Аналогичное свойство обеспечивает любая перестановка, не содержащая циклов длины 1. Для обеспечения примерно равномерного влияния каждого входного бита операционного блока $\mathbf{P}_{32/96}$ на значение всех выходных битов блока $\mathbf{P}^{-1}{}_{32/96}$, можно использовать обращаемую перестановку, содержащую только один цикл длины 1.

Обращаемая перестановка, например, может быть реализована с помощью однослойного БУП, содержащего 32 элементарных переключателя, на управляющий вход которых подается один и тот же бит e. Структура перестановки $\Pi^{(e)}$, реализующей операцию циклического сдвига вправо ($e = 1$) или влево ($e = 0$), показана на рис. 8.2. Время задержки, соответствующее выполнению этой операции, определяется прохождением сигнала через один активный слой и примерно равно времени задержки операции поразрядного суммирования по модулю два ($t_{\oplus}$). Время задержки, соответствующее выполнению операций переменных перестановок, если используются БУП $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$, равно $6t_{\oplus}$. Тогда время выполнения одного раунда составит $15t_{\oplus}$.

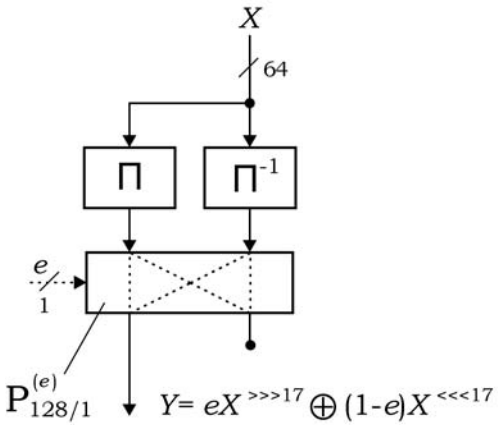

Рис. 8.2. Структура обращаемой одноцикловой перестановки

Данный вариант решения проблемы формирования управляющих векторов может быть применен не только для управляемых перестановочных операций, но и для пар взаимно обратных переменных операций других типов, построенных на основе управляемых УППС (т. е. для случая использования блоков $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$).

8.1.2. Особенности применения двух однотипных операций G

Достоинством управляемых перестановок является то, что влияние одного входного бита на все выходные обеспечивается за минимальное время задержки. Однако данное преобразование сохраняет значение веса Хэмминга. В связи с этим при построении шифров представляется разумным дополнительно с перестановочными

операциями использовать преобразования другого типа, которые изменяют вес и четность битов преобразуемых двоичных векторов. В качестве такой операции представляется целесообразным использовать операцию подобную управляемой двухместной операции **G,** апробированной в качестве криптографического примитива в шифре SPECTR-H64 [14] и хорошо себя зарекомендовавшую. Данная операция может быть применена и при использовании пар взаимно обратных операций других типов (построенных на основе УППС), для преобразования правого подблока.

С рассмотренной выше схемой раундового преобразования при использовании двух одинаковых **G**-операций связана следующая особенность. В отличие от шифра SPECTR-H64 в раундовом преобразовании, показанном на рис. 8.1 используется не одна, а две одинаковые операции **G**. Каждая из них осуществляет «смешивание» левого подблока данных с разными парами подключей. Из приводимого ниже описания структуры этой операции легко видеть, что изменение i-го бита L приводит к детерминированному изменению i-го бита и вероятностному изменению нескольких следующих за ним битов на выходе операции **G**. При непосредственном сложении выходных значений двух операций **G** лавинный эффект ослабляется. Использование операции перестановки (в частном случае – циклического сдвига на число битов, превышающее значение d), выполняемой над выходным значением одной из операций **G**, позволяет устранить подобное гашение лавинного эффекта. С этой целью можно использовать одноцикловую перестановку $\Pi^{(e)}$, соответствующую циклическому сдвигу, например, на 17 битов (вправо при e=0 и влево при e=1). В криптосхеме на рис. 8.1 гашение лавинного эффекта устраняется за счет выполнения фиксированной перестановочной операции над правым подблоком данных после выполнения одной операции **G** и до выполнения второй. Таким образом, используемая фиксированная перестановка в правой ветви криптосхемы играет двоякую роль. Она согласует требуемым образом две одинаковые операции **G** и обеспечивают оптимизацию механизма формирования управляющих векторов для БУП.

Тем не менее, с механизмом, представленным на рис. 8.1 и включающим две одинаковые операции **G**, связана еще одна важная особенность, заключающаяся в том, что и в первой и во второй **G**-операциях каждый бит левого подблока данных влияет на выходные биты, принадлежащие одним и тем же разрядам. Эти разряды разносятся примененной фиксированной битовой перестановкой в правой ветви, однако вероятности порождения активных битов разности в идентичных разрядах обеих операций **G** являются зависимыми, поскольку ключи являются фиксированными элементами. Эта особенность приводит к некоторому повышению вероятности прохождения по схеме шифрования разностей с малым весом. Для устранения этого может быть использовано выполнение операции фиксированной перестановочной инволюции **I** над левым подблоком данных. Использование такой операции также реализует и вторую цель – совершенствование механизма формирования управляющих векторов.

Перестановочная инволюция **I** выбирается с учетом структуры операции **G**. Пусть данная операция имеет такое устройство, что i-й входной бит x_i влияет на че-

тыре выходных бита y_i, y_{i+1}, y_{i+2} и y_{i+3}. Это означает, что на i-й выходной бит влияют входные биты с номерами i–1, i–2, i–3 (для операций **G** с начальными условиями исключение составляют значения i = 1, 2 и 3). Для такой операции **G** естественно выбрать такую перестановочную инволюцию, которая для всех i переставит биты левого подблока l_{i-1}, l_{i-2}, l_{i-3} на расстояние не менее чем четыре шага от переставленного бита l_i. Этот критерий обеспечит зависимость пары выходных битов верхней и нижней операций **G**, принадлежащих одноименному заданному разряду, например j-му, от семи различных битов левого подблока для максимального числа различных значений j (для случая операций **G** без начальных условий это выполняется для всех значений j). Для случая 32-битовых подблоков данных такому условию удовлетворяет, например, следующая перестановка:

$$\mathbf{I} = (1,17)(2,21)(3,25)(4,29)(5,18)(6,22)(7,26)(8,30)(9,19)$$

$$(10,23)(11,27)(12,31)(13,20)(14,24)(15,28)(16,32).$$

Если в левой ветви криптосхемы использовать вместо инволюции **I** перестановку общего вида, то это упростит разработку нужной перестановки, но для обеспечения возможности использования одного и того же алгоритма для выполнения зашифрования и расшифрования потребуется применить операционный блок, реализующий обращаемую фиксированную перестановку, аналогичную перестановке $\Pi^{(e)}$. В свою очередь в правой ветви вместо обращаемой перестановки $\Pi^{(e)}$ можно использовать соответствующую перестановочную инволюцию **I′**, отличную от **I**. Отметим, что использование фиксированных перестановок в левой и правой ветвях криптосхемы требует их согласования.

8.1.3. Другие механизмы согласования управляющих векторов

Рассмотрим другие варианты построения раундовых преобразований, предполагая, что на их основе строится итеративный шифр с приведенной на рис. 8.3 структурой. Чтобы избежать использования дополнительных активных элементов при схемной реализации в качестве промежуточной фиксированной перестановки можно использовать перестановочную инволюцию, содержащую только циклы длины 2. В этом случае, также как и в случае использования одноцикловой перестановки, для всех значений j в одном раунде не обеспечивается влияние j-го входного бита на j-й выходной. Эта неравномерность в следующем раунде выравнивается, что позволяет отказаться от наложения различных раундовых подключей при формировании управляющих векторов, соответствующих прямому и обратному БУП. Это упрощает аппаратную реализацию криптосистем со структурой раунда, подобной раунду шифрования в криптосистеме SPECTR-H64. Таким образом, добавление фиксированной перестановки между блоками $\mathbf{P}_{n/m}$ и $\mathbf{P}^{-1}_{n/m}$ является эффективным способом оптимизации механизма управления взаимно обратными перестановками, зависящими от преобразуемых данных.

В раундовом преобразовании, показанном на рис. 8.4, используется преобразование левого подблока, за счет чего при использовании одинаковых блоков расширения формируются различные управляющие вектора для управления взаимно обратными блоками перестановок.

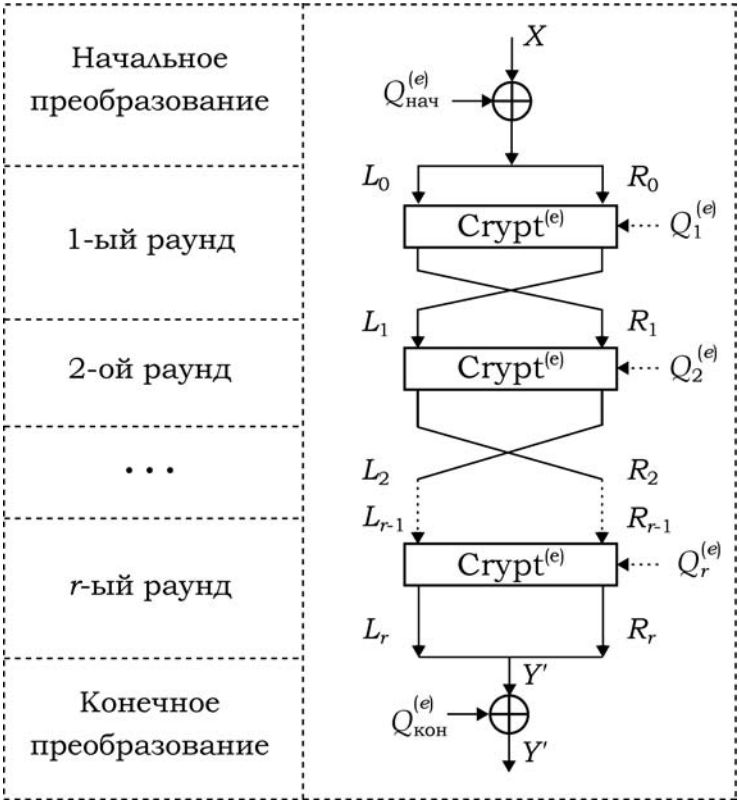

Рис. 8.3. Обобщенная схема итеративного шифра

Статистические исследования влияния входных битов блока $\mathbf{F}_{n/m}$ на выходные биты блока $\mathbf{F}^{-1}_{n/m}$ показало, что даже для несложного механизма, показанного на рис. 8.4а, может быть найдена достаточно простая перестановочная инволюция, дающая равномерное влияние.

Аналогичные исследования схемы, представленной на рис. 8.4б, показали, что одновременное использование перестановочных инволюций, выполняемых над правым и левым подблоком, также могут быть использованы, т. е. эффекты, вносимые этими двумя механизмами, не поглощают друг друга.

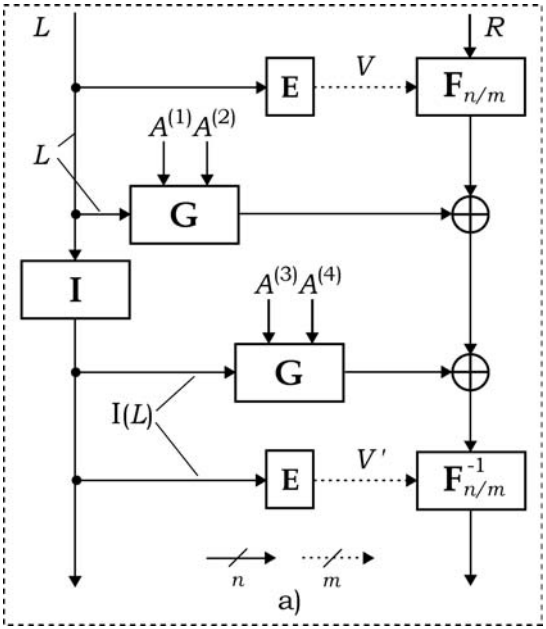

Рис. 8.4а. Раунд шифрования с использованием преобразования левого подблока данных с помощью фиксированных перестановочных инволюций

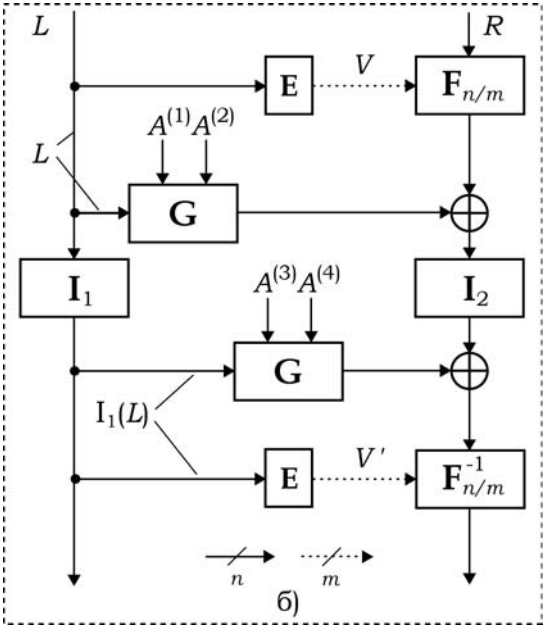

Рис. 8.4б. Раунд шифрования с использованием преобразования левого и правого подблоков данных с помощью фиксированных перестановочных инволюций

Аналогичный механизм показан на рис. 8.5, в котором в левой ветви вместо фиксированной перестановки применена переключаемая, что позволяет использовать перестановки общего типа, причем эта переключаемая операция не вводит каких-либо временных задержек, поскольку после выполнения верхней операции **G** выполняется еще операция XOR, благодаря чему выходное значение нижней операции **G** успевает сформироваться одновременно с формированием выходного значения перестановки **I**.

С целью уменьшения ключевого материала, используемого в одном раунде, можно применить схему, представленную на рис. 8.6, в которой показано применение двух управляемых операций $\mathbf{S}_{32/32}$, формирующих раундовую функцию. В этой схеме вместо блоков $\mathbf{S}_{32/32}$ могут быть применены блоки $\mathbf{S}_{32/96}$, которые при реализации в программируемых БИС будут согласованы по времени задержки с блоком $\mathbf{R}_{32/96}$, осуществляющим преобразование правого подблока данных. Подобная структура раундового преобразования допускает достаточно простое выполнение оценок дифференциальных характеристик с разностями малого веса, которые наиболее эффективны при проведении дифференциальной атаки. Этот вариант демонстрирует возможность использования минимального ключевого материала в одном раунде шифрования.

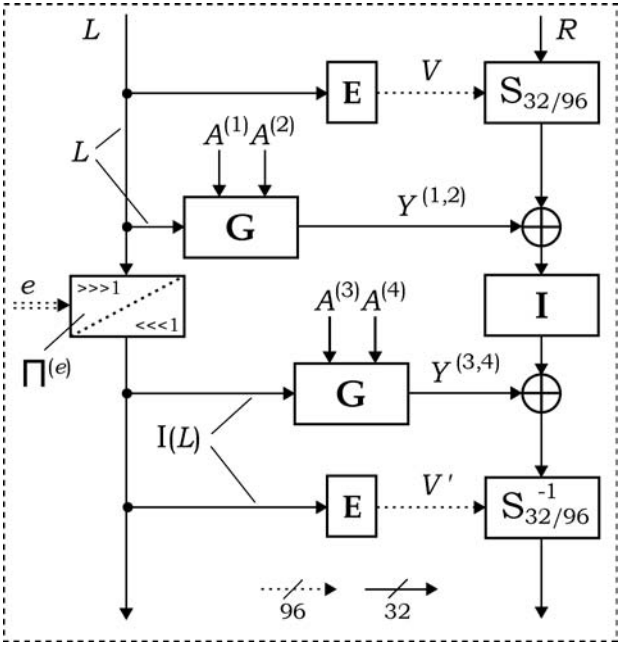

Рис. 8.5. Структура процедуры $\mathbf{Crypt}^{(e)}$ с переключаемой перестановкой в левой ветви криптосхемы

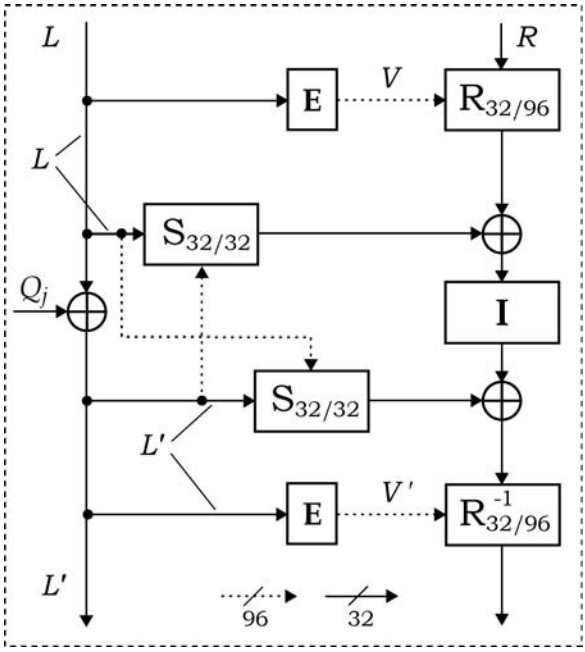

Рис. 8.6. Структура процедуры **Crypt**$^{(e)}$ с преобразованием левой ветви криптосхемы путем наложения на нее подключа

В рассмотренных выше примерах для осуществления внутренней оптимизации распределения управляющих битов по элементарным управляемым элементам управляемых операционных блоков $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}{}_{n/m}$ было использовано осуществление промежуточного обратимого преобразования над левым подблоком, которое вносило либо нулевую задержку, либо задержку примерно равную времени выполнения операции XOR. В принципе могли бы быть применены и другие более сложные преобразования левого подблока, однако они приводят к увеличению критического пути комбинационной схемы, реализующей раундовое преобразование, что не только потребляет дополнительные аппаратные ресурсы при изготовлении шифратора, но и снижает скорость. Очевидно, что более сложные преобразования, выполняемые над левым подблоком, не только приводят к наиболее эффективному устранению проблемы формирования управляющих векторов, но и к возможности уменьшения количества раундов, что потенциально может дать выигрыш в производительности и снижение стоимости реализации. Возможность уменьшения количества раундов связана с тем, что в одном раунде будут преобразованы сразу два подблока, т. е. весь преобразуемый блок данных. Но для реализации данной идеи требуется разработка других построений с высоким параллелизмом выполняемых преобразований.

В силу самой природы операции, зависящей от преобразуемых данных, она сама по себе задает параллелизм преобразования, поэтому могут быть применены две

схемы построения раундового преобразования, показанные на рис. 8.7а и 8.7б. Первая схема использует управляемые двухместные операции и требует большего объема ключевого материала в одном раунде шифрования. Особенностью второй схемы является согласование прямой и обратной операций $\mathbf{R}_{n/m}$ и $\mathbf{R}^{-1}_{n/m}$, операций, выполняемых параллельно до осуществления первой (верхней) операции XOR. Одна из указанных операций выполняется над правым подблоком, который затем переставляется с левым подблоком. Согласование здесь необходимо, поскольку операции выполняются с использованием одинаковых значений управляющих векторов. Оказываются согласованными также две последовательно выполняемые операции $\mathbf{R}_{n/m}$, между которыми выполняется вторая операция XOR. Особенностью применения двух внутренних операций $\mathbf{R}_{n/m}$ является то, что они выполняются над раундовыми подключами (одна из них – одновременно с преобразованием правого подблока). Проблема согласования снимается, если использовать различные блоки расширения для операций, выполняемых над подключами и над правым подблоком. Также эта проблема устраняется применением разнотипных операций, выполняемых над правым подблоком и подключами. Заметим, что в схеме, представленной на рис. 8.7б, неразумно две операции $\mathbf{R}_{n/m}$, выполняемые над подключами, заменить на операции $\mathbf{R}^{-1}_{n/m}$.

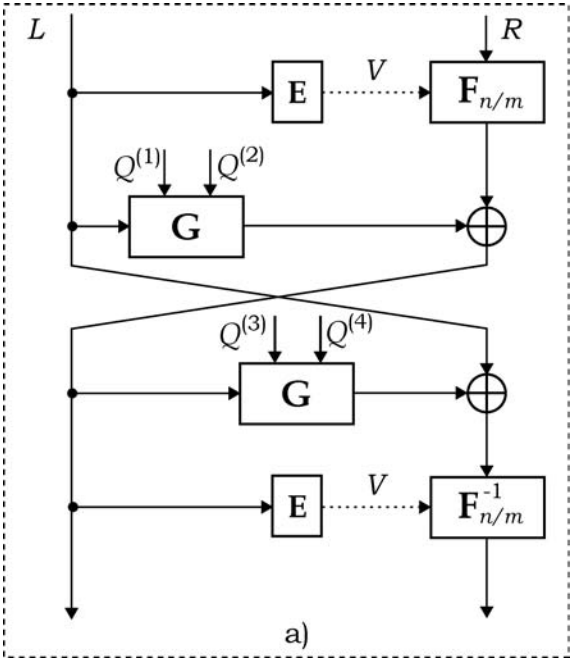

Рис. 8.7а. Механизм раундового шифрования с преобразованием обоих подблоков данных с использованием двух одинаковых операций **G**

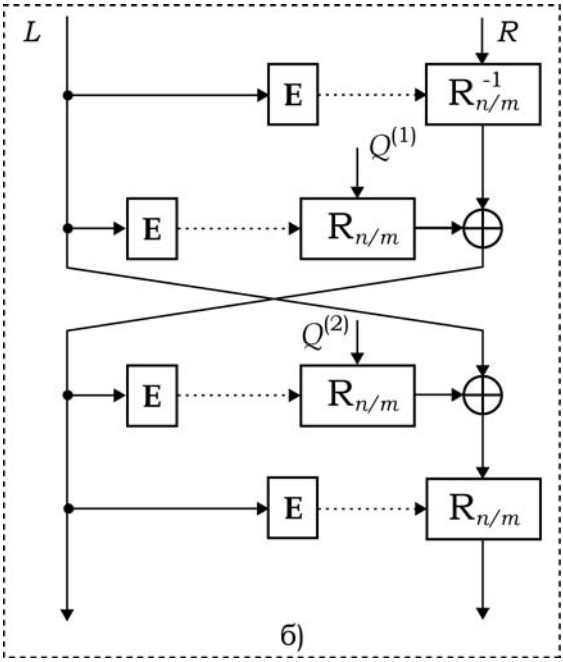

Рис. 8.7б. Механизм раундового шифрования с преобразованием обоих подблоков данных с использованием трех одинаковых операций $\mathbf{R}_{n/m}$

Эти две внешне элегантные схемы хотя и обеспечивают построение стойких шифров при числе раундов от 6 до 12 для различных конкретных вариантов используемых операционных блоков, но их критический путь примерно в полтора раза больше критического пути рассмотренных выше криптосхем, поскольку первая из двух последовательно выполняемых операций не может быть выполнена одновременно с парой параллельно выполняемых операций. Это приводит к времени выполнения раунда, равного примерно $6mt_{\oplus}/n + 2$. В рассмотренных выше криптосхемах уровень параллелизма можно оценить значением 2 (в среднем две параллельно выполняемые операции — три операции выполняются параллельно и затем выполняется четвертая операция), а в последних двух криптосхемах – значением 4/3 (две операции выполняются параллельно и затем выполняется третья, а потом четвертая операция). В следующем параграфе будут предложены более эффективные построения раундового преобразования с преобразованием обоих подблоков данных и значением уровня параллелизма равным 2.

Уместно отметить, что, построив эффективные механизмы формирования управляющих векторов, которые не требуют использования раундовых подключей, мы имеем возможность вернуться к применению последних, если это по каким-либо конструктивным соображениям потребуется. При этом устраняются некоторые статистические неравномерности влияния битов правого подблока, имеющие место при выполнении одного раунда шифрования алгоритма SPECTR-H64 даже при исполь-

зовании различных подключей при формировании управляемых операций $P_{32/80}$ и $P^{-1}{}_{32/80}$. Отметим, что для рассмотренных выше механизмов такие неравномерности сглаживаются и без использования различных подключей при формировании управляющих векторов.

8.1.4. Криптосхемы, сочетающие преобразование обоих подблоков с высоким уровнем параллелизма

Для реализации преобразования всего блока данных в рамках одного раунда с сохранением достаточно высокого параллелизма вычислений была разработана криптосхема, показанная на рис. 8.8.

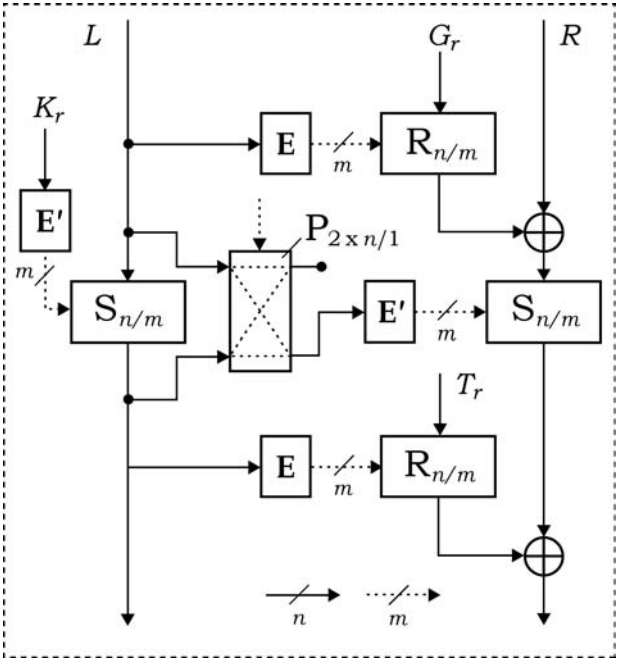

Рис. 8.8. Структура раундового шифрования с преобразованием обоих подблоков данных при сохранении высокого параллелизма вычислений

Данное построение основано на следующих идеях:

- используются две одинаковые операции преобразования, зависящие от левого подблока и располагаемые зеркально симметрично в структуре раунда шифрования; выходные значения данных операций складываются с правым подблоком данных как и в криптосхеме Фейстеля;

- одна из указанных выше операций вычисляется одновременно с преобразованием левого подблока данных, а вторая – с преобразованием правого подблока данных с помощью операции, зависящей от левого; таким образом достигаем уровня параллелизма, равного 2;
- для обеспечения универсальности криптосхемы левый подблок преобразуем с помощью операции, являющейся инволюцией или переключаемой управляемой операции; с той же целью управляющий вектор, используемый при выполнении управляемой операции над правым подблоком данных, формируется по исходному значению левого подблока при зашифровании и по преобразованному значению левого подблока при расшифровании.

В данной криптосхеме имеются две пары операций $\mathbf{S}_{n/m}$ и $\mathbf{R}_{n/m}$. Операции первой пары выполняются над левым подблоком данных ($\mathbf{S}_{n/m}$) и подключом G_r ($\mathbf{R}_{n/m}$), а операции второй пары – над правым подблоком данных ($\mathbf{S}_{n/m}$) и подключом T_r ($\mathbf{R}_{n/m}$). Причем только операция, выполняемая над левым подблоком, является фиксированной после установки ключа, а остальные управляемые операции являются переменными, поскольку зависят от левого подблока данных. В этой криптосхеме предполагается, что блоки расширения построены таким образом, что ни один бит левого подблока данных не влияет дважды ни на какой бит преобразуемого двоичного вектора (подключа или подблока данных). Операции, зависящие от левого подблока данных, задают нелинейное преобразование, а операция, выполняемая над левым подблоком, является линейной, если операция $\mathbf{S}_{n/m}$ построена с использованием управляемых элементов типа $\mathbf{F}_{2/1}$ (при использовании управляемых элементов с размером $\mathbf{F}_{3/1}$ и более данная операция становится также нелинейной). Операция $\mathbf{S}_{n/m}$ обеспечивает достаточно хороший лавинный эффект, а три нелинейные операции обеспечивают достаточно высокую нелинейность раундового преобразования (алгебраическая степень нелинейности равна 7). Далее будут рассмотрены варианты построения нелинейного преобразования левого подблока с использование управляемых элементов с минимальным размером, обеспечивающие алгебраическую степень нелинейности раундового преобразования превосходящую значение 20. Удачным решением представляется также вариант преобразования левого подблока с помощью параллельно выполняемых блоков подстановок размера 4×4.

Следует отметить, что для обеспечения свойства универсальности такой криптосхемы требуется наложить определенные ограничения на используемые управляемые операции для преобразования левого и правого подблоков данных. Возможны два типа таких операций:

- управляемые инволюции;
- переключаемые управляемые операции общего типа.

За счет изменения бита, задающего прямую или соответствующую ей обратную операцию, обеспечивается возможность выполнения зашифрования и расшифрования с помощью одного и того же алгоритма. Экономичные варианты построения управляемых операций, предложенные в разделе 4, делают их использование в рамках рассматриваемой криптосхемы весьма перспективным для разработки скоростных и экономичных в реализации блочных криптосистем.

Построение управляемых инволюций на основе УППС может быть легко осуществлено по аналогии с построением управляемых перестановочных инволюций [14, 15]. Для этого можно воспользоваться следующими двумя вариантами: 1) последовательным (рис. 8.9а) и 2) параллельным (рис. 8.9б). Достоинство первой схемы состоит в том, что задается преобразование входного вектора как единого целого, однако требуется выполнение двух взаимно обратных управляемых операций $\mathbf{F}_{n/m}$ и $\mathbf{F}^{-1}_{n/m}$. Достоинством второго варианта является параллелизм выполняемых операций, уменьшающий вносимое время задержки, однако входной двоичный вектор разбивается и преобразуется как два независимых значения. В первом случае формируется результирующий блок $\mathbf{F}_{n/2m}$, а во втором – блок $\mathbf{F}_{2n/m}$. Несмотря на отличие по принципу задания инволюций, при установленном числе активных слоев в результирующем блоке обе схемы обеспечивают примерно одинаковые конструктивные возможности при синтезе шифров.

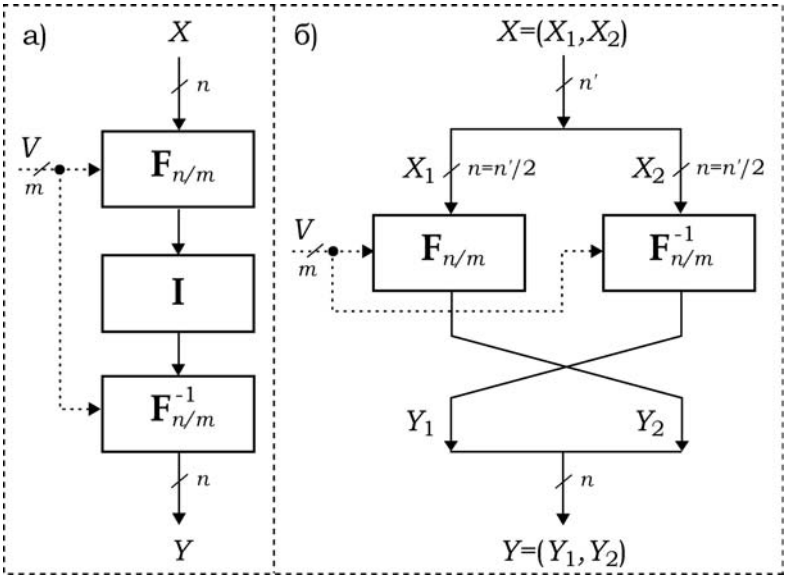

Рис. 8.9. Последовательная (а) и параллельная (б) схема построения блока управляемых инволюций, реализуемых на основе УППС

Легко показать, что данные схемы действительно приводят к построению управляемых инволюций. Действительно, осуществим преобразование выходного вектора Y с помощью операции $\mathbf{F}_{n/2m}$ при сохранении значения управляющего вектора. Данное преобразование для последовательной схемы имеет вид:

$$Y' = \mathbf{F}_{n/2m}(Y) = (\mathbf{F}^{-1}{}_{n/m})^{(V)}(\mathbf{I}(\mathbf{F}^{(V)}{}_{n/m}(Y))) = (Y)\mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} =$$

$$= ((X)\mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)})\mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} =$$

$$= (X)\mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} \bullet \mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} =$$

$$= (X)\mathbf{F}^{(V)}{}_{n/m} \bullet \mathbf{I} \bullet \mathbf{I} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} = (X)\mathbf{F}^{(V)}{}_{n/m} \bullet (\mathbf{F}^{-1}{}_{n/m})^{(V)} = X .$$

Таким образом, первая схема реализует управляемую операцию, являющуюся инволюцией. Для второй схемы имеем:

$$Y' = \mathbf{F}_{2n/m}(Y) = \mathbf{F}_{2n/m}(Y_1, Y_2) = (\mathbf{F}^{-1}{}_{n/m})^{(V)}(Y_2), \mathbf{F}^{(V)}{}_{n/m}(Y_1)) =$$

$$= ((\mathbf{F}^{-1}{}_{n/m})^{(V)}(\mathbf{F}^{(V)}{}_{n/m}(X_1)), \mathbf{F}^{(V)}{}_{n/m}((\mathbf{F}^{-1}{}_{n/m})^{(V)}(X_2))) = (X_1, X_2).$$

То есть и во втором случае мы имеем дело с управляемой инволюцией при произвольном блоке $\mathbf{F}_{n/m}$.

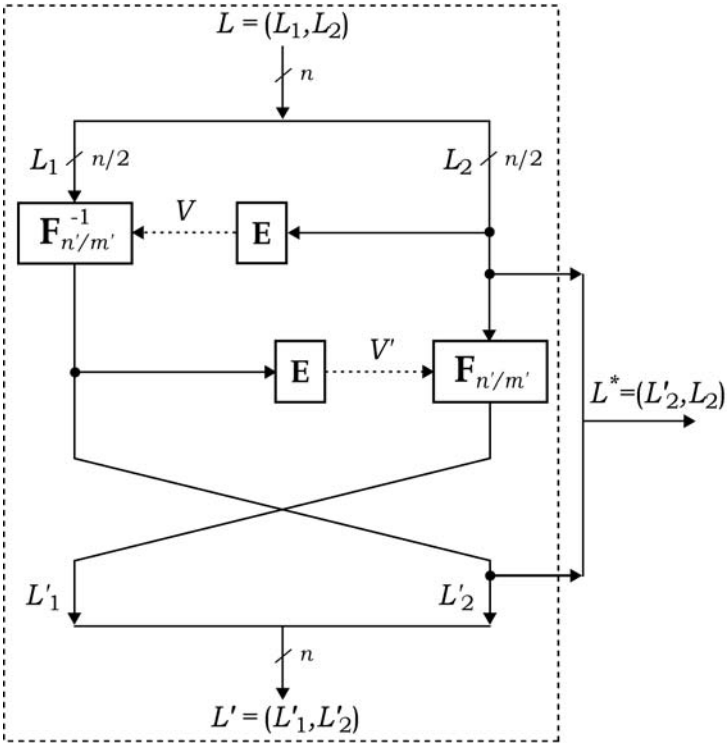

Рис. 8.10. Структура нелинейной переменной операции для преобразования левого подблока (операция $\mathbf{H}_{n/m}$)

Другой вариант криптосхемы с высоким уровнем параллелизма получается путем замены операций, используемых для преобразования раундовых подключей на **G**-операции. Представляет интерес также модифицирование операции, выполняемой над левым подблоком данных, которой может быть придано свойство нелинейности

не применением управляемых элементов большего размера, а путем использования преобразования, показанного на рис. 8.10. Данное преобразование заключается в разбиении левого подблока L на два подблока L_1 и L_2 половинного размера и последовательном преобразовании последних с использованием управляемых операций, которые являются взаимно обратными, после чего выполняется перестановка подблоков L_1 и L_2. В силу того, что при выполнении каждой из управляемых операций управляющий вектор формируется по одному из подблоков данных, каждая из них задает нелинейное преобразование, а, следовательно, преобразование в целом является нелинейным. Кроме того, данное преобразование $\mathbf{H}_{n/m}$ является инволюцией, что можно показать, подвергнув преобразованию выходное значение $L' = (L'_1, L'_2)$:

$$L'' = \mathbf{H}_{n/m}(L'_1, L'_2) = \mathbf{H}_{n/m}(\mathbf{H}_{n/m}(L_1, L_2)) = \mathbf{H}_{n/m}(\mathbf{F}^{(V')}_{n'/m'}(L_2), (\mathbf{F}^{-1})^{(V)}_{n'/m'}(L_1)) =$$

$$= (\mathbf{F}^{(V'')}_{n'/m'}((\mathbf{F}^{-1})^{(V)}_{n'/m'}(L_1)), (\mathbf{F}^{-1})^{(V')}_{n'/m'}(\mathbf{F}^{(V')}_{n'/m'}(L_2))) = (X_1, X_2),$$

поскольку $V'' = V$. Полезно сделать следующее наблюдение. Конкатенация входного и выходного значения операции $\mathbf{F}_{n'/m'}$ при преобразовании блока L есть $L^* = (L'_2, L_2)$, а при преобразовании блока L' эта же конкатенация равна $L^{**} = (L_2, L'_2)$. Таким образом, сконструированный нами нелинейный операционный блок на основе двух переменных операций формирует также двоичный вектор, в котором компоненты L_2 и L'_2 переставляются при повторном преобразовании выходного значения с помощью этого же блока.

Как показано в разделе 7, в экономичном варианте реализации переключаемой операции эти функции выполняются однослойным блоком, выполняющим транспозицию двух половин управляющего вектора. Таким образом, используя сконструированный операционный узел и беря вектор $L^* = (L'_2, L_2)$ в качестве управляющего значения для выполнения переключаемой операции $\mathbf{S}^{(e)}_{n/m}$, выполненной в экономичном варианте, мы можем блок транспозиции управляющего вектора вообще не применять, так как его функции автоматически реализуются рассмотренным операционным узлом по отношению к вектору L^*. Этим достигается еще большая экономия аппаратных ресурсов при использовании переключаемой операции $\mathbf{S}^{(e)}_{n/m}$.

В результате проведенного анализа можно предложить вариант построения раунда шифрования, показанный на рис. 8.11. При разработке конкретного шифра с целью получения максимального быстродействия требуется согласовать время задержки параллельно выполняемых операций. Это связано с ограничением числа активных слоев, которые могут быть использованы в операциях $\mathbf{F}_{n'/m'}$ и $\mathbf{F}^{-1}{}_{n'/m'}$, на основе которых сформирован узел $\mathbf{H}_{n/m}$. Это показывает, что нелинейность достигнута за счет уменьшения числа активных слоев, так как указанные прямая и обратная операции выполняются последовательно над подблоками L_1 и L_2. В зависимости от конкретного типа операции $\mathbf{F}_{n'/m'}$ это может привести уменьшению вклада операции, выполняемой над подблоком L, в лавинный эффект. Следовательно, необходимо учитывать и компромисс между нелинейностью и лавинным эффектом при выборе операции преобразования левого подблока данных.

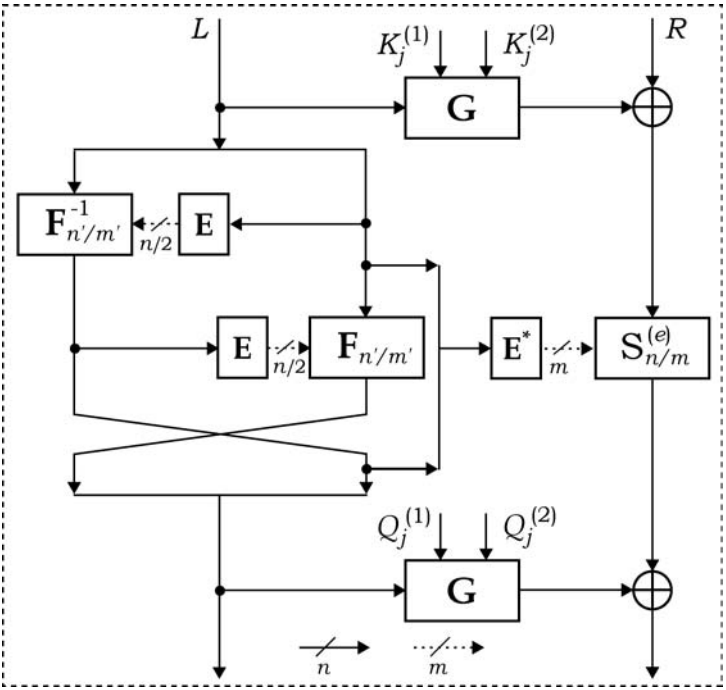

Рис. 8.11. Раунд шифрования с нелинейным преобразованием левого подблока и экономичной переключаемой операции в правой ветви криптосхемы

8.2. Криптосхема COBRA-H64

8.2.1. Общая схема шифрования

Блочный шифр COBRA-H64 спроектирован с учетом результатов линейного и дифференциального криптоанализа шифра SPECTR-H64 [14, 75, 82]. Последний оказался стойким к предложенным вариантам атак. Однако из проведенных исследований стойкости следует, что при совершенствовании блока расширения, формирующего управляющие векторы, используемые при выполнении переменных перестановок, и нелинейной операции **G** число раундов может быть сокращено, что позволит увеличить скорость шифрования при итеративной реализации и уменьшить схемотехническую сложность при конвейерной реализации. Другой поставленной задачей при проектировании системы COBRA-H64 было упрощение расписания ключа и уменьшение его длины. Основными отличиями этого шифра являются следующие:

1. В раундовом преобразовании шифра COBRA-H64 используются два БУП второго порядка $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}_{32/96}$, тогда как в SPECTR-H64 – три БУП первого

порядка: два блока $\mathbf{P}_{32/80}$ и один $\mathbf{P}_{32/80}^{-1}$. Это позволяет в первом шифре более равномерно распределить влияние управляющего подблока на выполнение переменных битовых перестановок над преобразуемым подблоком.

2. В раунде COBRA-H64 используются две одинаковые нелинейные операции **G**, а в SPECTR-H64 – только одна.

3. Благодаря предыдущей особенности в раундовом преобразовании COBRA-H64 оказалось возможным задать над управляющим подблоком выполнение перестановочной инволюции, которая позволила отказаться от использования ключей при формировании управляющих векторов соответствующим взаимно обратным БУП.

4. В COBRA-H64 используется новый криптографический примитив – переключаемая операция, хотя и в наиболее простом варианте. Ее использование позволило устранить наличие слабых и полуслабых ключей.

Общая схема шифрования и расшифрования в шифре COBRA-H64 определяется следующими преобразованиями:

$$Y = \mathbf{T}^{(0)}(X, K) \text{ и } X = \mathbf{T}^{(1)}(Y, K),$$

где $X \in \{0, 1\}^{64}$ – открытый текст (входной блок), $Y \in \{0, 1\}^{64}$ – шифртекст (выходной блок); $K \in \{0, 1\}^{128}$ – секретный ключ; $\mathbf{T}^{(e)}$ – функция преобразования блока данных; $e \in \{0, 1\}$ – параметр, определяющий режимы зашифрования ($e = 0$) и расшифрования ($e = 1$).

Секретный ключ рассматривается как объединение четырех подключей $K = (K_1, K_2, K_3, K_4)$, где $K_i \in \{0, 1\}^{32}$ для всех $i = 1, 2, 3, 4$. Общая схема шифрования представляет собой десятираундовую итеративную структуру с очень простыми начальным и конечным преобразованиями (см. рис. 8.12). При выполнении каждого j-го раунда ($j = 1, 2, \ldots, 10$) применяется раундовый ключ $Q_j^{(e)}$, формируемый на основе непосредственного использования всех четырех подключей K_1, K_2, K_3, K_4 без выполнения каких-либо специальных процедур преобразования (расширения) секретного ключа, т. е. каждый ключ $Q_j^{(e)}$ формируется как простая последовательность секретных подключей K_i, применяемых в порядке, заданном достаточно простым расписанием ключей.

Процедура шифрования начинается с начального преобразования **IT**. Затем выполняются 10 раундов шифрования в соответствии с процедурой $\mathbf{Crypt}^{(e)}$, за которыми следует конечное преобразование **FT**. Формально шифрующие преобразования записываются в виде следующего алгоритма.

1. Входной блок X разбивается на два 32-битовых равных подблока L и R: $X = (L, R)$.

2. Выполняется начальное преобразование **IT** в соответствии с формулами: $L_0 = L \oplus O_3$ и $R_0 = R \oplus O_4$.

3. Последовательно для $j = 1, 2, \ldots, 9$ выполнить процедуру
   $\{ (L_j, R_j) := \mathbf{Crypt}^{(e)}(L_{j-1}, R_{j-1}, Q_j^{(e)}),\ M := R_j, R_j := L_j, L_j := M; \}$

4. Выполняется последний раунд шифрования:
   $(L_{10}, R_{10}) := \mathbf{Crypt}^{(e)}(L_9, R_9, Q_{10}^{(e)})$.

5. Выполняется конечное преобразование **FT** по формулам:
   $L' = L_r \oplus O_1$ и $R' = R_r \oplus O_2$.

Выходной блок шифрованного текста имеет вид $Y = (L', R')$.

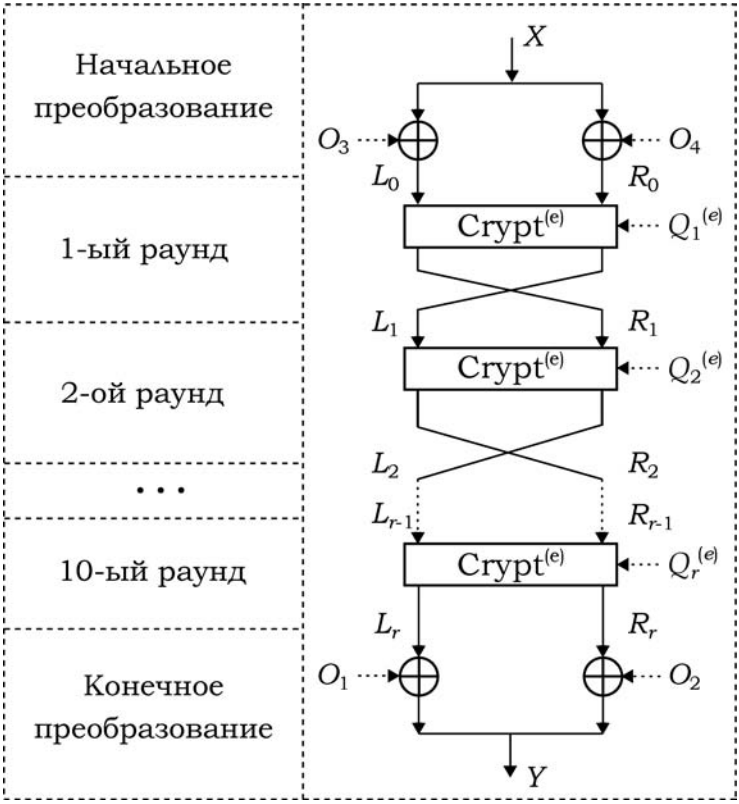

Рис. 8.12. Общая схема шифрования в системах COBRA-H64 ($r = 10$) и COBRA-H128 ($r = 12$)

Схема процедуры $\mathbf{Crypt}^{(e)}$ блочного шифра COBRA-H64 представлена на рисунке 8.13.

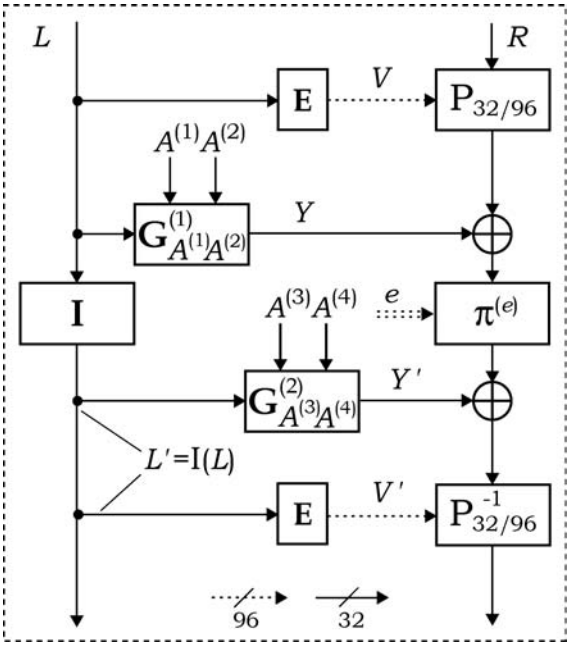

Рис. 8.13. Процедура **Crypt**$^{(e)}$ блочного шифра COBRA-H64

8.2.2. Формирование расписания использования ключа

Каждый раундовый ключ $Q_j^{(e)}$ состоит из четырех зависящих от параметра e раундовых подключей $A^{(i)} \in \{0, 1\}^{32}$, где $i = 1, 2, 3, 4$, и записывается в виде $Q_j^{(e)} = (A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)})_j^{(e)}$, где $j = 1, \ldots, 10$. Спецификация раундовых ключей шифра COBRA-H64 определяется таблицей 8.1 и рис. 8.14.

Таблица 8.1

Расписание ключа в шифре COBRA-H64

j	1	2	3	4	5	6	7	8	9	10
$A_j^{(1)}$	O_1	O_4	O_3	O_2	O_1	O_1	O_2	O_3	O_4	O_1
$A_j^{(2)}$	O_2	O_1	O_4	O_3	O_4	O_4	O_3	O_4	O_1	O_2
$A_j^{(3)}$	O_3	O_2	O_1	O_4	O_3	O_3	O_4	O_1	O_2	O_3
$A_j^{(4)}$	O_4	O_3	O_2	O_1	O_2	O_2	O_1	O_2	O_3	O_4

Изменение режима зашифрования на режим расшифрования проводится простым изменением параметра e, который управляет однослойным БУП $\mathbf{P}_{128/1}^{(e)}$, осуще-

ствляющим соответствующую перестановку подключей K_1, K_2, K_3, K_4. Блок $\mathbf{P}_{128/1}^{(e)}$ представляет собой каскад из двух блоков $\mathbf{P}_{64/1}^{(e)}$, представленных на рис. 8.14. На выходе первого блока $\mathbf{P}_{64/1}^{(e)}$ формируется пара подключей O_1 и O_3, а на выходе второго – пара O_2 и O_4. При $e = 0$ выполняется условие $O_i = K_i$ для всех $i = 1, 2, 3, 4$, а при $e = 1$ имеем $O_1 = K_3$, $O_3 = K_1$, $O_2 = K_4$ и $O_4 = K_2$.

Подключи O_1, O_2, O_3, O_4, зависящие от бита e, используются в каждом раунде в соответствии с таблицей 8.1 вместо формальных подключей $A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}$. Универсальность шифрования (возможность использования одного и того же алгоритма для зашифрования и для расшифрования) обеспечивается тем, что фиксированная перестановка $\boldsymbol{\pi}^{(0)}$ изменяется на обратную при изменении бита e: $\boldsymbol{\pi}^{(1)} = (\boldsymbol{\pi}^{(0)})^{-1}$, а также соответствующим изменением расписания использования подключей K_1, K_2, K_3, K_4. Раунд шифрования не является инволюцией, но при инвертировании бита e и перестановке подключей $A^{(1)}$ с $A^{(3)}$ и $A^{(2)}$ с $A^{(4)}$ раундовое преобразование обращается:

$$\mathrm{Crypt}^{(0)}_{A^{(1)},A^{(2)},A^{(3)},A^{(4)}} = (\mathrm{Crypt}^{(1)}_{A^{(3)},A^{(4)},A^{(1)},A^{(2)}})^{-1}.$$

При зашифровании используются раундовые ключи $Q_j^{(0)} = (A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)})_j^{(0)}$, где $j = 1, \ldots, 10$, а при расшифровании – $Q_j^{(1)} = (A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)})_j^{(1)}$. Для реализации корректного шифрования для всех $j = 1, \ldots, 10$ должны выполняться следующие условия:

$$(A^{(1)})_j^{(1)} = (A^{(3)})_{11-j}^{(0)},\ (A^{(2)})_j^{(1)} = (A^{(4)})_{11-j}^{(0)},\ (A^{(3)})_j^{(1)} = (A^{(1)})_{11-j}^{(0)},\ (A^{(4)})_j^{(1)} = (A^{(2)})_{11-j}^{(0)}.$$

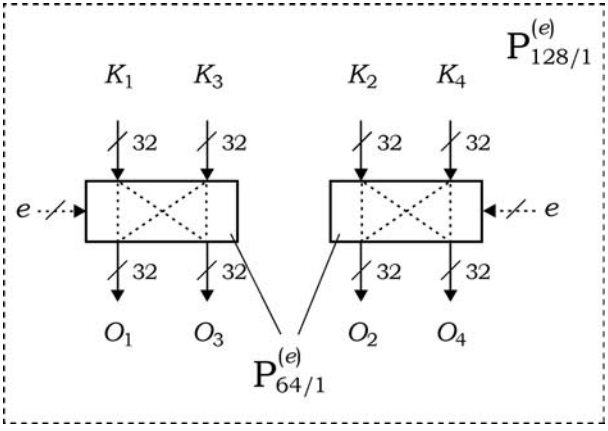

Рис. 8.14. Схема реализации транспозиции подключей при переходе от зашифрования ($e = 0$) к расшифрованию ($e = 1$)

Используя рис. 8.14 и табл. 8.1, легко записать расписание подключей K_1, K_2, K_3, K_4 в явном виде для случая зашифрования (табл. 8.2) и расшифрования (табл. 8.3). Легко видеть, что указанные условия выполняются.

Таблица 8.2

Расписание подключей K_1, K_2, K_3, K_4 при зашифровании ($e = 0$)

j	1	2	3	4	5	6	7	8	9	10
$A_j^{(1)}$	K_1	K_4	K_3	K_2	K_1	K_1	K_2	K_3	K_4	K_1
$A_j^{(2)}$	K_2	K_1	K_4	K_3	K_4	K_4	K_3	K_4	K_1	K_2
$A_j^{(3)}$	K_3	K_2	K_1	K_4	K_3	K_3	K_4	K_1	K_2	K_3
$A_j^{(4)}$	K_4	K_3	K_2	K_1	K_2	K_2	K_1	K_2	K_3	K_4

Таблица 8.3

Расписание подключей K_1, K_2, K_3, K_4 при расшифровании ($e = 1$)

j	1	2	3	4	5	6	7	8	9	10
$A_j^{(1)}$	K_3	K_2	K_1	K_4	K_3	K_3	K_4	K_1	K_2	K_3
$A_j^{(2)}$	K_4	K_3	K_2	K_1	K_2	K_2	K_1	K_2	K_3	K_4
$A_j^{(3)}$	K_1	K_4	K_3	K_2	K_1	K_1	K_2	K_3	K_4	K_1
$A_j^{(4)}$	K_2	K_1	K_4	K_3	K_4	K_4	K_3	K_4	K_1	K_2

8.2.3. Переменные перестановки

Перестановки, зависящие от преобразуемых данных, выполняются с помощью БУП $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}_{32/96}$, описанных ранее (см. описание шифра DDP-64, раздел 4). Текущая перестановка, выполняемая над правым 32-битовым подблоком данных, зависит от 96-битового управляющего вектора (УВ) $V \in \{0, 1\}^{96}$. Вектор V формируется по левому подблоку с помощью блока расширения **E**, представляющего собой простой узел разветвления проводников. Представим УВ в виде $V = (V_1, V_2, V_3, V_4, V_5, V_6)$, где каждая из компонентов управляет одним из шести активных слоев используемых БУП. Преобразование в блоке **E** осуществляется в соответствии со следующими формулами:

$$V_1 = L_{lo}\ V_2 = L_{lo}^{>>>6},\ V_3 = L_{lo}^{>>>12},\ V_4 = L_{hi}\ V_5 = L_{hi}^{>>>6},\ V_6 = L_{hi}^{>>>12},$$

где $L_{lo} = (l_1, l_2, \ldots, l_{16}) \in \{0, 1\}^{16}$, $L_{hi} = (l_{17}, l_{n/2+2}, \ldots, l_{32}) \in \{0, 1\}^{16}$ и "$>>> k$" – циклический сдвиг на k битов (для используемой нами записи битового представления двоичных векторов сдвиг осуществляется влево). Данное правило формирования управляющего вектора соответствует критериям формирования управляющих векторов, и

перестановка каждого входного бита в УП $\mathbf{P}_{32/96}$ зависит от шести различных битов из L. Причем ни при каких значениях управляющего подблока данных ни один из его битов не влияет дважды ни на какой бит преобразуемого блока данных.

8.2.4. Переключаемая перестановка $\pi^{(e)}$

Операция $\pi^{(e)}$ представляет собой переключаемую фиксированную перестановку. Ее использование устраняет слабые и полуслабые ключи. В зависимости от значения бита e данная переключаемая операция реализует либо прямую фиксированную битовую перестановку $\pi^{(0)}$, либо обратную $\pi^{(1)}$. Фиксированные перестановки $\pi^{(0)}$ и $\pi^{(1)}$ имеют следующее представление:

$$\pi^{(0)}(x_1, x_2, \ldots, x_{32}) = ((x_1, x_2, \ldots, x_{31})^{>>>5}, x_{32}) \quad \text{и}$$

$$\pi^{(1)}(x_1, x_2, \ldots, x_{32}) = ((x_1, x_2, \ldots, x_{31})^{>>>26}, x_{32}).$$

Переключение этих перестановок осуществляется однослойным БУП $\mathbf{P}^{(e)}_{64/1}$ в соответствии с рис. 8.15.

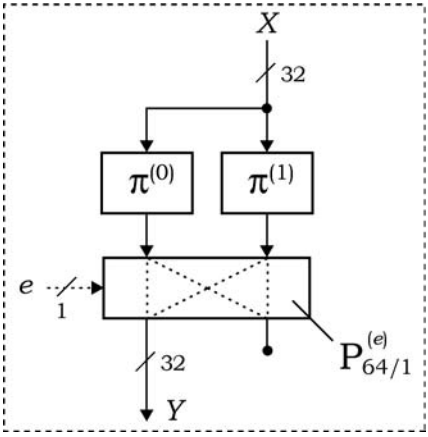

Рис. 8.15. Переключаемая перестановка $\pi^{(e)}$

Переключаемая перестановка $\pi^{(e)}$ обеспечивает влияние каждого входного бита БУП $\mathbf{P}_{32/96}$ на каждый выходной бит БУП $\mathbf{P}^{-1}_{32/96}$ даже в случае, когда в левой ветви отсутствует перестановочная инволюция $\mathbf{I}$, т. е. когда значения управляющих векторов, соответствующих блокам $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}_{32/96}$ одного раунда, равны между собой. Таким образом, назначение переключаемой перестановки состоит

- в устранении слабых и полуслабых ключей;
- в устранении необходимости использования подключей при формировании управляющих векторов.

8.2.5. Фиксированная перестановка I

Битовая перестановка **I**, выполняемая над левым подблоком данных, является инволюцией и предназначена для усиления лавинного эффекта, распространяющего изменения битов подблока данных L при одновременном выполнении двух одинаковых нелинейных преобразований $\mathbf{G}^{(1)}$ и $\mathbf{G}^{(2)}$. Перестановка **I** имеет следующую цикловую структуру:

$$\mathbf{I} = (1, 17)(2, 21)(3, 25)(4, 29)(5, 18)(6, 22)(7, 26)(8, 30)(9, 19)$$
$$(10, 23)(11, 27)(12, 31)(13, 20)(14, 24)(15, 28)(16, 32).$$

Критерии построения инволюции **I** связаны с конкретной структурой операции **G**. Пусть выходные биты y_k и y_l операции **I** соответствуют входным битам x_i и x_j. При выборе перестановочной инволюции **I** были использованы следующие принципы:

1. Для любого i и j из условия $|j - i| \le 3$ следует $|l - k| \ge 4$
2. Для любых i выполняется $|i - k| \ge 6$.

В результате применения инволюции **I** изменение одного бита в подблоке L приводит к изменению от 2 до 8 битов на выходе блока R после того, как выходные блоки $\mathbf{G}^{(1)}(L)$ и $\mathbf{G}^{(2)}(\mathbf{I}(L))$ суммируются с помощью операции "$\oplus$" с исходным блоком R.

8.2.6. Нелинейная операция G

Нелинейные операции $\mathbf{G}^{(1)}$ и $\mathbf{G}^{(2)}$ имеют одинаковую структуру и определяются в соответствии с формулой:

$$\mathbf{G}_{AB}(X) = X \oplus A \oplus X_3X_2 \oplus X_2X_1 \oplus X_3X_1 \oplus B_1X_2 \oplus A_1X_3 \oplus BX_2X_1, \text{ где:}$$

- $X, A, B \in \{0, 1\}^{32}$;
- AX – обозначает поразрядное умножение по модулю 2 векторов A и X;
- для всех $i = 1, 2, 3$ вектор X_i определен как $X_i = X^{\rightarrow i} \oplus X^{(0) \leftarrow (3-i)}$, где $X^{(0)} = (1, 1, 1, 0, \ldots, 0) \in \{0, 1\}^{32}$ – фиксированный блок начальных условий, "$\rightarrow k$" и "$\leftarrow k$" – логические сдвиги вектора на k позиций вправо и влево (освободившиеся позиции заполняются нулями);
- $A_1 = A^{\rightarrow 1} \oplus A^{(0)}$, где $A^{(0)} = (1, 0, \ldots, 0) \in \{0, 1\}^{32}$ – фиксированный блок начальных условий;
- $B_1 = B^{\rightarrow 1} \oplus B^{(0)}$, где $B^{(0)} = (1, 0, \ldots, 0) \in \{0, 1\}^{32}$ – фиксированный блок начальных условий.

Операции $\mathbf{G}^{(1)}$ и $\mathbf{G}^{(2)}$ предназначены для повышения нелинейности процедуры $\mathbf{Crypt}^{(e)}$ и усиления лавинного эффекта, распространяющего изменения битов входных данных на выходе процедуры. При рассмотрении влияния одного бита входного вектора X операции **G** на биты ее выходного вектора $Y = \mathbf{G}(X)$ полезна формула, описывающая зависимость отдельных выходных битов от входных:

$$y_i = x_i \oplus a_i \oplus x_{i-3}x_{i-2} \oplus x_{i-2}x_{i-1} \oplus x_{i-3}x_{i-1} \oplus b_{i-1}x_{i-2} \oplus a_{i-1}x_{i-3} \oplus b_i x_{i-2}x_{i-1},$$

где $(x_{-2}, x_{-1}, x_0) = (1, 1, 1)$; $a_0 = b_0 = 1$.

8.3. Блочный шифр COBRA-H128

Блочный шифр COBRA-H128, имеющий 128-битовую разрядность входа, во многом схож с криптосистемой CIKS-128, описанной ранее в [14]. Отличие состоит в небольшом видоизменении операции **G**, которое сделало вклад в лавинный эффект, имеющий место в одном раунде шифрования, одинаковым для всех битов левого подблока данных. В этом параграфе дается краткое описание шифра COBRA-H128, поскольку более подробное описание и обоснование использованных при его построении примитивов читатель может дополнительно найти в [14, 120]. Вместе с тем приводимое описание является полным и для понимания приводимого далее дифференциального криптоанализа этой криптосистемы не требуется обращения к каким-либо дополнительным литературным источникам.

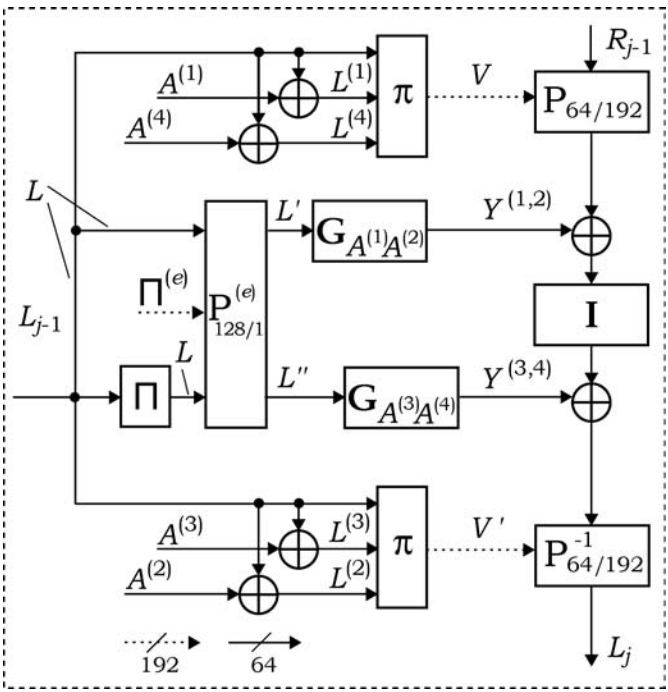

Рис. 8.16. Процедура **Crypt**$^{(e)}$ блочного шифра COBRA-H128

Общая схема шифрования COBRA-H128 соответствует рис. 8.12 за исключением того, что выполняются не десять, а двенадцать раундов преобразования. Секретный

ключ K имеет длину 256 бит и разбивается на четыре $K_i \in \{0, 1\}^{64}$: $K = (K_1, K_2, K_3, K_4)$. Раундовые ключи $Q_j^{(e)}$, где $j = 1, 2, \ldots, 12$, формируются как конкатенация секретных подключей без использования каких-либо специальных процедур преобразования секретного ключа. Начальное преобразование **IT** соответствует формулам $L_0 = L \oplus O_3$ и $R_0 = R \oplus O_4$, а конечное (**FT**) – формулам $L' = L_{12} \oplus O_1$ и $R' = R_{12} \oplus O_2$. Схема процедуры $\mathbf{Crypt}^{(e)}$ шифра COBRA-H128 представлена на рисунке 8.16.

Формирование раундового ключа

Каждый раундовый ключ, используемый в процедуре $\mathbf{Crypt}^{(e)}$, состоит из четырех подключей $(A^{(i)})_j^{(e)}$, зависящих от параметра e и номера раунда $j = 1, \ldots, 12$, и записывается в виде $Q_j^{(e)} = (A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)})_j^{(e)}$. Спецификация ключей $Q_j^{(e)}$ представлена в табл. 8.4 и 8.5.

Таблица 8.4

Расписание раундовых ключей шифра COBRA-H128
в режиме зашифрования ($e = 0$)

j	1	2	3	4	5	6	7	8	9	10	11	12
$A_j^{(1)}$	K_1	K_4	K_3	K_2	K_1	K_3	K_3	K_1	K_2	K_3	K_4	K_1
$A_j^{(2)}$	K_2	K_3	K_4	K_1	K_2	K_4	K_4	K_2	K_1	K_4	K_3	K_2
$A_j^{(3)}$	K_3	K_2	K_1	K_4	K_3	K_1	K_1	K_3	K_4	K_1	K_2	K_3
$A_j^{(4)}$	K_4	K_1	K_2	K_3	K_4	K_2	K_2	K_4	K_3	K_2	K_1	K_4

Таблица 8.5

Расписание раундовых ключей шифра COBRA-H128
в режиме расшифрования ($e = 1$)

j	1	2	3	4	5	6	7	8	9	10	11	12
$A_j^{(1)}$	K_3	K_2	K_1	K_4	K_3	K_1	K_1	K_3	K_4	K_1	K_2	K_3
$A_j^{(2)}$	K_4	K_1	K_2	K_3	K_4	K_2	K_2	K_4	K_3	K_2	K_1	K_4
$A_j^{(3)}$	K_1	K_4	K_3	K_2	K_1	K_3	K_3	K_1	K_2	K_3	K_4	K_1
$A_j^{(4)}$	K_2	K_3	K_4	K_1	K_2	K_4	K_4	K_2	K_1	K_4	K_3	K_2

Управляемые перестановки

Переменные перестановки в COBRA-H128 реализуются с помощью БУП первого порядка $\mathbf{P}_{64/192}$ и $\mathbf{P}^{-1}_{64/192}$, которые представлены на рис. 8.17.

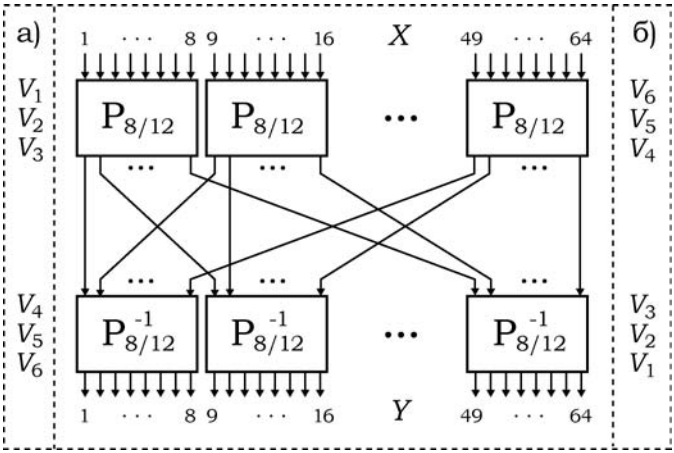

Рис. 8.17. Операционные блоки переменных перестановок, управляемые вектором $V = (V_1, V_2, V_3, V_4, V_5, V_6)$: а) $\mathbf{P}_{64/192}$ и б) $\mathbf{P}^{-1}{}_{64/192}$

Формирование управляющих векторов V и V' описывается табл. 8.6, которая показывает, что (см. рис. 8.17.) в случае вектора V (V') строки, соответствующие векторам V_1 и V_4 (V_1' и V_4'), содержат номера битов вектора L (L), строки, соответствующие векторам V_2 и V_5 (V_2' и V_5'), содержат номера битов вектора $L^{(1)}$ ($L^{(3)}$) и строки, соответствующие векторам V_1 и V_4 (V_1' и V_4'), содержат номера битов вектора $L^{(4)}$ ($L^{(2)}$).

Таблица 8.6

Распределение влияния битов управляющего подблока L в блоке $\mathbf{P}_{64/192}$

V_1	31	32	3	4	5	6	7	8	9	10	11	12	13	14	15	16
V_2	10	24	25	26	29	13	27	16	1	2	31	32	3	4	19	6
V_3	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
V_4	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
V_5	55	56	57	58	59	60	61	62	63	64	33	34	35	36	37	38
V_6	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	V_1
7	8	9	23	11	12	28	15	14	30	17	18	5	20	21	22	V_2
29	30	31	32	1	2	3	4	5	6	7	8	12	10	11	9	V_3
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	V_4
39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	V_5
61	62	63	64	33	34	35	36	37	38	39	40	41	42	43	44	V_6

Фиксированные перестановки

Перестановка **I** является инволюцией и описывается формулой

$$Y = (Y_1, Y_2, \ldots, Y_8) = \mathbf{I}(X_1, X_2, \ldots, X_8),$$

где $X_i, Y_i \in \{0, 1\}^8$; $Y_1 = X_6^{>>>4}$; $Y_2 = X_5^{>>>4}$; $Y_3 = X_4^{>>>4}$; $Y_4 = X_3^{>>>4}$; $Y_5 = X_2^{>>>4}$; $Y_6 = X_1^{>>>4}$; $Y_7 = X_8^{>>>4}$; $Y_8 = X_7^{>>>4}$.

Перестановка **П** предназначена для усиления лавинного эффекта, обусловливаемого выполнением двух идентичных нелинейных операций $\mathbf{G}^{(1)}$ и $\mathbf{G}^{(2)}$, и состоит из четырех циклов длины 16:

(1, 50, 9, 42, 17, 34, 25, 26, 33, 18, 41, 10, 49, 2, 57, 58)

(3, 64, 43, 24, 19, 48, 59, 8, 35, 32, 11, 56, 51, 16, 27, 40)

(4, 7, 28, 47, 52, 23, 12, 63, 36, 39, 60, 15, 20, 55, 44, 31)

(5, 14, 13, 6, 21, 62, 29, 54, 37, 46, 45, 38, 53, 30, 61, 22).

Перестановка $\boldsymbol{\pi}$ осуществляет распределение влияния битов управляющего подблока L, заданного табл. 8.6.

Переключаемый блок транспозиции

Блок $\mathbf{P}_{128/1}^{(e)}$ реализует транспозицию векторов L и $\mathbf{П}(L)$ при $e = 1$, благодаря чему обеспечивается корректность расшифрования.

Нелинейная операция G

Нелинейные операции $\mathbf{G}^{(1)}$ и $\mathbf{G}^{(2)}$ имеют одинаковую структуру и определяются в соответствии с формулой:

$$G_{AB}(X) = X \oplus A \oplus BX_1 \oplus X_2X_5 \oplus A_1X_6 \oplus B_1A_2 \oplus X_4X_3 \oplus X_1X_6X_4 \oplus B_1X_2X_6 \oplus B_1X_1X_2X_4,$$

где:

- $X, A, B \in \{0, 1\}^{64}$;
- AB – обозначает побитовое умножение по модулю 2 векторов A, B;
- для $i = 1, 2, 3, 4, 5, 60$ имеем $X_i = X^{>>>64-i}$;
- для $i = 1, 2$ имеем $A_i = A^{\rightarrow i} \oplus A^{(0)\leftarrow(2-i)}$, где $A^{(0)} = (1, 1, 0, \ldots, 0) \in \{0, 1\}^{64}$ – фиксированный блок начальных условий, "$\rightarrow k$" и "$\leftarrow k$" – логические сдвиги вектора на k позиций вправо и влево (освободившиеся позиции заполняются нулями);
- $B_1 = B^{\rightarrow 1} \oplus B^{(0)}$, где $B^{(0)} = (1, 0, \ldots, 0) \in \{0, 1\}^{64}$ – фиксированный блок начальных условий.

В дифференциальном анализе CIKS-128, описываемом дальше, используется следующее выражение для вычисления i-го выходного бита:

$$y_i = x_i \oplus a_i \oplus b_i x_{i-1} \oplus x_{i-2}x_{i-5} \oplus a_{i-1}x_{i-6} \oplus a_{i-2}b_{i-1} \oplus x_{i-3}x_{i-4} \oplus x_{i-1}x_{i-4}x_{i-6}$$
$$\oplus b_{i-1}x_{i-2}x_{i-6} \oplus b_{i-1}x_{i-1}x_{i-2}x_{i-4},$$

где $(x_{-5}, x_{-4}, x_{-3}, x_{-2}, x_{-1}, x_0) = (x_{59}, x_{60}, x_{61}, x_{62}, x_{63}, x_{64})$; $a_0 = a_{-1} = b_0 = 1$.

Исследование статистических свойств шифра COBRA-H128 было выполнено в соответствии со стандартными тестами, ранее примененными для оценки влияния битов исходного текста финалистов AES и битов ключа на шифртекст в шифрах SPECTR-H64 и SPECTR-128 (см. [14], с. 412-416). Полученные результаты показали, что COBRA-H128 обладает статистическими свойствами, аналогичными кандидатам AES, SPECTR-H64 и SPECTR-128. Результаты экспериментов приведены в табл. 8.7 и 8.8, где знак «*» обозначает эксперимент «1 ключ и 40000 текстов», знак «**» – эксперимент «200 ключей и 200 текстов», а знак «+» соответствует случаю «4000 ключей и 1 текст». Из результатов эксперимента видно, что после 6 раундов алгоритма COBRA–H128 статистические критерии удовлетворяются полностью как для влияния битов исходного текста, так и для влияния битов ключа. Исследование статистических свойств влияния битов ключа для данного шифра представляется весьма важным, поскольку в нем используется очень простое расписание ключа.

Таблица 8.7

Влияние битов входного текста

R	*	*	*	*	**	**	**	**
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
12	64.002	1.0000	0.9997	0.9960	64.001	1.0000	0.9996	0.9961
10	64.001	1.0000	0.9996	0.9960	64.002	1.0000	0.9997	0.9960
8	63.995	1.0000	0.9997	0.9960	64.000	1.0000	0.9996	0.9960
7	64.001	1.0000	0.9996	0.9960	64.002	1.0000	0.9997	0.9960
6	63.996	1.0000	0.9996	0.9960	63.999	1.0000	0.9996	0.9960
5	63.999	1.0000	0.9996	0.9960	63.996	1.0000	0.9996	0.9960
4	63.899	1.0000	0.9982	0.9953	63.895	1.0000	0.9982	0.9953
3	53.732	1.0000	0.8396	0.8382	53.726	1.0000	0.8395	0.8382
2	28.367	0.9983	0.4432	0.4430	28.270	1.0000	0.4417	0.4415
1	6.940	0.5005	0.1084	0.1006	6.934	0.5039	0.1083	0.1005

Таблица 8.8

Влияние битов ключа

R	+	+	+	+	**	**	**	**
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
12	63.997	1.0000	0.9997	0.9960	64.005	1.0000	0.9994	0.9921
10	64.002	1.0000	0.9996	0.9960	64.001	1.0000	0.9996	0.9960

R	+	+	+	+	**	**	**	**
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
8	64.003	1.0000	0.9997	0.9960	64.001	1.0000	0.9997	0.9960
7	64.001	1.0000	0.9996	0.9960	64.001	1.0000	0.9997	0.9960
6	63.996	1.0000	0.9996	0.9960	64.004	1.0000	0.9996	0.9960
5	63.928	1.0000	0.9986	0.9954	63.929	1.0000	0.9986	0.9954
4	63.405	1.0000	0.9906	0.9879	63.418	1.0000	0.9908	0.9881
3	55.604	1.0000	0.8688	0.8660	55.596	1.0000	0.8687	0.8660
2	29.417	0.9040	0.4596	0.4541	29.439	1.0000	0.4600	0.4545
1	5.803	0.4099	0.0907	0.0792	5.799	0.5040	0.0906	0.0792

8.4. Блочные шифры на основе управляемых подстановочно-перестановочных сетей

8.4.1. Общая схема

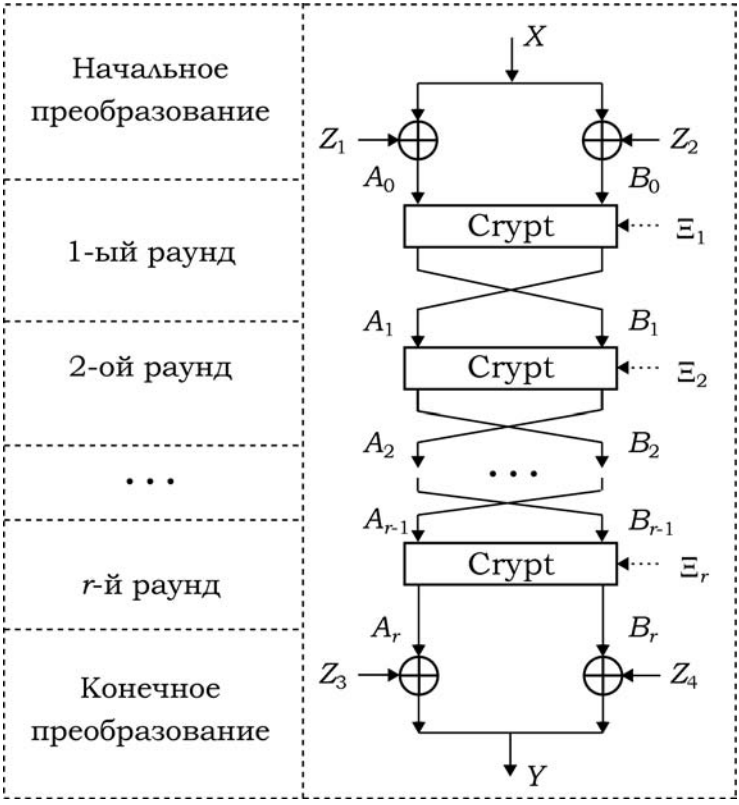

Рис. 8.18. Общая схема алгоритмов шифрования

В данном разделе разработаны варианты алгоритмов шифрования 128-битовых блоков данных [19, 23, 24]. При разработке алгоритмов используется идеология построения алгоритмов шифрования на основе БУП. Структура алгоритмов является комбинацией классов сбалансированных сетей Фейстеля и подстановочно-перестановочных сетей. Общая схема алгоритмов шифрования представлена на рис. 8.18.

Шифрование осуществляется в виде поочередного преобразования двух 64-битовых подблоков A и B:

1. Присвоить i:=1 и выполнить начальное преобразование: $A := A \oplus Z_4$, $B := B \oplus Z_3$.

2. Выполнить раундовое преобразование и, если $i < R$, переставить подблоки A и B.

3. Присвоить $i := i + 1$ и если $i \le R$, перейти к шагу 2 алгоритма.

4. Выполнить конечное преобразование: $A := A \oplus Z_2$, $B := B \oplus Z_1$.

Основу алгоритмов составляют блоки УППС $\mathbf{F}_{64/192}$ и $\mathbf{F}^{-1}{}_{64/192}$, в которых в качестве элементарных блоков $\mathbf{F}_{2/1}$ применяются блоки из табл. 4.13. В частности, при анализе стойкости исследуются блоки $\mathbf{F}_{2/1}$ синтезируемые из БФ вида $f_1 = x_2x_3 \oplus x_1$, $f_2 = x_1x_3 \oplus x_1 \oplus x_2$, (первая строка табл. 4.13).

8.4.2. Шифр SG-128 на основе УППС и векторной БФ

На рис. 8.19 представлен один раунд алгоритма (раундовая функция ***Crypt*** в соответствии с рис. 8.18). В целом алгоритм шифрования описывается следующими выражениями $Y = \mathbf{E}(X, K)$ и $X = \mathbf{D}(Y, K)$, где $X \in \mathrm{GF}(2)^{128}$ – блоки открытой информации, $K \in \mathrm{GF}(2)^{256}$ – секретный ключ, $Y \in \mathrm{GF}(2)^{128}$ – блоки зашифрованной информации, $\mathbf{E}$, $\mathbf{D}$ – функции зашифрования и расшифрования, соответственно. Секретный ключ $K = (K_1, K_2, K_3, K_4)$, где $K_i \in \mathrm{GF}(2)^{64}$.

В синтезированном шифре процедура формирования раундовых ключей $\Xi_j = Z_1 \| Z_2 \| Z_3 \| Z_4$, $j \in \{1, 2, \ldots, R\}$, из секретного ключа K заключается в использовании при зашифровании информации расписания ключей в соответствии с табл. 8.9. При расшифровании ключи используются в обратной последовательности.

Таблица 8.9

Ключи	Ξ_1	Ξ_2	Ξ_3	Ξ_4	Ξ_5	Ξ_6	Ξ_7	Ξ_8	Ξ_9	Ξ_{10}	Ξ_{11}	Ξ_{12}
Z_1	K_1	K_4	K_3	K_2	K_4	K_3	K_1	K_2	K_3	K_1	K_4	K_2
Z_2	K_2	K_1	K_4	K_3	K_2	K_4	K_3	K_1	K_2	K_3	K_1	K_4
Z_3	K_3	K_2	K_1	K_4	K_1	K_2	K_4	K_3	K_4	K_2	K_3	K_1
Z_4	K_4	K_3	K_2	K_1	K_3	K_1	K_2	K_4	K_1	K_4	K_2	K_3

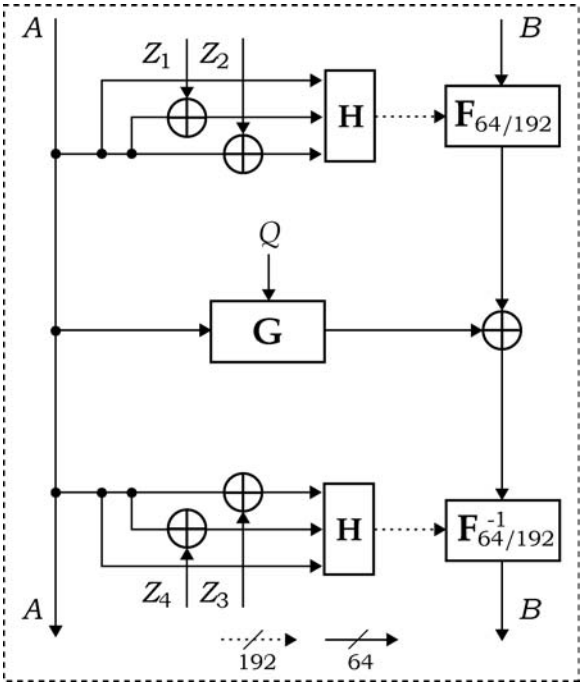

Рис. 8. 19. Структура раундовой функции ***Crypt***

Значения управляющих кодов V_1, $V_2 \in \mathrm{GF}(2)^{192}$ для блоков УППС $\mathbf{F}_{64/192}$ и $\mathbf{F}^{-1}{}_{64/192}$ формируются путем конкатенации векторов $A = A^{>>>23}$ и раундовых подключей Z_1, Z_2, Z_3, $Z_4 \in \mathrm{GF}(2)^{64}$: $V_1 = A \| A \oplus Z_1 \| A \oplus Z_2$; $V_2 = A \oplus Z_3 \| A \oplus Z_4 \| A$, где "$>>>$" – циклический сдвиг в сторону старших разрядов. Однако при последовательном использовании битов управляющих кодов в элементарных подстановочных узлах $\mathbf{F}_{2/1}$ происходит поглощение компонентов вектора A как переменных, являющихся аргументами БФ, относящихся ко всему блоку УППС и реализующих отдельные компоненты выходного вектора. Поэтому следует избегать повторного использования одного и того же бита вектора A в узлах $\mathbf{F}_{2/1}$, участвующих в формировании выходных БФ УППС $\mathbf{F}_{64/192}$ и $\mathbf{F}^{-1}{}_{64/192}$. В целях выполнения данного требования в алгоритм добавлена схема преобразования управляющего вектора – **H**. Входом схемы являются коды V_i, i=1, 2, представленные в следующем виде: $V_i = \{ W_1^{(i)} \| W_2^{(i)} \| W_3^{(i)} \| W_4^{(i)} \| W_5^{(i)} \| W_6^{(i)} \}$. Выходные значения векторов $W_j^{(i)} \in GF(2)^{32}$ определяются из соотношений: $W_1^{(1)} = A_l$, $W_2^{(1)} = A_h{}^{>>>1}$, $W_3^{(1)} = (A \oplus Z_1)_h{}^{>>>18}$, $W_4^{(1)} = (A \oplus Z_1)_l{}^{>>>4}$, $W_5^{(1)} = (A \oplus Z_2)_l{}^{>>>8}$, $W_6^{(1)} = (A \oplus Z_2)_h{}^{>>>16}$; $W_1^{(2)} = A_l$, $W_2^{(2)} = A_h{}^{>>>1}$, $W_3^{(2)} = (A \oplus Z_4)_h{}^{>>>18}$, $W_4^{(2)} = (A \oplus Z_4)_l{}^{>>>4}$, $W_5^{(2)} = (A \oplus Z_3)_l{}^{>>>8}$, $W_6^{(2)} = (A \oplus Z_3)_h{}^{>>>16}$, где индексы l, h означают младшие или старшие 32 разряда преобразуемого вектора.

В качестве подстановочного преобразования **G** используется векторная функция $\Psi(A, Q)$, где $A, Q \in \mathrm{GF}(2)^{64}$, $Q = Z_1 \oplus Z_3$ или $Q = Z_2 \oplus Z_4$ для нечетных и четных раундов, соответственно. Каждый компонент выходного вектора $\Psi(A, Q)$ вычисляется в соответствии с БФ вида:

$$f(x_j)=x_1x_4x_5x_8x_9x_{12}\oplus x_1x_4x_9x_{12}\oplus x_1x_5x_8x_{12}\oplus x_4x_5x_8x_9\oplus x_1x_5x_9\oplus x_4x_8x_{12}\oplus x_1x_4x_5 \oplus x_8x_9x_{12}\oplus x_1x_6\oplus x_2x_9\oplus x_3x_{12}\oplus x_4x_7\oplus x_5x_{11}\oplus x_8x_{10}\oplus x_{13}\oplus x_{14},$$

где $x_j \in GF(2)$. БФ $f(x_j)$ представляет собой прямую сумму бент-функции $\varphi(x_1, x_2, \ldots, x_{12})$ и аффинной функции $\phi(x_{13}, x_{14})$.

Выбор БФ $f(x_j)$ произведен с учетом оптимизации показателей эффективности, введенных в *главе 2*:

- спектр УА исследуемой БФ, определяемый в соответствии с выражением (2.3), имеет 12288 нулевых компонентов (соответствие корреляционной эффективности);
- максимальное значение спектра $\max\limits_{\alpha\in \mathrm{GF}(2)^n} |U_\alpha(f(X))| = 256$, откуда согласно (2.26) нелинейность функции $f(x_j)$ равна $\mathrm{NL}(f) = 8064$. При этом максимально возможное значение нелинейности для БФ от 14 переменных в соответствии с (2.27, 2.28) $\mathrm{NL}_{\max} = 8128$;
- автокорреляционная функция (2.10) БФ $f(x_j)$ имеет нулевые значения для любых векторов (соответствие критерию распространения) $\beta \in \mathrm{GF}(2)^{14}$, $\beta \neq \{(1, 0, 0, \ldots, 0), (0, 1, 0, \ldots, 0), (1, 1, 0, \ldots, 0)\}$.

Соответствие между координатами БФ и значениями векторов A, Q определяется подстановкой:

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} \\ a_{i-1} & q_{i-2} & a_{i-2} & a_{i-3} & q_{i-1} & a_{i-5} & q_{i-3} & a_{i-5} & a_{i-6} & q_{i-4} & a_{i-7} & a_{i-8} & a_i & q_i \end{pmatrix}$$

для $\forall\, i \in \{1, 2, \ldots, n\}$, $n = 64$. Подстановка подобного типа позволяет не снижать показатели нелинейности для любых линейных комбинаций компонентных БФ векторной функции $\Psi(A, Q)$ в сравнении с нелинейностью самих компонентных БФ. В этом случае векторная функция будет иметь вид

$$\Psi(A, Q) = (A_1\times A_3\times Q_2\times A_5\times A_6\times A_8) \oplus (A_1\times A_3\times A_6\times A_8) \oplus (A_1\times Q_2\times A_5\times A_8) \oplus (A_3\times Q_2\times A_5\times A_6) \oplus (A_1\times Q_2\times A_6) \oplus (A_3\times A_5\times A_8) \oplus (A_1\times A_3\times Q_2) \oplus (A_5\times A_6\times A_8) \oplus (A_1\times A_4) \oplus (Q_1\times A_6) \oplus (A_2\times A_8) \oplus (A_3\times Q_3) \oplus (Q_2\times A_7) \oplus (A_5\times Q_4) \oplus A_0 \oplus Q_0,$$

где $A_0 = (a_1, a_2, \ldots, a_n)$, $A_1 = (a_n, a_1, \ldots, a_{n-1})$, …, $A_j = (a_{n-j+1}, \ldots, a_n, a_1, \ldots, a_{n-j})$, $Q_0 = (q_1, q_2, \ldots, q_n)$, $Q_1 = (q_n, q_1, \ldots, q_{n-1})$, …, $Q_j = (q_{n-j+1}, \ldots, q_n, q_1, \ldots, q_{n-j})$, $\times$ и $\oplus$ – обозначают поразрядное умножение и сложение векторов по модулю два. В таком представлении векторная функция $\Psi(A, Q)$ в сравнении с вариантом, представленным в работе [74],

где проведено детальное исследование принципов построения векторных БФ подобного типа, не реализует биективное отображение относительно операнда A, однако, как будет показано ниже, оценивание стойкости такого рода векторной функции и всего алгоритма к разностному методу анализа существенно упрощается.

8.4.3. Шифр SS-128 на основе УППС, осуществляющих преобразование векторов различной размерности

В качестве альтернативы предлагается другой вариант раундовой функции ***Crypt***, полностью основанный на использовании УППС (рис. 8.20).

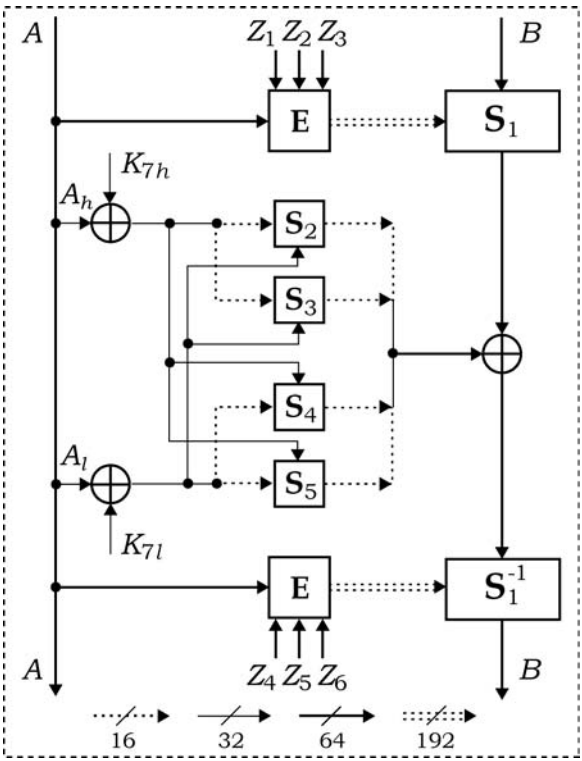

Рис. 8. 20. Раундовая функция алгоритма **SS-128**

Один раунд алгоритма включает блок УППС $\mathbf{S}_1=\mathbf{F}_{64/192}$ и $\mathbf{S}_1^{-1}=\mathbf{F}_{64/192}^{-1}$ с параметрами n=64, m=192, k=6 и блоки УППС $\mathbf{S}_2$, $\mathbf{S}_3$, $\mathbf{S}_4$, $\mathbf{S}_5$ вида $\mathbf{F}_{16/32}$ с параметрами n=16, m=32, k=4. Образующие БФ, используемые в каждом блоке $\mathbf{F}_{2/1}$, для различных УППС представлены в табл. 8.10.

Таблица 8.10

УППС	f_1	f_2
$\mathbf{S}_1, \mathbf{S}_1^{-1}$	$x_2x_3 \oplus x_1 \oplus x_3$	$x_1x_3 \oplus x_1 \oplus x_2$
$\mathbf{S}_2$	$x_2x_3 \oplus x_1$	$x_1x_3 \oplus x_1 \oplus x_2$
$\mathbf{S}_3$	$x_2x_3 \oplus x_1 \oplus x_2$	$x_1x_3 \oplus x_2$
$\mathbf{S}_4$	$x_2x_3 \oplus x_1 \oplus x_3$	$x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$
$\mathbf{S}_5$	$x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus 1$	$x_1x_3 \oplus x_2 \oplus x_3$

Формирование раундовых ключей $\Xi_j = Z_1 \| Z_2 \| \ldots \| Z_7, j \in \{1, 2, \ldots, R\}$ из секретного ключа $K = (K_1 \| K_2 \| K_3 \| K_4)$, где K_j и $Z_j \in \mathrm{GF}(2)^{64}$, заключается в использовании при шифровании информации расписания ключей в соответствии с табл. 8.11, где числитель соответствует зашифрованию, а знаменатель – расшифрованию.

Таблица 8.11

Ключи	Ξ_1	Ξ_2	Ξ_3	Ξ_4	Ξ_5	Ξ_6	Ξ_7	Ξ_8	Ξ_9	Ξ_{10}
Z_1	K_1/K_3	K_4/K_2	K_3/K_1	K_2/K_3	K_4/K_4	K_3/K_2	K_1/K_3	K_2/K_4	K_3/K_1	K_1/K_2
Z_2	K_2/K_2	K_1/K_4	K_4/K_3	K_3/K_4	K_2/K_2	K_4/K_1	K_3/K_4	K_1/K_1	K_2/K_2	K_3/K_3
Z_3	K_3/K_4	K_2/K_1	K_1/K_4	K_4/K_2	K_1/K_1	K_2/K_3	K_4/K_1	K_3/K_2	K_4/K_3	K_2/K_4
Z_4	K_2/K_1	K_1/K_3	K_4/K_2	K_3/K_1	K_2/K_3	K_4/K_4	K_3/K_2	K_1/K_3	K_2/K_4	K_3/K_1
Z_5	K_3/K_3	K_2/K_2	K_1/K_1	K_4/K_3	K_1/K_4	K_2/K_2	K_4/K_3	K_3/K_4	K_4/K_1	K_2/K_2
Z_6	K_4/K_2	K_3/K_4	K_2/K_3	K_1/K_4	K_3/K_2	K_1/K_1	K_2/K_4	K_4/K_1	K_1/K_2	K_4/K_3
Z_7	K_1/K_1	K_4/K_3	K_3/K_2	K_2/K_1	K_4/K_3	K_3/K_4	K_1/K_2	K_2/K_3	K_3/K_4	K_1/K_1

При расшифровании ключи начального и конечного преобразования используются в обратной последовательности.

Значения управляющих векторов $V_1, V_2 \in \mathrm{GF}(2)^{192}$ для УППС $\mathbf{S}_1$ и $\mathbf{S}_1^{-1}$ формируются в соответствии со схемой расширения $\mathbf{E}$, использование которой позволяет предотвратить поглощение компонентов вектора A как переменных, являющихся аргументами БФ, реализующих отдельные компоненты выходного вектора УППС. Выходом схемы являются векторы $V_i = \{W_1^{(i)} \| W_2^{(i)} \| W_3^{(i)} \| W_4^{(i)} \| W_5^{(i)} \| W_6^{(i)}\}$, где $i = 1, 2$. Значения векторов $W_j^{(i)} \in \mathrm{GF}(2)^{32}$ определяются из соотношений: $W_1^{(1)} = (A \oplus Z_1)_l$, $W_2^{(1)} = (A \oplus Z_1)_h^{<<<1}$, $W_3^{(1)} = (A \oplus Z_2)_h^{<<<18}$, $W_4^{(1)} = (A \oplus Z_2)_l^{<<<4}$, $W_5^{(1)} = (A \oplus Z_3)_l^{<<<8}$, $W_6^{(1)} = (A \oplus Z_3)_h^{<<<16}$; $W_1^{(2)} = (A \oplus Z_4)_l$, $W_2^{(2)} = (A \oplus Z_4)_h^{<<<1}$, $W_3^{(2)} = (A \oplus Z_5)_h^{<<<18}$,

$W_4^{(2)} = (A \oplus Z_5)_l^{<<<4}$, $W_5^{(2)} = (A \oplus Z_6)_l^{<<<8}$, $W_6^{(2)} = (A \oplus Z_6)_h^{<<<16}$. Для УППС $\mathbf{S}_2$, $\mathbf{S}_3$ и $\mathbf{S}_4$, $\mathbf{S}_5$ управляющими векторами являются младшие и старшие 32 разряда подблока A, соответственно, над которыми предварительно производится операция сложения по модулю 2 с ключами Z_{7l} и Z_{7h}, где индекс $7l$ означает младшие 32 разряда ключа Z_7, а $7h$ – старшие 32 разряда ключа Z_7. Биты управляющих векторов используются последовательно по слоям в соответствии с нумерацией блоков $\mathbf{F}_{2/1}$ (слева-направо, сверху-вниз), т.е. i-бит управляющего вектора попадает на i-й блок $\mathbf{F}_{2/1}$.

Другие варианты использования УППС в симметричных алгоритмах блочного шифрования информации рассмотрены в работах [18, 25].

Таким образом:

- рассмотрены общие принципы использования перспективных управляемых криптографических примитивов в блочных алгоритмах;
- предложена эффективная структура блочных алгоритмов на основе УППС и векторных БФ, а также рассмотрены примеры конкретных реализаций УППС в различных блочных криптографических алгоритмах;
- предложены схемы блочных криптографических алгоритмов, обеспечивающие высокое быстродействие, невысокую сложность аппаратной реализации и, как будет показано далее, высокую стойкость к различным методам анализа.

8.5. Анализ стойкости и статистическое тестирование шифров на основе управляемых и переключаемых операций

Оценивание эффективности применения любых криптографических примитивов необходимо не только при определении достигаемого уровня производительности или схемотехнической сложности аппаратной реализации, но и, в первую очередь, с учетом результатов их криптографической стойкости. При этом именно криптоанализ представляется наиболее ресурсоемким этапом разработки и внедрения новых криптоалгоритмов.

Одним из основных методом криптоанализа блочно-итерационных алгоритмов шифрования является разностный (дифференциальный) метод. Он основан на неравномерности распределения выходных разностей и состоит в поиске некоторой специально выбранной метрики, для которой имеет место высокая вероятность выходной разности [57, 123].

Дифференциальный метод применим для криптоанализа марковских криптоалгоритмов. Марковским называется криптоалгоритм, у которого уравнение шифрования на одном раунде удовлетворяет условию: вероятность появления разности не зависит от выбора открытых сообщений [48]. Тогда последовательность разностей на каждом раунде образует марковскую цепь, в которой последующее состояние опре-

деляется только предыдущим. В основу данного метода положены вероятностные характеристики алгоритма шифрования.

Пара (X, X') открытых текстов соответствует входной разности $\Delta X = X - X' = X \oplus X'$ и при последовательном выполнении преобразования $Y = F_{k_r} F_{k_{r-1}} \dots F_{k_1}(X)$ порождает последовательность выходных разностей

$$\Delta Y(1) = Y(1) \oplus Y'(1), Y(2) \oplus Y'(2), \dots, \Delta Y(r) = Y(r) \oplus Y'(r),$$

где $Y(i)$, $Y'(i)$ – результаты выполнения i раундов преобразования над X и X' соответственно. Для последовательности из s разностей будем рассматривать вероятность s-раундовой характеристики

$$\Pr\{\Delta Y(1) = \beta(1), \dots, \Delta Y(s) = \beta(s) \mid \Delta X = \alpha\},$$

$1 \le s \le r$, в предположении, что X, X' – случайные элементы с равномерным распределением на всем множестве открытых текстов, $k_1, k_2, \dots, k_r$ – случайные раундовые ключи, равномерно распределенные на всем множестве ключей. Для марковских алгоритмов шифрования вероятность s-раундового дифференциала определяется равенством

$$\Pr\{\Delta Y(s) = \beta(s) \mid \Delta X = \alpha\} =$$

$$\sum_{\beta(1)} \sum_{\beta(2)} \cdots \sum_{\beta(s-1)} \prod_{i=1}^{s} \Pr\{\Delta Y(i) = \beta(i) \mid \Delta Y(i-1) = \beta(i-1)\}$$

Целью атаки криптоаналитика противника является восстановление раундового ключа k_r. Предварительно находится $(r-1)$ - раундовая характеристика $(\alpha, \beta(r-1))$ с максимально возможной вероятностью p. После чего в ходе N экспериментов генерируются случайные открытые тексты X_j, вычисляются $X_j' = X_j \oplus \alpha$ и находятся Y_j, Y_j', где $j = 1, \dots, N$. В качестве оценки $\hat{k}$ искомого раундового ключа k_r выбирается значение

$$\hat{k} = \arg\max_{k \in \mathbf{K}} \sum_{j=1}^{N} I\left\{ F_k^{-1}(Y_j) \oplus F_k^{-1}(Y_j') = \beta(r-1) \right\},$$

где $I\{\bullet\}$ – индикатор случайного события. $I\{\Theta\} = 1$, если событие Θ наступает, и $I\{\Theta\} = 0$ – в противном случае. Рассмотренные операции повторяются необходимое число раз, до тех пор пока не будет установлен подлинный раундовый ключ как наиболее вероятный.

Вычислительная сложность атаки по методу дифференциального анализа составляет не менее $W_{\min} \ge \frac{1}{p}$ операций преобразований $Y = F_{k_r} F_{k_{r-1}} \dots F_{k_1}(X)$.

8.5.1. Анализ стойкости шифра DDP-64 к дифференциальному и линейному криптоанализу

Шифр DDP-64 (рассмотренный в разделе 4) является примером блочного шифра, основанного только на перестановках, зависящих от преобразуемых данных (в том смысле, что они являются единственным нелинейным криптографическим примитивом). Переменные перестановки используются для решения следующих задач:

- преобразования правого подблока данных с помощью операционных блоков $\mathbf{P}_{32/96}$, $\mathbf{P}^{-1}_{32/96}$, обеспечивающих возможность обратного преобразования;
- осуществления нелинейного преобразования (осуществляемого с помощью блоков **F**), не позволяющего однозначно выполнить обратное преобразования;
- транспозиции соответствующих пар подключей при смене режима зашифрования на режим расшифрования;
- реализации переключаемой перестановки $\Pi^{(e')}$.

Также как и SPECTR-H64, криптосистема DDP-64 имеет высокую производительность в случае частой смены ключей, так как в ней не используются предвычисления ключа. Другим сходством является также то, что для переменных перестановок, выполняемых операционными блоками **F**, $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}_{32/96}$ достаточно легко вычислить дифференциальные характеристики, соответствующие разностям с малым числом активных битов, причем такие характеристики имеют максимальную вероятность. По сравнению с SPECTR-H64 шифр DDP-64 имеет следующие особенности:

1. В шифре DDP-64 используется весь секретный ключ в каждом раунде.
2. В шифре DDP-64 используются два **F**-блока, выполняемых одновременно с операцией $\mathbf{P}_{32/96}$. Каждый из двух **F**-блоков представляет собой БУП специального типа.
3. Раундовое преобразование включает в себя специальные перестановочные инволюции, выполняемые над левым и правым подблоками данных.
4. Лавинный эффект распространяется за счет того, что изменяемые биты используются как управляющие. При преобразовании двоичных векторов, содержащих измененные биты, количество измененных битов сохраняется неизменным, т. е. лавинный эффект не имеет места. (Некоторое исключение можно видеть в случае **F**-блока, где 8 входных битов используются как внутренний управляющий вектор, обозначенный ранее как W_6).

Рассматривая стойкость шифра DDP-64 к дифференциальному анализу был получен результат, характерный для шифров, основанных на перестановках, зависящих от преобразуемых данных [14]. Этот результат заключается в том, что наибольшими вероятностями обладают дифференциальные характеристики с разностями, имеющими наименьший вес. Воспользуемся следующими обозначениями. Пусть Δ^{W_h} разность, содержащая h-активных (т. е. не равных нулю) битов и соответствующая век-

тору W. Пусть $\Delta_{h|i_1,\ldots i_n}$ – разность с активными битами, находящимися в разрядах $i_1, \ldots, i_n$. В отличие от предыдущего обозначения во второй записи фиксируются разряды, которым принадлежат активные биты. В первом случае рассматриваются множества разностей с заданным количеством активных битов (в различных конкретных вариантах разности $\Delta^W{}_h$ активные биты принадлежат в общем случае различным разрядам). Пусть $p(\Delta \to \Delta'/\mathbf{P})$ – вероятность того, что входная разность Δ, проходя через операцию $\mathbf{P}$, преобразуется в выходную разность Δ'.

Лавинный эффект, соответствующий операциям $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$, обусловливается использованием подблока данных L для задания значений V и V'. Каждый бит левого подблока данных влияет на три бита каждого из этих управляющих векторов. Каждый управляющий бит влияет на два бита правого подблока данных. Таким образом, благодаря УП, производимым над правым подблоком данных R операционными блоками $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$, один бит L влияет примерно на 12 бит R. В том случае, когда некоторая разность с одним активным битом $\Delta'' = \Delta_{1/i}$ проходит по левой ветви криптосхемы, она влияет на три элементарных переключающих элемента, переставляющих шесть различных битов правого подблока данных. Например, если входная разность БУП $\mathbf{P}_{32/96}$ не содержит активных битов (это случай нулевой разности), то разность $\Delta^L{}_{1/i}$ может вызвать следующие события, зависящие от правого подблока данных:

1. Активные биты в БУП не формируются (т. е. формируется выходная разность Δ'_0) с вероятностью 2^{-3};
2. С вероятностью $3\cdot2^{-3}$ формируется разность Δ'_2 на выходе БУП;
3. С вероятностью $3\cdot2^{-3}$ формируется выходная разность Δ'_4;
4. С вероятностью 2^{-3} формируется выходная разность Δ'_6.

Средние вероятности $p(\Delta_q \to \Delta'_g/\mathbf{P}_{32/96})$, соответствующие входным и выходным разностям блока $\mathbf{P}_{32/96}$ с несколькими активными битами ($q = 0, 1, 2$ и $g = q + a$, где $a = -2, 0, 2, 4$), представлены в табл. 8.12. Эти вероятности зависят от числа активных битов в разности, проходящей по левой ветви раунда шифрования. Вероятности $p(\Delta_{q|i_1,\ldots i_q} \to \Delta'_g)$ вычислены при допущении, что значения номеров $i_1, \ldots, i_q$, соответствующие позициям активных битов, равновероятны. При этом учитывается, что позиции активных битов в разностях, рассматриваемых в табл. 8.12, не фиксированы. Можно легко показать, что для произвольной входной разности Δ_q и соответствующей ей выходной разности Δ'_g сумма $q + g$ всегда является четной.

Таблица 8.12

Значения вероятностей некоторых характеристик БУП $\mathbf{P}_{32/96}$

Δ''	$\Delta_0 \to \Delta'_0$	$\Delta_0 \to \Delta'_2$	$\Delta_0 \to \Delta'_4$	$\Delta_1 \to \Delta'_1$	$\Delta_1 \to \Delta'_3$
$\Delta^L{}_0$	1	0	0	1	0
$\Delta^L{}_1$	2^{-3}	$1.5\cdot2^{-2}$	$1.5\cdot2^{-2}$	$1.17\cdot2^{-3}$	$1.59\cdot2^{-2}$
$\Delta^L{}_2$	2^{-6}	$1.5\cdot2^{-4}$	$1.88\cdot2^{-3}$	$1.38\cdot2^{-6}$	$1.88\cdot2^{-4}$

Δ''	$\Delta_1 \to \Delta'_5$	$\Delta_2 \to \Delta'_0$	$\Delta_2 \to \Delta'_2$	$\Delta_2 \to \Delta'_4$	$\Delta_2 \to \Delta'_6$
$\Delta^L{}_0$	0	0	1	0	0
$\Delta^L{}_1$	$1.41 \cdot 2^{-2}$	$1.55 \cdot 2^{-9}$	$1.08 \cdot 2^{-3}$	$1.36 \cdot 2^{-2}$	$1.15 \cdot 2^{-2}$
$\Delta^L{}_2$	$1.06 \cdot 2^{-2}$	$1.55 \cdot 2^{-8}$	$1.38 \cdot 2^{-6}$	$1.69 \cdot 2^{-4}$	$1.72 \cdot 2^{-3}$

Вклад операции **F** в лавинный эффект определяется в основном благодаря использованию левого подблока данных для задания управляющих векторов W и W'. Дополнительный вклад вносится за счет зависимости выходного вектора W_6, формируемого блоком расширения «Ext» (см. рис.4.6), от L. Рассмотрим вектор $L = (L_l, L_h)$ перед операцией "<<< 16" (см. рис.4). Каждый бит l_i из L_l, где $1 \le i \le 16$, влияет на три элементарных блока $\mathbf{P}_{2/1}$ блока $\mathbf{P}_{32/48}$ в нижнем **F**-блоке и на два $\mathbf{P}_{2/1}$ блока $\mathbf{P}^{-1}{}_{32/48}$ в верхнем **F**-блоке (после выполнения операции "<<< 16" бит l_i перемещается в старшую половину разрядов L). Кроме того, с вероятностью 2^{-2} (вероятность того, что l_i будут перемещены в один из разрядов, соответствующих вектору H_5) бит l_i влияет на два блока $\mathbf{P}_{2/1}$, принадлежащих первому активному слою БУП $\mathbf{P}^{-1}{}_{32/48}$ в верхнем **F**-блоке и с той же вероятностью l_i влияет на два аналогичных блока $\mathbf{P}_{2/1}$ в нижнем **F**-блоке. Такие же свойства имеют все биты L_h, поскольку после операции "<<< 16" компоненты входного вектора L_l и L_h меняются местами, а преобразования в верхнем и нижнем блоках **F** являются «симметричными».

Рассмотрим механизм формирования итеративной двухраундовой характеристики с разностью $(\Delta^L{}_1, \Delta^R{}_0)$. Разность с одним активным битом $\Delta^L{}_1$, проходя по левой ветви криптосхемы, может породить нулевую разность на выходе обоих **F**-блоков. Это может произойти в двух следующих наиболее вероятных случаях. Случай 1 связан с реализацией всех перечисленных ниже элементарных событий:

1) в обоих **F**-блоках активный бит перемещается в один из восьми разрядов вектора H_5 на выходе перестановки Π' (вероятность этого события равна $p_1 = 2^{-2} \cdot 2^{-2} = 2^{-4}$);
2) в обоих **F**-блоках активный бит не порождает никаких пар активных битов в операционных блоках $\mathbf{P}_{32/48}$ и $\mathbf{P}^{-1}{}_{32/48}$ (вероятность данного события равна $p_2 = 2^{-2} \cdot 2^{-3} \cdot (2^{-2})^2 = 2^{-9}$);
3) в блоке $\mathbf{P}_{32/96}$ активный бит левого подблока данных не порождает дополнительных активных битов в правом подблоке данных (вероятность этого события $p_3 = 2^{-3}$);
4) в блоке $\mathbf{P}^{-1}{}_{32/96}$ активный бит не порождает активных битов (вероятность этого события $p_4 = 2^{-3}$).

В случае 2 мы имеем следующие события:

1) в обоих **F**-блоках активный бит перемещается на выходе перестановки Π' в один из тридцати двух разрядов векторов вектора (H_1, H_2, H_3, H_4) (вероятность этого события равна $p'_1 = (1 - 2^{-2})^2 \approx 1.12 \cdot 2^{-1}$);

2) в обоих **F**-блоках активный бит левой ветви не вызывает порождение пар активных битов в блоках $\mathbf{P}_{32/48}$ и $\mathbf{P}^{-1}{}_{32/48}$ (вероятность этого события равна $p'_2 = 2^{-2} \cdot 2^{-3} = 2^{-5}$);

3) в блоке $\mathbf{P}_{32/96}$ активный бит левого подблока данных не порождает дополнительных активных битов в правом подблоке (вероятность этого события $p'_3 = 2^{-3}$);

4) в блоке $\mathbf{P}^{-1}{}_{32/96}$ активный бит левого подблока данных не порождает активных битов (вероятность этого события $p'_4 = 2^{-3}$).

5) на выходе верхнего и нижнего **F**-блоков возникают, соответственно, разности $\Delta'_{1|i}$ и $\Delta'_{1|j}$, где $i = \mathbf{I}_2(j)$ (вероятность этого события $p'_5 = 2^{-5}$).

Обозначим вероятности реализации случаев 1 и 2 через P' и P'', соответственно. Рассматривая вероятности элементарных событий, легко получить:

$$P' = p_1 p_2 p_3 p_4 = 2^{-4} \cdot 2^{-9} \cdot 2^{-3} \cdot 2^{-3} = 2^{-19} \text{ и}$$

$$P'' = p'_1 p'_2 p'_3 p'_4 p'_5 = 2^{-1} \cdot 2^{-5} \cdot 2^{-3} \cdot 2^{-3} \cdot 2^{-5} = 1.12 \cdot 2^{-17}.$$

Существует несколько других возможных механизмов получения нулевой разности на выходе правой ветви криптосхемы, однако их вклад в вероятность итеративной двухраундовой характеристики существенно ниже по сравнению с вкладом случаев 1 и 2 и ими можно пренебречь. Таким образом, наиболее значимые случаи определяют вероятность $P(2) = P' + P'' = 1.37 \cdot 2^{-17}$. Разность (Δ^L_1, Δ^R_0) проходит один раунд с вероятностью $P(2) = 1.37 \cdot 2^{-17}$. После перестановки подблоков данных на входе второго раунда мы имеем разность (Δ^L_0, Δ^R_1), которая проходит его с вероятностью 1 и после перестановки подблоков данных дает начальную разность (Δ^L_1, Δ^R_0). В результате для двухраундовой характеристики мы получили вероятность $P(2) \approx P = 1.37 \cdot 2^{-17}$.

Наилучшими представляются характеристики с разностями (Δ^L_0, Δ^R_1) и (Δ^L_1, Δ^R_0). Рассматривая другие дифференциальные характеристики с различным числом активных битов в соответствующих им разностях, мы установили, что добавление активных битов существенно уменьшает вероятность характеристики. Разность (Δ^L_1, Δ^R_0) проходит восемь или десять раундов шифра DDP-64 с вероятностью

$$P(8) = P^4(2) \approx 1.76 \cdot 2^{-67} \quad \text{и} \quad P(10) = P^5(2) \approx 1.2 \cdot 2^{-83}.$$

Для случайного шифра мы имеем $P((\Delta^L_1, \Delta^R_0) \to (\Delta^L_1, \Delta^R_0)') = 2^{-64} \cdot 2^5 = 2^{-59} > P(8) > P(10)$. Таким образом, DDP-64 с восьмью и десятью раундами неотличим от случайного шифра при дифференциальной атаке, использующей наиболее эффективные итеративные характеристики.

Использование линейного криптоанализа для выявления отличия шифра DDP-64 от случайного представляется менее эффективным по сравнению с дифференциальной атакой. Наши исследования показали, что наибольшее смещение имеют ЛХ с малым числом активных битов, а максимальное – ЛХ с двумя активными битами, которые строятся с учетом событий, учитывающих замену битов преобразуемых данных битами константы $C = (10101010)$. Пусть $A = (A^L, A^R)$ и $B = (B^L, B^R)$ являются

входной и выходной масками, соответственно. Верхние индексы L и R указывают, соответственно, на левую и правую половины маски. В силу замысла, заложенного при проектировании шифра DDP-64, ЛХ с масками $A = B = (111\ldots1)$ имеют очень малое смещение, поскольку **F**-блоки реализуют преобразование с достаточно высоким значением нелинейности.

Используя формулы для расчета линейных характеристик, приведенные в [2, 14], легко установить, что смещение линейной характеристики (ЛХ) с числом активных битов $z \le 31$ имеет значение $b \le 2^{-6}$ для каждого из блоков $\mathbf{P}_{32/96}$, $\mathbf{P}^{-1}{}_{32/96}$ и **F**. Максимальное значение $b = 2^{-6}$ соответствует случаю $z = 1$. Вычисление линейных характеристик БУП удобно проводить, рассматривая «физическое» движение битов по перестановочной сети. Поскольку шифр DDP-64 преимущественно построен на основе перестановочных операций, то данный подход может быть применен и для него. Введем нижние индексы в обозначения масок, которые по аналогии с обозначением дифференциальных характеристик будут указывать на количество активных (ненулевых) битов и разряды, которым принадлежат активные биты. Например, A_2 и $A_{2|5,7}$ означают, соответственно, произвольную маску с двумя активными битами и маску с двумя активными битами, расположенными в пятом и седьмом разрядах, отсчитываемых слева.

Рассмотрим однораундовую ЛХ с масками $A = (A^L{}_{1|i}, A^R{}_{1|j})$ и $B = (B^L{}_{1|i'}, B^R{}_{1|g})$, где значение i' определяется значением i (i' есть номер разряда, в который перемещается i-ый бит левого подблока на входе первого раунда). Смещение указанной ЛХ определяется тем, что имеется вероятность того, что активный бит левого подблока будет дважды налагаться при выполнении операции XOR с активным битом правого подблока. Для этого он должен быть перемещен верхним **F**-блоком в тот же разряд, в который попадает активный бит правого подблока на выходе операции $\mathbf{P}_{32/96}$ (вероятность этого события $p_1 = 2^{-5}$). Активный бит левого подблока также должен быть перемещен нижним **F**-блоком в разряд, в который попадает активный бит правого подблока на выходе операции **I** (вероятность этого события $p_2 \approx 0.75 \cdot 2^{-5}$). После этого активный бит в правой ветви должен попасть на выходе блока $\mathbf{P}^{-1}{}_{32/96}$ в g-ый разряд (вероятность этого элементарного события $p_3 = 2^{-5}$). Таким образом, выделенные на входе первого раунда два бита с вероятностью $P(1) = p_1 p_2 p_3 \approx 0.75 \cdot 2^{-15}$ попадают в известные разряды на выходе. Зная эту вероятность, легко вычислить смещение $b(1)$ однораундовой характеристики $((A^L{}_{1|i}, A^R{}_{1|j}); (B^L{}_{1|i'}, B^R{}_{1|g}); b(1))$:

$$b(1) = 0.5(1 - P(1)) + P(1) = 0.5P(1) = 0.75 \cdot 2^{-16}.$$

Для r-раундовой характеристики $((A^L{}_{1|i}, A^R{}_{1|j}); (B^L{}_{1|i'}, B^R{}_{1|g}); b(r))$ аналогичным путем легко получить следующую оценку:

$$b(r) = 0.5P(r) < 2^{-15r-1}.$$

Из полученных оценок видно, что трех раундов шифра DDP-64 с избытком достаточно, чтобы предотвратить линейный криптоанализ.

Следует отметить, что значение константы $C = (10101010)$ выбрано таким образом, что ее вес $\phi(C)$ имеет значение 4. Это связано с тем, что для другого веса стано-

вится преобладающим другой механизм формирования линейной характеристики $((A^L_0, A^R_{1|j}); (B^L_0, B^R_{1|g}); b(2))$, связанный с разной вероятностью замены активного бита, переходящего из левой ветви раундового преобразования в правую, на нулевое и единичное значение. Этот механизм определяет следующее значение смещения указанной двухраундовой ЛХ:

$$b(2) = 2^{-2r-6}\left(|\phi(C) - 4|/4\right)^{r/2} = 2^{-3r-6}\,|\phi(C) - 4|^{r/2}.$$

Из данной формулы видно, что при $\phi(C) \neq 4$ линейный криптоанализ становится существенно эффективным. Благодаря указанному выбору веса $\phi(C)$ обеспечивается высокая стойкость к линейному анализу.

Рассмотрим некоторые соображения относительно других атак. Высокая степень алгебраической нормальной формы и сложность булевых функций (содержащих более 100 000 членов), описывающих раундовое преобразование DDP-64, предотвращают алгебраическую атаку. Несмотря на очень простое расписание ключей, DDP-64 является стойким к специальным атакам (slide attacks) [59] на шифры, в которых не используются процедуры предварительного преобразования ключа, благодаря тому, что рассматриваемый шифр включает 1) непериодическое использование подключей и 2) раундовое преобразование, которое не является инволюцией и содержит переключаемую операцию, расписание режимов которой также не является периодическим. В частности, последний фактор предотвращает атаки типа «slide attack» в случае, когда все подключи имеют равные значения. Несмотря на простоту расписания ключей, «симметричные» ключи $K' = (X, Y, Y, X)$ и $K'' = (X, X, X, X)$ не являются ни слабыми, ни полуслабыми, так как дешифрование требует осуществления соответствующего переключения операции $\Pi^{(e')}$. Например, рассматривая раундовое преобразование, легко заметить, что $T^{(e=0)}(C, K'') \neq M$, где $C = T^{(e=0)}(M, K'')$). Представляется вычислительно сложным нахождение полуслабых пар ключей для DDP-64 – если это вообще возможно.

С последними замечаниями связан следующий важный вывод: перестановки, зависимые от e, играют достаточно важную роль в DDP-64, не включающем процедуры предварительного преобразования ключа. Для сравнения укажем, что для шифра SPECTR-H64, в котором нет переключаемых операций, для любого значения X 256-битный ключ $K = (X, X, X, X, X, X, X, X)$ является слабым, кроме того использование такого ключа создает предпосылки к успешному проведению атак типа «slide attack».

8.5.2. Оценка аппаратной реализации шифра DDP-64

Интерес к оценке различных вариантов аппаратной реализации шифра DDP-64 связан с тем, что этот шифр в «чистом виде» демонстрирует эффективность переменных перестановок при их использовании в качестве криптографического примитива. Следовательно, и используемые при реализации аппаратные ресурсы, и скоростные показатели определяются только реализацией переменных перестановок. На примере

этого шифра можно получить характерные оценки аппаратных реализаций. Временная задержка, соответствующая одному активному слою многослойных БУП, примерно равна τ, где τ – время задержки операции XOR (⊕). Время задержки (T) блока $\mathbf{P}_{m/n}$ может быть оценено как $T \approx 2m\tau/n$. Критический путь комбинационных схем, реализующих различные элементы шифра трудно оценить в единицах τ, поскольку в данном случае нет привязки к конкретной микроэлектронной технологии. В выбранных единицах критический путь одного раунда DDP-64 равен 16τ. Критический путь десятираундового шифра DDP-64 составляет 162τ (две единицы связаны с выполнением начального и конечного преобразований).

Таблица 8.13

Сравнительная оценка аппаратной реализации DDP-64.

Шифр	Количество вентилей		Критический путь, τ	
	Установка ключа	Шифрование	Установка ключа	Шифрование
DDP-64 (10 раундов)	320	34 500	–	162
DDP-64 (8 раундов)	320	28 500	–	130
DES	12 000*	42 000*	–	80
Triple-DES	23 000*	120 000*	–	220
Rijndael	94 000*	520 000*	83	95
RC6	900 000*	740 000*	3000	880
TwoFish	230 000*	200 000*	23	470

Сложность аппаратной реализации блока $\mathbf{P}_{n/m}$ составляет $6m$ вентилей И-НЕ. Реализация 10 раундов DDP-64 требует менее 30 000 вентилей И-НЕ. К этому числу следует добавить некоторое количество вентилей, соответствующих 160-битовому регистру ключа и двум 64-битовым регистрам входных и выходных данных. Это составляет примерно 4 500 вентилей. По нашим оценкам суммарная сложность реализации десяти (восьми) раундов DDP-64, включающей затраты на регистры для хранения ключей и данных, составляет менее 35 000 (29 000) вентилей И-НЕ при использовании варианта схемотехнической реализации, описанного в [80] и заключающегося в исполнении комбинационной схемы, включающей полное число раундов шифрования. Таблица 8.13 представляет оценку сложности схемотехнической реализации, дающую сравнение различных шифров (числа, помеченные *, относятся к результатам работы [80]). Производительность шифра удобно представить в количестве преобразованных битов, приходящемся на время τ. Из приведенных в таблице данных видно, что самая быстрая реализация соответствует 128-битовому шифру Rijndael (производительность $f \approx 1.35$ бит/τ), что достигается сравнительно высокой стоимостью реализации, а самая дешевая соответствует DDP-64 (453 – 547 вентилей/бит).

Шифр DDP-64 имеет производительность $f \approx 0.42 - 0.52$ бит/τ, что превосходит значения производительности многих широко применяемых криптосистем, например, RC6 ($f \approx 0.15$ бит/τ), Triple-DES ($f \approx 0.29$ бит/τ) и Twofish ($f \approx 0.27$ бит/τ). Примечательно, что реализация DDP-64 требует существенно меньше схемотехнических ресурсов по сравнению с DES. Эти результаты показывают, что криптосистема DDP-64 очень хорошо подходит для встраивания в интеллектуальные электронные карточки и микроконтроллеры различного типа. В работе [20] показано также, что благодаря низкой сложности схемотехнической реализации, эффективности как криптографического примитива и общей значимости различных типов операций битовых перестановок, операция УП является также хорошей кандидатурой в качестве новой быстрой команды для встраивания в систему стандартных операций процессоров общего назначения.

Для получения более полной картины о достигаемых параметрах аппаратной реализации различных шифров были проведены работы по проектированию устройств шифрования на базе ряда криптоалгоритмов, основанных на перестановках, зависящих от преобразуемых данных. Были выбраны следующие два варианта реализации: 1) с использованием программируемых логических матриц типа FPGA фирмы **Xilinx Vitrex** и 2) с использованием зарубежной микроэлектронной технологии с разрешением 0.33 мкм для изготовления заказных СБИС. Данные работы выполнены совместно с Патрасским университетом (Греция). Были исследованы параметры реализации алгоритмов DDP-64, CIKS-1 и SPECTR-H64. Реализация осуществлена для следующих двух архитектур:

1. Схемная реализация одного раунда и ее использование для выполнения всех раундов шифрования при смене раундовых ключей (итеративная архитектура – ИА).
2. Конвейерная реализация с числом уровней, равным числу раундов шифрования, в которой схемно реализуется полное число раундов (конвейерная архитектура – КА).

Конвейерная архитектура обеспечивает существенно более высокую производительность, однако требует сравнительно больших аппаратных затрат. Итеративная архитектура обеспечивает минимальную стоимость реализации, однако при этом значительно уменьшается производительность. Тем не менее, ИА обладает весьма существенным преимуществом, которое заключается в возможности использования блочных шифров в режиме сцепления блоков шифра практически с сохранением того же уровня производительности, который обеспечивается в режиме электронной кодовой книги (независимое шифрование блоков данных).

Результаты реализации представлены в табл. 8.14. Сравнение с аналогичной реализацией других шифров (см. табл. 8.15) показывает, что DDP-64, CIKS-1 и SPECTR-H64 обеспечивают более высокую скорость при меньших аппаратных затратах по сравнению с криптосистемами AES и IDEA. Их стоимость реализации несколько превышает стоимость реализации алгоритма DES, однако при этом они обеспечивают многократное превышение по скорости шифрования.

Таблица 8.14

Результаты аппаратной реализации шифров DDP-64, CIKS-1 и SPECTR-H64 с использованием программируемых и заказных СБИС

Шифр	Архи-тектура	ПЛИС (Xilinx Vitrex)			Заказные СБИС (0.33 мкм)		
		К-во блоков CLB*	Частота, Мбит/с	Скорость шифрования Гбит/с	Площадь, sqmil**	Частота Мбит/с	Скорость шифрования Гбит/с
DDP-64	ИА	615	85	0.544	2 620	92	0.589
DDP-64	КА	3 440	95	6.100	14 050	101	6.500
CIKS-1	ИА	907	81	0.648	3456	93	0.744
CIKS-1	КА	6346	81	5.184	21036	95	5.824
SPECTR-H64	ИА	713	83	0.443	3194	91	0.485
SPECTR-H64	КА	7021	83	5.312	32123	94	6.016

* Типовые логические элементы данного типа программируемых логических схем (ПЛИС): Configurable Logic Blocks (CLB) – конфигурируемые логические блоки.

** Указана площадь используемой поверхности полупроводникового кристалла в единицах sqmil; 1 sqmil = $7.45\ 10^{-4}$ мм^2

Таблица 8.15

Сравнение результатов аппаратной реализации различных шифров с использованием программируемых СБИС

Шифр	Архитектура реализации	ПЛИС (Xilinx Vitrex)		
		К-во блоков CLB	Частота, Мбит/с	Скорость шифрования Гбит/с
DDP-64	Итеративная	615	85	0.544
DDP-64	Конвейерная	3 440	95	6.100
CIKS-1 [14]	Итеративная	907	81	0.648
CIKS-1 [14]	Конвейерная	6 346	81	5.184
SPECTR-H64 [14]	Итеративная	713	83	0.443
SPECTR-H64 [14]	Конвейерная	7 021	83	5.312
AES [117]	Итеративная	2 358	22	0.259
AES [117]	Конвейерная	17 314	28.5	3.650
IDEA [119]	Итеративная	2 878	150	0.600
DES [119]	Итеративная	722	11	0.181

Шифр DDP-64 имеет меньшие затраты аппаратных ресурсов по сравнению с криптосистемами CIKS-1 и SPECTR-H64 для всех вариантов реализации, что обеспечивается особенностями построения раунда шифрования. По производительности он во всех случаях превосходит криптосистему SPECTR-H64. Только CIKS-1 в итеративной архитектуре несколько превосходит шифр DDP-64 по производительности. Тем не менее, все три шифра основанных на переменных перестановках, обладают близкими параметрами и дают представление об эффективности аппаратной реализации подобных шифров.

Таким образом, относительно шифра DDP-64 можно сделать следующие выводы:

1. Шифр DDP-64 имеет высокую производительность при недорогой аппаратной реализации;
2. Структура DDP-64 очень хорошо подходит для выполнения детального дифференциального анализа, в частности для вычисления дифференциальных характеристик с малым числом активных битов, которые имеют наиболее высокие значения вероятности.
3. Этот шифр стоек к дифференциальным, линейным и другим атакам.
4. Криптосистема DDP-64 представляет собой пример шифров, основанных только на УП, иллюстрирующих высокую эффективность УП как криптографического примитива.

8.5.3. Дифференциальный криптоанализ шифра COBRA-H64

Также как и в случае ряда других шифров, основанных на перестановках, зависящих от преобразуемых данных, в криптосистеме COBRA-H64 переменные битовые перестановки, выполняемые над правым подблоком данных, определяют то обстоятельство, что наибольшую вероятность имеют дифференциальные характеристики с разностями, имеющими малый вес. При увеличении веса разности вероятность характеристики существенно уменьшается, что связано с попаданием активных битов в произвольные разряды при прохождении разности через правую ветвь. Рассматриваемый ниже дифференциальный анализ является эмпирическим. В его основе лежит изучение различных разностей, имеющих вес от 1 до 6. Несмотря на то, что наблюдается закономерность существенного уменьшения вероятности с ростом веса разности, для полного завершения дифференциального анализа требуется выполнение обобщенного теоретического исследования, позволяющего получить формальное доказательство того, что не существует характеристик с вероятностью, превышающей некоторое заданное значение. В общем случае такая задача представляется достаточно трудоемкой даже для малого числа раундов. Эмпирический анализ имеет значение как один из этапов анализа стойкости. Это замечание также относится и к анализу, выполненному для других шифров и приводимому в следующих разделах.

Разности, соответствующие левому и правому подблокам данных, обозначим как Δ^L и Δ^R. Разность итеративной дифференциальной характеристики имеет вид (Δ^L, Δ^R).

При рассмотрении некоторой отдельной операции **F** относящиеся к ней входная и выходная разности будут обозначены как Δ^{F} и $\Delta^{(\mathrm{F})}$, соответственно. Число активных битов разности и разряды, которым они принадлежат, будем указывать в нижнем регистре. Значения разрядов будем записывать после вертикальной черты. При этом будем полагать, что записи $\Delta_{2|i,j}$ и Δ_2 являются существенно различными: первая обозначает конкретную разность с двумя активными битами, а вторая – одну из класса разностей с двумя активными битами.

Механизм формирования дифференциальной характеристики определяется свойствами операций, выполняемых в раундовом преобразовании, и структурой последнего. Дифференциальные характеристики элементарного блока $\mathbf{P}_{2/1}$ и прохождение разностей малого веса через БУП рассмотрены детально в книге [14] (см. с. 423-428). Используя их, легко вычислить характеристики БУП $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}{}_{32/96}$, которые соответствуют малым значениям z в разности Δ_z^L, определяющей разность на управляющем входе БУП. Значения вероятностей различных характеристик представлены в табл. 8.16 и 8.17 , где Δ^X и Δ^Y – входная и выходная разность БУП.

Таблица 8.16

Значения вероятностей $p(\Delta^X \to \Delta^Y / \Delta_1^L)$ для блока $\mathbf{P}_{32/64}$

$z=1$	Δ_0^Y	Δ_2^Y	Δ_4^Y	Δ_6^Y
Δ_0^X	2^{-3}	$1.5{\cdot}2^{-2}$	$1.5{\cdot}2^{-2}$	2^{-3}
Δ_2^X	$1.5{\cdot}2^{-11}$	$1.1{\cdot}2^{-3}$	$1.4{\cdot}2^{-2}$	$1.1{\cdot}2^{-2}$

Таблица 8.17

Значения вероятностей $p(\Delta^X \to \Delta^Y / \Delta_2^L)$ для блока $\mathbf{P}_{32/64}$

$z=2$	Δ_0^Y	Δ_2^Y	Δ_4^Y	Δ_6^Y
Δ_0^X	2^{-6}	$1.5{\cdot}2^{-4}$	$1.9{\cdot}2^{-3}$	$1.2{\cdot}2^{-2}$
Δ_2^X	$1.5{\cdot}2^{-13}$	$1.4{\cdot}2^{-6}$	$1.7{\cdot}2^{-4}$	$1.7{\cdot}2^{-3}$

При выполнении операции **G** биты, принадлежащие разрядам 1, 2, …, 29, влияют на четыре выходных бита. При этом изменение входного бита в i-том разряде приводит к детерминированному изменению выходного бита, принадлежащего этому разряду, и вероятностному изменению выходных битов в разрядах $i + k$, где $k = 1, 2, 3$. В табл. 8.18 приведены формулы, описывающие зависимость изменения выходных битов при изменении входного бита x_i. Легко заметить, что разность $\Delta_{1|i}$ проходит без изменения операцию **G** с вероятностью 2^{-3} при $1 \le i \le 29$, 2^{-2} при $i = 30$, 2^{-1} при $i = 31$ и 1 при $i = 32$.

Таблица 8.18

Вероятности порождения активного бита в различных разрядах на выходе функции **G** при изменении *i*-го входного бита

Формула	Вероятность	Примечание
$\Delta y_i=\Delta l_i$	$p(\Delta y_i=1)=1$	$1 \le i \le 32$
$\Delta y_{i+1} = \Delta l_i(l_{i-1} \oplus l_{i-2} \oplus l_{i-1}b_{i+1})$	$p(\Delta y_{i+1}=1)=1/2$	$1 \le i \le 31$
$\Delta y_{i+2} = \Delta l_i(l_{i-1} \oplus l_{i+1} \oplus b_{i+1} \oplus l_{i+1}b_{i+1})$	$p(\Delta y_{i+2}=1)=1/2$	$1 \le i \le 30$
$\Delta y_{i+3} = \Delta l_i(l_{i+1} \oplus l_{i+2} \oplus a_{i+2})$	$p(\Delta y_{i+3}=1)=1/2$	$1 \le i \le 29$

Мы нашли, что наиболее эффективными являются двухраундовые итеративные дифференциальные характеристики с разностью $(0,\Delta_1^R)$ или $(\Delta_1^L,0)$. Механизмы их формирования идентичны, поэтому рассмотрим прохождение через два раунда только первой разности. Поскольку мы не указываем конкретного номера разряда, которому принадлежит активный бит, то имеется в виду одна из существующих однобитовых разностей в правом подблоке данных.

Пусть, например, в первом раунде через правую ветвь распространяется разность Δ_1^R. С вероятностью $p(i) = 2^{-5}$ на выходе БУП $\mathbf{P}^{-1}_{32/96}$ мы будем иметь разность $\Delta_{1|i}^R$, которая после перестановки подблоков данных преобразуется в разность $\Delta_{1|i}^L$. Последняя, распространяясь через левую ветвь криптосхемы, вносит не менее двух активных битов в правую ветвь за счет выполнения двух операций **G** и наложения их выходных значений на правый блок с помощью операции суммирования по модулю два. Каждая из них с вероятностью 2^{-3} вносит только один бит, а с вероятностью 2^{-6} в правую ветвь вносятся только два активных бита. Основной вклад в формирование двухраундовой характеристики вносят следующие два случая.

Событие A

- Верхняя операция **G** порождает на своем выходе только один активный бит с вероятностью $p_1 = 2^{-3}$.
- Нижняя операция **G** порождает на своем выходе только один активный бит с вероятностью $p_1 = 2^{-3}$.
- Благодаря наличию разности управляющего вектора, порожденного разностью $\Delta_{1|i}^L$, операционный блок $\mathbf{P}_{32/96}$ формирует на своем выходе разность $\Delta_{2|i,j'}^R$, где $j' = \pi^{(e \oplus 1)}(\mathbf{I}(i))$, с вероятностью p_3, зависящей от i.
- Разность $\Delta_{2|i,j'}^R$ после суммирования с активными битами, формируемыми на выходах операций **G**, преобразуется в нулевую разность, которая проходит БУП $\mathbf{P}^{-1}_{32/96}$ без изменения с вероятностью $p_4 = 2^{-3}$.

Событие B

- Разность Δ_0^R проходит БУП $\mathbf{P}_{32/96}$ без изменения с вероятностью $p_3 = 2^{-3}$.
- Верхняя операция **G** порождает на своем выходе только один активный бит с вероятностью $p_1 = 2^{-3}$.
- Нижняя операция **G** порождает на своем выходе только один активный бит с вероятностью $p_2 = 2^{-3}$.
- Благодаря наличию разности на управляющем входе блока $\mathbf{P}^{-1}_{32/96}$, последний обнуляет оба активных бита разности $\Delta^R_{2|i',j}$, где $i' = \pi^{(e)}(i)$, поступающей на его вход, с вероятностью p_3, зависящей от i.

Вкладом других механизмов в вероятность двухраундовой характеристики $P(2)$ можно пренебречь, поскольку он значительно меньше вклада событий A и B, которые обозначим как P' и P''. Таким образом, имеем $P(2) \approx P' + P''$. Для расчета вероятностей P' и P'' требуется учесть распределение управляющих битов по обоим БУП, присутствующим в правой ветви. При этом различные разряды левого подблока, которым принадлежит активный бит разности $\Delta^L_{1|i}$, вносят различный вклад. Рассмотрение указанных двух событий является аналогичным. Отличие состоит только в следующем. В событии A рассматривается вероятность попадания двух активных битов, порожденных активным переключающим элементом (т. е. элементом, к которому приложен активный бит разности управляющего вектора) блока $\mathbf{P}_{32/96}$, в разряды j' и i. В событии B рассматривается вероятность попадания активных битов, присутствующих в разрядах j и $i' = \pi^{(e)}(i)$ на входе нижнего БУП, на вход одного и того же активного переключающего элемента блока $\mathbf{P}^{-1}_{32/96}$. Ввиду симметрии топологической структуры блоков $\mathbf{P}_{32/96}$ и $\mathbf{P}^{-1}_{32/96}$, рассмотрение события B можно свести к рассмотрению события A, в котором операционный блок $\mathbf{P}_{32/96}$ формирует на своем выходе разность $\Delta^R_{2|i',j}$ при наличии разности $\Delta^L_{1|j}$ в левой ветви.

Рассмотрим нахождение вероятности P' с использованием схемы формирования двухраундовой характеристики, показанной на рис. 8.21. Разность левого подблока $\Delta^L_{1|i}$, являющаяся одновременно и входной разностью перестановочной инволюции **I**, преобразуется после операции **I** в разность $\Delta^L_{1|j}$. В связи с этим верхняя операция **G** вносит в правую ветвь активный бит, принадлежащий i-ому разряду, а нижняя – активный бит, принадлежащий j-ому разряду.

Вероятность того, что верхняя (нижняя) операция **G** формирует на своем выходе только один активный бит разности, равна величине p_1 (p_2), которая зависит от i (j). Один из трех активных переключающих элементов верхнего БУП может породить пару активных битов, которые при движении к выходу могут быть перемещены в i-ый и j'-ый разряды.

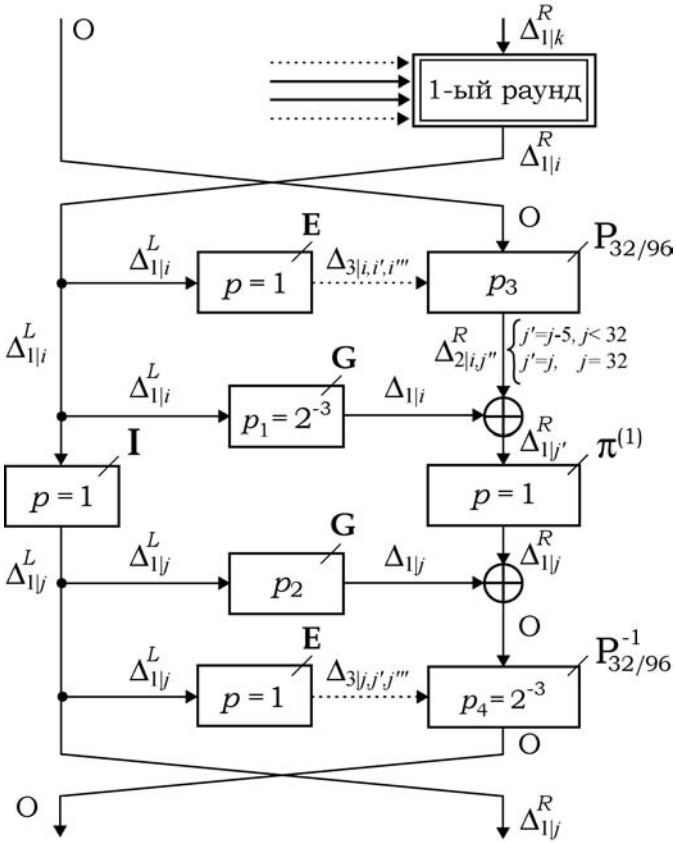

Рис. 8.21. Двухраундовая дифференциальная характеристика $0|\delta_{1,k}$ шифра COBRA-H64

Вероятность этого события (p_3) зависит от значения i. В результате суммирования с активным битом, вносимым верхней операцией **G**, на входе переключаемой перестановки $\pi^{(e)}$ остается один j'-ый активный бит. Последний, после выполнения операции $\pi^{(e)}$, попадает в разряд с номером $\pi^{(e)}(j') = \pi^{(e)}(\pi^{(e \oplus 1)}(j)) = j$, где он аннигилирует совместно с активным битом, вносимым нижней операцией **G**. В результате на входе нижнего БУП формируется нулевая разность, которая проходит его с вероятностью $p_4 = 2^{-3}$. Значения вероятностей p_1, p_2, p_3 и p_4 приведены в таблице 8.18. Отметим особенность случая $i = 13$, для которого активные биты, порождаемые обеими операциями **G,** могут взаимно уничтожаться. Благодаря последнему обстоятельству в случае $i = 13$ следующие выходные разности верхнего БУП вносят существенный вклад в вероятность P': $\Delta^R_{2|13,15}$, $\Delta^R_{2|13,16}$, $\Delta^R_{2|13,17}$ и $\Delta^R_{2|13,18}$. Таблица 8.18 дает для этого случая некоторые усредненные интегральные значения величин p_1, p_2 и p_3. Вероятность P' может быть рассчитана по следующей формуле:

$$P' \approx \sum_{i=1}^{i=32} p(i)p_1p_2p_3p_4 = p(i)p_4 \sum_{i=1}^{i=32} p_1p_2p_3 \approx 1.33\cdot 2^{-20},$$

где первый знак приближенного равенства учитывает пренебрежение слабой зависимости между событиями, к которым относятся вероятности p_3 и p_4. Значения вероятности p_3 рассчитываются с учетом распределения битов управляющего левого подблока данных по элементарным переключателям верхнего БУП, т. е. с учетом структуры блока расширения **E**. Например, в случае $i = 5$ мы имеем $j = 18$ и $j' = 15$, и активными являются переключающие элементы с номерами 5, 31 и 41.

Следующие три случая генерации верхним БУП разности $\Delta^R_{2|5,13}$ должны быть приняты во внимание:

1. В зависимости от значения правого подблока данных R пятый переключающий элемент с вероятностью 0.5 формирует на своем выходе пару активных битов (это происходит в случае, когда два его входных бита различны), с вероятностью 0.5 элемент 31 формирует на своем выходе нулевую разность и с вероятностью 0.5 на выходе элемента 41 появляется нулевая разность (это имеет место, когда оба входных бита соответствующего элемента равны).
2. Пятый и сорок первый переключающие элементы с вероятностью 0.5 формируют на своем выходе нулевую разность, а тридцать первый – два активных бита.
3. Пятый и тридцать первый переключающие элементы с вероятностью 0.5 формируют на своем выходе нулевую разность, а сорок первый – два активных бита.

Поскольку, с учетом структуры блока **E**, каждый бит подблока L управляет перестановкой шести различных битов подблока R, то вероятность каждого из указанных выше трех событий равна строго 2^{-3}. Один из выходных битов элемента 5 с вероятностью 2^{-5} (он проходит пять активных слоев блока $\mathbf{P}_{32/96}$) попадает в пятый разряд на выходе операции $\mathbf{P}_{32/96}$, с такой же вероятностью второй выходной бит элемента 5 попадает в тринадцатый разряд. В результате получаем, что с вероятностью $p^{(1)} = 2^{-3}(2^{-5})^2 = 2^{-13}$ первое событие приводит к формированию разности $\Delta^R_{2|5,13}$.

Второе событие не может привести к формированию такой разности на выходе блока $\mathbf{P}_{32/96}$, т. е. $p^{(2)} = 0$. Для третьего события получаем: $p^{(3)} = 2^{-3}(2^{-3})^2 = 2^{-9}$. Таким образом, для $i = 5$ получаем $p_3 = p^{(1)} + p^{(2)} + p^{(3)} \approx 1.06\cdot 2^{-9}$. Аналогичным способом вычисляются значения вероятности для всех других значений i. Заметим, что случаи со значениями $17 \le i \le 32$ вносят нулевой вклад в вероятность P'.

Для случая B можно аналогичным путем получить значение вероятности $P'' \approx 2^{-20}$, а затем вычислить значение

$$P(2) \approx P' + P'' \approx 1.33\cdot 2^{-20} + 2^{-20} \approx 1.16\cdot 2^{-19}.$$

Для восьми и десяти раундов шифра Cobra-H128 получаем $P(8) = P^4(2) \approx 1.82 \cdot 2^{-76}$ и $P(10) = P^5(2) \approx 1.05 \cdot 2^{-94}$.

Учитывая, что для случайного преобразования на выходе формируется однобитовая разность $(0, \Delta_1^R)$ с вероятностью $P = 32 \cdot 2^{-64} = 2^{-59} > P(8) > P(10)$, можно сделать заключение, что шифр COBRA-H64 является стойким к дифференциальному криптоанализу, поскольку он неотличим от случайного преобразования с использованием характеристик с наибольшей вероятностью.

Использование переключаемой операции $\pi^{(e)}$ устраняет слабые и полуслабые ключи, что делает более безопасным применение простого расписания ключа. Если использовать различные значения параметра e в различных раундах шифрования, то это может обеспечить снятие периодичности процедуры шифрования, а, следовательно, и защиту от нападений типа «slide attacks» даже в случае использования одинаковых раундовых ключей во всех раундах, что может иметь место в случае секретных ключей, имеющих структуру типа $K = (X, X, \ldots, X)$.

8.5.4. Дифференциальный криптоанализ шифра COBRA-H128

Шифр COBRA-H128 сходен по своей структуре с алгоритмом COBRA-H64. Это прослеживается также в дифференциальных свойствах этих шифров. Для шифра COBRA-H128 дифференциальные характеристики с малым числом активных битов также имеют более высокие вероятности по сравнению с характеристиками, включающими разности большого веса. Наибольшую вероятность имеет характеристика, соответствующая прохождению разности $(0, \Delta_1^R)$ через два раунда. Механизм ее прохождения через процедуру шифрования аналогичен случаю шифра COBRA-H64. Отличие состоит в различии размерностей использованных БУП и операции **G**.

Первый раунд активный бит проходит с вероятностью 1. При этом он перемещается в другой разряд, но в случае, когда в обозначении разности не указываются номера разрядов, нас будет интересовать любая выходная разность с заданным числом активных битов (независимо от номеров разрядов, которым принадлежат активные биты). Это означает, что в указанном случае мы будем рассматривать не одну разность, а совокупности всех разностей с заданным числом активных битов. При изменении только одного входного бита операции **G** на ее выходе обязательно изменится только один бит, соответствующий тому же разряду, что и измененный входной бит. Кроме того, с вероятностью 0.5 могут измениться еще шесть последующих выходных битов. Основные различные варианты формирования двухраундовой характеристики связаны с рассмотрением появления на выходе операции **G** разностей $\Delta^{(G)}_{2|i,i+1}$, $\Delta^{(G)}_{2|i,i+2}$, $\Delta^{(G)}_{2|i,i+3}$, $\Delta^{(G)}_{2|i,i+4}$, $\Delta^{(G)}_{2|i,i+5}$ и $\Delta^{(G)}_{2|i,i+6}$ (для значений $i > 58$ индексы $i + k$ имеют значения больше 64; в качестве таких индексов следует взять значение $i + k - 64$). Используя задание операции **G** через булевы функции, легко записать формулы, описывающие изменение выходных битов в разряде $i+k$, где k=0, 1, …, 6 (см. табл. 8.19.).

Таблица 8.19

Вероятности порождения активного бита на выходе функции **G** при изменении *i*-го входного бита

Формула	**Вероятность**
$\Delta y_i=\Delta l_i$	$p(\Delta y_i=1)=1$
$\Delta y_{i+1} = \Delta l_i(a''_{i+1} \oplus l_{i-3}(l_{i-5} \oplus l_{i-1}a''_i))$	$p(\Delta y_{i+1}=1)=1/2$
$\Delta y_{i+2} = \Delta l_i(l_{i-3} \oplus a''_{i+1}(l_{i+1}l_{i-2} \oplus l_{i-4}))$	$p(\Delta y_{i+2}=1)=1/2$
$\Delta y_{i+3} = \Delta l_i\, l_{i-1}$	$p(\Delta y_{i+3}=1)=1/2$
$\Delta y_{i+4} = \Delta l_i(l_{i+1} \oplus l_{i+3}(l_{i-2} \oplus l_{i+2}a''_{i+3}))$	$p(\Delta y_{i+4}=1)=1/2$
$\Delta y_{i+5} = \Delta l_i\, l_{i+3}$	$p(\Delta y_{i+5}=1)=1/2$
$\Delta y_{i+6} = \Delta l_i(a'_{i+5} \oplus l_{i+5}l_{i+2} \oplus l_{i+4}a''_{i+5})$	$p(\Delta y_{i+6}=1)=1/2$

Ниже описываются события A1 и A2, вносящие наибольший вклад в вероятность двухраундовой характеристики шифра COBRA-H128. Рассмотрим механизм формирования двухраундовой характеристики с разностью $(0, \Delta_1^R)$, представленный на рис. 8.22. Разность Δ_1^R после выполнения первого раунда преобразуется в разность $\Delta_{1|i}^R$ с вероятностью $p' = 2^{-6}$, а после транспозиции подблоков данных разность преобразуется в разность $\Delta_{1|i}^L$, т. е. на вход второго раунда с вероятностью $p = 2^{-6}$ поступает разность $(\Delta_{1|i}^L, 0)$. Активный бит из левой ветви проходит через верхнюю операцию **G** с вероятностью 2^{-6}, порождая на выходе этой операции разность $\Delta_{1|i}^G$. Активный бит из левой ветви проходит также через нижнюю операцию **G**, но перед этим он проходит через фиксированную перестановку П, преобразуясь в разность $\Delta_{1|j}$, где $j = \Pi(i)$. Разность $\Delta_{1|j}$ проходит через нижнюю операцию **G** также с вероятностью 2^{-6}. При этом, благодаря перестановке П, при выполнении операций **G** различные биты левого подблока влияют на порождение новых активных битов в разрядах $i + k$ (для верхней операции **G**) и разрядах $j + k$ (для нижней операции **G**), где k=0, 1, …, 6. Таким образом, рассмотренные два события являются независимыми. Разности $\Delta_{1|i}^G$ и $\Delta_{1|j}^G$ накладываются на правый подблок. Если в верхнем БУП будет порождена пара активных битов и они будут перемещены в разряды i и j, то, накладываясь на единичные биты, вносимые из левой ветви криптосхемы, они обнулят последние, формируя на входе нижнего БУП нулевую разность. Порождение активных битов в БУП может произойти, поскольку активный бит левого подблока порождает три единичных разности управляющего вектора V, т. е. на управляющем входе БУП имеем разность Δ_3^V. Если нулевая разность правого подблока проходит верхний БУП, то два активных бита, вносимые из левой ветви, могут взаимо уничтожиться в

нижнем БУП, если они попадут одновременно на один и тот же элементарный переключатель, на управляющий вход которого попадает единичный бит управляющего вектора.

Имеются и другие механизмы формирования двухраундовой характеристики с разностью $(0, \Delta_1^R)$, но они вносят существенно меньший вклад в ее вероятность. Мы имеем следующие два существенных события.

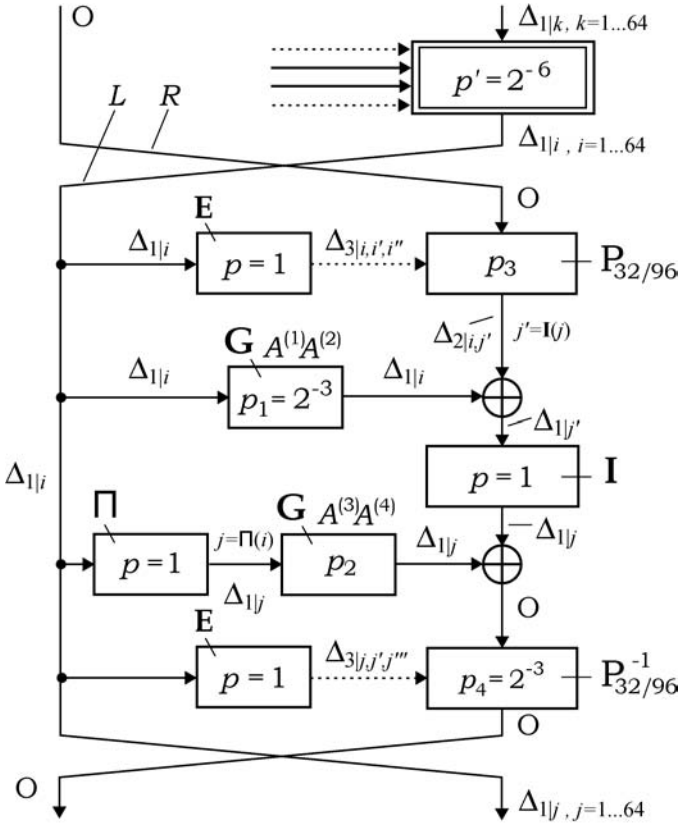

Рис. 8.22. Двухраундовая дифференциальная характеристика $0|\delta_{1,k}$ шифра COBRA-H128

Событие A1 (см. рис. 8.22):

- с вероятностью $p_1 = 2^{-6}$ на выходе верхней операции **G** формируется разность $\Delta^{\mathrm{G}}_{1|i}$;
- с вероятностью $p_2 = 2^{-6}$ на выходе нижней операции **G** формируется разность $\Delta^{\mathrm{G}}_{1|j}$;

- с вероятностью $p_3^{(i,j')}$ на выходе верхнего БУП ($\mathbf{P}_{64/192}$) формируется разность $\Delta^{P'}_{2|i,j'}$, где $j' = \mathbf{I}(j)$;
- на входе нижнего БУП ($\mathbf{P}^{-1}_{64/192}$) формируется разность $\Delta^{G}_{1|i} \oplus \Delta^{P'}_{2|i,j'} \oplus \Delta^{G}_{1|i} = \Delta_0$;
- с вероятностью $p_4 = 2^{-3}$ нулевая разность проходит нижний БУП.

Событие A2:

- с вероятностью $p_1 = 2^{-6}$ на выходе верхней операции **G** формируется разность $\Delta^{G}_{1|i}$, которая ,проходя операцию I, превращается в разность $\Delta^{G}_{1|i}$, где $i' = \mathrm{I}(i)$;
- с вероятностью $p_3 = 2^{-3}$ нулевая разность проходит верхний БУП;
- с вероятностью $p_2 = 2^{-6}$ на выходе нижней операции **G** формируется разность $\Delta^{G}_{1|j}$;
- с вероятностью $p_4^{(i',j)}$ разность $\Delta^{R}_{2|i',j}$ обнуляется при прохождении через нижний БУП.

Расчет вероятностей $p_3^{(i,j')}$ и $p_4^{(i',j)}$ требует рассмотрения структуры блока расширения и топологии БУП. Легко могут быть получены данные, представленные в табл. 8.20.

Таблица 8.20

Вероятности элементарных событий случая A1

i	p_1	p_2	$p_3^{(i,j')}$	p_4	$P^{(i)}$	p'
$i = i^* \in \{1,\ldots,5,10,\ldots,14,$ $20,21,22,23,29,\ldots,32$	2^{-6}	2^{-6}	2^{-13}	2^{-3}	2^{-28}	2^{-6}
$\forall i \neq i^*$	2^{-6}	2^{-6}	0	2^{-3}	0	2^{-6}

Вероятность события A1, при принадлежности активного бита левого подблока i-му разряду, обозначим как $P^{(i)} = p_1 p_2 p_3^{(i,j')} p_4$. Учитывая зависимость вероятности $P^{(i)}$ от номера i, получаем вклад события A1 в вероятность двухраундовой характеристики:

$$P' = p' \sum_{i=1}^{i=64} P^{(i)} = p' p_1 p_2 p_4 \sum_{i=1}^{i=64} p_3^{(i,j')} \approx 1.125 \cdot 2^{-30},$$

где $p' = 2^{-6}$– вероятность того, что после первого раунда активный бит разности попадает в i-ый разряд. События A1 и A2 являются симметричными. Расчет вклада второго события $P'' = 1.125 \cdot 2^{-30}$ осуществляется аналогичным образом. Для вероятности двухраундовой характеристики получаем следующее значение:

$$P(2) \approx P' + P'' \approx 1.125 \cdot 2^{-29}.$$

Вероятность прохождение разностей с одним активным битом через полное число раундов равно $P(12) = P^6(2) \approx 2^{-173}$. Для десяти раундов шифра COBRA-H128 получаем $P(10) = P^5(2) \approx 2^{-144}$. Учитывая, что для случайного преобразования на выходе формируется однобитовая разность $(0, \Delta_1^R)$ с вероятностью $P = 64 \cdot 2^{-128} = 2^{-122}$ > $P(10)$ > $P(12)$, делаем вывод, что COBRA-H128 является стойким к рассмотренной атаке, поскольку по отношению к ней этот шифр неотличим от случайного преобразования.

Использование переключаемой операции $\Pi^{(e)}$ устраняет слабые и полуслабые ключи, что делает более безопасным применение простого расписания ключа. Применение различных значений параметра e в различных раундах зашифрования (и расшифрования) обеспечивает снятие периодичностей и защиту от нападений типа «slide attacks» в случае использования одинаковых раундовых ключей во всех раундах, а значит — и в случае секретных ключей, имеющих структуру типа $K = (X, X, \ldots, X)$.

8.5.5. Интегральные статистические оценки шифров SG-128, SS-128

При оценивании статистических свойств алгоритмов шифрования на основе УППС $\mathbf{F}_{n/m}$ определялись следующие параметры [106]:

- среднее число битов выходного вектора после одного раунда преобразования, изменяющихся при изменении одного бита открытого текста – d_1;
- степень полноты преобразования – d_2;
- степень лавинного эффекта – d_3;
- степень соответствия строгому лавинному критерию – d_4.

Использование данных показателей позволяет выявить возможные слабости алгоритма шифрования к разностному методу криптоанализа. Для их определения формируются две матрицы: матрица зависимостей $\|\mathbf{A}\|$ и матрица расстояний $\|\mathbf{B}\|$, состоящие из элементов a_{ij}, b_{ij}:

$$a_{ij} = \#\{X \in \mathbf{X} \mid (\mathbf{E}(X^{(i)}))_j \neq (\mathbf{E}(X))_j\},$$

$$b_{ij} = \#\{X \in \mathbf{X} \mid \boldsymbol{wt}(\mathbf{E}(X^{(i)}) \oplus \mathbf{E}(X)) = j\},$$

где $\mathbf{E}(\bullet)$ – функция зашифрования, $\boldsymbol{wt}(X)$ – вес Хемминга вектора X, а $X^{(i)}$ – вектор, полученный в результате инвертирования i-го бита вектора X. Значения показателей $d_1 – d_4$ вычислялись следующим образом:

$$d_1 = \frac{1}{n} \sum\nolimits_{i=1}^{n} \frac{\sum_{j=1}^{m} j b_{ij}}{\#\mathbf{X}},$$

$$d_2 = 1 - \frac{\#\{(i, j) \mid a_{ij} = 0\}}{nm},$$

$$d_3 = 1 - \frac{\sum_{i=1}^{n} \left| \frac{1}{\#\mathbf{X}} \sum_{j=1}^{m} 2\, jb_{ij} - m \right|}{nm},$$

$$d_4 = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \left| \frac{2a_{ij}}{\#\mathbf{X}} - 1 \right|}{nm},$$

где #X – количество исследуемых открытых текстов, генерируемых датчиком случайных чисел. В проведенном тестировании выбиралось #X = 10000, количество раундов шифрования $R = 12$.

Результаты тестирования для алгоритма SG-128 представлены в табл. 8.21.

Таблица 8.21

R	d_1	d_2	d_3	d_4
1	25.533	0.503784	0.398954	0.388307
2	57.020	0.999878	0.890936	0.884261
3	63.984	1	0.999302	0.991924
4	63.994	1	0.994801	0.992000
5	63.997	1	0.999332	0.992090
6	63.998	1	0.999324	0.992077
7	64.001	1	0.999280	0.992095
8	64.000	1	0.999224	0.992121
9	63.999	1	0.999434	0.992072
10	63.998	1	0.999324	0.992089
11	64.000	1	0.999233	0.991935
12	64.001	1	0.999336	0.991909

Таким образом, анализ результатов исследования показал, что алгоритмы шифрования информации на основе блоков УППС $\mathbf{F}_{n/m}$ обеспечивают достаточно сильное рассеивающее влияние каждого из битов открытого текста на все биты преобразованного текста. В целом, для всех раундов, начиная с четвертого, значения параметров соответственно равны $d_1 \approx 0.5n$, $d_2 = 1$, $d_3 \approx 1$, $d_4 \approx 1$.

8.5.6. Оценивание стойкости шифра SG-128 к дифференциальному методу анализа

При исследовании стойкости описанного в п.п. 8.4 шифра, основанного на УППС, и векторной БФ к дифференциальному (разностному) анализу будем исхо-

дить из гипотезы, что наибольшими вероятностями обладают характеристики, включающие разности с малым числом (весом) активных битов. Введем обозначения: w – вес разности на выходе $\mathbf{F}_{64/192}$, ζ – вес разности после подстановочного преобразования $\mathbf{G}$. Вероятность s-раундовой характеристики $p^*(s)$ можно определить путем оценивания вероятностей прохождения разностей через два раунда преобразования и рассмотрения вклада каждого примитива преобразования, используемого в алгоритме, в данные вероятности: $p^*(2) = p^{(1)} p^{(2)}$, в общем случае

$$p^{(1)} = \Pr\left\{\bigcup_{w=1}^{n}\left((\Delta_1^x \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_w^y) \cap (\Delta_w^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_1^y)\right)\right\}, \tag{8.1}$$

$$\begin{aligned} p^{(2)} = \Pr\Bigg\{\bigcup_{w=0}^{n}\Bigg[&\left((\Delta_0^x \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_w^y) \bigcap_{\zeta=1}^{9} (\Delta_1^x \overset{\mathbf{G}}{\longrightarrow} \Delta_\zeta^y) \bigcap_{w \ge \zeta} (\Delta_{w-\zeta}^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_0^y)\right) \bigcup \\ &\left((\Delta_0^x \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_w^y) \bigcap_{\zeta=1}^{9} (\Delta_1^x \overset{\mathbf{G}}{\longrightarrow} \Delta_\zeta^y) \bigcap_{w+\zeta \le n} (\Delta_{w+\zeta}^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_0^y)\right)\Bigg]\Bigg\}, \end{aligned} \tag{8.2}$$

где обозначение $\Delta_i^x \overset{F}{\longrightarrow} \Delta_j^y$ означает событие, при котором разность с весом Хэмминга, равным i во входном векторе x, после выполнения преобразования F переходит в разность веса j в выходном векторе y. Значения вероятностей прохождения разностей через элементарные блоки $\mathbf{F}_{2/1}$ могут быть вычислены из табл. 8.22, 8.23, которые описывают прохождение разностей через вход данных (табл. 8.22) и управляющий вход (табл. 8.23) блока $\mathbf{F}_{2/1}$.

Из описания алгоритма и структуры раундовой функции ***Crypt*** (рис. 8.19) следует, что на первых и последних двух слоях $\mathbf{F}_{64/192}$ и $\mathbf{F}_{64/192}^{-1}$, соответственно, векторы управления идентичны: $W_1^{(1)} = W_1^{(2)}$, $W_2^{(1)} = W_2^{(2)}$, поэтому вероятности прохождения единичной разности из i-го входного разряда в i-й выходной разряд: $(i) \to (i)$, $i \in \{0, 2, \ldots, n-2\}$ и $(i') \to ((i'-2) \bmod n)$, $i' \in \{1, 3, \ldots, n-1\}$ через каждую пару слоев равна вероятности прохождения единичной разности через один слой блока $\mathbf{F}_{64/192}$, т.е. с учетом того, что $p(\Delta_1^x \overset{\sigma_1}{\longrightarrow} \Delta_1^y) = p(\Delta_1^x \overset{\sigma_2}{\longrightarrow} \Delta_1^y) = \Pr(\Delta_1^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y) = \frac{1}{2}$ (см. табл. 8.22), получим

$$p_1 = \Pr\left\{(\Delta_1^x \overset{\sigma_1}{\longrightarrow} \Delta_1^y) \cap (\Delta_1^x \overset{\sigma_1^{-1}}{\longrightarrow} \Delta_1^y)\right\} = \frac{1}{2}, \quad p_2 = \Pr\left\{(\Delta_1^x \overset{\sigma_2}{\longrightarrow} \Delta_1^y) \cap (\Delta_1^x \overset{\sigma_2^{-1}}{\longrightarrow} \Delta_1^y)\right\} = \frac{1}{2}.$$

Таблица 8.22

Прохождение разностей через вход данных блока $\mathbf{F}_{2/1}$

x_3	Δx_1	Δx_2	Δx_3	Δy_1	Δy_2
0	0	1	0	0	1
0	1	0	0	1	1
0	1	0	0	1	1
0	0	1	0	0	1
1	0	1	0	1	1
1	1	0	0	1	0
1	1	0	0	1	0
1	0	1	0	1	1
0	1	1	0	1	0
0	1	1	0	1	0
1	1	1	0	0	1
1	1	1	0	0	1

Для всех остальных слоев блоков $\mathbf{F}_{64/192}$ и $\mathbf{F}^{-1}_{64/192}$ векторы $W_3^{(1)}$, $W_4^{(1)}$, $W_5^{(1)}$, $W_6^{(1)}$ и $W_4^{(2)}$, $W_3^{(2)}$, $W_2^{(2)}$, $W_1^{(2)}$ отличны друг от друга, поэтому вероятность прохождения единичной разности через данные слои равна: $p_3 = \Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y)^8 = 2^{-8}$.

В соответствии с (8.1) необходимо рассмотреть и другие возможные варианты появления единичной разности после первого раунда преобразования. Наиболее вероятным является событие $\left\{(\Delta_1^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_2^y) \bigcap (\Delta_2^x \xrightarrow{\mathbf{F}^{-1}_{64/192}} \Delta_1^y)\right\}$, причем такое, что разность веса $w = 2$ после прохождения слоя σ_6 появляется на выходе одного из блоков $\mathbf{F}_{2/1}$. Другие события исключим из рассмотрения, ввиду малой доли вклада в общую вероятность прохождения единичной разности. При таких ограничениях $\Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_2^y) = \Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y)^5 \Pr\left(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_2^y\right) = 2^{-6}$, где в соответствии с данными табл. 8.22 $\Pr\left(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_2^y\right) = \frac{1}{2}$. Разность веса $w = 2$ может быть преобразована в раз-

ность веса $w = 1$ на выходе $\mathbf{F}_{64/192}^{-1}$ только тогда, когда разность веса $w = 1$ появляется на выходе первого слоя $\mathbf{F}_{64/192}^{-1}$, следовательно, принимая во внимание $W_1^{(1)} = W_1^{(2)}$, $W_2^{(1)} = W_2^{(2)}$, получим

$$\Pr(\Delta_2^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_1^y) = \Pr(\Delta_2^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y) \Pr\left(\Delta_1^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y\right)^3 = 2^{-4},$$

где согласно табл. 8.22 и с учетом выполнения условия перехода разности в нужный разряд: $\Pr(\Delta_2^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y) = \frac{1}{2}$, откуда $p_4 = \Pr\{(\Delta_1^x \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_2^y) \bigcap (\Delta_2^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_1^y)\} = 2^{-10}$.

Вероятность перехода разности веса $w = 2$ в другой разряд также равна 1/2, однако появление данного события приводит к тому, что $\Pr(\Delta_2^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_1^y) = \Pr(\Delta_2^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y) \Pr\left(\Delta_1^x \overset{\mathbf{F}_{2/1}}{\longrightarrow} \Delta_1^y\right)^5 = 2^{-6}$, так как на последних двух слоях $W_1^{(1)} \neq W_1^{(2)}$, $W_2^{(1)} \neq W_2^{(2)}$, а $\Pr\{(\Delta_1^x \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_2^y) \bigcap (\Delta_2^x \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_1^y)\} = 2^{-12}$.

Следовательно, вероятность прохождения единичной разности при выполнении условия $(i) \to (i)$, $i \in \{0, 2, \ldots, n-2\}$ или $(i') \to ((i'-2) \bmod n)$, $i' \in \{1, 3, \ldots, n-1\}$:

$$p^{(1)} = (p_1 p_2 p_3 + p_4) = 2^{-9},$$

а вероятностями появления единичной разности в других разрядах можно пренебречь.

На втором раунде алгоритма шифрования разность веса $w = 1$ является входом для левой ветви преобразования, а веса $w = 0$ – для правой. Необходимо определить вероятность события, при котором на выходе правой ветви преобразования не появится ни одна разность с весом $w > 0$. В общем случае вероятность такого события определяется путем согласования событий, заключающихся в прохождении разностей различного веса через УППС $\mathbf{F}_{64/192}$, подстановочное отображение $\mathbf{G}$ и преобразовании данных разностей в нулевую на выходе $\mathbf{F}_{64/192}^{-1}$. В соответствии с (8.2) $p^{(2)}$ в общем виде определяется как

$$p^{(2)} = \sum_{w=0}^{n} \sum_{i=1}^{\mathbf{C}_n^w} \Pr_i(\Delta_0 \overset{\mathbf{F}_{64/192}}{\longrightarrow} \Delta_w) \sum_{\zeta=1}^{9} \sum_{j=1}^{\mathbf{C}_8^{\zeta-1}} \Pr_j(\Delta_1 \overset{\mathbf{G}}{\longrightarrow} \Delta_\zeta) \left[\sum_{\substack{k=1, \\ w \geq \zeta}}^{\mathbf{C}_n^{w-\zeta}} \Pr_k(\Delta_{w-\zeta} \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_0) + \sum_{\substack{k=1, \\ w+\zeta \leq n}}^{\mathbf{C}_n^{w+\zeta}} \Pr_k(\Delta_{w+\zeta} \overset{\mathbf{F}_{64/192}^{-1}}{\longrightarrow} \Delta_0) \right]. \quad (8.3)$$

Очевидно, наибольший вклад в суммарную вероятность $p^{(2)}$ вносят вероятности появления следующих событий:

- $\{(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_1^y) \bigcap (\Delta_1^x \xrightarrow{\mathbf{G}} \Delta_1^y) \bigcap (\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}^{-1}} \Delta_0^y)\}$;
- $\{(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_0^y) \bigcap (\Delta_1^x \xrightarrow{\mathbf{G}} \Delta_1^y) \bigcap (\Delta_1^x \xrightarrow{\mathbf{F}_{64/192}^{-1}} \Delta_0^y)\}$;

где $\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_0^y) = \Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}^{-1}} \Delta_0^y)$, $\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_1^y) = \Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{64/192}^{-1}} \Delta_0^y)$. При определении вероятностей $\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_1^y)$ и $\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}^{-1}} \Delta_0^y)$ следует учитывать равенства $W_1^{(1)} = W_1^{(2)}$, $W_2^{(1)} = W_2^{(2)}$. Тогда

$$\begin{aligned}\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_1^y) = \Pr\Bigg\{\Bigg[\Bigg[\Bigg((\Delta_0 \xrightarrow{\sigma_{1,5}} \Delta_0) \bigcap (\Delta_0 \xrightarrow{\sigma_4} \Delta_1) \bigcap (\Delta_1 \xrightarrow{\sigma_{5,6}} \Delta_1)\Bigg) \bigcup \\ \Bigg((\Delta_0 \xrightarrow{\sigma_{1,4}} \Delta_0) \bigcap (\Delta_0 \xrightarrow{\sigma_5} \Delta_1) \bigcap (\Delta_1 \xrightarrow{\sigma_6} \Delta_1)\Bigg) \Bigg| \; i \in \{0,\ 1,\ ...,\ 31\}\Bigg] \bigcup \\ \Bigg[\Bigg((\Delta_0 \xrightarrow{\sigma_{2,3}} \Delta_0) \bigcap (\Delta_0 \xrightarrow{\sigma_6} \Delta_1)\Bigg) \bigcup \Bigg((\Delta_0 \xrightarrow{\sigma_2} \Delta_0) \bigcap (\Delta_0 \xrightarrow{\sigma_3} \Delta_1) \bigcap \\ (\Delta_1 \xrightarrow{\sigma_{4-6}} \Delta_1) \bigcap (\Delta_0 \xrightarrow{\sigma_6} \Delta_0)\Bigg) \Bigg| \; i \in \{32,\ 33,\ ...,\ 63\}\Bigg]\Bigg\},\end{aligned} \tag{8.4}$$

$$\begin{aligned}\Pr(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_0^y) = \Pr\Bigg\{\Bigg((\Delta_0 \xrightarrow{\sigma_{1,4,5}} \Delta_0) \Bigg| \; i \in \{0,\ 1,\ ...,\ 31\}\Bigg) \bigcup \\ \Bigg((\Delta_0 \xrightarrow{\sigma_{2,3,6}} \Delta_0) \Bigg| \; i \in \{32,\ 33,\ ...,\ 63\}\Bigg)\Bigg\},\end{aligned} \tag{8.5}$$

где, в соответствии с данными табл. 8.23,

$$\Pr(\Delta_0 \xrightarrow{\sigma_j} \Delta_0) = \Pr(\Delta_0 \xrightarrow{\mathbf{F}_{2/1}} \Delta_0) = 2^{-2},\ \Pr(\Delta_0 \xrightarrow{\sigma_j} \Delta_1) = \Pr(\Delta_0 \xrightarrow{\mathbf{F}_{2/1}} \Delta_1) = 2^{-1},$$

а в соответствии с табл. 8.22

$$\Pr(\Delta_1 \xrightarrow{\sigma_j} \Delta_1) = \Pr(\Delta_1 \xrightarrow{\mathbf{F}_{2/1}} \Delta_1) = 2^{-1}.$$

Таблица 8.23

Прохождение разностей через управляющий вход блока $\mathbf{F}_{2/1}$

x_1	x_2	Δx_3	Δy_1	Δy_2
0	0	1	0	0
0	1	1	1	0
1	0	1	0	1
1	1	1	1	1

С учетом данных табл. 8.24, где представлены значения вероятностей прохождения разности веса $w = 1$ через подстановочное преобразование **G**, получим $\Pr(\Delta_1^x \xrightarrow{\mathbf{G}} \Delta_1^y) = 2^{-8}$.

Согласование разностей на выходах $\mathbf{F}_{64/192}$, **G** и $\mathbf{F}^{-1}_{64/192}$ показало, что только в 6-ти случаях из 64 разности попадают в нужные разряды, откуда, используя свойство симметричности алгоритма шифрования, получим оценку

$$p^{(2)} = 2\Pr\left\{(\Delta_0^x \xrightarrow{\mathbf{F}_{64/192}} \Delta_1^y) \bigcap (\Delta_1^x \xrightarrow{\mathbf{G}} \Delta_1^y) \bigcap (\Delta_0^x \xrightarrow{\mathbf{F}^{-1}_{64/192}} \Delta_0^y)\right\} \approx 2^{-23}.$$

Таким образом, вероятность прохождения разности с весом $w = 1$ оценивается значением $p^* = p^{(1)}p^{(2)} \approx 2^{-32}$. Для 10-ти раундов шифрования $p^*(10) = 2^{-160}$, следовательно, вычислительная сложность дифференциального анализа $W \geq 2^{160}$ шифрований 128-битовых блоков информации, что позволяет говорить о высокой стойкости предлагаемого алгоритма шифрования к дифференциальному методу анализа и делает возможным уменьшение количества раундов шифрования до уровня R=8, при котором стойкость к дифференциальному методу анализа не превышает стойкости к анализу на основе перебора всего ключевого множества.

Таблица 8.24

Вероятности появления разности различных разрядов на выходе функции **G** при изменении *i*–го входного бита

Появление разности в *i*-м разряде	**Вероятности**
$\Delta y_i = \Delta a_i$	$p(\Delta y_i = 1) = 1$
$\Delta y_{i+1} = \Delta a_i(q_{i-1}(a_{i-2}a_{i-4}a_{i-5}a_{i-7} \oplus a_{i-4}a_{i-7} \oplus a_{i-2} \oplus a_{i-5}) \oplus a_{i-2}a_{i-5}a_{i-7} \oplus a_{i-3})$	$p(\Delta y_{i+1} = 1) = 1/2$

Появление разности в *i*-м разряде	**Вероятности**
$\Delta y_{i+2} = \Delta a_i a_{i-6}$	$p(\Delta y_{i+2} = 1) = 1/2$
$\Delta y_{i+3} = \Delta a_i(q_{i+1}(a_{i+2}a_{i-2}a_{i-3}a_{i-5} \oplus a_{i-2}a_{i-3} \oplus a_{i+2}) \oplus a_{i+2}a_{i-3}a_{i-5} \oplus a_{i-2}a_{i-5} \oplus q_i)$	$p(\Delta y_{i+3} = 1) = 1/2$
$\Delta y_{i+4} = \Delta a_i a_{i+3}$	$p(\Delta y_{i+4} = 1) = 1/2$
$\Delta y_{i+5} = \Delta a_i(q_{i+3}(a_{i+4}a_{i+2}a_{i-1}a_{i-3} \oplus a_{i+4}a_{i-3} \oplus a_{i+2}a_{i-1}) \oplus a_{i+2}a_{i-3} \oplus a_{i-1}a_{i-3} \oplus q_{i+1})$	$p(\Delta y_{i+5} = 1) = 1/2$
$\Delta y_{i+6} = \Delta a_i(q_{i+4}(a_{i+5}a_{i+3}a_{i+1}a_{i-2} \oplus a_{i+3}a_{i+1} \oplus a_{i+5}) \oplus a_{i+5}a_{i+3}a_{i+2} \oplus a_{i+1}a_{i-2} \oplus q_{i+5})$	$p(\Delta y_{i+6} = 1) = 1/2$
$\Delta y_{i+7} = \Delta a_i q_{i+5}$	$p(\Delta y_{i+7} = 1) = 1/2$
$\Delta y_{i+8} = \Delta a_i(q_{i+6}(a_{i+7}a_{i+5}a_{i+3}a_{i+2} \oplus a_{i+7}a_{i+2}) \oplus a_{i+7}a_{i+5}a_{i+2} \oplus a_{i+5}a_{i+3} \oplus a_{i+3}a_{i+2} \oplus a_{i+6})$	$p(\Delta y_{i+8} = 1) = 1/2$

Осуществлен сравнительный анализ результатов аналитического оценивания с имитационным. При этом экспериментально определялась вероятность появления единичной разности в выходном векторе Y после двух раундов шифрования при условии появления единичной разности во входном векторе $X = A \| B$. В целях уменьшения объема вычислений моделирование проводилось для одного раунда шифрования при появлении единичной разности вначале в правом подблоке B, а затем в левом подблоке A. Результирующая вероятность появления разности веса $w = 1$ на выходе алгоритма после двух раундов шифрования вычислялась так же как и при аналитическом оценивании: $p^* = p^{(1)}p^{(2)}$.

Для первого варианта количество тестируемых текстов $\#X = 10^5$, для второго варианта $\#X = 10^6$. Причем в каждый из текстов последовательно вводилась единичная разность в разряд $i \in \{64, 65, \ldots, 128\}$ или $i \in \{0, 1, \ldots, 63\}$ вектора X, т. е. общий объем тестируемых текстов для первого правого подблока $\#X^* = 6.4 \cdot 10^6 \approx 2^{22.6}$, а для левого подблока $\#X^* = 6.4 \cdot 10^7 \approx 2^{26}$. Для $i \in \{64, 65, \ldots, 128\}$ – $p_1 = 2.308 \cdot 10^{-3} \approx 2^{-8.76}$, $i \in \{0, 1, \ldots, 63\}$ – $p_2 = 2.969 \cdot 10^{-7} \approx 2^{-22.68}$, следовательно, $p^* \approx 2^{-31.44}$, что подтверждает аналитические оценки вероятности появления единичной разности на выходе алгоритма шифрования после двух раундов.

8.5.7. Анализ стойкости алгоритма SS-128

Анализ стойкости данного алгоритма шифрования осуществляется аналогично исследованию стойкости алгоритма шифрования на основе УППС и векторной БФ.

Оценим вероятность прохождения разностей через два раунда преобразования и вклад каждого примитива преобразования, используемого в алгоритме, в данные вероятности: $p^*(2) = p^{(1)}p^{(2)}$, где

$$p^{(1)} = \Pr\left\{\bigcup_{w=1}^{n}\left((\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_w^y) \cap (\Delta_w^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_1^y)\right)\right\}, \tag{8.6}$$

$$p^{(2)} = \Pr\left\{\bigcup_{w=0}^{n}\left[\left((\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_w^y)\bigcap_{\varsigma=1}^{n}(\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_\varsigma^y)\bigcap_{\substack{\eta=0 \\ w\geq\varsigma}}^{\varsigma}(\Delta_{w-\varsigma+\eta}^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y)\right)\bigcup \right.\right.$$
$$\left.\left.\left((\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_w^y)\bigcap_{\varsigma=1}^{n}(\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_\varsigma^y)\bigcap_{\substack{\eta=0 \\ w+\varsigma\leq n}}^{\varsigma}(\Delta_{w+\varsigma-\eta}^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y)\right)\right]\right\}. \tag{8.7}$$

Из симметричности блоков $\mathbf{S}_1$ и $\mathbf{S}_1^{-1}$ следует, что $\Pr(\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_1^y) = \Pr(\Delta_1^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_1^y)$. Для выбранных БФ в блоке $\mathbf{F}_{2/1}$ $\Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y) = \frac{1}{2}$, следовательно, $\Pr(\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_1^y) = 2^{-6}$, откуда $p_1 = 2^{-12}$. Согласно (8.6) необходимо рассмотреть и другие возможные варианты появления единичной разности после первого раунда преобразования. Наиболее вероятным является событие $\{(\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_2^y) \cap (\Delta_2^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_1^y)\}$, причем такое, что разность веса $w = 2$ после прохождения последнего слоя блока S_1 появляется на выходе одного из блоков $\mathbf{F}_{2/1.}$ Другие события исключим из рассмотрения, ввиду малой доли вклада в общую вероятность прохождения единичной разности. При таких ограничениях $\Pr(\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_2^y) = \Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y)^5 \Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_2^y) = 2^{-6}$. Разность веса $w = 2$ может быть преобразована в разность веса $w = 1$ на выходе $\mathbf{S}_1^{-1}$ только тогда, когда разность веса $w = 1$ появляется на выходе первого слоя блока $\mathbf{S}_1^{-1}$, следовательно, $\Pr(\Delta_2^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_1^y) = \Pr(\Delta_2^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y)\Pr(\Delta_1^x \xrightarrow{\mathbf{F}_{2/1}} \Delta_1^y)^5 = 2^{-6}$, откуда $p_2 = \Pr\{(\Delta_1^x \xrightarrow{\mathbf{S}_1} \Delta_2^y) \cap (\Delta_2^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_1^y)\} = 2^{-12}$ Таким образом, вероятность прохождения единичной разности $p^{(1)} \approx 2^{-11}$.

На втором раунде алгоритма шифрования оценивается вероятность события, при котором на выходе правой ветви преобразования не появится ни одна разность с весом $w > 0$. Вероятность такого события определяется путем согласования событий, заключающихся в прохождении разностей различного веса через $\mathbf{S}_1$, $\mathbf{S}_2$, …, $\mathbf{S}_5$ и преобразовании данных разностей в нулевую на выходе $\mathbf{S}_1^{-1}$. Из выражения (8.7) следует, что наибольший вклад в суммарную вероятность $p^{(2)}$ вносят вероятности появления событий:

$$\{(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_1^y) \cap (\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_1^y) \cap (\Delta_0^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y)\} \text{ и } \{(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_0^y) \cap (\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_1^y) \cap (\Delta_1^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y)\}.$$

Вероятности $\Pr(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_1^y)$ и $\Pr(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_0^y)$ определяются из следующих соотношений:

$$\Pr(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_1^y) = \Pr(\Delta_1^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y) = \Pr\left\{\left[\left[\left((\Delta_0 \xrightarrow{\sigma_1} \Delta_1) \cap (\Delta_0 \xrightarrow{\sigma_{4,5}} \Delta_0) \cap (\Delta_1 \xrightarrow{\sigma_{2-6}} \Delta_1)\right) \cup \right.\right.\right.$$

$$\left.\left.\left((\Delta_0 \xrightarrow{\sigma_{1,5}} \Delta_0) \cap (\Delta_0 \xrightarrow{\sigma_4} \Delta_1) \cap (\Delta_1 \xrightarrow{\sigma_{5,6}} \Delta_1)\right) \cup \left((\Delta_0 \xrightarrow{\sigma_{1,4}} \Delta_0) \cap (\Delta_0 \xrightarrow{\sigma_5} \Delta_1) \cap (\Delta_1 \xrightarrow{\sigma_6} \Delta_1)\right)\right.$$

$$\left.\left. \mid i \in \{0,\ \ldots,\ 31\}\right] \cup \left[\left((\Delta_0 \xrightarrow{\sigma_2} \Delta_1) \cap (\Delta_1 \xrightarrow{\sigma_{3-6}} \Delta_1)(\Delta_0 \xrightarrow{\sigma_{3,6}} \Delta_0)\right) \cup \left((\Delta_0 \xrightarrow{\sigma_{2,3}} \Delta_0) \cap (\Delta_0 \xrightarrow{\sigma_6} \Delta_1)\right)\right.$$

$$\left.\left. \cup \left((\Delta_0 \xrightarrow{\sigma_2} \Delta_0) \cap (\Delta_0 \xrightarrow{\sigma_3} \Delta_1) \cap (\Delta_1 \xrightarrow{\sigma_{4-6}} \Delta_1) \cap (\Delta_0 \xrightarrow{\sigma_6} \Delta_0)\right) \mid i \in \{32,\ \ldots,\ 63\}\right]\right\},$$

$$\Pr(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_0^y) = \Pr(\Delta_0^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y) = \Pr\left\{\left((\Delta_0 \xrightarrow{\sigma_{1,4,5}} \Delta_0) \mid i \in \{0,\ldots,31\}\right) \cup \left((\Delta_0 \xrightarrow{\sigma_{2,3,6}} \Delta_0) \mid i \in \{32,\ldots,63\}\right)\right\},$$

где $\Pr(\Delta_0 \xrightarrow{\sigma_j} \Delta_0) = \Pr(\Delta_0 \xrightarrow{\mathbf{F}_{2/1}} \Delta_0) = 2^{-2}$,

$\Pr(\Delta_0 \xrightarrow{\sigma_j} \Delta_1) = \Pr(\Delta_0 \xrightarrow{\mathbf{F}_{2/1}} \Delta_1) = \frac{1}{2}$,

$\Pr(\Delta_1 \xrightarrow{\sigma_j} \Delta_1) = \Pr(\Delta_1 \xrightarrow{\mathbf{F}_{2/1}} \Delta_1) = \frac{1}{2}$.

Анализируя прохождение разности веса $w = 1$ через преобразования $\mathbf{S}_2$, $\mathbf{S}_3$, $\mathbf{S}_4$, $\mathbf{S}_5$, получим $\Pr(\Delta_1^x \xrightarrow{\mathbf{S}_2} \Delta_1^y) = \ldots = \Pr(\Delta_1^x \xrightarrow{\mathbf{S}_5} \Delta_1^y) = 2^{-4}$, однако следует учитывать, что разность веса $w = 1$ на информационных входах $\mathbf{S}_2$, $\mathbf{S}_3$ одновременно появляется на управляющих входах $\mathbf{S}_4$, $\mathbf{S}_5$ и, наоборот. Следовательно, событие $(\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_1^y)$ является объединением следующих групп пересечения независимых событий

$$\left\{\left[\left(\Delta_1^x \xrightarrow{\mathbf{S}_2,\mathbf{S}_3} \Delta_1^y\right) \cap \left(\Delta_1^v \xrightarrow{\mathbf{S}_4} \Delta_0^y\right) \cap \left(\Delta_1^v \xrightarrow{\mathbf{S}_5} \Delta_0^y\right)\right] \cup \left[\left(\Delta_1^x \xrightarrow{\mathbf{S}_4,\mathbf{S}_5} \Delta_1^y\right) \cap \left(\Delta_1^v \xrightarrow{\mathbf{S}_2} \Delta_0^y\right) \cap \left(\Delta_1^v \xrightarrow{\mathbf{S}_3} \Delta_0^y\right)\right]\right\}$$

где $\Pr(\Delta_1^x \xrightarrow{\mathbf{S}_2} \Delta_0^y) = \ldots = \Pr(\Delta_1^x \xrightarrow{\mathbf{S}_5} \Delta_0^y) = 2^{-2}$.

Вероятность появления каждого из объединяемых событий (попадания разности в разряды $i \in \{0, \ldots, 31\}$ или $i \in \{32, \ldots, 64\}$) равна 1/2. На основе формулы полной вероятности $\Pr\{\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_1^y\} = 2^{-8}$. Согласование разностей на выходах $\mathbf{S}_1, \ldots, \mathbf{S}_5$ и $\mathbf{S}_1^{-1}$ показало, что только в 6-ти случаях из 64 разности попадают в нужные разряды, откуда, используя свойство симметричности алгоритма шифрования, получим

$$p^{(2)} = 2\Pr\{(\Delta_0^x \xrightarrow{\mathbf{S}_1} \Delta_1^y) \bigcap (\Delta_1^x \xrightarrow{\mathbf{S}_2,\ldots,\mathbf{S}_5} \Delta_1^y) \bigcap (\Delta_0^x \xrightarrow{\mathbf{S}_1^{-1}} \Delta_0^y)\} \approx 2^{-23}.$$

Таким образом, вероятность прохождения разности с весом $w = 1$ через два раунда шифрования оценивается значением $p^*(2) = p^{(1)}p^{(2)} \approx 2^{-34}$. Для 10-ти раундов шифрования $p^*(10) \approx 2^{-170}$, следовательно, вычислительная сложность дифференциального анализа $W \geq 2^{170}$ преобразований 128-битовых блоков информации, что позволяет говорить о высокой стойкости исследуемого алгоритма шифрования к дифференциальному методу анализа и делает возможным сокращение количества раундов шифрования до уровня $R = 8$, поскольку при таком значении R рассмотренный алгоритм неотличим от случайного шифра. Экспериментальная проверка теоретического значения вероятности двухраундовой характеристики (при общем объеме пар зашифрованных текстов $\#X = 6.4 \cdot 10^6 \approx 2^{22.6}$ для правого и $\#X = 6.4 \cdot 10^7 \approx 2^{26}$ для левого подблока) дала следующие результаты: $p^{(1)} = 6.1 \cdot 10^{-4} \approx 2^{-10.68}$, $p^{(2)} = 1.7 \cdot 10^{-7} \approx 2^{-22.49}$, следовательно, вероятность двухраундовой характеристики $p^*(2) \approx 2^{-33.17}$ (аналитическая оценка $p^*(2) \approx 2^{-34}$).

В заключение представим сравнительные оценки стойкости рассмотренных шифров к дифференциальному методу анализа, которые сведены в табл. 8.25.

Таблица 8.25

Шифр	r	Характеристика		$P(r)^*$
		Разность	Вероятность	
COBRA-H64	10	$(0, \Delta_1^R)$	$P(2) \approx 1.13 \cdot 2^{-19}$	2^{-75}
COBRA-H64	8	$(0, \Delta_1^R)$	$P(2) \approx 1.13 \cdot 2^{-19}$	2^{-94}
COBRA-H128	12	$(0, \Delta_1^R)$	$P(2) \approx 1.125 \cdot 2^{-29}$	2^{-173}
COBRA-H128	10	$(0, \Delta_1^R)$	$P(2) \approx 1.125 \cdot 2^{-29}$	2^{-144}
SG-128	12	$(0, \Delta_1^R)$	$P(2) \approx 2^{-32}$	2^{-192}
SG-128	10	$(0, \Delta_1^R)$	$P(2) \approx 2^{-32}$	2^{-160}
SS-128	12	$(0, \Delta_1^R)$	$P(2) \approx 2^{-34}$	2^{-204}
SS-128	10	$(0, \Delta_1^R)$	$P(2) \approx 2^{-34}$	2^{-170}

* вклад характеристики в вероятность прохождения разности через r раундов

ГЛАВА 9

Хэш-функции и их построение на основе управляемых операций

9.1. Защита от модифицирования данных

Защита информации от модифицирования требует решения комплекса задач, связанных с надежностью аппаратного и программного обеспечения, организационными вопросами, резервированием данных, защитой от компьютерных вирусов, проверкой целостности данных, оперативностью их восстановления и др. В реальных информационных системах всегда имеется существенная вероятность модифицирования информации. Одной из важнейших задач защиты от модифицирования данных является обнаружение факта искажения информации. Во многих случаях обнаружение этого факта является достаточно сложной задачей. Последствия от использования модифицированных программ и данных в информационных автоматизированных системах могут привести к большому экономическому ущербу [26, 113].

Причинами нарушения целостности информации являются ненадежность аппаратных средств, используемых в информационных технологиях, воздействие внешних электромагнитных излучений, наличие естественных помех в каналах связи, ошибки операторов и программистов, компьютерные вирусы и действия нарушителей. Многие из этих причин вызывают появление непреднамеренных ошибок, которые имеют случайный характер, а другие связаны с умышленными воздействиями на информацию. Это может быть осуществлено специально внесенной в компьютерную систему программой или специально разработанным вирусом. Нарушитель, получая возможность несанкционированного доступа к информационным ресурсам, может внести изменения в электронные документы, модифицировать технологическую программу, удалить информацию из базы данных или внести дополнительную информацию. Умышленные воздействия имеют целенаправленный характер. При целенаправленном внесении изменений нарушитель может учитывать используемые механизмы обнаружения модификаций с целью осуществления такого модифицирования, которое нельзя было бы обнаружить. В некоторых ситуациях умышленные воздействия будут приводить к случайному модифицированию данных, например, при модифицировании зашифрованных данных. Очевидно, что если решается задача обнаружения целенаправленного модифицирования, то одновременно решается и задача обнаружения случайных искажений информации.

Одним из способов проверки целостности информации является хранение эталонных копий, используемых для сравнения. Однако, такая процедура при необхо-

димости частых проверок целостности данных большого объема приводит к существенным временным задержкам, поэтому она применяется только в специальных случаях. Более технологичным является способ, основанный на вычислении по сложным законам некоторых контрольных сумм. Вычисляются контрольные суммы для массивов данных (например, файлов, каталогов), находящихся в так называемом эталонном состоянии. Эти эталонные контрольные суммы записываются в таблицы, которые затем используются для проверки целостности информации. В этом случае проверка целостности состоит в вычислении по заданному алгоритму контрольной сумма для данного блока информации и сопоставлении полученного значения с эталонным значением контрольной суммы. Выигрыш состоит в том, что теперь нет необходимости проводить сравнение двух больших массивов данных. Целостность проверяется путем сравнения, например, 64-битовых, 128-битовых или 256-битовых контрольных сумм.

Такие контрольные суммы называются защитными контрольными суммами (ЗКС). Для их вычисления необходимо использовать алгоритмы, которые обеспечивают влияние каждого бита сообщения на значение выходного кода, описываемое некоторым «сложным» законом. Иными словами требуется алгоритм с хорошими свойствами рассеивания. В связи с этим криптографические преобразования используются не только для закрытия передаваемых или хранимых сообщений, но и для контроля целостности информации. Применяются различные типы алгоритмов вычисления ЗКС в зависимости от конкретных угроз целостности информации и требуемого уровня необнаружения модификаций данных. Двумя основными типами алгоритмов вычисления ЗКС являются:

- секретные (или использующие секретный ключ);
- открытые (или бесключевые).

Первые используют секретный ключ или сами являются секретными (наиболее удобным является использование секретного ключа). Вторые не содержат никаких секретных элементов и известны потенциальным нарушителям. Применяются алгоритмы вычисления ЗКС первого или второго типа в зависимости от конкретной решаемой задачи. Для защиты от непреднамеренных модификаций могут быть применены оба варианта. Для защиты от преднамеренных искажений также могут быть использованы секретные или открытые алгоритмы, однако во втором случае требуется обеспечить целостность самой контрольной суммы, чтобы нарушитель не мог осуществить замену исходного значения ЗКС на новое, вычисленное от модифицированного сообщения. Для этого контрольные суммы могут храниться в зашифрованном виде (т.е. опять приходим к использованию секретного ключа) или на носителях, которые являются недоступными для потенциальных нарушителей. При использовании алгоритмов с хорошими криптографическими свойствами 64-битовые ЗКС являются вполне достаточными для обнаружения умышленного модифицирования информации в случае, когда значение ЗКС недоступно нарушителю (вероятность необнаружения равна 2^{-64}).

В случае открытых алгоритмов нарушитель может вычислить эталонное значение ЗКС и попытаться осуществить очень большое число экспериментов с целью получения модифицированного сообщения, значение ЗКС которого равно эталонно-

му. В случае m-битовых ЗКС при количестве таких экспериментов 2^m имеется вероятность более 1/2, что нарушителем будет обнаружено нужное ему модифицированное сообщение. Следовательно, если указанная угроза актуальна и потенциальный нарушитель имеет доступ к большим вычислительным ресурсам, то можно рекомендовать использование значения $m \geq 96$.

Разновидностью контрольной суммы является ***имитовставка***, которая представляет собой некоторую дополнительную информацию, имеющую сравнительно малый размер и присоединяемую к передаваемому шифртексту. Имитовставка может быть вычислена по следующей итеративной формуле

$$S_i = \mathrm{E}_K(M_i \oplus S_{i-1}),$$

где K – секретный ключ (не обязательно совпадающий с ключом, использованным для шифрования сообщения), M_i – совокупность блоков данных, на которые разбивается сообщение, преобразуемое с помощью некоторой функции шифрования E, $i = 1, 2, ..., n$.

Если используется имитовставка длиной m бит, то вероятность того, что имевшая место модификация данных не будет обнаружена приемной стороной, составляет 2^{-m}. Свойство защищенности криптосистемы от навязывания ложных сообщений называется *имитостойкостью*, которую можно оценить вероятностью необнаружения искажений криптограммы. Во многих практических случаях значения $m = 64$ или даже $m = 32$ обеспечивают приемлемую имитостойкость. Приведенная выше формула фактически описывает шифрование в режиме сцепления блоков шифра, но выходом является не все зашифрованное сообщение, а только последний блок, значение которого и служит контрольной суммой.

9.2. Хэш-функции

Важнейшим типом ЗКС являются хэш-функции, относящиеся к бесключевым ЗКС. Хэш-функции иногда называют криптографическими контрольными суммами. В связи с их применением в системах ЭЦП в качестве цифрового «отпечатка пальцев» малого размера (128, 160 или 256 бит) от электронного документа большого размера (например, 10^8 бит) к ним предъявляются наиболее жесткие требования по сравнению с другими типами алгоритмов ЗКС. Вместо подписывания большого числа фрагментов документа вырабатывается подпись к хэш-функции, вычисленной от документа. Это значительно ускоряет процедуру подписывания документа, поскольку алгоритмы хэширования работают очень быстро, но основным достоинством является многократное сокращение длины подписи, что имеет важное практическое значение при работе с большим числом подписанных электронных документов. Хэш-функция должна удовлетворять следующим требованиям:

- аргументом для хэш-функции может быть сообщение произвольной длины;
- значение хэш-функции имеет фиксированную разрядность (фиксированный размер);

- хэш-функция является эффективно вычислимой.
- хэш-функция является криптографически стойкой.

Эффективная вычислимость означает, что алгоритмы хэш-функций при программной и/или аппаратной реализации должны иметь достаточно высокую производительность. По стойкости к хэш-функциям предъявляются наиболее высокие требования по сравнению с различными вариантами ЗКС. Особые требования к хэш-функциям связаны с тем, что ряд атак на ЭЦП основаны на нападениях на используемые хэш-функции. В частности так называемая атака с помощью секретарши представляет собой попытку сформировать два документа, – легальный и модифицированный – имеющие одинаковое значение хэш-функции. После подписывания хэш-функции проверяющему направляется модифицированный документ, который успешно пройдет процедуру проверки подписи, так как полученное от него значение хэш-функции на самом деле подписано владельцем секретного ключа, хотя последний полагал, что подписывал легальный документ.

Потенциальная возможность осуществления такой атаки диктует наиболее жесткое требование к хэш-функциям, которое формулируется следующим образом: *хэш-функция должна быть стойкой к коллизиям*. Это означает, что нахождение двух сообщений (документов), имеющих одно и то же значение хэш-функции является вычислительно сложной задачей, которая не может быть решена за обозримое время. При этом предполагается, что атакующий может использовать очень большие вычислительные ресурсы, например, организовать атаку с использованием миллионов процессоров и память порядка 10^{16} байт. Пусть H есть хэш-функция, а M – сообщение произвольного размера. Формально требование коллизионной устойчивости можно сформулировать так:

Вычислительно неосуществимо нахождение двух сообщений M' и $M'' \neq M'$, для которых с существенной вероятностью выполняется условие $H(M') = H(M'')$.

Оно называется также строгим (или сильным) требованием коллизионной стойкости. Формулируют также требование слабой коллизионной стойкости:

Для данного сообщения M, хэш-функция от которого равна $H(M)$*, вычислительно неосуществимо нахождение другого сообщения $M' \neq M$, для которого с существенной вероятностью выполняется условие $H(M') = H(M)$.*

Очевидно, что хэш-функция, стойкая к коллизиям в строгом смысле, является стойкой и в слабом смысле. Но хэш-функция, стойкая к коллизиям в слабом смысле, не всегда является стойкой к коллизиям в строгом смысле. Для того, чтобы хэш функция была устойчивой к коллизиям необходимо, но не достаточно, чтобы она была труднообратимой. Функция называется труднообратимой, если выполняется следующее условие.

Для случайно выбранного значения H вычислительно неосуществимо нахождение сообщения M, хэш-функция от которого с существенной вероятностью была бы равна H, т.е. $H = H(M)$.

Иными словами хэш-функция должна быть труднообратимой. Функция может быть труднообратимой, но не быть коллизионно устойчивой. Примером может слу-

жить функция возведения в дискретную степень по модулю большого простого числа: $Y = \alpha^X \bmod p$, где α – первообразный корень в поле вычетов по модулю p. При заданном случайном Y вычислительно сложно найти какое-либо значение X, которое бы соответствовало данному Y, хотя таких значений чрезвычайно много, если не ограничивать размер аргумента, что и имеет место в случае хэш-функций. Наибольший интерес представляют хэш-функции, являющиеся стойкими к коллизиям в строгом смысле.

Если хэш-функция стойкая, то при любом изменении аргумента (т. е. входного сообщения) ее значение изменится псевдослучайным образом, а именно, с вероятностью 1/2 изменится каждый из m битов двоичного вектора H. Простейшей атакой на труднообратимую хэш-функцию является подбор сообщения, имеющего заданное значение хэш-функции H. Оценим количество различных текстов (N), от которых надо вычислить хэш-функцию, чтобы с вероятностью 1/2 среди них оказался текст с заданным значением хэш-функции. Вероятность того, что хэш-функция от некоторого произвольно выбранного текста не совпадет с заданным значением H, равна $1–2^{-m}$. Вероятность, что хэш-функция ни от одного из N различных текстов не совпадает с H, равна $p' = (1 - 2^{-m})^N$. Вероятность, что хэш-функция, по крайней мере, от одного из этих сообщений совпадает с H, равна

$$p = 1 - p' = 1 - (1 - 2^{-m})^N.$$

Используя формулу бинома Ньютона, получаем следующие приближения

$$(1-x)^N = 1 - Nx + \frac{N(N-1)}{2!}x^2 - \frac{N(N-1)(N-2)}{3!}x^3 \ldots \approx 1 - Nx \text{ , если } x \text{ мало,}$$

$$p \approx N \cdot 2^{-m} \quad \text{и} \quad N \approx p \cdot 2^m.$$

При $p = 1/2$ имеем $N \approx 2^{m-1}$. В настоящее время для значений $m = 64$ такая лобовая (силовая) атака может быть реализована организацией, имеющей сравнительно большие вычислительные ресурсы. При $m > 96$ битов труднообратимая хэш-функция является стойкой к такой атаке, однако существуют и другие варианты атак (см. раздел 9.4), которые диктуют необходимость использования значений $m \geq 128$.

Для того чтобы иметь возможность вычислять хэш-функции от сообщений произвольного размера, используют разбиение блока данных на подблоки одинакового размера: $M = (M_1, M_2, ..., M_n)$, где n – число подблоков длины m бит. Если длина исходного сообщения не кратна m, то оно дополняется до кратной длины по некоторому заранее оговоренному правилу, задающему однозначность дополнения.

Один из способов построения хэш-функций состоит в использовании некоторого итеративного механизма на базе стойкой шифрующей (возможно также использование блочного одностороннего преобразования) функции E с m–битовым входом:

$$H_i = \mathrm{E}(H_{i-1}, M_i), \quad i = 1, 2, ..., n,$$

где H_0 – некоторое специфицированное начальное значение, M_i – блоки сообщения, используемые в качестве ключа. Значение $H = H_n$ принимается за значение хэш-

функции, которую можно записать в виде $H = H(H_0, M)$, что подчеркивает зависимость H от начального параметра H_0. В таких схемах построения стойкость хэш-функций не превышает стойкости базовой итеративной функции E, что делает необходимым применение криптографически стойких базовых функций.

Не представляет труда изменить преобразования, используемые в некоторой стойкой хэш-функции, таким образом, чтобы они зависели от секретного ключа или использовали некоторые секретные параметры, однако такие хэш-функции не настолько широко применяются как бесключевые хэш-функции. Если имеется стойкая бесключевая хэш-функция, то очень просто она может быть преобразована в хэш-функцию с ключом. Достаточно дополнительно использовать зашифрование полученного значения хэш-функции с помощью некоторого известного блочного шифра. Несомненно, что использование секретных элементов в хэш-функциях делает их более стойкими, однако во многих случаях требуются бесключевые хэш-функции. В первую очередь это касается систем ЭЦП, прочно вошедших в широкое применение.

9.3. Построение хэш-функций на основе блочных преобразований

В связи с тем, что аргументом хэш-функции является сообщение произвольной длины, хэш-функции удобно строить в виде некоторого итеративного механизма, последовательно преобразующего блоки данных, на которые разбивается сообщение. Преобразование, выполняемое над одним блоком, называется раундовым преобразованием, а реализуемая им функция – раундовой хэш-функцией. При этом преобразования строятся таким образом, что результат, получаемый на текущем шаге вычисления раундового значения хэш-функции, должен зависеть от всех предыдущих значений и от текущего блока данных. Это достигается применением механизма сцепления. Таким образом, некоторая раундовая блочная хэш-функция F должна зависеть от двух аргументов – предыдущего значения раундовой хэш-функции H_{i-1} и текущего подблока M_i. Это можно записать в виде:

$$H_i = \mathrm{F}(H_{i-1}, M_i), \quad i = 1, 2, ..., n,$$

где H_0 – некоторое специфицированное начальное значение. В качестве блочного преобразования F может использоваться, например, блочный шифр (обратимое преобразование) или некоторая труднообратимая функция. Может быть использована сжимающая функция F, получившая такое название ввиду того, что разрядность ее выхода меньше разрядности подаваемых на вход блоков данных. Переменная H_i называется переменной сцепления. Она задает зависимость результата преобразования на текущем шаге от подблоков данных, участвовавших в преобразованиях на предыдущих шагах.

Ввиду наличия достаточного количества стойких блочных шифров различного типа представляет интерес разработка хэш-функций на основе блочных алгоритмов. Простейшая конструкция на основе блочных шифров носит название схемы Рабина, которая описывается следующей формулой:

$$H_i = \mathrm{E}_{M_i}(H_{i-1}), \quad i = 1, 2, ..., n,$$

где в качестве индекса указан параметр, используемый как ключ. Универсальной атакой на хэш-функцию, построенную по схеме Рабина, является атака «встреча посередине», которая рассмотрена в разделе 9.5. Данная атака является универсальной в том смысле, что она не зависит от конкретного вида раундового преобразования. Данная атака устраняется выбором шифрующей функции с размером входа $m \geq 128$ бит.

В других схемах построения хэш-функций дополнительно в составе раундового преобразования используется поразрядное суммирование по модулю два. На рис. 9.1 показаны три варианта общей структуры таких раундовых преобразований. Может быть использовано также комбинирование этих схем.

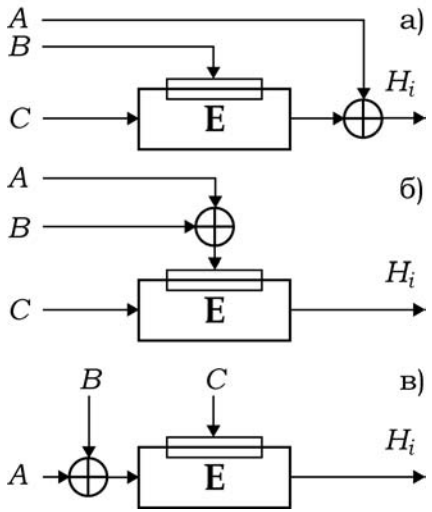

Рис. 9.1. Варианты обобщенной схемы построения раундовой хэш-функции:
а – сложение на выходе функции **E**, б – сложение на входе для ключа,
в – сложение на входе функции **E**

В качестве параметров A, B и C в различных схемах могут использоваться значения H_{i-1}, M_{i-1} или M_i. Исследование многочисленных схем показало, что использование стойкого шифра не является достаточным условием для построения стойкой хэш-функции. Большую роль играет конкретная схема построения [113]. Установлено, что стойкими являются хэш-функции, построенные на основе стойкого m-битового шифра при $m \geq 128$ в соответствии с формулами, представленными в таблице 9.1. Рисунок 9.2 иллюстрирует схемное представление первого, пятого и десятого вариантов данной таблицы. Иногда в схемном представлении показывают только i-ый шаг вычислений, обведенный на рис. 9.2 пунктирной рамкой.

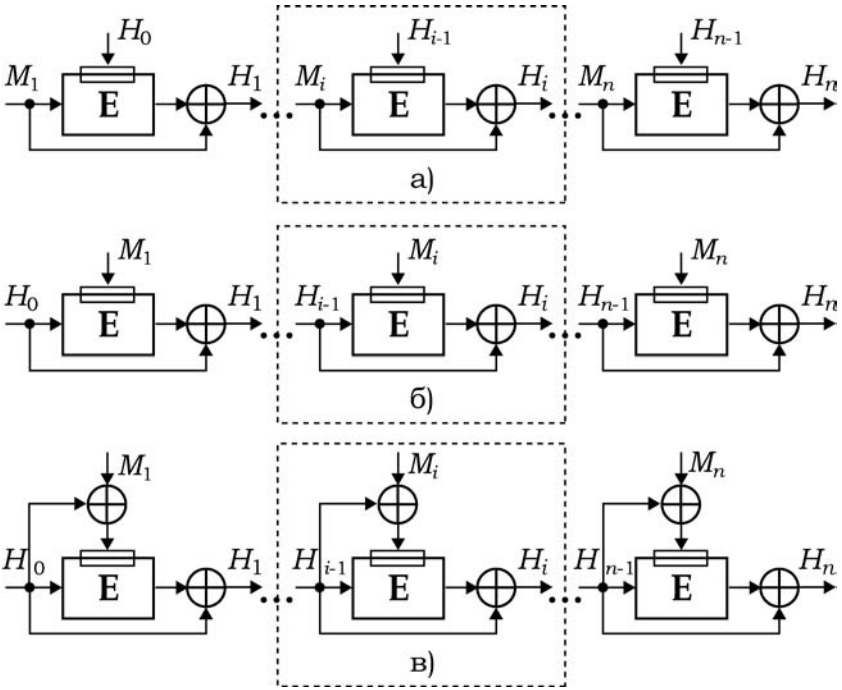

Рис. 9.2. Схематическое представление хэш-функции, построенной в соответствии с вариантами 1 (а), 5 (б) и 10 (в) таблицы 9.1

Таблица 9.1

Приемлемые схемы построения хэш-функций на основе стойких блочных шифрующих функций E (случай использования параметров H_{i-1} и M_i).

№ п/п	Формула
1	$H_i = \mathrm{E}_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = \mathrm{E}_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = \mathrm{E}_{H_{i-1}}(M_i) \oplus M_i \oplus H_{i-1}$
4	$H_i = \mathrm{E}_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = \mathrm{E}_{M_i}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = \mathrm{E}_{M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
7	$H_i = \mathrm{E}_{M_i}(H_{i-1}) \oplus M_i \oplus H_{i-1}$
8	$H_i = \mathrm{E}_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$

№ п/п	Формула
9	$H_i = \mathrm{E}_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = \mathrm{E}_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = \mathrm{E}_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = \mathrm{E}_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_{i-1}$

Большое число других вариантов раундовых хэш-функций может быть построено с использованием трех параметров H_{i-1}, M_{i-1} и M_i. В таких схемах используется на один регистр больше, однако они дают большее разнообразие стойких схем построения. При использовании однонаправленной блочной функции вместо блочного шифра обычно имеется вход и выход, что приводит к уменьшению числа различных приемлемых схем и целесообразности использования значения M_{i-1} в качестве одного из параметров, участвующих в раундовом преобразовании (в этом случае требуется дополнительно специфицировать значение M_0). Как правило, в однонаправленную функцию легко ввести дополнительный вход, что позволяет применить указанные в табл. 9.1 двенадцать конструктивных схем.

Более интересным представляется построение хэш-функции на основе однонаправленных функций без дополнительного входа. Такие варианты показаны на рис. 9.3, где предполагается, что размер входа функции F равен $m \geq 128$ бит.

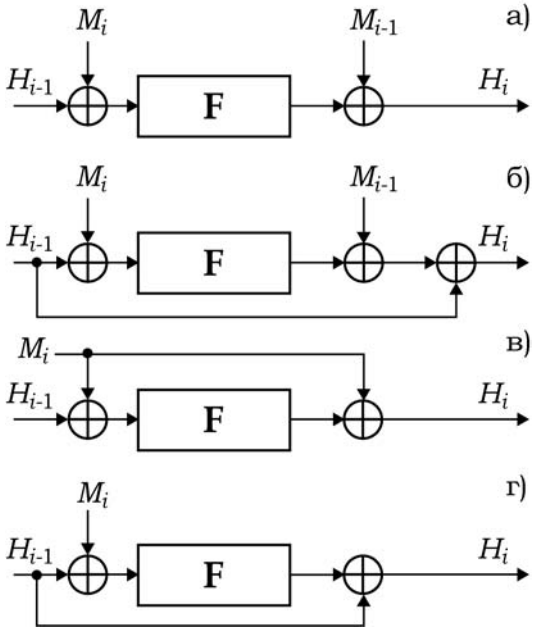

Рис. 9.3. Схемы построения хэш-функций на основе блочных односторонних преобразований

Однонаправленную функцию можно построить на основе стойкого блочного алгоритма E, например, в соответствии со следующей формулой:

$$F(X) = X \oplus E_K(X),$$

где K – некоторый фиксированный известный ключ. Для стойкого алгоритма E_K данную функцию обратить трудно, поскольку мы не знаем вектора, формируемого на выходе E_K. Подобный прием часто используется при построении труднообратимых блочных преобразований, используемых в хэш-функциях. В нашем примере аргумент используется как входное значение для блочной шифрующей функции и как операнд завершающего преобразования, заключающегося в выполнении операции сложения. В применяемых однонаправленных функциях аргумент иногда многократно (целиком или по частям) входит как параметр преобразования.

При использовании значения M_i в качестве аргумента и H_{i-1} в качестве ключа мы получаем первую схему из табл. 9.1. Этот пример показывает, что в различных схемах из данной таблицы используется аналогичный прием преобразования шифрующего преобразования в одностороннее. При использовании указанной односторонней функции схема, показанная на рис. 9.3г, преобразуется последовательно к виду, показанному на рис. 9.4а и 9.4б.

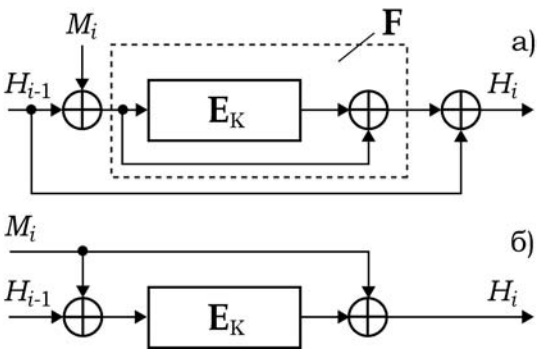

Рис. 9.4. Пример использования шифра E вместо однонаправленного преобразования при построении хэш-функции

Последняя схема напоминает конструкцию, показанную на рис. 9.3в, но в первом случае хэш-функция построена на основе одностороннего преобразования, а во втором – на основе шифрующего. Следует заметить, что различные элементы построения, использованные в схемах, представленных на рис. 9.3, могут быть объединены в различных комбинациях, однако некоторые сочетания могут привести к построению слабой хэш-функции. Например, на рис. 9.5 показан вариант комбинирования элементов, используемых на рис. 9.3в и 9.3г. Такая конструкция однонаправленной функции выглядит несколько искусственной или намеренно выбранной, однако этот пример полезен тем, что показывает необходимость конкретного рассмотрения каждого построения, даже если используются базовые примитивы, являющиеся стойкими как самостоятельные преобразования.

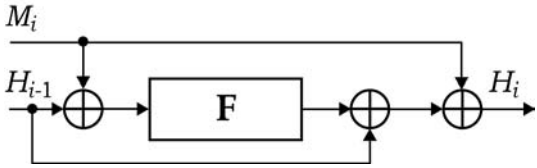

Рис. 9.5. Пример нестойкой хэш-функции с использованием стойкого блочного преобразования F

Очевидно, что в последней схеме $H_i = \mathrm{F}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$. Из данной формулы нетрудно увидеть возможность следующего простого модифицирования хэшируемого сообщения с сохранением исходного значения хэш-функции.

Пусть дано сообщение $M = (M_1, M_2, \ldots, M_n)$. Вычисляя значение хэш-функции от этого сообщения, можно сохранить все промежуточные значения раундовой хэш-функции $H_1, H_2, \ldots, H_n$. Теперь модифицируем некоторые блоки данных M_{i+1}, M_{i+2}, ...,M_j и получаем вместо них новую последовательность блоков M'_{i+1}, M'_{i+2}, ...,M'_j. Это приводит к получению новых значений $H'_{i+1}, H'_{i+2}, \ldots, H'_j$. Теперь вместо блока M_{j+1} возьмем блок $M'_{j+1} = H_j \oplus H'_j \oplus M_j$ и получим значение

$$H'_{j+1} = \mathrm{F}(M'_{j+1} \oplus H'_j) \oplus M'_{j+1} \oplus H_j{}' = \mathrm{F}(M_{j+1} \oplus H_j) \oplus M_{j+1} \oplus H_j = H_{j+1}.$$

Очевидно, что при вычислении хэш-функции от сообщения

$$M' = (M_1, M_2, \ldots, M_i, M'_{i+1}, M'_{i+2}, \ldots, M'_j, M'_{j+1}, M_{j+2}, \ldots, M_n)$$

мы получим следующую последовательность промежуточных значений раундовой хэш-функции: $H_1, H_2, \ldots, H_{i-1}, H_i, H'_{i+1}, H'_{i+2}, \ldots, H'_j, H_{j+1}, H_{j+2}, \ldots, H_n$, т. е. имеем $H(M') = H(M) = H_n = H$. Таким образом, по данному сообщению и хэш-функции от него мы легко вычислили другое сообщение, хэш-функция от которого равна хэш-функции исходного сообщения. Но для некоторого данного случайно выбранного значения H'' нахождение документа M'', для которого с большой вероятностью выполняется условие $H(M'') = H''$, является вычислительно сложным.

Таким образом, рассмотренный пример представляет труднообратимую хэш-функцию, которая, однако, не является стойкой к коллизиям даже в слабом смысле. Более того, построенная атака легко находит коллизию практически для любой разрядности блочного преобразования F. Эту схему нельзя исправить ни выбором хороших однонаправленных блочных преобразований, ни использованием большой разрядности хэш-функции.

9.4. Нахождение коллизий в общем случае

Используемые в системах ЭЦП хэш-функции должны быть стойкими к атакам на основе нахождения двух различных сообщений, имеющих одинаковые значения хэш-функции. Это связано с тем, что на практике документы для подписывания могут

готовиться техническим персоналом, среди которого может оказаться потенциальный нарушитель. Если хэш-функция не является стойкой к нахождению коллизий, то тогда нарушитель может составить два различных документа, имеющих одинаковое значение хэш-функции. Один из документов, содержание которого предварительно согласовано с подписывающим, представляется для подписи. После получения подписи этот документ подменяется вторым, имеющим фальсифицированное содержание. Подложный документ и подпись направляются проверяющему. Очевидно, что в этом случае проверка подписи даст положительный результат и фальсифицированный документ будет принят за истинный.

Если размер значения хэш-функции сравнительно мал, то независимо от качества преобразований, обеспечиваемых алгоритмом хэширования, коллизии могут быть найдены с помощью атаки, основанной на парадоксе о днях рождения. Парадокс дней рождения состоит в ответе на следующий вопрос. Какой должна быть численность группы случайно выбранных людей, чтобы с вероятностью 0.5 в этой группе нашлось два человека, у которых дни рождения совпадают? Оказывается, что численность этой группы должна составлять всего примерно 20 человек. Дело в том, что вероятность совпадения дней рождения для случайной пары составляет $p' = 1/365$, а в группе из n человек имеется $n(n-1)/2 \approx n^2/2$ разных пар. Отсюда легко получить следующую приближенную оценку. Вероятность того, что хотя бы для одной из этих пар будет иметь место совпадение дней рождения, составляет $p \approx p'n^2/2$, откуда для $p = 1/2$ получаем $n \approx \sqrt{1/p'}$.

Рассмотрим, как парадокс о днях рождения может быть использован для нахождения коллизий. Пусть дана хэш-функция H с m-битовым значением выхода. Составим N различных документов $M^{(i)}$, $i = 1, 2, \ldots, N$, и вычислим соответствующие им хэш-функции $H^{(i)}$. Если хэш-функция задает псевдослучайное преобразование (а стойкие хэш-функции обладают этим свойством), то тогда легко подсчитать вероятность того, что среди полученных N значений $H^{(i)}$ не найдутся два одинаковых.

Сначала рассмотрим оценку, которая изначально опирается на приближение, учитывающее, что число коллизий достаточно мало. Выберем некоторое значение $H^{(1)}$. Вероятность того, что оно не совпадает с остальными равна

$$p^{(1)} \approx (1 - 2^{-m})^{N-1}.$$

В этом соотношении как раз и используется исходное предположение. Выберем некоторое новое значение $H^{(2)}$. Вероятность того, что оно не совпадает с остальными, равна

$$p^{(2)} \approx (1 - 2^{-m})^{N-2}.$$

Выбирая каждый раз новые значения хэш-функции, на i-ом шаге получаем вероятность несовпадения значений

$$p^{(i)} \approx (1 - 2^{-m})^{N-i}.$$

Всего мы должны выполнить $N - 1$ шагов проверки на совпадение. Вероятность того, что ни на одном из них не будет найдено равных значений, составляет

$$p' \approx (1-2^{-m})^{N-1}\cdot(1-2^{-m})^{N-2}\cdot\ ...\cdot\ (1-2^{-m})^{N-i}\cdot\ ...\cdot\ (1-2^{-m}) =$$

$$= (1-2^{-m})^{\Sigma}, \text{ где } \Sigma = 1 + 2 + \ldots + (N-1) = N(N-1)/2.$$

Вероятность нахождения коллизий (т. е. двух одинаковых значений $H^{(i)}$) составляет

$$p \approx 1 - p' = 1 - (1-2^{-m})^{\Sigma} \approx 1 - (1 - \Sigma\cdot 2^{-m}) \approx$$

$$\approx N^2\cdot 2^{-m-1}.$$

Из условия $N^2\cdot 2^{-m-1} = 1/2$ определим значение $N_{1/2}$, при котором вероятность наличия коллизий составляет 1/2:

$$N_{1/2} \approx \sqrt{2^m}\ .$$

Точное вычисление вероятности наличия коллизий в множестве значений $H^{(i)}$, $i = 1, 2, \ldots, N$, может быть также получено достаточно просто. Возьмем $H^{(2)}$. Вероятность того, что $H^{(2)}$ не равно $H^{(1)}$, равна $p^{(2)} = (1 - 2^{-m})$. Возьмем $H^{(3)}$. Вероятность того, что $H^{(3)}$ не совпадает с $H^{(1)}$ и $H^{(2)}$ при условии, что $H^{(1)} \neq H^{(2)}$, равна $p^{(3)} = (1 - 2^{-m}) \cdot (1 - 2\cdot 2^{-m})$. Продолжая такое рассмотрение, определим вероятность $p^{(N)}$ того, что $H^{(N)}$ не совпадает ни с одним из следующих значений $H^{(1)}, H^{(2)}, \ldots, H^{(N-1)}$ при условии, что среди последних нет равных значений. Получаем $p^{(N)} = p^{(2)}\ p^{(3)}\cdot\ ...\cdot\ p^{(N-1)} \cdot [1 - (N-1)\cdot 2^{-m}]$. Таким образом, точное значение вероятности отсутствия коллизий составляет

$$p' = p^{(N)} = (1-2^{-m})\cdot(1-2\cdot 2^{-m})\cdot\ ...\cdot\ [1-(i-1)\cdot 2^{-m}]\cdot\ ...\cdot$$

$$\cdot\,[1-(N-1)\cdot 2^{-m}] = \prod_{i=1}^{i=N-1}(1-i\cdot 2^{-m}).$$

Из него можно точно определить вероятность наличия коллизий $p = 1 - p'$. Однако для получения формулы, дающей более наглядную связь между вероятностью коллизий и числом N, можно воспользоваться следующей приближенной формулой $1 - x \approx e^{-x}$, где e – основание натуральных логарифмов. Откуда получаем

$$p' = \prod_{i=1}^{i=N-1}(1-i\cdot 2^{-m}) \approx \prod_{i=1}^{i=N-1}\exp(-i\cdot 2^{-m}) =$$

$$= \exp\Big(-\sum_{i=1}^{i=N-1} i\cdot 2^{-m}\Big) = \exp[-N(N-1)\cdot 2^{-m-1}].$$

Вероятность наличия, по крайней мере, одной коллизии равна

$$p \approx 1 - p' = 1 - \exp[-N(N-1)\cdot 2^{-m-1}].$$

Из последнего соотношения далее легко получить

$$N^2 + N \approx 2^{m+1} \ln\left(\frac{1}{1-p}\right).$$

Пренебрегая значением N по сравнению с N^2, получаем

$$N^2 \approx 2^{m+1} \ln\left(\frac{1}{1-p}\right),$$

$$N \approx \sqrt{2^{m+1} \ln\left(\frac{1}{1-p}\right)}.$$

Для $p = 0.5$ имеем $N_{1/2} \approx 1.17 \cdot \sqrt{2^m}$. Таким образом, более точный расчет $N_{1/2}$ дал примерно то же значение, что и первый вариант. В случае задачи о днях рождения получаем, что среди 23 случайных человек с вероятностью 0.5 у двух из них дни рождения будут совпадать.

Используя формулу для $N_{1/2}$ и учитывая, что задача сортировки N элементов имеет трудоемкость порядка $N \ln N$, можно оценить ресурсы, необходимые для нахождения коллизии с вероятностью 1/2. Легко установить, что требуется иметь память объемом $1.17 \cdot 2^{m/2} m$ бит, а также необходимо выполнить $1.17 \cdot 2^{m/2}$ вычислений хэш-функции и осуществить сортировку $N = 1.17 \cdot 2^{m/2}$ чисел. Сложность выполненных процедур хэширования можно оценить как NnW, где W – число операций, необходимых для вычисления раундовой хэш-функции. При $m = 64$ бит эта задача может быть решена за приемлемое время при использовании достаточно умеренных вычислительных ресурсов. Это означает, что стойкой к коллизиям может быть признана хэш-функция для значений $m \geq 128$ бит. При значении $m = 64$ бит никакие идеальные алгоритмы не обеспечат высокую стойкость хэш-функции к рассмотренной атаке на основе парадокса о днях рождения.

Однако если подписывающий будет вносить случайную корректировку, перед тем как подписать документ, то такая атака может быть устранена. Конечно, это требует дополнительных действий от подписывающего, о которых он может забыть или просто игнорировать их. При использовании хэш-функций с разрядностью не менее 128 эти дополнительные действия не потребуются. Важно и то, что при увеличении разрядности можно разработать алгоритмы, которые потенциально обладают более высокой стойкостью не только к атаке на основе парадокса о днях рождения, но и к другим атакам.

В рассмотренном выше варианте атаки осуществляется нахождение двух сообщений с равными значениями хэш-функции, однако, найденные сообщения являются случайными и маловероятно, что атакующий сможет убедить подписывающего в необходимости подписать одно из них. Нахождение коллизии по формальным требованиям означает, что хэш-функция не является стойкой. Для успешной реализации атаки на практике полезно иметь в качестве одного из сообщений некоторый документ, который выглядит естественным для подписывающего. Кроме того, атакующий может иметь желание, чтобы подделанное сообщение имело определенное содержа-

ние, выгодное ему. Это означает, что на практике желательно найти коллизию не к произвольным документам, а к таким, которые являются осмысленными. В этом случае построение атаки принципиально не меняется. Она может быть проведена следующим образом.

Атакующий составляет два исходных документа M и M', первый из которых имеет законное содержание, а второй – фальсифицированное. Затем он разрабатывает некоторый генератор, формирующий путем видоизменения форматирования документов два множества осмысленных документов $\{M\}$ и $\{M'\}$. Первое множество представляет собой варианты представления законного документа, а второе – варианты представления фальсифицированного документа. Внутри каждого из множеств смысловое содержание остается фиксированным. Пусть мощности этих множеств равны $N = \#\{M\}$ и $N' = \#\{M'\}$. Для всех документов из обоих множеств вычисляется хэш-функция (см. рис. 9.6) и формируются два их подмножества $\{H\}$ и $\{H'\}$. После этого каждое значение из множества $\{H'\}$ сравнивается с каждым значением из множества $\{H\}$ до тех пор, пока не будут найдены два равных между собой значения H и $H' = H$ (на самом деле все значения H и H' сортируются, находятся два равных и проверяется принадлежат ли они разным множествам $\{H\}$ и $\{H'\}$). Условием нахождения двух таких значений является достаточно большая вероятность p их наличия в указанных множествах. Оценим, какие минимальные значения должны иметь N и N', чтобы получить значение $p = 0.5$.

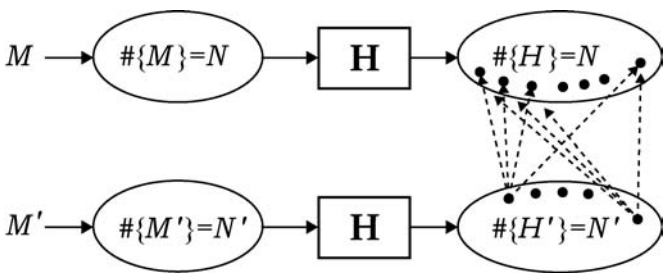

Рис. 9.6. Поиск коллизии, соответствующей осмысленным сообщениям M и M'

Учитывая парадокс дней рождения, можно ожидать, что N и N' существенно меньше значения 2^m. Тогда с достаточно хорошим приближением можно принять, что вероятность несовпадения некоторого первого значения $H'^{(1)}$ из множества $\{H'\}$ ни с одним из значений множества $\{H\}$, равна

$$p_1 \approx (1 - 2^{-m})^N.$$

Вероятность того, что ни одно из значений множества $\{H'\}$ не совпадает ни с одним из значений множества $\{H\}$, равна

$$p' \approx (1 - 2^{-m})^{N'N}.$$

Вероятность того, что найдется, по крайней мере, одна пара одинаковых значений H и H', равна

$$p \approx 1 - p' = 1 - (1 - 2^{-m})^{N'N}.$$

Воспользуемся следующим приближением для малых значений x: $(1-x)^{N''} \approx 1 - N'x$. Получим:

$$p \approx 1 - (1 - N'N \cdot 2^{-m}) \approx N'N \cdot 2^{-m}.$$

При фиксированной сумме $N + N'$ произведение NN' максимально при $N = N'$. Из условия $p \approx 0.5$ легко получить:

$$N \approx \sqrt{2^{m-1}}.$$

Это значение определяет сложность атаки и требуемую память. Таким образом, атака с использованием осмысленных сообщений имеет тот же порядок сложности, что и исходный вариант атаки на основе парадокса дней рождения.

Рассмотрим еще один вариант атаки с использованием парадокса дней рождения. Выберем достаточно длинное сообщение $M = (M_1, M_2, \ldots, M_n)$, содержащее $n \geq 2^{m/2}$ блоков M_i, и вычислим промежуточные значения хэш-функции: $H_1, H_2, \ldots, H_n$. В соответствии с парадоксом дней рождения с вероятностью 0.5 среди этих значений найдутся два равных. Пусть это будут H_i и H_j. Удалив из M блоки $M_{i+1}, M_{i+2}, ,...,M_j$, получаем новое сообщение $M' = (M_1, M_2, \ldots, M_i, M_{j+1}, M_{j+2}, \ldots, M_n)$, которое имеет значение хэш-функции, равное значению хэш-функции сообщения M (см. рис. 9.7). Даже при $m = 64$ получаем, что длина используемого сообщения чрезмерно велика, однако по формальным критериям хэш-функция с такой размерностью должна быть признана нестойкой, поскольку рассмотренная атака вычислительно реализуема.

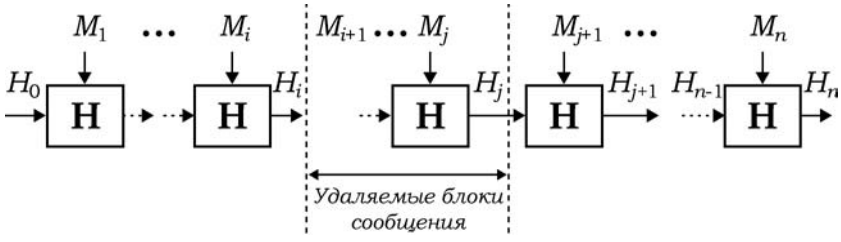

Рис. 9.7. Атака с удалением части сообщения

9.5. Атака «встреча посередине»

Атака, получившая название «встреча посередине» (meet-in-the-middle attack), может быть предпринята к хэш-функциям, построенным на основе использования блочного шифра в качестве раундовой хэш-функции. Она дает более сильный ре-

зультат по сравнению с атакой на основе парадокса о днях рождения – коллизия находится к заданному наперед сообщению. В атаке с использованием парадокса о днях рождения находится коллизия, но получаемое значение хэш-функции для найденной коллизии является случайным. Однако атака «встреча посередине» может быть применена только к частным типам итеративных хэш-функций. Впервые такая атака была предложена для нападения на хэш-функцию, построенную по схеме Рабина (см. рис. 9.8).

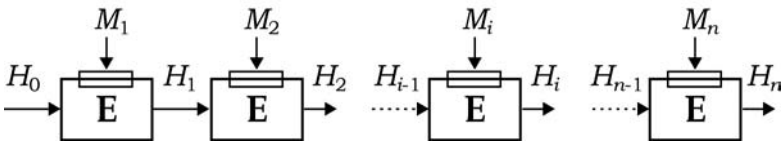

Рис. 9.8. Схема Рабина

Схема Рабина представляет собой простой и естественный механизм построения итеративной хэш-функции с использованием в качестве раундовой хэш-функции некоторого предположительно стойкого блочного шифра. Схема основана на идее о вычислительной сложности определения ключа при известных входном и выходном блоках данных. Блоки данных M_i используются в качестве ключей на соответствующих раундах вычисления хэш-функции. Нахождение коллизии связано с задачей вычисления ключа. Например, атакующий может заменить некоторый блок данных M_k на M'_k. Это приведет к получению нового значения раундовой хэш-функции H'_k. Может существовать некоторый ключ шифрования M'_{k+1}, для которого получим

$$H'_{k+1} = \mathrm{E}_{M'_{k+1}}(H'_k) = H_{k+1}.$$

Если атакующему удастся найти указанное M'_{k+1}, то, заменяя в исходном сообщении блоки данных M_k и M_{k+1} на M'_k и M'_{k+1}, он получает модифицированное сообщение, хэш-функция от которого имеет то же самое значение, что и исходное сообщение. Если блочный алгоритм является стойким к атакам на основе известного текста, то указанное нападение на хэш-функцию имеет высокую вычислительную сложность. Однако атака «встреча посередине» является универсальной по отношению к схеме Рабина и диктует необходимость использования блочных шифров с разрядностью не менее 128 бит.

Рассмотрим данную атаку. Пусть дано сообщение M, хэш-функция от которого равна H. Целью атакующего является нахождение другого сообщения M', хэш-функция от которого также равна H. Для этого он делит сообщение $M = (M_1, M_2, \ldots, M_k, M_{k+1}, \ldots, M_n)$ на две части $M^{(1)} = (M_1, M_2, \ldots, M_k)$ и $M^{(2)} = (M_{k+1}, \ldots, M_n)$. Первая часть сообщения многократно модифицируется и от каждого модифицированного значения вычисляется хэш-функция $H^{(1)}$. Пусть получено N_1 значений $H^{(1)}$ от N_1 различных вариантов первой части (см. рис. 9.9). Сообщение $M^{(2)}$ также многократно модифицируется и от него вычисляется хэш-функция по другому алгоритму, а именно, в обратном порядке и с использованием алгоритма расшифрования D, соот-

ветствующего алгоритму шифрования E (см. рис. 9.10). В качестве начального значения хэш-функции берется H. Пусть получено N_2 значений $H^{(2)}$ от N_2 различных вариантов второй части.

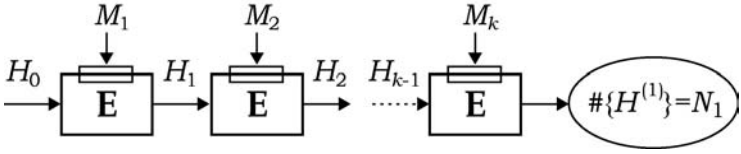

Рис. 9.9. Вычисление промежуточных значений хэш-функции с использованием алгоритма зашифрования

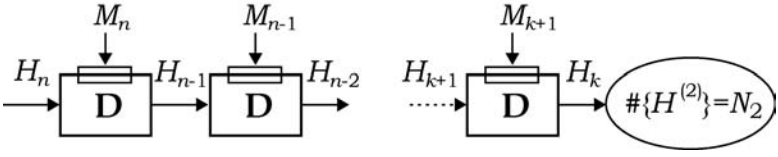

Рис. 9.10. Вычисление промежуточных значений хэш-функции с использованием алгоритма расшифрования

При достаточно больших N_1 и N_2 с большой вероятностью можно найти равные между собой значения $H^{(1)}$ и $H^{(2)}$. Пусть они соответствуют сообщениям $M'^{(1)}$ и $M'^{(2)}$. Очевидно, что для сообщения $M' = (M'^{(1)}, M'^{(2)}) \neq M$ мы имеем $H(M') = H(M)$, т. е. мы нашли коллизию для наперед заданного значения хэш-функции. Определим необходимую память и трудоемкость атаки «встреча посередине».

Будем предполагать, что N_1 и N_2 существенно меньше значения 2^m, где m – разрядность хэш-функции. Тогда с достаточно хорошим приближением можно принять, что вероятность несовпадения первого значения $H^{(1)}$ из множества $\{H^{(1)}\}$ ни с одним из значений множества $\{H^{(2)}\}$ равна

$$p_1 \approx (1 - 2^{-m})^{N_2}.$$

Вероятность того, что ни одно из значений множества $\{H^{(1)}\}$ не совпадает ни с одним из значений множества $\{H^{(2)}\}$, равна

$$p' \approx (1 - 2^{-m})^{N_2 N_1}.$$

Вероятность того, что найдется, по крайней мере, одна пара равных значений $H^{(1)}$ и $H^{(2)}$ составляет

$$p \approx 1 - p' = 1 - (1 - 2^{-m})^{N_2 N_1} \approx 1 - (1 - N_1 N_2 \cdot 2^{-m}) \approx N_1 N_2 \cdot 2^{-m}.$$

Теперь из условия $N_1 N_2 \cdot 2^{-m} = 1/2$ для случая $N_1 = N_2 = N$ (при $N_1 = N_2$ трудоемкость атаки минимальна) легко определить значение $N_{1/2}$, при котором вероятность наличия коллизий составляет 1/2:

$$N_{1/2} \approx \sqrt{2^{m-1}}\ .$$

Получена примерно такая же оценка как и в случае атаки на основе парадокса дней рождения. Более точная оценка несколько превышает полученное значение для $N_{1/2}$. Для более точной оценки при $N_1 = N_2 = 2^{-m/2}$ можно получить значение $p \approx 0.63$. Таким образом, необходимая память составляет примерно $2m\sqrt{2^{m-1}}$ бит, а трудоемкость атаки «встреча посередине» составляет порядка $N_1 + N_2 \approx \sqrt{2^{m+1}}$ вычислений хэш-функции.

Данная атака показывает, что даже если подписывающий будет модифицировать документ, подготовленный третьими лицами, то в схеме Рабина все равно требуется использовать 128-разрядный блочный алгоритм шифрования. Были предложены модифицированные версии построения итеративных хэш-функций на основе блочных алгоритмов. Однако было показано, что и к ним можно применить нападения с помощью несколько модифицированной атаки типа «встреча посередине». Другим способом противодействия такой атаке является применение хэш-функций, построенных с использованием односторонних блочных преобразований.

Таким образом, наиболее надежным и общим методом противодействия атаке «встреча посередине» является использование разрядности хэш-функции не менее 128 бит. Тем более, что этот способ устраняет также атаки на основе парадокса дней рождения, рассмотренные в предыдущем параграфе.

9.6. О построении хэш-функций на основе управляемых операций

В разделе 9.3 были рассмотрены общие схемы построения хэш-функций на основе блочных криптографических преобразований. Управляемые операции представляют собой удобный примитив для конструирования блочных шифров и односторонних преобразований. В *главе 8* рассматриваются различные варианты построения скоростных шифров, основанных на управляемых операциях различного типа и ориентированных на недорогую аппаратную реализацию. Поскольку все они используют простое расписание ключа, то могут быть использованы в качестве базового блочного преобразования при построении хэш-функций в соответствии с известными конструктивными схемами. Виду того, что простое расписание ключей предполагает использование частей секретного ключа как непосредственных операндов выполняемых при шифровании операций, то следует учитывать возможность существования слабых ключей. Наличие последних в шифрах не является критическим, если их доля мала, однако при построении хэш-функций на основе таких шифров могут появиться серьезные слабости. Некоторые схемы, обеспечивающие построение стойких хэш-функций при использование шифров, свободных от слабых ключей, могут оказаться нестойкими при использовании шифров, имеющих даже очень малую долю слабых ключей. Примером является схема Рабина (см. раздел 9.5). Действительно при наличии слабых ключей у преобразования E возможна следующая простейшая атака. В

произвольное сообщение $M = (M_1, M_2, ..., M_n)$ после i-го блока встраиваются еще два таких же блока M_i. Очевидно, что для сообщения

$$M' = (M_1, M_2, ..., M_{i-1}, M_i, M_i, M_i, M_{i+1} ..., M_n)$$

выполняется условие $H(M') = H(M)$, т. е. мы нашли коллизию. Этот пример показывает, что наличие слабых ключей вносит определенные ограничения на используемые конструктивные схемы. Таким образом, предпочтение можно отдать переключаемым управляемым операциям при проектировании шифров, поскольку их применение обеспечивает устранение слабых ключей при использовании простого расписания ключа (ПРК). Следует учесть, что ПРК позволяет получить высокие скорости вычисления хэш-функций и упрощение аппаратной реализации, что представляет существенный практический интерес.

Управляемые операции, построенные на основе УППС, являются экономичными в реализации и вносят малое время задержки, поэтому они могут быть применены вместо операции поразрядного суммирования по модулю два в общих схемах построения хэш-функций. В этом случае комбинирование входных и выходных параметров блочного преобразования, выполняемого на текущем раунде шифрования, осуществляется с одновременным внесением дополнительного вклада в лавинный эффект и нелинейность преобразования. В результате мы получаем три механизма, показанные на рис. 9.11, на основе которых могут быть построены различные варианты раундовой хэш-функции, включая и комбинированные. Вместо формальных величин *A*, *B* и *C*, указанных на рисунке, предполагается использование значений H_{i-1}, M_{i-1} или M_i. В качестве прототипов могут применяться варианты конкретных конструктивных схем, описанные в разделе 9.3. При использовании нескольких управляемых операций, обрамляющих блочное преобразование E, они могут относиться к различному типу. В общем случае использование управляемых операций позволяет существенно расширить множество вариантов построения хэш-функций для некоторого фиксированного блочного преобразования.

Таким образом, управляемые операции могут быть применены

- для построения базового (шифрующего или одностороннего) блочного преобразования;
- в качестве вспомогательных обрамляющих операций, формирующих различные механизмы сцепления промежуточных значений раундовой хэш-функции.

Поскольку криптографические преобразования на основе управляемых операций экономично реализуются в виде быстродействующих электронных схем, то при построении базового блочного преобразования может быть применено одновременное преобразование значений M_i и H_{i-1} с помощью шифрующего преобразования, управляемого ключом *K*, в качестве которого используется одна из величин M_{i-1}, M_i, H_{i-1} или некоторая их комбинация. В данном случае блочное преобразование E имеет удвоенную разрядность входа ($2m$ бит). После его выполнения выходной блок делится на два равных по размеру подблока и они складываются, давая в результате текущее m-битовое значение раундовой хэш-функции (см. рис. 9.12а).

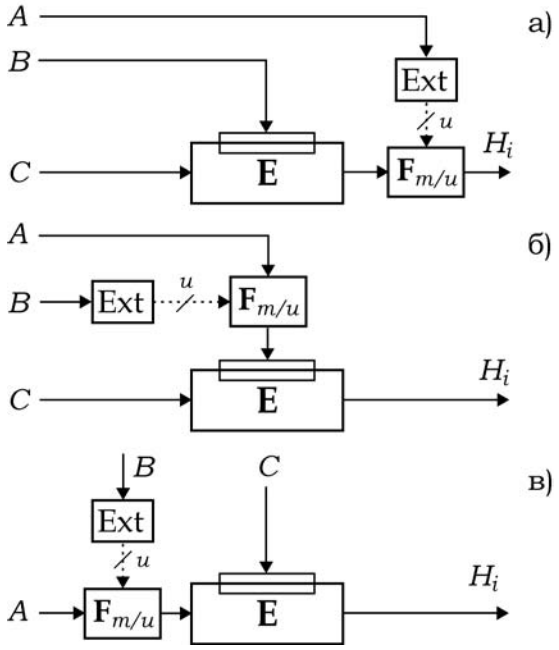

Рис. 9.11. Варианты обобщенной схемы построения раундовой хэш-функции с использованием управляемых операций на выходе (а), входе для ключа (б) и входе (в) функции **E**

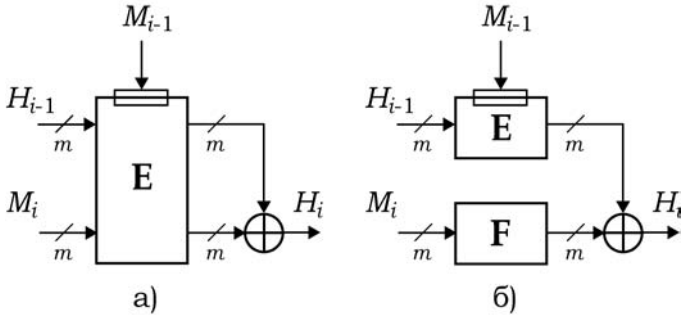

Рис. 9.12. Варианты с одновременным преобразованием значений M_i и H_{i-1}:
а – использование единого блочного преобразования удвоенной размерности,
б – использование двух различных блочных преобразований с m-разрядным входом

Для одновременного преобразования значений M_i и H_{i-1} можно применить два независимых преобразования, например, одно шифрующее, а другое– одностороннее, как это показано на рис. 9.12б. При выполнении большого числа параллельных опе-

раций лавинный эффект развивается в каждом из них, поэтому необходимых свойств преобразования можно достичь за меньшее время, что ведет к повышению скорости вычисления хэш-функции. Это приводит к идее построения раундовой хэш-функции с предварительным расширением векторов M_i и H_{i-1}, за которым следует блочное преобразование с расширенным входом или несколько одновременно выполняемых блочных преобразований (в общем случае различного типа). Преобразованное расширенное значение суммируется таким образом, что получается m-битовое значение раундовой хэш-функции, в котором аккумулируются накопленные изменения (рис. 9.13).

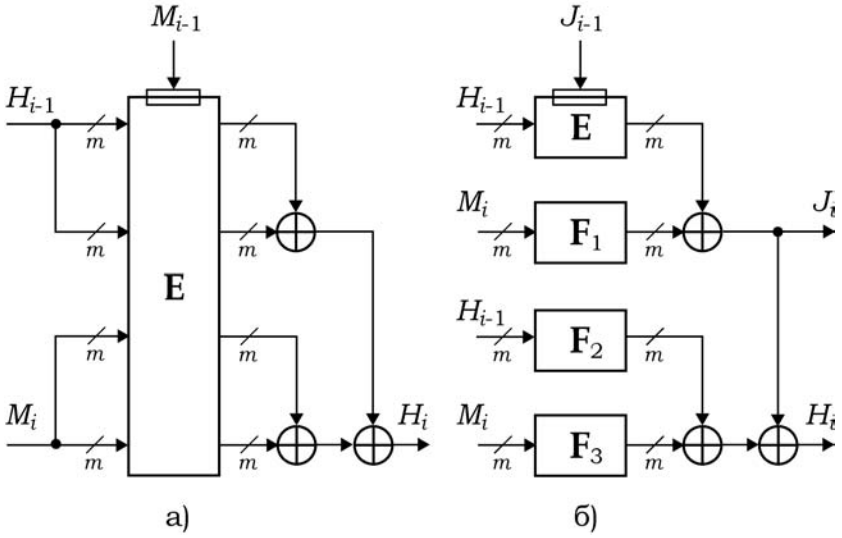

Рис. 9.13. Построение раундового преобразования с предварительным расширением преобразуемых значений M_i и H_{i-1}:
а) использование единого блочного преобразования,
б) использование четырех преобразований с m-разрядным входом

Особенностью схемы параллельного построения вычислений, показанной на рис. 9.13б, является использование двух значений раундовой функции H_i и J_i. Это дает возможность задать хэш-функцию с разрядностью $2m$ при использовании разбиения сообщения на m-битовые блоки данных. При этом стойкость к атакам на основе парадокса дней рождения будет определяться расширенным размером хэш-функции $H(M) = (H_n, J_n)$. Хэш-функции с расширенным входом в принципе позволяют получить более высокую производительность при повышении уровня сложности схемотехнической реализации.

На практике встречается и другая задача – встраивание функций хэширования в телекоммуникационные устройства при минимальной сложности реализации. В данном случае применение управляемых операций также представляет интерес, поскольку одновременно с экономией аппаратных ресурсов они обеспечивают доста-

точную скорость шифрования и хэширования. При решении задачи экономии аппаратных ресурсов, затрачиваемых на встраивание криптографических функций, представляет интерес реализация шифрования и хэширования с использованием одного и того же алгоритма, например, имеющего 128-битовый размер входа. Однако здесь возникает следующее противоречие. Для стойкого шифрования достаточно выбрать 64-разрядный шифр с длиной ключа не менее 128 бит, но разрядность хэш-функции должна быть не менее 128 бит, причем практически используются значения $m = 160$, $m = 256$ и даже $m = 512$. Последнее значение является уже избыточным, но первые два являются вполне разумными. Это означает, что, решая вопрос о реализации функций шифрования и хэширования данных, мы должны решить вопрос как с помощью блочного преобразования меньшей разрядности получить хэш-функцию большей разрядности. Решить данную задачу можно путем использования рассмотренных выше конструктивных схем с одновременным (параллельным) преобразованием отдельных частей хэш-функции. Но при таком подходе мы возвращаемся к использованию существенных дополнительных аппаратных ресурсов, т. е. к тому, чего мы хотим избежать. Поэтому поставленная цель может быть достигнута с помощью следующего технического решения.

Используется 128-битовый шифр с четным числом раундов. Преобразованный входной блок после выполнения половины раундов принимается в качестве первой части раундового значения хэш-функции (J_i), а после выполнения оставшихся раундов – в качестве второй части (H_i). Вариант такого построения показан на рис. 9.14. Следует иметь в виду, что использование двух последовательных значений H_n и H_{n-1} не приводит к существенному повышению стойкости. Действительно, они являются последовательными значениями одного и того же механизма сцепления, за счет которого обеспечивается влияние каждого бита сообщения на значение хэш-функции, поэтому атаку можно построить так, чтобы найти коллизию по значению H_{n-1} и взять в двух разных сообщениях одинаковый блок M_i. Если коллизия по H_{n-1} найдена, то таким способом обеспечится и совпадение значений H_n. В схеме, представленной на рис. 9.14, используются два независимых механизма сцепления, т. е. влияние всех битов хэшируемого сообщения накапливается в двух значениях H_i и J_i, связанных между собой сложным образом.

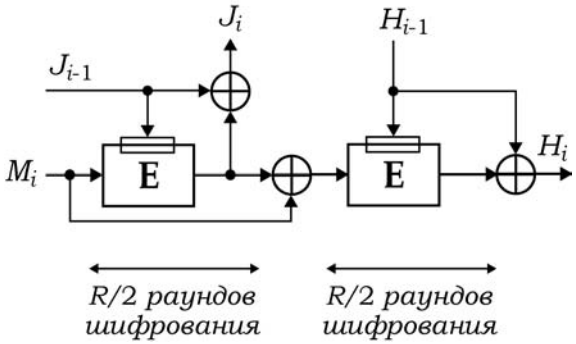

Рис. 9.14. Последовательная схема раундовой хэш-функции с расширенной разрядностью

9.7. Хэш-функции на основе выборок значений из таблицы в зависимости от преобразуемых данных

Рассмотрим некоторые особенности использования схем построения хэш-функций на основе блочных шифров в соответствии с вариантами табл. 9.1. Каждый раунд вычисления промежуточных значений хэш-функции связан с заменой значений, используемых вместо ключа шифрования. В случае использования шифров с простым расписанием ключа это не приводит к существенному уменьшению скорости хэширующего преобразования по сравнению с процедурой шифрования. Однако для шифров с предвычислениями падение производительности может быть весьма значительным. Это делает криптосистемы, включающие построение расширенного ключа большого размера, малоприменимыми для построения на их основе производительных хэш-функций. Эта проблема может быть решена модифицированием таких алгоритмов путем:

- включения в них дополнительного ключа, части которого непосредственно используются при шифровании;
- использования вместо расширенного ключа некоторой известной таблицы, которая теперь уже будет служить для выполнения операций подстановки или отображения (в зависимости от конкретной реализации).

При таком видоизменении анализ стойкости шифра должен быть выполнен заново. В случае программных шифров, рассмотренных в *главе 1* настоящей книги или в *главе 2* монографии [14], этот способ позволяет получить стойкие шифры. В них уже можно говорить не о выборке подключей в зависимости от преобразуемых данных, а о выборке табличных значений, зависящей от входного блока данных и от ключа. Использование известной и фиксированной таблицы псевдослучайных значений устраняет необходимость выполнения сравнительно длительных предвычислений, что позволяет использовать данные модифицированные шифры для построения программных хэш-функций, обладающих достаточно высокой производительностью.

Рассмотрим вариант такого шифра, построенный с использованием шифрующих преобразований, описанных в разделе 1 и примененных в *первой главе* для построения доказуемо недетерминированных шифров. Пусть имеется некоторая известная специфицированная псевдослучайная последовательность 32-битовых значений $Q(j)$: $\{Q(j)\}$, где $j = 0, 1, ..., 255$. Дополнительный 128-битовый ключ представлен в виде конкатенации четырех 32-битовых подключей K_i: $K = K_4||K_3||K_2||K_1$. Входной блок представлен как конкатенация четырех 32-битовых подблоков. Все 32-битовые слова (подблоки и подключи) V представляются в виде конкатенации четырех 8-битовых слов: $V = v_4||v_3||v_2||v_1$, где $V(v)$ обозначает переменные X, Y, Z и W (x, y, z и w).Запись $V^{>>>b}$ обозначает операцию циклического сдвига слова V вправо на b битов.

Программный алгоритм шифрования с фиксированным расписанием использования подключей

Вход 1: 128-битовый блок исходных данных $M = A||B||D||E$.

Вход 2: 128-битовый ключ $K = K_4||K_3||K_2||K_1$.

1. Установить число раундов преобразования $R = 3$, инициализировать счетчик $r := 1$ и переменные X, Y, Z, W:

$$X := A; \quad Y := B; \quad Z := D; \quad W := E.$$

2. Преобразовать переменные:

$$X := [(X + K_1) \bmod 2^{32}] \oplus Q(w_1);$$
$$Y := [(\mathrm{Y} \oplus K_2) + Q(x_1)] \bmod 2^{32};$$
$$Z := [(Z - K_3) \bmod 2^{32}] \oplus Q(y_1);$$
$$W := [(W \oplus K_4)^{>>>8} - Q(z_1)] \bmod 2^{32}.$$

3. Преобразовать переменные:

$$X := [(X \oplus K_3)^{>>>8} + Q(w_1)] \bmod 2^{32};$$
$$Y := [(Y + K_4) \bmod 2^{32}]^{>>>8} \oplus Q(x_1);$$
$$Z := [(Z \oplus K_1)^{>>>8} - Q(y_1)] \bmod 2^{32};$$
$$W := [(W - K_2) \bmod 2^{32}]^{>>>8} \oplus Q(z_1).$$

4. Преобразовать переменные:

$$X := [(X - K_4) \bmod 2^{32}]^{>>>8} \oplus Q(w_1);$$
$$Y := [(Y \oplus K_1)^{>>>8} - Q(x_1)] \bmod 2^{32};$$
$$Z := [(Z + K_2) \bmod 2^{32}]^{>>>8} \oplus Q(y_1);$$
$$W := [(W \oplus K_3)^{>>>8} + Q(z_1)] \bmod 2^{32}.$$

5. Преобразовать переменные:

$$X := [(X \oplus K_2)^{>>>Y} - Q(w_1)] \bmod 2^{32};$$
$$Y := [(Y + K_3) \bmod 2^{32}]^{>>>8} \oplus Q(x_1);$$
$$Z := [(Z \oplus K_4)^{>>>Y} + Q(y_1)] \bmod 2^{32};$$
$$W := [(W - K_1) \bmod 2^{32}]^{>>>8} \oplus Q(z_1).$$

6. Прирастить: $r := r + 1$;

7. Если $r < R$, то перейти к шагу 2, в противном случае СТОП.

Выход: 128-битовый блок шифртекста $C = X||Y||Z||W$.

Если данные подавать на вход 1, то мы будем иметь алгоритм шифрования, а если на вход 2 – то алгоритм блочного одностороннего преобразования. Преобразования такого типа обеспечивают возможность построения достаточно производительных программных хэш-функций.

Рассмотрим другой вариант построения раундовой хэш-функции, ориентированной на программную реализацию. Воспользуемся механизмом шифрования, описанным в *главе 2* книги [14] и основанным на выборке подключей в зависимости от пре-

образуемых данных с использованием аккумулирующих переменных. Последнее обеспечивает зависимость выбираемых подключей не только от текущего преобразуемого подблока данных, но и от всех подблоков, преобразованных на предыдущих шагах. Каждая аккумулирующая переменная реализует индивидуальный механизм сцепления, и ее значение на текущем шаге преобразования зависит от значений всех предыдущих подблоков. Особенно эффективен данный механизм при сравнительно больших значениях входных блоков данных. Это делает его удобным для построения хэш-функций, в которых мы хотим избежать процедуры предвычислений. С этой целью мы будем использовать некоторую известную псевдослучайную последовательность байтов $q[i]$, где $i = 0, 1, \dots, 2050$. При осуществлении раундовых преобразований будем использовать 32-битовые элементы $Q[j] = q[j+3]||q[j+2]||q[j+1]||q[j]$, где $j = 0, 1, \dots, 2047$. Благодаря такой выборке мы имеем возможность выбрать 2^{11} различных (в общем случае) 32-битовых значений, имея таблицу, включающую только 2051 байт. Она может храниться в памяти постоянно или генерироваться по некоторому алгоритму, например в соответствии с процедурой ***Table_Q***, описанной в [14].

Блоки данных M_i длины $b = 256$ бит будем рассматривать как конкатенацию 32-битовых слов $T_0, T_1, \dots, T_7$. Ключ шифрования представим в виде конкатенации пяти 32-битовых подключей: $K = K_5||K_4||K_3||K_2||K_1$. Следующий алгоритм осуществляет зашифрование 256-битовых блоков данных (алгоритм расшифрования легко составить, причем обеспечивается одинаковая производительность с алгоритмом зашифрования). Поскольку в нем отсутствуют предвычисления, то он может быть использован для построения хэш-функций.

Программный алгоритм шифрования

Алгоритм формирования раундовой хэш-функции состоит из следующих шагов:

Вход 1: 256-битовый блок данных $T_7||T_6||\dots||T_1||T_0$.

Вход 2: 160-битовый ключ $K_5||K_4||K_3||K_2||K_1$.

1. Инициализировать значение счетчика циклов $j := 4$ и вычислить начальные значения внутренних переменных $R := K_1$, $U := K_2$, $Y := K_3$, $V := K_4$, $N := K_5$. Инициализировать внутренний счетчик $i := 0$.
2. Выполнить процедуру ***Change_NVYU***.

   {*Change_NVYU*:

   - $N := N \oplus R$; $\quad V := V +_{32} N$;
   - $n := N +_{11} 0$; $\quad V := (V +_{32} Q[n])^{>>>11}$;
   - $N := N \oplus V$; $\quad Y := (Y +_{32} N)$;
   - $n := N +_{11} 0$; $\quad Y := (Y +_{32} Q[n])^{>>>11}$;
   - $N := N +_{32} Y$; $\quad n := N +_{11} 0$;
   - $U := [(U \oplus Q[n]) +_{32} R)]^{>>>V}$.

   END}

3. Осуществить преобразование очередного слова текста:
   $T_i := (T_i -_{32} V) \oplus U$.
4. Преобразовать переменную R: $R := R +_{32} T_i$.
5. Завершить преобразование слова T_i: $T_i := T_i^{<<<V} -_{32} Y$.
6. Прирастить $i := i + 1$. Если $i \neq 8$, то перейти к шагу 3.
7. Обменять значениями слова T_0 и T_7: $X := T_0$; $T_0 := T_7$; $T_7 := X$.
8. Уменьшить значение счетчика циклов: $j := j - 1$.
   Если $j \neq 0$, то перейти к шагу 1.

СТОП.

Выход 1: блок шифртекста $T_7||T_6||\ldots||T_1||T_0$.

Выход 2: 160-битовое значение переменной конкатенации $G = N||V||Y||U||R$.

При применении данного алгоритма для шифрования второй выход не используется. Он нужен при применении алгоритма в качестве блочного сжимающего преобразования. Значение G может служить переменной сцепления, подаваемой на вход для ключа. Раунд получаемой хэш-функции представлен на рис. 9.15а (заметим, что данная схема подвержена атаке методом «встреча посередине» и только большая разрядность хэш-функции делает эту атаку вычислительно сложной и практически нереализуемой). Отметим, что данный алгоритм легко видоизменяется под любой размер входа 1, кратный 32 битам. Заметим, что при переходе к последующему раунду хэширования не требуется ввод начальных значений переменных R, U, Y, V и N, поскольку они уже сформированы на последнем шаге блочного преобразования предыдущего раунда. При увеличении размера входного блока данных может быть существенно увеличена степень сжатия. В вариантах со сравнительно большой степенью сжатия представляет интерес объединять блоки данных со значением переменной конкатенации и подавать такой объединенный блок на вход блочного преобразования. Для хэширования данных, обрабатываемых с помощью ЭВМ, представляет интерес в одном раунде обрабатывать блоки данных размером 4096 бит, что соответствует размеру секторов на встроенном магнитном носителе. В случае сильного сжатия переменные R, U, Y, V и N могут инициализироваться нулевыми (или другими специфицированными) значениями или служить в качестве второй переменной сцепления. При использовании двух переменных сцепления появляются возможности разработки новых стойких схем построения хэш-функций. Одна из схем представлена на рис. 9.15б. В описанном выше механизме шифрования перед выполнением каждого внутреннего цикла преобразования переменные R, U, Y, V и N инициализируются заново, однако при хэшировании такой необходимости нет и на шаге 9 можно задать переход не к шагу 1, а к шагу 2. Это устранит необходимость хранить в памяти значение G_{i-1} до окончания преобразований i-го раунда хэширования. Возможны и другие видоизменения, в которых читатель может поупражняться самостоятельно. Однако целесообразно сохранение базового варианта механизма выборки подключей в зависимости от преобразуемых данных.

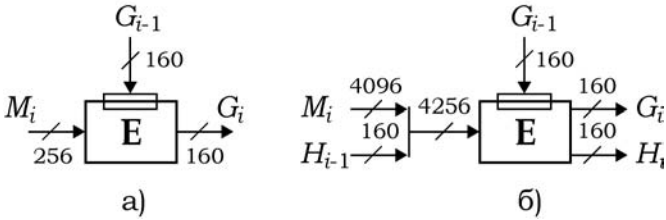

Рис. 9.15. Хэш-функция на основе сжимающего блочного преобразования: а – слабая степень сжатия, б – сильная степень сжатия

9.8. Алгоритмы формирования расширенного ключа на основе УППС

Кроме рассмотренных ранее хэш-функций, другим вариантом одностороннего преобразования, точнее – областью приложения хэш-функций, является алгоритм формирования расширенного ключа для блочных алгоритмов шифрования.

9.8.1. Принципы построения алгоритмов формирования расширенного ключа

Большинство современных алгоритмов блочного шифрования в качестве секретного параметра используют ключ небольшого размера – 128, 192, 256 бит [7, 26, 113]. Из данного ключа различными способами формируется массив подключей, элементы которого используются в каждом раунде шифрования. Существуют специальные типы криптоаналитических атак, направленные на алгоритмы формирования подключей, к числу которых относятся поиск эквивалентных, частично эквивалентных, слабых и других видов подключей.

В связи с этим одним из требований, предъявляемых к построению блочных шифров, является наличие стойкого алгоритма (процедуры) расширения ключа [81]. Усложнение алгоритма формирования расширенного ключа (АФРК) может привести к повышению или понижению стойкости всего шифра, но при этом затруднить его эффективную реализацию. Поэтому в работе [58] рассматривается специальная классификация шифров в зависимости от используемого типа АФРК.

В работе [89] с целью построения стойких процедур генерации расширенного ключа предложено использовать однонаправленные функции. Однако скорость их выполнения недостаточно высока ввиду применения большого количества итераций. Следовательно, разработка стойких высокоскоростных процедур формирования расширенного ключа является одной из актуальных задач практической криптографии.

В данном разделе рассматривается алгоритм генерации расширенного ключа в виде последовательности раундовых подключей $S = Q^{(1)} \| Q^{(2)} \| \ldots \| Q^{(r)}$, использующий оригинальную однонаправленную функцию $\alpha(\bullet)$, построенную на основе БУП или блоков УППС. Функциональная схема алгоритма представлена на рис. 9.16 [13].

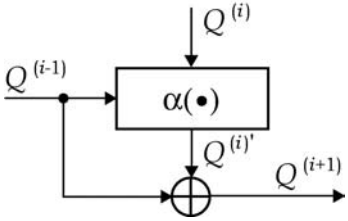

Рис. 9.16. Однораундовый алгоритм формирования подключа

В соответствии со схемой не только каждое выходное значение $Q^{(i+1)}$ алгоритма в i-ом раунде зависит от двух предыдущих, но и каждое выходное значение $Q^{(i)'}$ функции **α(•)** также зависит от значения подключей текущего и предыдущего раундов $(Q^{(i)}, Q^{(i-1)})$.

Модель функции **α(•)** представлена на рис. 9.17. Входом функции **α(•)** является 256-битовый подключ $Q^{(i)}$ текущего раунда шифрования, который можно представить в виде конкатенации четырех двоичных векторов одинаковой длины, то есть $Q^{(i)} = K^{(i)}_1 \| K^{(i)}_2 \| K^{(i)}_3 \| K^{(i)}_4$.

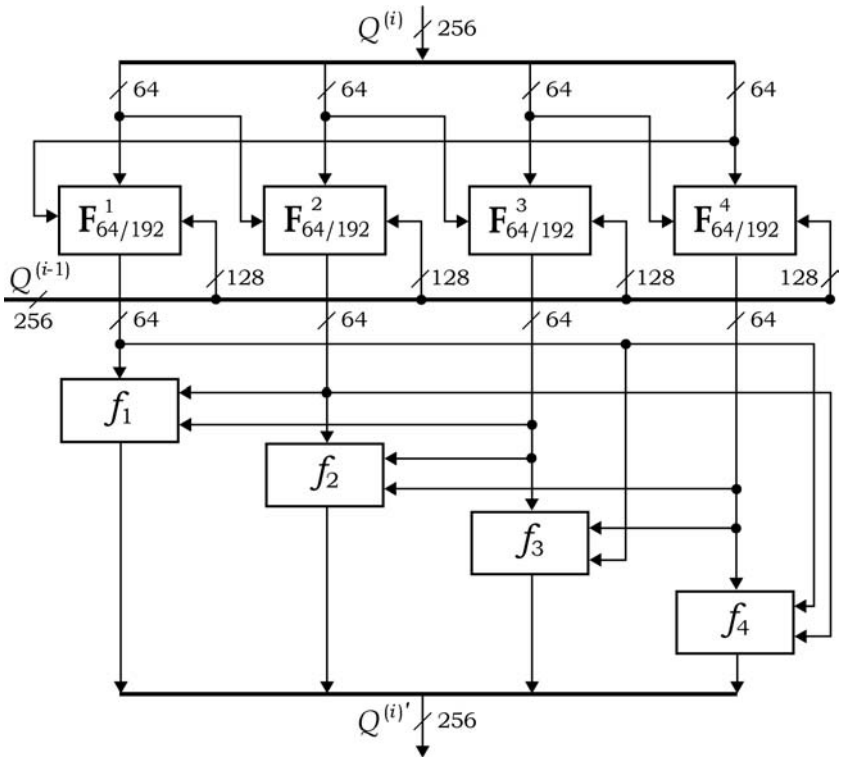

Рис. 9.17. Модель функции **α(•)**

Над каждым вектором $K^{(i)}_j$ (j=1, …, 4) осуществляется преобразование $\mathbf{F}^j$ (см. рис. 9.17), зависящее от 192 битового вектора $V^{(j)}$, значение которого для каждого j различное. Вектор $V^{(j)}$ формируется путем конкатенации одного из входных векторов $K^{(i)}{}_s$ ($s \neq j$) и 128 бит подключа $Q^{(i-1)}$, где $Q^{(i-1)} = K^{(i-1)}{}_1 \,||\, K^{(i-1)}{}_2 \,||\, K^{(i-1)}{}_3 \,||\, K^{(i-1)}{}_4$, а именно: $V^{(j)} = K^{(i)}{}_s \,||\, K^{(i-1)}{}_u \,||\, K^{(i-1)}{}_v$. В целях обеспечения равномерного влияния всех битов подключей текущего и предыдущего раундов шифрования индексы s, u и v выбираются в соответствии с расписанием табл. 9.2.

Таблица 9.2

F^1	$V_1 = K^{(i)}{}_4, K^{(i-1)}{}_1, K^{(i-1)}{}_3$
F^2	$V_2 = K^{(i)}{}_1, K^{(i-1)}{}_2, K^{(i-1)}{}_4$
F^3	$V_3 = K^{(i)}{}_2, K^{(i-1)}{}_3, K^{(i-1)}{}_4$
F^4	$V_4 = K^{(i)}{}_3, K^{(i-1)}{}_1, K^{(i-1)}{}_2$

Каждое преобразование $\mathbf{F}^j$ представляет собой управляемую перестановочную или подстановочно-перестановочную сеть. Ниже рассматривается конкретная модель функции $\boldsymbol{\alpha}(\bullet)$, построенная с использованием шестислойных блоков управляемых перестановок $\mathbf{P}_{64/192}$, в котором первые три слоя являются объединением восьми блоков $\mathbf{P}_{8/12}$, а остальные три – объединением восьми блоков $\mathbf{P}^{-1}{}_{8/12}$. Коммутация между блоками $\mathbf{P}_{8/12}$ и $\mathbf{P}^{-1}{}_{8/12}$ формируется по принципу «каждый с каждым», т.е. k-й разряд выхода l-го блока $\mathbf{P}_{8/12}$ коммутируется с l-м разрядом входа k-го блока $\mathbf{P}^{-1}{}_{8/12}$ при соответствующей нумерации блоков $\mathbf{P}^{-1}{}_{8/12}$. Более подробно построение блоков $\mathbf{P}_{8/12}$ и $\mathbf{P}^{-1}{}_{8/12}$ рассмотрено в работе [15]. Блок $\mathbf{P}_{64/192}$, построенный подобным образом, является блоком первого порядка, который обеспечивает отображение любого входного бита в любой выходной. При такой структуре блока управляющий вектор $V^{(j)}$, например, для $\mathbf{F}^1$ имеет вид: $V^{(1)} = (V^{(1)}{}_1 \,||\, \ldots \,||\, V^{(1)}{}_6) = (K^{(i)}{}_4 \,||\, K^{(i-1)}{}_1 \,||K^{(i-1)}{}_3)$, где $V^{(1)} \in \mathrm{GF}(2)^{192}$, $V^{(1)}{}_1, \ldots, V^{(1)}{}_6 \in \mathrm{GF}(2)^{32}$, $K^{(i)}{}_4, K^{(i\text{-}1)}{}_1, K^{(i-1)}{}_3 \in \mathrm{GF}(2)^{64}$.

Таким образом, j-й блок $\mathbf{F}^j$ реализует преобразование следующего вида $Y_j = \mathbf{P}_{64/192}(K^{(i)}{}_j, V^{(j)})$. Полученные значения векторов Y_j дополнительно преобразуются с использованием логических функций от трех переменных, а именно:

$$Y_1 = \boldsymbol{f}_1(Y_1, Y_2, Y_3) = (Y_1 \wedge Y_2) \vee (\overline{Y_1} \wedge Y_3);$$

$$Y_2 = \boldsymbol{f}_2(Y_2, Y_3, Y_4) = Y_2 \oplus Y_3 \oplus Y_4;$$

$$Y_3 = \boldsymbol{f}_3(Y_3, Y_4, Y_1) = (Y_3 \wedge Y_4) \vee (Y_3 \wedge Y_1) \vee (Y_4 \wedge Y_1);$$

$$Y_4 = \boldsymbol{f}_4(Y_4, Y_1, Y_2) = Y_4 \oplus Y_1 \oplus Y_2,$$

причем используются функции алгоритма хэширования SHA [113]. Кроме функций данного типа, целесообразнее использовать векторные БФ, подобные тем, что исследованы в разделе 8 при синтезе шифров, с учетом оптимизации их математических свойств (нелинейности и корреляционной иммунности).

Функции $\boldsymbol{f}_1$ и $\boldsymbol{f}_3$ выполняются не более чем за три последовательных операции типа XOR. Поскольку блок $\mathbf{P}_{64/192}$ выполняется за шесть последовательных операций типа XOR, то вся процедура генерации одного раундового подключа, с учетом заключительной операции XOR, требует не более десяти тактов.

В целом, алгоритм процедуры расширения ключа выглядит следующим образом:

Вход: значение секретного ключа Q_c длиной 256 бит;

1. Установить начальные значения: $i := 0$, $Q^{(i-1)} = Q^{(i)} = Q_c$, $r := R$ (значение R определяется количеством раундов шифрования в используемом блочном криптоалгоритме);
2. Выполнить преобразование $Q^{(i+1)} = \boldsymbol{\alpha}(Q^{(i)}, Q^{(i-1)}) \oplus Q^{(i-1)}$;
3. Если $i < r - 1$, тогда $i := i + 1$, перейти к шагу 3;

Выход: последовательность раундовых подключей $S = Q^{(1)} \| Q^{(2)} \| \ldots \| Q^{(r)}$.

9.8.2. Исследование свойств АФРК

Для оценки качества процедур генерации раундовых подключей использовались методы, основанные на статистических свойствах входных и выходных ключевых последовательностей. В частности, исследовались:

- статистическая независимость подключей между собой, то есть невозможность восстановления всех подключей при знании одного или нескольких подключей;
- высокая сложность восстановления секретного ключа при знании всех или нескольких раундовых подключей.

При определении количественных показателей применялись: критерий лавинного эффекта [111] и частотный тест проверки равномерности распределения битов в сгенерированной ключевой последовательности [26].

Анализ лавинного эффекта заключался в подсчете количества битов в генерируемой ключевой последовательности, изменившихся при изменении одного бита в секретном ключе. В соответствии с частотным тестом вычислялось значение $f(S)$, согласно выражению

$$f(S) = \frac{2}{n\sqrt{m}} \sum_{j=1}^{n} \left(\sum_{i=1}^{m} s_i - \frac{m}{2} \right), \tag{9.1}$$

где $s_i \in S$, $m = 256r$ – длина ключевой последовательности, n – количество испытаний. Значение статистики $f(S)$ при достаточно больших значениях m является оценкой нормально распределенной случайной величины с параметрами N(0, 1). Данный тест позволяет проверить равномерность распределения битов в сгенерированной последовательности подключей. Для уровня значимости $\beta = 0.01$ значение $|f(S)|$ не должно превышать 2,57, а для $\beta = 0.001$ – значения 3,2 [5, 30].

Анализ независимости подключей между собой осуществлялся с использованием автокорреляционного теста. Данный тест позволяет выявить наличие линейных зависимостей между сгенерированной последовательностью S и ее сдвигом на фиксированное значение τ. Для последовательности $S = s_1, s_2, \ldots, s_m$ вычислялись вероятности появления единичного бита в последовательностях $s_1 \oplus s_{1+\tau}, s_2 \oplus s_{2+\tau}, \ldots, s_m \oplus s_{m+\tau}$ при значениях τ от 1 до m–1, где $m = 256$. Чем меньше отклонение значения вероятности от значения 0.5, тем меньше зависимость между битами раундовых подключей, а также между массивом раундовых подключей и секретным ключом.

В табл. 9.3 представлены результаты тестирования применительно к процедуре расширения ключа для различных типов функций $\alpha(\bullet)$. Тестирование осуществлялось методом Монте-Карло для количества секретных ключей #**Q** = 1000. Количество раундов $R = 10$.

Таблица 9.3

№ п/п	Модель функции $\alpha(\bullet)$	Лавинный эффект	Частотный тест	Автокорреляционный тест
1	Только функции $\boldsymbol{f}_1$–$\boldsymbol{f}_4$	0.006	375.5	0.482
2	Функции $\boldsymbol{f}_1$–$\boldsymbol{f}_4$ и $Q^{(i-1)} \oplus Q^{(i)'}$	0.007	–215.4	0.485
3	Только $\mathbf{F}^1$–$\mathbf{F}^4$	0.314	1.328	0.49996
4	$\mathbf{F}^1$–$\mathbf{F}^4$ и $Q^{(i-1)} \oplus Q^{(i)'}$	0.343	–4.03	0.49988
5	$\mathbf{F}^1$–$\mathbf{F}^4$, функции $\boldsymbol{f}_1$–$\boldsymbol{f}_4$ и $Q^{(i-1)} \oplus Q^{(i)'}$	0.409	0.809	0.50004

Из табл. 9.3 видно, что первые две схемы не удовлетворяют критериям лавинного эффекта и равномерности (частотный тест). Для третьей схемы значения лавинного эффекта сравнимы с результатами некоторых процедур расширения ключа [39]. Четвертая схема не подходит по критерию частотного теста. Пятая схема, описание которой представлено выше, обладает наилучшими показателями по всем рассматриваемым критериям.

Более точное представление о мере зависимости битов раундовых подключей относительно битов других подключей или секретного ключа можно получить путем определения корреляционных функций и коэффициента корреляции между этими ключами. Коэффициент корреляции $\rho_{i,i-1}$ между двумя ключами $Q^{(i)}$, $Q^{(i-1)}$ можно вычислить согласно выражению [28]

$$\rho_{i,i-1} = \frac{n\sum_{j=0}^{n-1} Q_j^{(i)} Q_j^{(i-1)} - \left(\sum_{j=0}^{n-1} Q_j^{(i)}\right)\left(\sum_{j=0}^{n-1} Q_j^{(i-1)}\right)}{\sqrt{\left(n\sum_{j=0}^{n-1}\left(Q_j^{(i)}\right)^2 - \left(\sum_{j=0}^{n-1} Q_j^{(i)}\right)^2\right)\left(n\sum_{j=0}^{n-1}\left(Q_j^{(i-1)}\right)^2 - \left(\sum_{j=0}^{n-1} Q_j^{(i-1)}\right)^2\right)}}, \qquad (9.2)$$

где $n = 256$. Значение коэффициента корреляции $|\rho_{i,\,i-1}| \le 1$, при этом, если $\rho_{i,\,i-1} \approx 0$, тогда величины $Q_j^{(i)}$ и $Q_j^{(i-1)}$ независимы, если же $|\rho_{i,i-1}| \approx 1$ – тогда биты ключей связаны между собой линейной зависимостью. Значение коэффициента корреляции должно находиться в пределах

$$(\mu_n - 2\sigma_n, \mu_n + 2\sigma_n), \qquad (9.3)$$

где

$$\mu_n = -\frac{1}{n-1},\ \sigma_n = \frac{1}{n-1}\sqrt{\frac{n(n-3)}{n+1}}. \qquad (9.4)$$

При $n = 256$ получим $\mu_n \approx -0.004$, $\sigma_n \approx 0.06$, $\mu_n - 2\sigma_n = -0.13$, $\mu_n + 2\sigma_n = 0.12$.

Значение коэффициента корреляции определялось имитационным моделированием для всех пар ключей, включая секретный. С учетом необходимого числа раундовых ключей R количество таких пар равно $\binom{R+1}{2} = \frac{R\cdot(R+1)}{2}$. Все пары были упорядочены по возрастанию значений $\eta(i, j) = Ri + j$, где i, j – номера ключей ($1 \le i < j \le R$). Объем выборки секретных ключей, как и в предыдущем случае, равен 1000, число раундов $R = 12$. При данных начальных условиях последовательно проанализировано 78 пар ключей для двух вариантов модели функции $\boldsymbol{\alpha}(\bullet)$: с БУП (УППС) $\mathbf{F}^1$**–** $\mathbf{F}^4$ и без них. Исследования показали, если функция $\boldsymbol{\alpha}(\bullet)$ использует БУП (УППС), тогда коэффициент корреляции $\rho_{i,\,i-1}$ для всех возможных пар ключей попадает в диапазон значений (– 0.13; 0.12), что продемонстрировано на рис. 9.18.

Если БУП (УППС) отсутствуют, то в большинстве случаев значение коэффициента корреляции превышает заданный диапазон (рис. 9.19).

Полученные результаты свидетельствуют, что наибольшее влияние на все показатели оказывают БУП (УППС). Добавление БФ $\boldsymbol{f}_1 - \boldsymbol{f}_4$ и заключительной операции XOR позволяет увеличить рассеивание битов и понизить усредненное значение частотного теста.

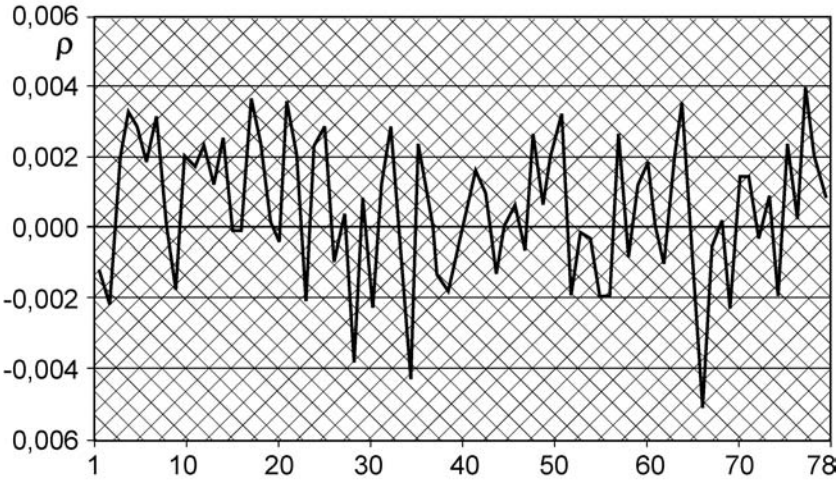

Рис.9.18. Коэффициент корреляции при использовании БУП

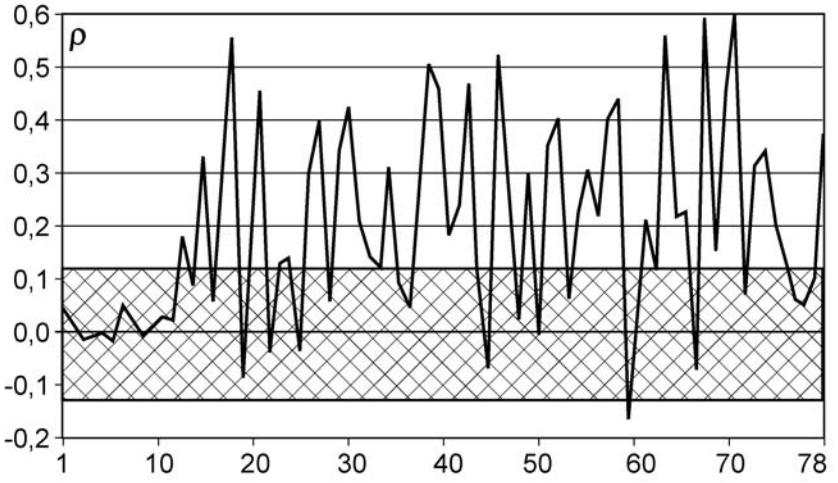

Рис.9.19. Коэффициент корреляции без использования БУП

Дополнительно, с целью проверки рассеивающих свойств алгоритма расширения ключа, тестирование проводилось по методике, предложенной в работе [106] для оценки качества алгоритмов блочного шифрования. При этом оценивались следующие показатели:

- среднее число битов сгенерированных подключей, изменяющихся при изменении одного бита секретного ключа – d_1;

- степень полноты преобразования – d_2;
- степень лавинного эффекта – d_3;
- степень соответствия строгому лавинному критерию – d_4.

Данные показатели позволяют выявить возможные слабости АФРК, которые могут быть использованы при применении разностного и корреляционного методов анализа. При их использовании формируются две матрицы: матрица зависимостей **A** и матрица расстояний **B**, элементы которых заданы следующим образом:

$$a_{ij} = \#\{Q \in \mathrm{Q} \mid (f(Q^{(i)}))_j \neq (f(Q))_j\},$$

$$b_{ij} = \#\{Q \in \mathrm{Q} \mid w(f(Q^{(i)}) \oplus f(Q)) = j\},$$

где $w(Q)$ – число единичных компонент вектора Q (вес Хемминга), а $Q^{(i)}$ – вектор, полученный инвертированием i-го бита вектора Q. Матрица **A** отражает зависимость j-го разряда выходного вектора от i-го разряда входного вектора. Матрица **B** является маркировкой веса Хэмминга лавинных векторов.

Значения $d_1 – d_4$ вычислялись по формулам:

$$d_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{m} j b_{ij}}{\#\mathbf{Q}}, \quad d_2 = 1 - \frac{\#\{(i,j) \mid a_{ij} = 0\}}{nm},$$

$$d_3 = 1 - \frac{\sum_{i=1}^{n}\left|\frac{1}{\#\mathbf{Q}}\sum_{j=1}^{m} 2 j b_{ij} - m\right|}{nm}, \qquad d_4 = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\frac{2a_{ij}}{\#\mathbf{Q}} - 1\right|}{nm}.$$

По результатам тестирования получена табл. 9.4, в которой проанализированы два варианта алгоритма формирования расширенного ключа. Первый вариант соответствует алгоритму, представленному на рис. 9.16. Второй вариант является модификацией первого (рис. 9.20), но отличается от него тем, что управляющий вектор функции $\alpha(\bullet)$ есть сумма вектора $Q^{(i-1)}$ и константы C_0.

Таблица 9.4

Номер ключа (раунда)	Схема 1				Схема 2			
	d_1	d_2	d_3	d_4	d_1	d_2	d_3	d_4
10	128,009	1,000000	0,998453	0,974784	128,003	1,000000	0,998406	0,974782
9	127,992	1,000000	0,998422	0,974831	128,002	1,000000	0,998390	0,974633
8	127,985	1,000000	0,998519	0,974707	128,001	1,000000	0,998411	0,974820
7	127,960	1,000000	0,998408	0,974851	127,988	1,000000	0,998459	0,974706
6	128,002	1,000000	0,998279	0,974697	128,007	1,000000	0,998501	0,974783

Номер ключа (раунда)	Схема 1				Схема 2			
	d_1	d_2	d_3	d_4	d_1	d_2	d_3	d_4
5	127,951	1,000000	0,998305	0,974651	127,925	1,000000	0,998318	0,974893
4	125,195	1,000000	0,978085	0,964997	125,185	1,000000	0,978007	0,964934
3	97,449	1,000000	0,761325	0,759495	97,591	1,000000	0,762428	0,760619
2	38,027	1,000000	0,297083	0,291320	38,158	1,000000	0,298111	0,292352
1	9,346	0,925323	0,073018	0,065582	9,361	0,925339	0,073130	0,065704

Результаты показывают, что предложенные алгоритмы расширения ключа обеспечивают достаточно сильное рассеивающее влияние каждого из битов секретного ключа на все биты сгенерированных подключей. Однако введение константы в схему алгоритма к существенному улучшению его статистических свойств не приводит.

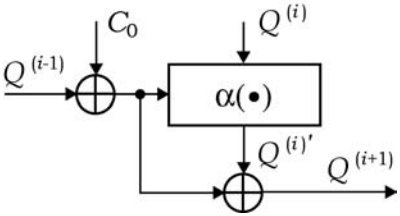

Рис. 9.20. Модифицированный одноразундовый АФРК

В целом, значения критериев для обоих алгоритмов соответственно равны $d_1 \approx \frac{1}{2}n$, $d_2 = 1$, $d_3 \approx 1$, $d_4 \approx 1$ для всех подключей, начиная с $i = 3$. Поэтому по возможности необходимо использовать последовательность без подключей $Q^{(1)}$ и $Q^{(2)}$.

Следует отметить, что предложенные АФРК, удовлетворяют основным критериям, предъявляемым к построению такого рода алгоритмов, и обладает высокой скоростью формирования выходной последовательности раундовых подключей. В первую очередь это обусловлено тем, что в качестве базового преобразования используются БУП (блок УППС), позволяющие выполнять преобразование над векторами большой длины (n=64), благодаря чему достигается высокая скорость распространения влияния битов секретного ключа на каждый бит генерируемой последовательности расширенного ключа. Реализация такой процедуры в виде компонента современной микросхемы не приведет к существенному усложнению криптосистемы, а также к значительному возрастанию стоимостных затрат. Кроме того, существующие схемы распределения раундовых подключей в блочных алгоритмах позволяют частично распараллелить процессы шифрования и генерации ключевой последовательности, что позволяет дополнительно повысить скорость шифрования в режиме частой смены ключей.

ГЛАВА 10

Криптографический практикум

10.1. Задания для практических занятий

Предлагаемые темы для выполнения практических работ направлены на закрепление материала, относящегося к двухключевой криптографии – системе открытого распределения ключей Диффи–Хеллмана, открытому шифрованию и системам цифровой электронной подписи. Для выполнения данных практических работ требуется программа, реализующая арифметические операции, алгоритм возведения в дискретную степень по модулю и вычисление мультипликативно обратного элемента в поле вычетов. Программа, включающая указанные функции, может быть разработана студентами в рамках самостоятельного задания на курсовое проектирование или на практических занятиях. При небольших длинах чисел составление такой программы обычно является посильной задачей.

При выполнении заданий на практических занятиях по открытому шифрованию и открытому распределению ключей целесообразно акцентировать внимание слушателей на сходство системы открытого распределения ключей Диффи-Хеллмана и алгоритма открытого шифрования Эль-Гамаля, которое заключается в том, что в обоих случаях используется открытый ключ отправителя. Во втором случае он направляется получателю как часть криптограммы. Специфика открытого шифрования Эль-Гамаля состоит в использовании общего секрета K (т. е. ключа, формируемого по открытому ключу отправителя и секретному ключу получателя) в качестве разового ключа шифрования, осуществляемого путем модульного умножения блока открытого текста на K.

При выполнении практической работы по ЭЦП Эль-Гамаля целесообразно подчеркнуть важность сохранения генерируемого случайного числа в тайне, поскольку знание этого числа, которое используется при генерации цифровой подписи, позволяет вычислить секретный ключ. Целесообразно также подчеркнуть, что система цифровой подписи Эль-Гамаля основана на идеях Диффи-Хеллмана и является существенным ее развитием.

При выполнении практической работы по «слепой подписи» Чаума следует акцентировать внимание обучаемых на том, что образец подписи к сообщению M должен храниться в секрете от подписывающего, поскольку его знание позволяет подписывающему легко вычислить значение M.

10.1.1. Открытое шифрование

Тема: "Система открытого распределения ключей Диффи – Хеллмана".

Теоретическая часть. В данной криптосистеме каждый абонент выбирает случайный секретный ключ x и вырабатывает открытый ключ y в соответствии с формулой

$$y = \alpha^x \pmod p.$$

Все абоненты размещают свои открытые ключи в общедоступном справочнике, который должен быть заверен специально созданным доверительным центром, чтобы исключить возможные нападения путем подмены открытых ключей или навязывания ложных открытых ключей. Если два абонента А и В хотят установить секретную связь, то они поступают следующим образом. Абонент А берет из справочника открытый ключ абонента В и, используя свой секретный ключ, вычисляет общий секретный ключ:

$$Z_{AB} = (y_B)^{x_A} = (\alpha^{x_B})^{x_A} = \alpha^{x_B x_A} \pmod p,$$

где y_A и y_B – открытые ключи абонентов А и В; x_A и x_B – соответствующие секретные ключи. Общий секретный ключ Z_{AB} нет необходимости передавать по сети связи, поскольку абонент В по известному из справочника открытому ключу абонента А аналогичным способом вычисляет значение

$$Z_{AB} = (y_A)^{x_B} = (\alpha^{x_A})^{x_B} = \alpha^{x_B x_A} \pmod p.$$

Предполагается, что оппоненту (потенциальному нарушителю) могут быть известны значения $y_B = \alpha^{x_B} \pmod p$ и $y_A = \alpha^{x_A} \pmod p$, передаваемые по открытому каналу, но для того чтобы вычислить Z_{AB}, он должен решить трудную задачу дискретного логарифмирования. Общий секрет Z_{AB} может использоваться абонентами для шифрования сеансовых секретных ключей, а последние – для шифрования сообщений с использованием симметричных методов шифрования.

Экспериментальная часть. Преподаватель задает обучаемым индивидуальные значения модуля p и параметра α. Для предполагаемых 10 пользователей обучаемые выбирают 10 значений секретного ключа $x_1, x_2, \ldots, x_{10}$. Вычисляют соответствующие им открытые ключи $y_1, y_2, \ldots, y_{10}$. Для всех возможных пар значений (i,j), где $i = 1, 2, \ldots, 10$ и $j = 1, 2, \ldots, 10$, вычисляется общий секретный ключ Z_{ij}. Полученные результаты оформляются в виде таблицы. Проверяется выполнение условия $Z_{ij} = Z_{ji}$.

Тема: "Вычисление мультипликативно обратных элементов в поле вычетов".

Теоретическая часть. Для вычисления обратных элементов в поле вычетов используется расширенный алгоритм Евклида. Известно, что если числа n и x являются взаимно простыми и $x < n$, то для x существует единственное значение x' – такое, что

$x' < n$ и $xx' = 1 \pmod n$. Это число называется мультипликативно обратным элементом в поле вычетов по модулю n и обозначается как $x^{-1} \pmod n$. Используя теорему Эйлера, можно записать

$$x^{\varphi(n)-1}x = x^{-1}x = 1 \pmod n,$$

откуда получаем

$$x^{\varphi(n)-1} = x^{-1} \pmod n,$$

т.е. мультипликативно обратный элемент можно вычислить по значению функции Эйлера.

Экспериментальная часть. Выполняется вычисление мультипликативно обратных элементов для ряда чисел x_i , $i = 1, 2, \ldots 10$, для трех простых модулей p_1, p_2, p_3 и трех составных модулей n_1, n_2 и n_3. Вычисления выполняются двумя способами: 1) с использованием расширенного алгоритма Евклида и 2) с использованием формулы $x^{\varphi(n)-1} = x^{-1} \pmod n$. Составляется сопоставительная таблица, показывающаяся идентичность результатов вычисления по двум способам.

Тема: "Открытое шифрование Эль-Гамаля".

Теоретическая часть. Способ открытого шифрования Эль-Гамаля включает в себя составной частью систему открытого распределения ключей Диффи-Хеллмана. Каждый пользователь секретной сети выбирает секретный ключ x, вычисляет свой открытый ключ $y = \alpha^x$ и помещает y в заверенный справочник. Шифрование сообщения T, отправляемого пользователю i, осуществляется с помощью следующего алгоритма:

- выбрать случайное число R;
- вычислить $C' = \alpha^R \pmod p$, которое по своей сути является разовым открытым ключом;
- используя открытый ключ i-го пользователя, вычислить $C'' = y^R T \pmod p$;
- отправить блок шифртекста (C', C'') пользователю i.

Расшифрование осуществляется пользователем i по следующему алгоритму:

- вычислить значение $(C')^x = (\alpha^R)^x = \alpha^{Rx} \pmod p$, которое по своей сути является разовым общим секретом (Z_{AB}) получателя и отправителя;
- вычислить значение $Z^{-1} = (\alpha^{Rx})^{-1} \pmod p$;
- расшифровать криптограмму C'': $T = C''Z^{-1} \pmod p$.

Экспериментальная часть. По указанию преподавателя обучающимся индивидуально задаются значения простого числа p и параметра α. Студенты формируют сек-

ретный ключ x_A и, используя заданные значения p и α, вычисляют открытый ключ y_A. Используя открытый ключ, осуществляется зашифрование 10 различных сообщений, фиксируя для каждого из них значения R, R^{-1}, C', C'', Z, Z^{-1}. Проверяется правильность расшифрования сообщений. Полученные результаты оформляются в виде таблицы.

Тема: "Открытое распределения ключей с использованием криптосистемы RSA".

Теоретическая часть. В криптосистеме RSA сеансовые ключи шифруются по открытому ключу получателя и распределяются по открытому каналу. Процедура зашифрования выражается формулой:

$$C = K^d \pmod{n}.$$

Получатель расшифровывает сеансовый ключ с использованием своего секретного ключа:

$$K = C^e \pmod{n}.$$

Экспериментальная часть. Используя заданные значения простых чисел p и q, вычислить модуль n, затем сформировать открытый и закрытый ключи e и d. Используя открытый ключ, зашифровать 10 различных ключей и, используя закрытый ключ, осуществить процедуру их расшифрования. Проверить корректность расшифрования. Результаты оформить в виде таблицы.

10.1.2. Системы цифровой подписи

Тема: "Электронная цифровая подпись Эль-Гамаля".

Теоретическая часть. Пусть для абонента А имеем секретный ключ x_A и открытый ключ $y_A = \alpha^{x_A}$. Подписью абонента A под документом M, где $M < p$ служит пара чисел (r,s), где $0 \le r < p - 1$ и $0 \le s < p - 1$, которая удовлетворяет уравнению

$$\alpha^M = y_A^r r^s \pmod{p}.$$

Это *уравнение проверки подписи абонента* А. Данная система ЭЦП основана на том, что только действительный владелец секретного ключа a может выработать пару чисел (r, s), удовлетворяющую уравнению проверки подписи, по следующему алгоритму:

1. Сгенерировать случайное число k, удовлетворяющее условию:
   $0 < k < p - 1$ и НОД$(k, p - 1) = 1$.

2. Вычислить $r = \alpha^k \pmod p$.
3. Вычислить s из уравнения $M = x_A r + ks \pmod{(p-1)}$.

Из теории чисел известно, что последнее уравнение имеет решение для s, если НОД(k, $p-1$) = 1. Это уравнение легко получить путем подстановки в уравнение проверки подписи значения $r = \alpha^k \bmod p$:

$$\alpha^M = \alpha^{x_A r} \alpha^{ks} = y_A^r r^s \pmod p.$$

Следует отметить, что для данного сообщения может быть выработано большое число различных подписей, соответствующих различным k. Однако выработать правильную подпись может только владелец секретного ключа.

Особенностью данной ЭЦП является то, что одно и то же значение k не допускается использовать для формирования подписи для двух разных сообщений, поскольку это делает возможным вычисление секретного ключа. Использованные значения k должны храниться в секрете. Обычно после выработки подписи они уничтожаются.

Экспериментальная часть. Используя заданные значения простого числа p и параметра α, сформировать секретный ключ x_A, вычислить соответствующий ему открытый ключ y_A и вычислить значение электронной подписи для 10 различных сообщений, фиксируя получаемые значения параметров k, $r = \alpha^k \pmod p$, s, α^M, y_A^r, r^s, $y_A^r r^s$. Осуществить проверку подписи по открытому ключу. Результаты оформить в виде таблицы.

Тема: "Нахождение чисел, относящихся к заданному показателю ".

Теоретическая часть. Для любого числа a, взаимно простого с модулем m, существуют числа δ, такие что $a^\delta = 1 \bmod p$. Минимальное из чисел δ, т.е. число $\gamma = \min\{\delta\}$, называется показателем числа a. Если a по модулю m принадлежит показателю δ, то числа $a^0, a^1, \ldots, a^{\delta-1}$ по модулю m несравнимы. Показателями могут быть только делители числа $p-1$. Если модуль является простым числом p, то число чисел $\psi(\gamma)$, относящихся к показателю γ, равно функции Эйлера от числа γ, т.е. $\psi(\gamma) = \varphi(\gamma)$. Если длина числа γ существенно меньше длины простого модуля, то нахождение чисел, относящихся к показателю γ, путем случайного выбора чисел a и проверки соотношения $a^\delta = 1 \bmod p$ является вычислительно неэффективным. В реальных системах ЭЦП с сокращенной длиной подписи простой модуль $p-1$ имеет размер 1024 или 2048 бит, а размер показателя γ составляет около 160 бит. Для нахождения числа α, относящегося к показателю γ, используется следующий вычислительно эффективный способ.

1. Выбирается число a, превосходящее 1 и меньшее числа p.
2. Вычисляется значение $\gamma' = (p-1)/\gamma$ и число $g = a^{\gamma'} \bmod p$.

3. Если $g \neq 1$, то в качестве числа α взять число g. В противном случае повторить шаги 1 – 3.

Действительно для полученного числа $\alpha \neq 1$ имеем $\alpha = a^{(p-1)/\gamma} \bmod p$. Следовательно, согласно теореме Ферма, имеем $\alpha^{\gamma} = a^{(p-1)} = 1 \bmod p$, т. е. число α относится к показателю γ.

Экспериментальная часть. Задаются несколько простых чисел p. Для каждого из них требуется найти разложение числа $p-1$, выбрать несколько значений в качестве показателей и для каждого из показателей найти несколько чисел, относящихся к нему, по модулю p. Другим вариантом является следующее задание. Требуется сформировать большое простое число p, для которого можно найти разложение $p-1$. Для модуля p находятся несколько чисел, относящихся к показателям, в качестве которых берутся делители $p-1$.

Тема: "Цифровая подпись Эль-Гамаля с сокращенной длиной параметра s".

Теоретическая часть. Уравнение проверки подписи

$$\alpha^{M} = y_A^{r} r^{s} \pmod p$$

может выполняться также в случае, когда в качестве α берется число, относящееся к показателю q, где $q \mid p-1$. Для этого s должно быть вычислено из следующего соотношения

$$M = x_A r + ks \pmod q.$$

Можно выбрать простой модуль p таким, чтобы разложение $p-1$ содержало простой множитель q, размер которого существенно меньше размера p. Например, для 2048-битового модуля p длина q может составлять160 бит. В этом случае вычисляемое значение s будет иметь размер, не превышающий 160 бит. Благодаря этому достигается сокращение длины подписи почти в 2 раза (длина параметра r остается равной размеру модуля).

Экспериментальная часть. Используя заданные значения простого числа p, найти разложение числа $p-1$. Выбрать в качестве показателя q один из делителей $p-1$. Найти первообразный корень α и число g, относящееся к показателю q. Сформировать секретный ключ x_A, вычислить соответствующий ему открытый ключ $y_A = \alpha^{x_A} \pmod p$. Вычислить значение электронной подписи для 5 различных сообщений, фиксируя получаемые значения параметров k, $r = \alpha^{k}$, s, α^{M}, y_A^{r}, r^{s}, $y_A^{r} r^{s} \pmod p$. Вычислить значение сокращенной электронной подписи для тех же сообщений, используя новое значение открытого ключа $y_A = g^{x_A} \pmod p$ и фиксируя

получаемые значения параметров k, $r = g^k$, s, g^M, y_A^r, r^s, $y_A^r r^s \pmod p$. Осуществить проверку подписи по открытому ключу. Результаты оформить в виде таблицы.

Тема: "Цифровая подпись Эль-Гамаля с сокращенной длиной параметров s и r".

Теоретическая часть. Уравнение проверки подписи

$$\alpha^M = y_A^r r^s \pmod p$$

может быть преобразовано к следующему виду

$$r = \alpha^{M/s} y_A^{-r/s} \pmod p.$$

При этом вместо r в степени при y_A можно использовать значение некоторой хэш-функции от значения r, т. е. $H(r)$. В этом случае уравнение проверки подписи имеет вид $r = \alpha^{M/s} y_A^{-H(r)/s} \pmod p$. Чтобы проверка была корректной, владелец секретного ключа должен вычислить параметр s из следующего уравнения

$$M = x_A H(r) + ks \pmod q.$$

Поскольку при проверке подписи не требуется выполнять никаких вычислений с использованием параметра r, то проверка подписи может быть осуществлена в соответствии с уравнением

$$H(r) = H(\alpha^{M/s} y_A^{-H(r)/s} \bmod p).$$

В этом случае нет необходимости представлять проверяющему значение r, имеющее сравнительно большую длину. Достаточно для проверки представить значение $H(r)$, где размер значения хэш-функции равен, например, 160 бит. Этим достигается существенное сокращение длины подписи. Если используется вариант с сокращенной длиной параметра s, то общая длина подписи составляет порядка 320 бит вместо исходной длины 2048 или 4096 бит при 1024-битовом или 2048-битовом модуле p, соответственно. Сокращение длины подписи не уменьшает стойкости системы ЭЦП, поскольку сложность задачи дискретного логарифмирования не изменяется, так как вычисления ведутся по модулю одинакового размера.

В качестве хэш-функции $H(r)$ можно взять следующую: $H(r) = r \bmod q$, где q показатель, используемый при сокращении параметра s. Тогда приходим к следующему уравнению проверки подписи:

$$r' = (\alpha^{M/s} y_A^{-r''/s} \bmod p) \bmod q,$$

где (r', s) есть подпись к сообщению M, а параметр r' вычисляется после выбора случайного числа k в соответствии с формулой $r' = (\alpha^k \bmod p) \bmod q$. Уравнение для вычисления параметра s имеет вид:

$$M = x_A r' + ks \pmod{p-1}.$$

Экспериментальная часть. Используя заданные значения простого числа p, найти разложение числа $p-1$. Выбрать в качестве показателя q один из делителей $p-1$. Найти первообразный корень α и число g, относящееся к показателю q. Сформировать секретный ключ x_A, вычислить соответствующий ему открытый ключ $y_A = \alpha^{x_A} \pmod{p}$. Вычислить значение электронной подписи для 5 различных сообщений, фиксируя получаемые значения следующих значений

$$k,\ r' = (\alpha^k \bmod p) \bmod q,\ \ s,\ \ \alpha^{M/s} \bmod p,\ \ y_A^{r'/s} \bmod p,\ \ \alpha^{M/s} y_A^{-r'/s} \bmod p.$$

Осуществить проверку подписи по открытому ключу. Результаты оформить в виде таблицы. Выполнить аналогичные вычисления в варианте с сокращенным размером параметра s, используя в качестве основания число g.

Тема: "Электронная цифровая подпись RSA".

Теоретическая часть. Теорема Эйлера: для любых взаимно простых целых чисел M и n, где $M<n$, выполняется соотношение

$$M^{\varphi(n)} = 1 \pmod{n}.$$

В криптосистеме RSA в качестве числа M используется сообщение, которое необходимо подписать или зашифровать. Будем полагать, что условие взаимной простоты чисел M и n выполняется. Например, это обеспечивается тем, что в данной криптосистеме выбирается число n, равное произведению двух больших простых множителей, поэтому вероятность того, что случайное сообщение не будет взаимно простым с модулем, является пренебрежимо малым.

Формирование ключей. Каждый пользователь выбирает два больших не равных между собой простых числа p и q, находит их произведение $n = pq$ и вычисляет значение функции Эйлера от n:

$$\varphi(n) = (p-1)(q-1).$$

Значение n является частью открытого ключа. Числа p и q являются частью закрытого ключа. Числа p и q должны иметь специальную структуру, в частности, по крайней мере одно из чисел $(p-1)$ или $(q-1)$ должно иметь один большой простой множитель. Размер модуля n должен быть не менее 1024 бит. Затем выбирается целое

число d такое, что $d < \varphi(n)$ и НОД(d, $\varphi(n)$) = 1 и вычисляется e, удовлетворяющее условию

$$ed = 1 \pmod{\varphi(n)}.$$

Секретным ключом является тройка чисел p, q, d. *Открытым ключом* является пара чисел n, e, которая сообщается всем пользователям.

Процедура подписывания:

$$S = M^{d} \pmod{n}.$$

Процедура проверки подписи:

$$M' = S^{e} \pmod{n}.$$

Если $M' = M$, то сообщение M признается подписанным.

Экспериментальная часть. Используя заданные значения простых чисел p и q, вычислить модуль n, затем сформировать открытый и закрытый ключи e и d. Используя закрытый ключ, подписать 10 различных сообщений и осуществить проверку подписи по открытому ключу. Результаты оформить в виде таблицы.

Тема: "Электронная цифровая подпись ГОСТ Р.34-2001".

Теоретическая часть. Будем использовать следующие параметры подписи:

- простое число p – модуль ЭК;
- эллиптическую кривую E, задаваемую коэффициентами $a, b \in GF_p$;
- простое число q – порядок циклической подгруппы группы точек ЭК;
- точку $P \in E(GF_p)$ с координатами (x_p, y_p): $P \neq O$, $qP = O$;
- секретный ключ – ключ подписи, являющийся целым числом d: $0 < d < q$;
- открытый ключ – ключ проверки подписи, являющийся точкой $Q \in E(GF_p)$ с координатами (x_q, y_q): $dP = Q$.

Формирование подписи осуществляется в соответствии с приведенным алгоритмом:

1. Генерируется случайное целое число k, удовлетворяющее неравенству $0 < k < q$.
2. Вычисляется точка ЭК $C = kP$ и определяется значение $r \equiv x_C \pmod{q}$, где x_C – координата точки C.
3. Вычисляется значение $s \equiv (rd + ke) \pmod{q}$, где $e \equiv M \pmod{q}$, M – это сообщение или хэш-функция от сообщения, которое необходимо подписать.

Подписью являются два двоичных вектора r и s.

Проверка подписи заключается в вычислении координат точки ЭК:

$$C = (se^{-1} \pmod q)P + ((q\text{-}r)e^{-1} \pmod q)Q,$$

определении значения $R \equiv x_C \pmod q$ и проверке выполнения равенства $R = r$.

Экспериментальная часть. Используя заданные значения простого числа p и коэффициентов уравнения ЭК, определить порядок q циклической подгруппы группы точек ЭК, генератор циклической подгруппы – точку $P \in E(GF_p)$, сформировать секретный ключ d, вычислить соответствующий ему открытый ключ Q и вычислить значение электронной подписи для 10 различных сообщений, фиксируя получаемые значения параметров k, r, s, e, C. Осуществить проверку подписи по открытому ключу. Результаты оформить в виде таблицы.

Тема: "Слепая" подпись Чаума".

Теоретическая часть. Слепая подпись Чаума основана на криптосистеме RSA. Пусть пользователь А желает подписать некоторое сообщение M у пользователя В таким образом, чтобы последний не мог прочесть подписываемое сообщение. Для этого необходимо осуществить следующие шаги:

Пользователь А генерирует случайное простое число k – такое, что НОД(k, n) = 1, где n – часть открытого ключа пользователя В. Затем А вычисляет значение $M' = k^e M \pmod n$ и предъявляет его пользователю В, чтобы последний подписал M' в соответствии со стандартной процедурой подписывания в системе RSA. Подписывающий не может прочесть сообщение M, поскольку оно преобразовано путем наложения на него "разового" ключа k^e с использованием операции модульного умножения.

Пользователь В подписывает сообщение M': $S' = (k^e M)^d = kM^d \pmod n$. Заметим, что по значению подписи S' к сообщению M' подписывающий не имеет возможности вычислить M^d. Заметим также, что по значению M^d легко вычислить M: $(M^d)^e = M \pmod n$. Это означает, что после получения значения $S = M^d \pmod n$, пользователь А должен держать его в секрете от подписавшего.

После получения от пользователя В значения S', используя расширенный алгоритм Евклида, пользователь А вычисляет для числа k мультипликативно обратный элемент (k^{-1}) в поле вычетов по модулю n и формирует подпись пользователя В к сообщению M: $S = k^{-1}S' = k^{-1}kM^d = M^d \pmod n$.

Экспериментальная часть. Используя заданные значения простых чисел p и q, вычислить модуль n, затем сформировать открытый и закрытый ключи e и d. Осуществить процедуру выработки подписи «вслепую» в соответствии с протоколом Чаума для 6 различных сообщений, фиксируя для каждого из них значения k, $k^{-1} \pmod n$, M, M', S' и S. Осуществить проверку правильности полученной подписи путем непосредственного вычисления значения $S = M^d \pmod n$. Результаты оформить в виде таблицы.

10.1.3. Генерация простых чисел

Тема: "Генерация больших простых и псевдопростых чисел".

Теоретическая часть. Для генерации больших простых чисел могут быть использованы следующие три подхода:

- формируются случайные числа заданного размера и проверяется, являются ли они простыми, с помощью вероятностных тестов (псевдопростые числа);
- по определенной процедуре генерируются простые числа, проверка которых осуществляется с помощью детерминированных тестов на простоту;
- комбинированная генерация простых чисел, при которой формируются псевдопростые числа (по первому варианту) промежуточного размера, на основе которых затем формируются псевдопростые числа, тестируемые с помощью детерминистических тестов (этот подход обеспечивает ускорение процедуры генерации псевдопростых чисел p с известным разложением функции Эйлера от него).

В первом случае тесты строятся на основе определенных теорем из теории чисел, сформулированных и доказанных для простых чисел. Если число не удовлетворяет тесту, то оно не является простым и отбрасывается. Для проверки берется следующее случайное число требуемого размера. Если число проходит тест, то некоторый переменный параметр, используемый для тестирования, изменяется и тест повторяется снова. Число прошедшее большое число опытов определенного типа, считается псевдопростым, поскольку вероятность, что составное число может пройти все тесты пренебрежимо мала. Для того, чтобы исключить некоторые возможные классы составных чисел, которые могут проходить тесты конкретного типа, используют несколько различных тестов, по каждому из которых выполняется большое число опытов. Достоинством генерации псевдопростых чисел является сравнительная простота процедуры. Недостатком первого подхода является то, что после генерации большого псевдопростого числа p может оказаться достаточно сложным определение разложения числа $p-1$, которое необходимо знать, например, в случае ЭЦП на основе сложности задачи дискретного логарифмирования с сокращенной длиной подписи. Разложение числа $p-1$ представляет интерес также и для отсеивания некоторых классов слабых простых чисел. Следующие два вероятностных теста могут быть применены совместно. Пусть мы хотим проверить, является ли число p простым.

- *Тест Ферма* заключается в проверке соотношения $b^{p-1} = 1 \pmod{p}$ для большого числа различных значений b. Число различных использованных при тестировании значений b, для которых выполняется указанное соотношение, определяет число выполненных опытов по тесту Ферма. Однако известен класс составных чисел, которые проходят тест Ферма (числа Кармайкла). Примеры чисел из этого класса приведены в таблице 10.1.

- *Тест Соловея-Штрассена* заключается в проверке равенств $\left(\frac{b}{p}\right)=1$, где $\left(\frac{b}{p}\right)$ – символ Лежандра для значений b являющихся квадратичными вычетами по модулю p, и $\left(\frac{b}{p}\right)=-1$ для значений b, являющихся квадратичными невычетами по модулю p (квадратичным вычетом называется число, являющееся квадратом некоторого числа x по модулю p; т. е. для квадратичного вычета существует квадратный корень: $b = x^2 \bmod p$).

Второй тест хорошо отсеивает числа Кармайкла. Вероятность того, что составное число пройдет один опыт по тесту Соловея-Штрассена, не превышает значения 0.5. Это позволяет получить оценку числа опытов, которые следует выполнить в соответствии с данным тестом, чтобы получить необходимо низкую вероятность принять составное число в качестве псевдопростого. Первый тест используется в качестве предварительной отбраковки чисел. Второму тесту подвергают только числа, прошедшие первый.

Таблица 10.1

Примеры чисел Кармайкла

Число	**Разложение**	**Число**	**Разложение**
561	3·11·17	6601	7·23·41
1105	5·13·17	8911	7·19·67
1729	7·13·19	41041	7·11·13·41
2465	5·17·29	825 265	5·7·17·19·73
2821	7·13·31	413 631 505	5·7·17·73·89·107

Примером второго подхода можно принять следующий.

В качестве комбинированного подхода к формированию псевдопростых чисел можно использовать выбор псевдопростых чисел $q_1, q_2, \ldots, q_k$ промежуточной (но достаточно большой) длины, которые используются в качестве начального набора $\{q_1, q_2, \ldots, q_k\}$ для генерации псевдопростых чисел увеличенной длины, используя на втором этапе рассмотренный ниже вариант формирования простых чисел с детерминистической проверкой на простоту (в результате формируются псевдопростые числа). Достоинством комбинированного способа является возможность формирования простых чисел p с заданным разложением числа $p - 1$.

Экспериментальная часть. Формируются несколько псевдопростых чисел p, имеющих заданную длину и заданное разложение числа $p - 1$. Возможны два типа заданий, соответствующих двум вариантам генерации псевдопростых чисел – веро-

ятностному и комбинированному. Во втором случае формируются псевдопростые числа малой длины, а в качестве детерминистического теста используется метод пробного деления. В случае генерации псевдопростых чисел с использованием вероятностного теста Соловея-Штрассена оценивается вероятность принять составное число в качестве псевдопростого. Чтобы обойти проблему определения значения символа Лежандра $\left(\frac{b}{p}\right)$ для случайно выбираемого значения b, в качестве чисел b можно брать заведомо квадратичные вычеты, генерируя случайные числа x и вычисляя $b = x^2 \bmod p$. Для квадратичных вычетов имеем $\left(\frac{b}{p}\right) = 1$. По определению символ Лежандра есть значение $b^{\frac{p-1}{2}} \bmod p$.

Тема: "Генерация (детерминистическая) больших простых чисел с подбором разложения функции Эйлера".

Теоретическая часть. Формируется набор k простых чисел $\{q_1, q_2, \ldots, q_k\}$ сравнительно малой длины (например, имеющих 8 – 10 десятичных знаков). Причем числа $q_1, q_2, \ldots, q_k$ проверяются детерминистическим тестом на простоту, в качестве которого можно взять проверку на делимость на все натуральные числа от 2 до $\lceil \sqrt{q_i} \rceil$ (метод пробного деления; $\lceil g \rceil$ обозначает наименьшее целое число, не меньшее чем число g). Из указанного набора случайным образом выбираются h простых чисел $m_1, m_2, \ldots, m_h$, вычисляется число p_1, имеющее следующую структуру:

$$p_1 = 1 + 2\prod_{i=1}^{i=h} m_i \ .$$

Затем выбирается некоторое число b и проверяется, выполняются ли для данного p_1 следующие два условия:

1) $b^{p_1 - 1} = 1 \pmod{p}$ и

2) $b^{\frac{p_1-1}{m_i}} \neq 1 \bmod p$ для всех $m_i \in \{m_1, m_2, \ldots, m_h\}$.

Если после нескольких попыток найдется некоторое b, которое удовлетворяет указанным выше двум соотношениям, то p является простым числом. Если такое число не найдено, то выбирается другой случайный набор простых чисел $m_1, m_2, \ldots, m_h$ из набора $q_1, q_2, \ldots, q_k$. Сформированное таким образом число p_1 имеет длину примерно в h раз большее средней длины чисел $q_1, q_2, \ldots, q_k$ (например, от $8h$ до $10h$ десятичных знаков). Можно сформировать аналогичным образом следующий набор простых чисел $\{p_1, p_2, \ldots, p_k\}$ и, используя их в качестве исходных, повторить рассматриваемую процедуру, формируя еще более длинные простые числа. Достоинством данного способа является то, что мы заведомо знаем разложение $p - 1$, кроме

того, мы можем формировать это разложение таким образом, чтобы в нем содержались простые числа требуемой длины. Основным недостатком такого способа является то, что формируется только некоторый подкласс простых чисел заданной большой длины. Однако мощность этого подкласса может быть задана такой, что этим обстоятельством атакующий не сможет воспользоваться для раскрытия той или иной двухключевой криптосистемы, в которой данная процедура детерминистической генерации простых чисел будет использоваться.

Данный детерминистический тест основан на следующей теореме:

Пусть p – целое нечетное число, превышающее 1. *Если существует* $b \le p-1$, *такое, что выполняются следующие условия* 1) $b^{p-1} = 1 \bmod p$ *и* 2) $b^{\frac{p-1}{m_i}} \neq 1 \bmod p$ *для каждого простого делителя* m_i *числа* $p-1$, *то число* p *является простым.*

Доказательство

Допустим, что p не является простым. Тогда функция Эйлера от p имеет значение меньшее, чем $p-1$, т. е. $\varphi(p) < p-1$. Рассмотрим два случая а) НОД(b, p) = 1 и б) НОД(b, p) $\neq 1$. В случае а) порядок числа b должен делить $\varphi(p) < p-1$, но по условию теоремы порядок b равен $p-1$. В случае б) не существует целых степеней n, для которых выполняется условие $b^n = 1 \bmod p$. В обоих случаях приходим к противоречию, которое доказывает утверждение теоремы. (Пояснение к случаю б): если НОД(b, p) = $\delta \neq 1$, то $\delta \mid b^n \bmod p$ для любого n, поскольку для остатка r от деления b^n на p имеем НОД(b^n, r) = δ).

Экспериментальная часть. Формируются несколько простых (псевдопростых) чисел p, имеющих заданную длину и заданное разложение числа $p-1$. Возможны два типа заданий, соответствующих двум вариантам генерации простых и псевдопростых чисел (детерминистическому и комбинированному). Оценивается примерное число возможных чисел, которые могут быть сформированы в соответствии с детерминистическим и комбинированным подходами.

Тема: " Генерация (детерминистическая) больших простых чисел по стандарту ГОСТ Р 34.10-94".

Теоретическая часть. Для генерации больших простых чисел в ГОСТ Р 34.10-94 используется детерминистический тест, основанный на следующей теореме:

Пусть $p = qN + 1$, где q – нечетное простое число, N – четное, и $p < (2q+1)^2$. Число p является простым, если выполняются следующие два условия:

1) $2^{qN} = 1 \bmod p$,

2) $2^{N} \neq 1 \bmod p$.

Доказательство

Пусть γ есть порядок числа 2 по модулю p и p имеет следующее каноническое разложение: $p = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_h^{\alpha_h}$. Ввиду условия 1) γ делит $p - 1$, т. е. $\gamma \mid p - 1$. В силу условия 2) γ не является делителем числа $\frac{p-1}{q}$. Отсюда следует, что $q \mid \gamma$. Согласно теореме Эйлера $2^{\varphi(p)} = 1 \bmod p$, следовательно, $\gamma \mid \varphi(p) \Rightarrow q \mid \varphi(p)$, где $\varphi(p) = p_1^{\alpha_1 - 1} p_2^{\alpha_2 - 1} \ldots p_h^{\alpha_h - 1} (p_1 - 1)(p_2 - 1)\ldots(p_h - 1)$. Пусть q совпадает с простым множителем p_i. Из такого допущения следует, что $p = qn'$ для некоторого натурального числа n. Однако по условию теоремы имеем $p = qN + 1$. Поскольку $q > 1$ не может делить число 1, то приходим к противоречию, из которого следует, что q должно делить число $p_i - 1$, по крайней мере, для некоторого одного из значений $i \in \{1, 2, \ldots, h\}$.

Таким образом, существует некоторое натуральное $n \geq 2$, такое, что имеем $p_i - 1 = qn$ и $p_i = qn + 1$. Следовательно, при некотором натуральном m получим:

$$p = mp_i = m(qn + 1) = qN + 1 \quad \Rightarrow \quad m = q(N - mn) + 1.$$

При некотором натуральном $s \geq 0$ имеем $m = qs + 1$ и

$$p = (qn + 1)(qs + 1).$$

Пусть p есть составное число, тогда $s \geq 2$ (поскольку N и n – четные числа, а $s = N - mn$), из чего следует $p \geq (2q + 1)^2$. Это противоречит условию теоремы, следовательно, $s = 0$ и число p является простым.

Экспериментальная часть. Заключается в выполнении нескольких шагов алгоритма детерминистического формирования простых чисел заданной длины по ГОСТ Р 34.10-94. Схема построения алгоритма описывается следующим образом. Пусть требуется сформировать простое число p длины $t \geq 17$ бит. С этой целью строится убывающий набор натуральных чисел $t_0, t_1, \ldots, t_s$, где $t_0 = t$ и $t_s < 17$ бит, для которых выполняется условие $t_i = [t_{i-1}/2]$. Последовательно вырабатываются простые числа $p_s, p_{s-1}, \ldots, p_0$, причем длина числа p_i равна значению t_i для всех $i = 1, \ldots, s$. Исходное простое значение p_s формируется путем случайного выбора числа размером менее 17 бит и проверкой на простоту методом пробного деления.

Генерация простого числа p_{i-1} по простому числу p_i осуществляется с использованием формулы

$$p_{i-1} = p_i N + 1,$$

где N – случайное четное число, такое, что длина числа $p_i N + 1$ равна значению t_i. Число p_{i-1} считается полученным, если одновременно выполнены следующие два условия:

1) $2^{p_i N} = 1 \bmod p_{i-1}$,

2) $2^{N} \neq 1 \bmod p_{i-1}$.

Если хотя бы одно из условий не выполнено, то значение N увеличивается на два, вычисляется новое значение p_{i-1}, которое снова проверяется на простоту по указанным двум условиям. Такая процедура выполняется до тех пор, пока не будет получено простое число p_{i-1}.

10.2. Задачи для решения на практических занятиях

10.2.1. Примитивы и схемы одноключевых криптосистем

1. Показать, что рекурсивный механизм построения БУП, показанный на рис. 10.1, позволяет построить БУП равновероятного смещения с размером входа $n=2^g$, где g – натуральное число ($g \geq 2$). Определить минимально необходимое число активных слоев.

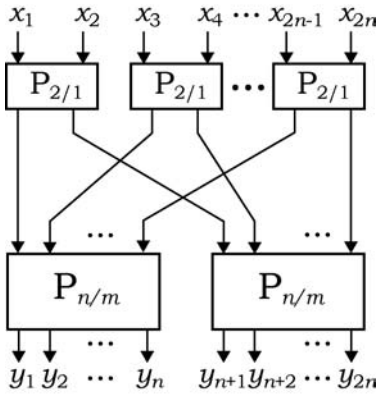

Рис.10.1. Схема построения блока $\mathbf{P}_{2n/2m+n}$ на основе двух БУП $\mathbf{P}_{n/m}$ равновероятного смещения

2. Построить блоки первого порядка $\mathbf{P}_{4/4}$, $\mathbf{P}_{8/12}$, $\mathbf{P}_{16/32}$, $\mathbf{P}_{32/80}$, $\mathbf{P}_{64/192}$, обладающие свойством равновероятного смещения произвольного входного бита в произвольный разряд на выходе.
3. Построить блоки равновероятного смещения $\mathbf{P}_{4/4}$, $\mathbf{P}_{8/12}$, $\mathbf{P}_{16/32}$, $\mathbf{P}_{32/80}$, $\mathbf{P}_{64/192}$ с симметричной структурой, для которых соответствующие обратные БУП $\mathbf{P}^{-1}{}_{4/4}$, $\mathbf{P}^{-1}{}_{16/32}$, $\mathbf{P}^{-1}{}_{64/192}$ получаются путем перераспределения управляющих битов.
4. Показать, что рекурсивный механизм построения БУП, показанный на рис. 10.2, позволяет построить БУП $\mathbf{P}_{2n/2m+2n}$ порядка $2h$, если блоки $\mathbf{P}_{n/m}$ имеют порядок $h \leq n$.

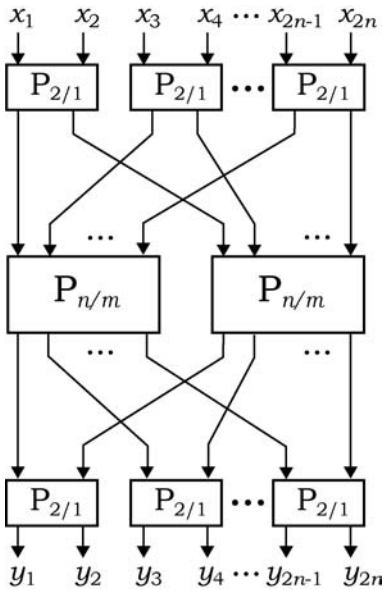

Рис.10.2. Схема удвоения порядка БУП

5. БУП, построенные в соответствии с механизмом, описанным в задаче 4, начиная на первом шаге с блока $\mathbf{P}_{2/1}$, называются рекурсивной сетью Бенеша. Показать, что БУП Бенеша $\mathbf{P}_{4/6}$, $\mathbf{P}_{8/20}$, $\mathbf{P}_{16/56}$, $\mathbf{P}_{32/144}$, $\mathbf{P}_{64/352}$ являются БУП максимального порядка. Определить число слоев в сети Бенеша для различных значений n.

6. Оценить время задержки слоистого БУП в единицах $t_{\oplus}$, где $t_{\oplus}$ – время задержки операции поразрядного суммирования по модулю два (XOR).

7. На основе решения задачи 5 построить симметричный БУП максимального порядка с размером входа $n=2^g$, где $g \geq 2$, т.е. такой блок $\mathbf{P}_{n/m}$ порядка n, который представим в виде суперпозиции двух взаимно обратных БУП $\mathbf{P}_{n/m}= \mathbf{P}_{n/m'} \bullet \mathbf{P}^{-1}_{n/m'}$.

8. На основе рекурсивной переключающей сети, описанной в задаче 1, построить БУП второго порядка для размера входа $n = 2^g$, где g – произвольное натуральное число ($g \geq 2$). Определить минимально необходимое число активных слоев.

9. Правильным ли будет определение операции управляемой битовой перестановки, как операции преобразования двоичного вектора, сохраняющей вес Хемминга? Почему?

10. На основе рекурсивной переключающей сети, описанной в задаче 4, построить блоки порядка 4, 8, 16,…, n/4. Определить необходимое число активных слоев для реализации БУП такого типа.

11. Показать, что суперпозиция двух взаимно обратных БУП порядка h = 4, 8,…,n/4, полученных в решении задачи 9, реализует БУП максимального порядка.

12. В БУП порядка $n/4$, соответствующих задаче 10, добавление одного внутреннего слоя образует БУП максимального порядка. Объясните, почему добавление только одного слоя увеличивает порядок в четыре раза.
13. Используя структуру БУП максимального порядка, дать оценку числа $n!$ степенью числа 2. Сравнить полученное соотношение с формулой Стирлинга.
14. Составить диаграмму перестановочного преобразования, соответствующего формуле $Y = (X)\ \mathbf{P}_{n/m(V)}\bullet\mathbf{T}\bullet\mathbf{P}^{-1}{}_{n/m(V)}$, где $X = X_1|X_2$ – входной блок данных, Y – преобразованные данные, $\mathbf{T}$ – операция транспозиции: $\mathbf{T}(X_1|X_2) = X_2|X_1$. Показать, что БУП, представленный суперпозицией $\mathbf{P}_{n/m(V)}\bullet\mathbf{T}\bullet\mathbf{P}^{-1}{}_{n/m(V)}$, выполняет операцию управляемой перестановочной инволюции.
15. Показать, что БУП, представленный суперпозицией $\mathbf{P}_{n/m(V)}\bullet\mathbf{I}\bullet\mathbf{P}^{-1}{}_{n/m(V)}$, где $\mathbf{I}$ – фиксированная перестановочная инволюция, является операционным блоком, выполняющим перестановочные инволюции при любом значении V, т. е. данная суперпозиция позволяет реализовать операцию управляемых перестановочных инволюций.
16. Показать, что БУП, представленный суперпозицией $\mathbf{P}_{n/m(V)}\bullet\Pi\bullet\mathbf{P}^{-1}{}_{n/m(V)}$, где П – некоторая фиксированная перестановка, реализует только перестановки, распадающиеся на циклы, имеющие ту же длину, что и циклы перестановки П, т.е. при любом V реализуется перестановка, которая обладает той же цикловой структурой, что и перестановка П.
17. Пользуясь решением задачи 16, составить БУП, реализующий только одноцикловые перестановки.
18. Записать в аналитической форме преобразование, представленное на рис. 10.3. Показать, что данная структура реализует управляемые перестановочные инволюции.

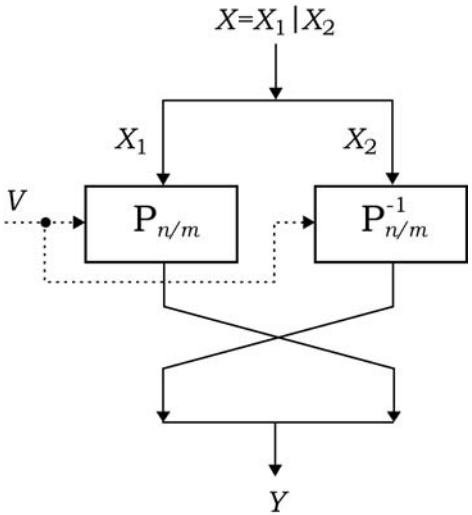

Рис.10.3. Структура управляемых перестановочных инволюций

19. Записать в аналитической форме преобразование, представленное на рис. 10.4. Показать, что данная структура реализует управляемые перестановочные инволюции.

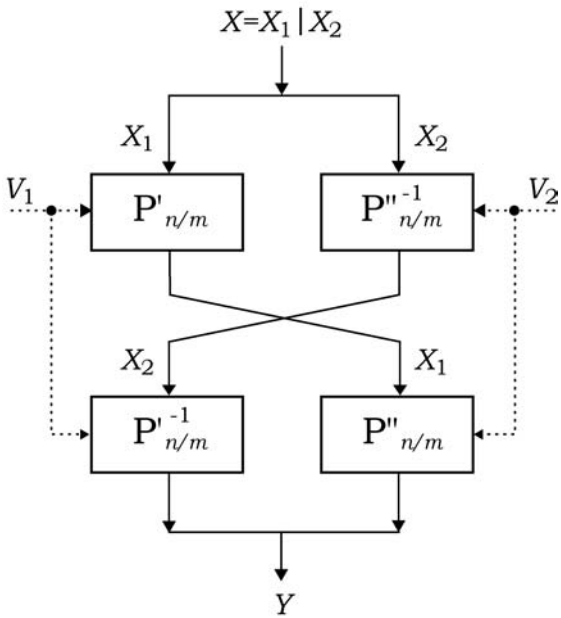

Рис. 10.4. Вариант структуры БУП, реализующего управляемые перестановочные инволюции

20. Показать, что если операционный блок $\mathbf{P}_{n/m(V_1)}$ имеет симметричную структуру и суперпозиция $\mathbf{P}'_{n/2m(V)} = \mathbf{P}_{n/m(V_1)} \bullet \mathbf{P}^{-1}{}_{n/m(V_2)}$ является БУП максимального порядка, то и суперпозиция $\mathbf{P}''_{n/2m(V_1)} = \mathbf{P}^{-1}{}_{n/m(V_1)} \bullet \mathbf{P}_{n/m(V_2)}$ также является блоком максимального порядка.

21. Показать, что блоки $\mathbf{P}_{8/12}$ и $\mathbf{P}^{-1}{}_{8/12}$, построенные в соответствии с рекурсивным механизмом задачи 1, включают одно и то же множество реализуемых модификаций перестановок.

22. Пользуясь решением задачи 21, показать, что блоки $\mathbf{P}_{16/32}$ и $\mathbf{P}^{-1}{}_{16/32}$, построенные в соответствии с рекурсивным механизмом задачи 1, включают одно и то же множество реализуемых модификаций перестановок.

23. Дать обобщение задачам 21 и 22 для произвольного размера БУП первого порядка, построенного в соответствии с рекурсивным механизмом задачи 1.

24. Доказать, что если БУП, имеющий структуру $\mathbf{P}'_{n/2m(V)} = \mathbf{P}_{n/m(V_1)} \bullet \mathbf{P}^{-1}{}_{n/m(V_2)}$, является блоком максимального порядка и блоки $\mathbf{P}_{n/m}$ и $\mathbf{P}^{-1}{}_{n/m}$ построены по рекурсивной схеме задачи 1, то тогда и суперпозиция этих БУП, взятых в обратном порядке, $\mathbf{P}''_{n/2m(V)} = \mathbf{P}^{-1}{}_{n/m(V_1)} \bullet \mathbf{P}_{n/m(V_2)}$ также является блоком максимального порядка.

25. Показать, что блоки $\Pi \bullet \mathbf{P}_{n/m(V)}$ и $\mathbf{P}_{n/m(V)} \bullet \Pi$ имеют максимальный порядок, если $\mathbf{P}_{n/m(V)}$ является БУП максимального порядка.

26. Показать, что БУП, имеющий структуру $\mathbf{P}_{n/m(V)} \bullet \Pi \bullet \mathbf{P}^{-1}{}_{n/m(V)}$, где Π – произвольная фиксированная одноцикловая перестановка и $\mathbf{P}_{n/m(V)}$ – управляемая перестановка максимального порядка, реализует все возможные одноцикловые перестановки.

27. Показать, что БУП, имеющий структуру $\mathbf{P}_{n/m(V)} \bullet \mathbf{I} \bullet \mathbf{P}^{-1}{}_{n/m(V)}$, где $\mathbf{I}$ – произвольная фиксированная перестановочная инволюция, не содержащая циклы единичной длины, реализует все возможные перестановки, содержащие только циклы длины два, если $\mathbf{P}_{n/m(V)}$ является управляемой перестановкой максимального порядка.

28. Обобщить задачи 26 и 27 на случай перестановки Π с произвольной цикловой структурой.

29. Показать, что преобразование на основе управляемых перестановок, показанное на рис. 10.5, является биективным по K при фиксированном X и по X при фиксированном K.

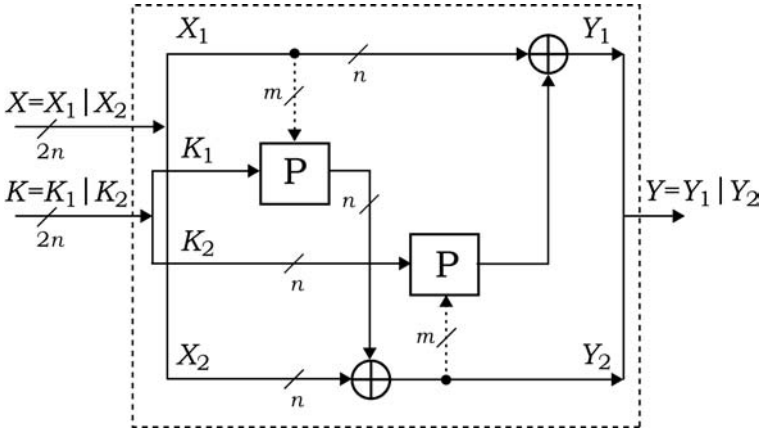

Рис.10.5. "Биективное" наложение двоичных векторов K и X.

30. Показать, что преобразование на основе управляемых перестановок, показанное на рис. 10.6, является биективным по K при фиксированном X и по X при фиксированном K.

31. Показать, что преобразование n-битового блока данных X, описываемое соотношением $Y = \Pi(X) \oplus X$, не является биективным и число возможных значений n-битового вектора Y равно 2^{n-k}, где k – число циклов в фиксированной перестановке Π. При каком условии достигается максимальное число различных значений Y?

32. Показать, что преобразование n-битового блока данных X, описываемое соотношением $Y = \mathbf{P}_{n/m(V)}(X) \oplus X$, не является биективным и число возможных значений n-битового вектора Y не превышает 2^{n-1}.

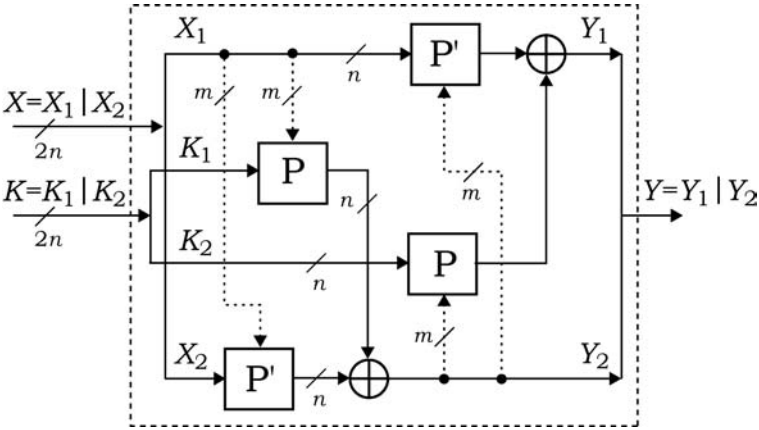

Рис.10.6. Вариант “биективного” наложение векторов K и X.

33. Показать, что для четного числа переставляемых элементов n не существует перестановки П, для которой перестановка $\Pi^2 = \Pi \bullet \Pi$ является одноцикловой.

34. Показать, что для числа переставляемых элементов $n = 2^g$, где g – натуральное число большее 0, последовательное выполнение одноцикловой перестановки П за g шагов приводит к вырождению перестановки П в тождественную перестановку.

35. Чему равно число различных перестановочных инволюций, не содержащих циклов единичной длины?

36. Для блоков равновероятного смещения определить, на сколько отличается от 1/2 вероятность того, что в заданном разряде на выходе будет единичный бит, если в заданном разряде на входе расположен единичный бит. Предположить, что остальные входные биты и управляющие биты имеют значение 0 с вероятностью 1/2.

37. В управляющем векторе изменяется один бит. С какой вероятностью значение на выходе слоистого БУП не изменяется, если значение на входе фиксировано. Предположить, что биты преобразуемого двоичного вектора имеют значение 1 с вероятностью 1/2. Зависит ли эта вероятность от значения управляющего вектора и от структуры БУП?

38. Показать, что итеративный шифр, один раунд которого представлен на рис. 10.7, позволяет выполнить зашифровывающие и расшифровывающие процедуры с помощью одного и того же алгоритма путем изменения расписания использования подключей Q_i и K_i .

39. Показать, что любой блочный шифр при фиксированном ключе реализует подстановку на множестве двоичных векторов $\{0,1\}^n$, где n – длина преобразуемых подблоков данных. Оценить необходимую длину ключа, для того чтобы шифр мог реализовать все возможные перестановки.

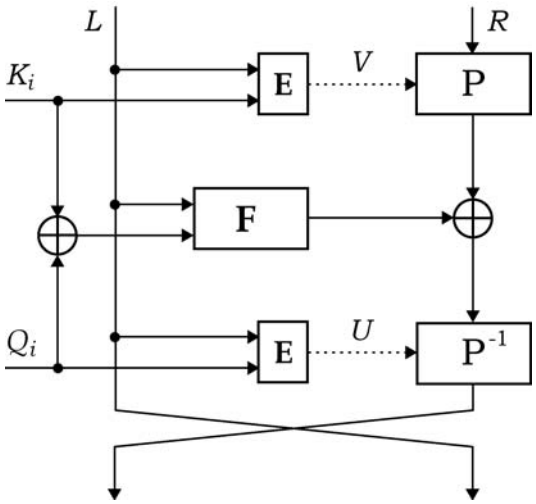

Рис. 10.7. Раунд шифрования с двумя взаимно-обратными БУП

40. Построить блок $\mathbf{P}_{16/32}$ с симметричной структурой и показать, что блоки $\mathbf{P}^{-1}{}_{16/32}$ и $\mathbf{P}_{16/32}$ реализуют одинаковые множества модификаций перестановок, причем вероятности реализации каждой фиксированной перестановки одинаковы для обоих БУП (для равновероятного случайного значения управляющего вектора).
41. Построить слоистый БУП (по аналогии с рис. 10.2) максимального порядка, в котором все фиксированные коммутации между активными слоями соответствуют фиксированным перестановкам, являющимся инволюциями.
42. Построить слоистый БУП $\mathbf{P}_{32/176}$ максимального порядка на основе четырех блоков $\mathbf{P}_{8/20}$ и 16 блоков $\mathbf{P}_{4/6}$. Показать, что построенный блок выполняет любую из 32! перестановок над 32 битами.
43. Показать, что структура раунда шифрования, представленная на рис. 10.8, при любой функции **F** задает преобразование, являющееся инволюцией.
44. Показать, что обобщенный раунд шифрования, представленный на рис. 10.9, при любых функциях $\mathbf{F_1}$ и $\mathbf{F_2}$ задает преобразование, являющееся инволюцией.
45. Используя управляемые операции и схему на рис. 10.9 построить 128-битовый блочный шифр. Обосновать критерии построения.
46. Вычислить среднюю вероятность прохождения разности с одним активным битом через операцию суммирования (вычитания) по модулю 2^n.
47. Показать, что шифр COBRA-H64 корректно осуществляет процедуру расшифрования при соответствующем изменении расписания использования ключа. Объяснить роль конечного преобразования.
48. Показать, что шифр COBRA-H128 корректно осуществляет процедуру расшифрования при соответствующем изменении расписания использования ключа. Объяснить роль конечного преобразования.

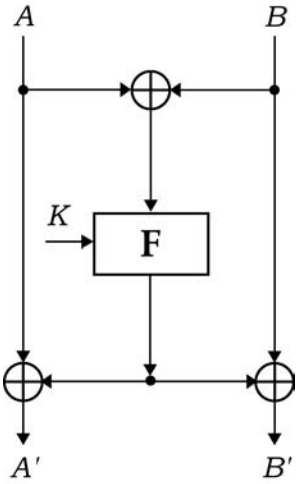

Рис.10.8. Структура раунда шифрования, являющегося инволюцией

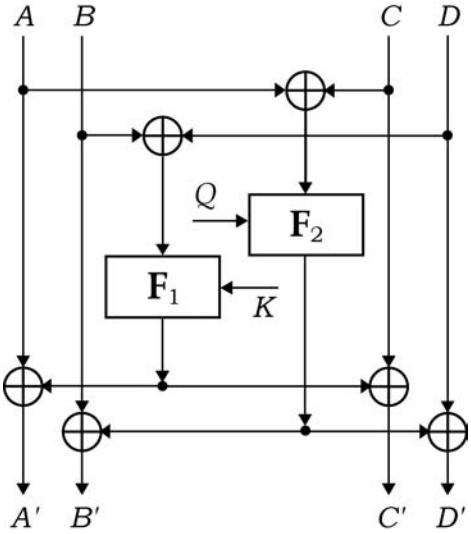

Рис. 10.9. Фрагмент структуры раунда шифрования алгоритма IDEA

49. Доказать, что шифр DDP-64 корректно осуществляет процедуру расшифрования при соответствующем изменении расписания использования ключа.
50. Показать слабость итеративной хэш-функции, основанной на раундовой функции $h_i = a^{h_{i-1} \oplus M_i} \bmod p$, где M_i – блоки данных, h_i – раундовое значение хэш-функции, a и p – известные параметры.

51. Показать слабость итеративной хэш-функции, основанной на раундовой функции $h_i = (a^{h_{i-1}})^{M_i} \bmod p$, где M_i – блоки данных, h_i – раундовое значение хэш-функции, a и p – известные параметры.
52. Показать слабость итеративной хэш-функции, основанной на раундовой функции $h_i = (h_{i-1} \oplus M_i)^a \bmod p$, где M_i – блоки данных, h_i – раундовое значение хэш-функции, a и p – известные параметры.

10.2.2. Булевы функции

53. Показать, что с помощью многочленов Жегалкина описываются все возможные БФ от произвольного конечного числа переменных.
54. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (e, k), определить значение нелинейности БФ и их линейной комбинации.
55. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (f, n), определить значение нелинейности БФ и их линейной комбинации.
56. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (q, x), определить корреляционную иммунность БФ и их линейной комбинации.
57. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (u, i), определить корреляционную иммунность БФ и их линейной комбинации.
58. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (p, l), определить автокорреляционную функцию БФ и их линейной комбинации.
59. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/1}$, реализуемого модификациями (o, h), определить автокорреляционную функцию БФ и их линейной комбинации.
60. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/2}$, реализуемого модификациями (e, h, j, f), определить корреляционную иммунность БФ и их линейной комбинации.
61. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/2}$, реализуемого модификациями (h, e, f, g), определить автокорреляционную функцию БФ и их линейной комбинации.
62. Записать БФ для каждого выхода управляемого элемента $\mathbf{F}_{2/2}$, реализуемого модификациями (h, f, f, g), определить значение нелинейности БФ и их линейной комбинации.
63. Показать, что управляемый элемент типа $\mathbf{F}_{2/2}$, описываемый следующими двумя БФ от четырех переменных, представляет собой нелинейную инволюцию: $y_1 = v_1v_2x_1 \oplus v_2x_1 \oplus v_2x_2 \oplus x_1$; $y_2 = v_1v_2x_1 \oplus v_1v_2x_2 \oplus v_2x_1 \oplus v_2x_2 \oplus x_2$.

64. Записать БФ для каждого выхода УЭ $\mathbf{F}_{3/1}$, представленного на рис. 6.2, показать, что преобразование, выполняемое данным УЭ, представляет собой инволюцию.

65. Записать БФ для каждого выхода УЭ $\mathbf{F}_{3/1}$, реализованного двумя инволюциями (10, 435) из табл. 6.1, 6.2. Определить значения нелинейности БФ и всех их линейных комбинаций.

66. Записать БФ для каждого выхода УЭ $\mathbf{F}_{3/1}$, реализованного двумя инволюциями (101, 556) из табл. 6.1, 6.2. Определить автокорреляционные функции БФ и всех их линейных комбинаций.

67. Записать БФ для каждого выхода УЭ $\mathbf{F}_{3/1}$, реализованного двумя инволюциями (175, 208) из табл. 6.1, 6.2. Определить корреляционную иммунность БФ и всех их линейных комбинаций.

10.2.3. Двухключевые криптосистемы

Арифметические задачи

68. Пусть известно разложение числа $p-1$, где p есть большое простое число размера 1024 бит, причем в разложении имеется простой множитель d размера 160 бит. Указать вычислительно эффективный способ нахождения числа α, относящегося к показателю.

69. Пусть известно разложение числа $p-1$, где p есть простое число. Как проверить то, что число α является первообразным корнем?

70. Вычислить функцию Эйлера от чисел N_1=567 и N_2=1024.

71. Вычислить функцию Эйлера от чисел N_1=1280 и N_2=512.

72. Вычислить функцию Эйлера от чисел N_1=213 и N_2=2025.

73. Вычислить $a/b \mod p$ для значений 1) $a = 79$, $b = 11$, $p = 97$ и 2) $a = 5$, $b = 17$, $p = 131$.

74. Вычислить $a^b \mod p$ для значений 1) $a = 13$, $b = 17$, $p = 131$ и 2) $a = 5$, $b = 17$, $p = 131$.

75. Вычислить $a^b \mod p$ для значений 1) $a = 7$, $b = 20$, $p = 109$ и 2) $a = 38$, $b = 7$, $p = 47$.

76. Вычислить $a/b \mod p$ для значений 1) $a = 79$, $b = 11$, $p = 97$ и 2) $a = 5$, $b = 17$, $p = 131$.

77. Вычислить $a/b \mod p$ для значений 1) $a = 6$, $b = 18$, $p = 123$ и 2) $a = 8$, $b = 24$, $p = 107$.

78. Найти линейное представление с целыми коэффициентами для наибольшего общего делителя чисел $a = 13$; $в = 87$.

79. Найти линейное представление с целыми коэффициентами для наибольшего общего делителя чисел $a = 11$; $в = 97$.

80. Показать, что для значения b являющегося взаимно простым с n, выполняется соотношение $a/b = ab^{\varphi(n)-1} \mod n$, где $\varphi(n)$ – функция Эйлера.

Системы ЭЦП

81. Вычислить закрытый d ключ криптосистемы RSA, соответствующий открытому ключу $e = 97$, для значений модуля $n_1 = 299$ и $n_2 = 527$.
82. Вычислить открытый d ключ криптосистемы RSA, соответствующий закрытому ключу $e = 91$ для значений модуля $n_1 = 187$ и $n_2 = 319$.
83. Вычислить открытый d ключ криптосистемы RSA, соответствующий закрытому ключу $e = 91$ для значений модуля $n_1 = 187$ и $n_2 = 319$.
84. Сформировать соответствующие друг другу открытый d и закрытый ключ e криптосистемы RSA при значениях модуля $n_1 = 377$ и $n_2 = 451$.
85. Предложить вариант слепой подписи с использованием системы ЭЦП Эль-Гамаля.
86. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $m = \alpha^{rs} yr \pmod p$
87. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $m = \alpha^{f(r)s} yr \pmod p$
88. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{hs} = y^r r^s \pmod p$.
89. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{hr} = y^s r \pmod p$.
90. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{rs} = y^h r \pmod p$.
91. Преобразовать ЭЦП с уравнением проверки подписи $\alpha^h = y^r r^s \bmod p$, где $r = \alpha^k \bmod p$, в ЭЦП с сокращенной длиной подписи.
92. Преобразовать ЭЦП с уравнением проверки подписи $\alpha^s = y^r r^h \bmod p$, где $r = \alpha^k \bmod p$, в ЭЦП с сокращенной длиной подписи.
93. Найти произвольную тройку чисел m, r и s, таких, что (r, s,) является правильной подписью к сообщению m для ЭЦП со следующим уравнением проверки подписи $\alpha^m = y^r r^s \bmod p$, где $r = \alpha^k \bmod p$.
94. Найти произвольную тройку чисел m, r и s, таких, что (r, s,) является правильной подписью к сообщению m для ЭЦП со следующим уравнением проверки подписи $\alpha^s = y^r r^m \bmod p$, где $r = \alpha^k \bmod p$.
95. Найти произвольную тройку чисел m, r и s, таких, что (r, s,) является правильной подписью к сообщению m для ЭЦП со следующим уравнением проверки подписи $\alpha^{mF(r)} = y^s r \bmod p$, где $r = \alpha^k \bmod p$.

96. Найти произвольную тройку чисел m, r и s, таких, что (r, s,) является правильной подписью к сообщению m для ЭЦП со следующим уравнением проверки подписи $\alpha^r = y^s r^m \bmod p$, где $r = \alpha^k \bmod p$.

97. Найти произвольную тройку чисел m, r и s, таких, что (r, s,) является правильной подписью к сообщению m для ЭЦП со следующим уравнением проверки подписи $\alpha^r = y^m r^s \bmod p$, где $r = \alpha^k \bmod p$.

98. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $\alpha^{hs} = y^r r^{sF(h)} \bmod p$, где h – значение хэш-функции от сообщения m, $F(h)$ – некоторая функция от h и $r = \alpha^k \bmod p$.

99. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $\alpha^{hs} = y^r r^{sF(h)} \bmod p$, где h – значение хэш-функции от сообщения m, $F(h)$ – некоторая функция от h и $r = \alpha^k \bmod p$.

100. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $\alpha^h = y r^s \bmod p$, где $r = \alpha^k \bmod p$.

101. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r = y^{rs} \alpha^h \bmod p$, где $r = \alpha^k \bmod p$.

102. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r = y^{rsF(h)} \alpha^h \bmod p$, где $r = \alpha^k \bmod p$.

103. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r^{F(h)} = y^{rs} \alpha^h \bmod p$, где $r = \alpha^k \bmod p$.

104. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $\alpha^{sF(h)} = y^h r \bmod p$, где $r = \alpha^k \bmod p$.

105. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $\alpha^{sF(h,r)} = y^h r \bmod p$, где $r = \alpha^k \bmod p$.

106. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r^{F(r)s} = y \alpha^h \bmod p$, где $r = \alpha^k \bmod p$.

107. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r^{F(h)} = y^s \alpha^{hs} \bmod p$, где $r = \alpha^k \bmod p$.

108. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r^{F(r)\,f(h)} = y^s \alpha^{hs} \bmod p$, где $r = \alpha^k \bmod p$.

109. Вычислить подпись (r, s) к заданному сообщению m в случае слабой ЭЦП со следующим уравнением проверки подписи $r^{F(r)} = y^{sf(h)} \alpha^{hs} \bmod p$, где $r = \alpha^k \bmod p$.

10.3. Как запатентовать алгоритм

10.3.1. Можно ли запатентовать алгоритм

Патентование представляет собой наиболее значительную форму защиты интеллектуальной собственности, и во всех цивилизованных странах этому вопросу уделяется значительное внимание. Стимулирование новых изобретений оказывает крайне благоприятное влияние на развитие новых технологий и научно-технический прогресс вообще. Существуют некоторые различия в патентном законе различных стран (в Индии, например, способы не составляют объект патентования), но в целом можно выделить следующие главные объекты патентования:

- устройства;
- вещества;
- способы.

Главной особенностью объекта патентования является его материальность, в частности способ должен включать в себя последовательность действий материальных объектов над материальными объектами. Не подлежит патентованию аксиома, теорема или некоторый закон природы, хотя они могут лежать в основе тех или иных изобретений. Не подлежат патентованию изобретения, противоречащие морали и нравственности общества. Одно и то же изобретение может быть использовано как на пользу, так и во вред человеку. Например, алгоритмические способы преобразования цифровых сигналов в компьютерных сетях могут быть предназначены как для решения полезных и важных для общества задач, так и во вред ему, если используются при создании вирусных программ. Несомненно, что последние не могут рассматриваться как объект патентования.

Вопрос правовой защиты технических решений имеет важное практическое значение. Авторы настоящей книги имеют в нем определенный опыт. Данный вопрос не является предельно простым. По некоторым заявкам на новые способы шифрования авторы получили из патентных ведомств ряд заключений, что в заявке якобы осуществляется попытка запатентовать алгоритм, а алгоритмы патентованию не подлежат. В начале своей изобретательской деятельности авторы неоднократно получали разъяснения патентоведов, что алгоритмы не патентуются. Это задержало начало активного патентования нами новых способов шифрования до тех пор, пока не была получена консультация опытного патентоведа. Так как же быть с патентованием алгоритмов? Некоторые алгоритмы действительно патентованию не подлежат. Примером может служить некоторый алгоритм, обеспечивающий релаксацию после напряженного физического труда, предусматривающий ряд шагов по самовнушению, который не может повторить любой желающий это сделать, что обусловливается индивидуальностью человеческой психики. Но нас интересуют технические решения, а в большинстве таких случаев алгоритмы связаны с вычислениями и преобразованиями, которые реально выполняются некоторыми устройствами и включают действия над

электрическими сигналами. Вряд ли кто-то станет оспаривать материальность электрических сигналов. И не важно то, что для осуществления воздействия на электрические сигналы необходимы специальные инструменты, в частности, отдельный микропроцессор, ЭВМ или другие электронные устройства. Таким образом, алгоритмы могут быть запатентованы как способы преобразования (шифрования, кодирования, формирования защитных контрольных сумм и электронной цифровой подписи, и т. п.) материального объекта – электрического сигнала – с помощью материального объекта. Кроме того, можно запатентовать устройство, реализующее такое преобразование.

10.3.2. Стратегия патентования

В предыдущем параграфе мы уяснили, что алгоритмы преобразования информации, в том числе и чисел, могут быть успешно запатентованы. Для этого необходимо, чтобы алгоритм преобразования сигналов отвечал условиям патентоспособности: обладал новой совокупностью существенных признаков, имел практическую применимость и имел изобретательский уровень (для детального ознакомления с требованиями к объектам изобретения мы отсылаем читателя к специальной патентоведческой литературе). Заметим, что простота технического решения не есть основание для заключения, что оно не соответствует изобретательскому уровню. Часто новые простые решения инициируют технологический переворот. В области криптографии таким примером служит метод открытого распределения ключей Диффи-Хелмана. Допустим, что мы придумали патентоспособный алгоритм и хотим его запатентовать. Какие стратегии патентования можно выбрать? Патентовать способ, устройство или то и другое?

Если изобретение связано с перспективами широкого применения, то целесообразным может оказаться патентование как способа, составляющего суть алгоритма, так и устройства, реализующего алгоритм. Здесь мы умышленно делаем акцент на алгоритме, но в действительности он находится в подчиненном отношении к способу. Алгоритм является только частным примером реализации способа, но обычно первоначально разрабатывается новый алгоритм. Затем приходит понимание того, что он решает важную для практики техническую задачу, и принимается решение о его патентовании. Та же техническая задача может быть решена также и другими аналогичными алгоритмами, т. е. другой совокупностью действий над сигналами. Некоторые признаки будут присутствовать во всех этих алгоритмах. Некоторые признаки не будут иметь принципиального значения и будут характеризовать более форму реализации, чем существо способа. Для патентования технического решения как нового способа целесообразно выделить минимальную новую совокупность признаков.

Новизна заключается в том, что в этой совокупности включены существенные отличительные признаки по сравнению с любым известным способом. Некоторый запатентованный способ может быть реализован в виде большого числа различных алгоритмов. Конечно, если у нас появится желание запатентовать конкретный частный вариант способа, то это сделать можно. Более того, в этом случае может оказаться, что его новизна обосновывается более просто. Это связано с тем, что частные случаи могут включать дополнительные существенные признаки.

Однако следует иметь в виду, что в этом случае другие частные варианты реализации способа, которые также будут решать аналогичную техническую задачу, уже не будут попадать под патентную охрану.

Если изобретение запатентовано как способ, то в этом случае устройства, реализующие этот способ, также попадают под патентную охрану. Но проверка факта использования запатентованного способа может оказаться достаточно проблематичной.

Изобретатели, активно работающие в некоторой узкой области, обычно имеют много новых технических решений некоторой важной практической задачи. При этом новые решения могут быть альтернативными и дающими примерно одинаковый технический результат. Более того, если запатентовать и обнародовать одно решение, то оно окажется путеводной нитью к другим решениям. В этом случае целесообразно найти одну или несколько обобщающих формул изобретения, которые бы охватили все решения. В этом случае уменьшается число патентов, по которым потребуется выплата соответствующих пошлин. Эта стратегия ориентирована на уменьшение выплат по поддержанию патентной охраны тех или иных патентных решений. Однако защита меньшим числом патентов имеет недостаток, заключающийся в снижении гибкости при переуступке патентных прав на новые решения.

ЗАКЛЮЧЕНИЕ

В данной книге авторы преследовали цель в доступной форме изложить вопросы прикладной криптографии, связанные с разработкой скоростных шифров на основе управляемых операций, реализуемых на основе подстановочно-перестановочных сетей. Последние являются криптографическим примитивом, который широко исследован, начиная с середины 70-х годов прошлого столетия, однако до последнего времени данный тип сетей применялся при разработке блочных шифров как статический элемент, а именно – сеть рассматривалась как последовательность чередующихся фиксированных подстановок и перестановок. Даже при использовании управляемых сетей предусматривался выбор ее модификации только в зависимости от секретного ключа. Внесение элемента зависимости от преобразуемых данных делает эту управляемую сеть переменной операцией, изменяющейся при переходе от одного шифруемого блока к другому, и привносит новое качество, делающее сеть нелинейным элементом криптографического преобразования даже в том случае, если все отдельные модификации ее реализации являются линейными операциями.

Этот подход достаточно широко апробирован на примере битовых перестановок, зависящих от преобразуемых данных и реализуемых с помощью управляемых перестановочных сетей. Подробный материал можно найти в книге [14], ознакомление с которой даст более полную картину использования управляемых преобразований в проектировании блочных шифров. Переход от перестановочных сетей к подстановочно-перестановочным позволил устранить ряд недостатков, связанных с управляемыми перестановками, которые заключаются в сохранении веса Хемминга и сравнительно слабом распространении лавинного эффекта, которое происходит только за счет использования измененных битов в качестве управляющих битов.

Тот факт, что даже при наличии таких особенностей перестановки, зависящие от преобразуемых данных, позволяют построить стойкие шифры, реализуемые аппаратно с малой затратой схемотехнических ресурсов, показывает, что при переходе к управляемым операционным подстановкам могут быть получены еще более экономичные решения.

Описанные в книге классы операционных подстановок значительно расширили класс управляемых примитивов и сделали постановку задачи разработки переключаемых управляемых операций не случайным моментом в концепции управляемых преобразований, а одним из ее направлений, имеющим важное значении для разработки блочных шифров с простым расписанием ключей и обеспечивающим устранение слабых ключей и гомогенность шифрующего преобразования. Использование переключаемых управляемых операций совместно с симметричным расписанием

ключа позволяет реализовать универсальные криптосхемы, режим шифрования которых сменяется только путем переключения операций такого типа. Благодаря тому, что расписание использования ключа сохраняется в режиме расшифрования таким же, каким оно было в режиме зашифрования, достигается дополнительная экономия схемотехнических ресурсов, затрачиваемых на аппаратную реализацию шифра. Для массовых применений технологий шифрования это имеет большое значение. Последнее делает практически важным проведение дальнейших исследований этого вопроса и более полное обоснование подобных криптосхем.

Описанные в книге новые криптографические примитивы открывают новое продолжение в практическом синтезе скоростных блочных шифров. Полученные результаты, по мнению авторов, составляют существенный вклад в развитие подхода к практическому проектированию блочных шифров, основанному на использовании операций, зависящих от преобразуемых данных, и закрепляют в нем приоритет российских исследователей, ведущих работу в области открытой криптографии. Наиболее широко исследованным примитивом, используемым при разработке блочных шифров, являются подстановки размера 4 ×4, которые можно рассмотреть как управляемый элемент $\mathbf{F}_{4/0}$, реализующий единственную модификацию. До последнего времени этот элемент был минимальным среди тех, на основе которых были предложены практически значимые блочные шифры. Недавно в монографии [1] было дано обоснование разработки шифров с использованием управляемых перестановочных сетей, реализуемых с использованием УЭ $\mathbf{F}_{2/1}$ частного вида, а именно, переключающих элементов $\mathbf{P}_{2/1}$. В настоящей книге дана полная классификация нелинейных УЭ типа $\mathbf{F}_{2/1}$ и рассмотрены свойства операционных блоков, проектируемых на основе таких элементов. Они являются минимальными по размеру УЭ, представляющими интерес как криптографический примитив. Упорядочивая по возрастанию размера, можно привести следующий ряд УЭ $\mathbf{F}_{2/1}$, $\mathbf{F}_{3/0}$, $\mathbf{F}_{2/2}$, $\mathbf{F}_{3/1}$, $\mathbf{F}_{4/0}$, $\mathbf{F}_{2/3}$, $\mathbf{F}_{3/2}$, $\mathbf{F}_{4/1}$, $\mathbf{F}_{4/2}$, $\mathbf{F}_{4/3}$ и $\mathbf{F}_{4/4}$, которые эффективно реализуются на современной элементной базе. При этом для УЭ $\mathbf{F}_{2/1}$, $\mathbf{F}_{3/0}$, $\mathbf{F}_{2/2}$, $\mathbf{F}_{3/1}$, $\mathbf{F}_{4/0}$, $\mathbf{F}_{2/3}$, $\mathbf{F}_{3/2}$, $\mathbf{F}_{4/1}$ и $\mathbf{F}_{4/2}$ может быть дана достаточно полная классификация путем формулирования тех или иных критериев отбора и полного экспериментального опробования всех возможных вариантов. Из последнего ряда данная книга дает характеризацию УЭ $\mathbf{F}_{2/1}$, $\mathbf{F}_{2/2}$ и $\mathbf{F}_{3/1}$. Элементы $\mathbf{F}_{3/0}$ и $\mathbf{F}_{4/0}$ не были исследованы как самостоятельный примитив, поскольку они реализуют единственную модификацию. Элементы $\mathbf{F}_{3/0}$ были частично охарактеризованы как модификации УЭ $\mathbf{F}_{3/1}$. Классификация УЭ $\mathbf{F}_{2/3}$, $\mathbf{F}_{3/2}$, $\mathbf{F}_{4/1}$ и $\mathbf{F}_{4/2}$ представляет собой самостоятельную достаточно трудоемкую задачу, для достаточно полного решения которой потребуются усилия не одного исследователя. Авторы надеются, что появление данной книги будет стимулировать исследования свойств УЭ этих типов. Определенная классификация лучших УЭ $\mathbf{F}_{4/2}$ по определенным критериям как блоков подстановки 6×4, была дана ранее в связи с исследованием вариантов блоков подстановок, используемым в шифре DES.

Элементы типа $\mathbf{F}_{2/1}$ и $\mathbf{F}_{2/2}$ являются УЭ с модификациями, являющимися блоками подстановок минимального размера, представляющими интерес для построения операционных блоков, предназначенных для использования в криптографических алгоритмах. При этом УЭ $\mathbf{F}_{2/2}$ описываются двумя булевыми функциями от четырех переменных. Два таких элемента описываются четырьмя указанными функциями. Блок

подстановки размером 4×4 также описывается четырьмя булевыми функциями от четырех переменных. Данная аналогия обусловливает интерес рассмотрения особенностей проектирования шифров на основе УЭ $\mathbf{F}_{2/2}$ и подстановок типа 4×4. Ознакомившись с содержанием данной книги, можно видеть, что использование УЭ $\mathbf{F}_{2/2}$ позволяет реализовать следующие новые возможности:

1) разработка комбинированных универсальных криптосхем, сочетающих в себе подстановочно-перестановочную сеть и криптосхему Фейстеля,
2) выполнение преобразования подключей в зависимости от преобразуемых данных,
3) большое разнообразие универсальных криптосхем,
4) синтез криптографических примитивов нового типа – переключаемых операций, зависящих от преобразуемых данных; данный тип имеет важное практическое значение, поскольку он позволяет устранить ряд проблем связанных с использованием простого расписания ключа в блочных шифрах,
5) повышение доверия пользователей к шифрам, поскольку управляемые операции с «однородной» топологией, синтезируемые на основе УЭ минимального размера, являются естественными примитивами, на которые не падает подозрение о специально встроенных «потайных ходах»,
6) при использовании УЭ типа $\mathbf{F}_{2/2}$ могут быть естественным образом построены экономичные в реализации управляемые операции, выполняемые в зависимости от ключа и от преобразуемых данных. Это позволяет при необходимости избежать применения дополнительных операций для задания зависимости хода шифрования от секретного ключа. Одновременно с выполнением нелинейного преобразования подблока данных на него будет накладываться ключ.

Перечисленные моменты делают управляемые операции самостоятельным новым примитивом, весьма привлекательным для разработчиков шифров, ориентированных на широкое использование в современных информационных и телекоммуникационных системах. Заметим также, что управляемая операция сочетает в себе свойства, которые реализуются несколькими операциями «обычного» типа. Например, управляемые операции представляют собой частные варианты подстановок, выполняемых над всем преобразуемым блоком данных. Управляемые операции реализуют также своеобразное наложение одного подблока данных на другой, причем одновременно выполняется преобразование второго подблока данных.

Новые типы криптографических преобразований, зависящих от преобразуемых данных, имеют не только практическое, но и теоретическое значение. Последнее состоит в классификации минимальных УЭ, на основе которых могут быть построены эффективные криптографические операции, в введении нового примитива – переключаемых управляемых операций, в возможности разработки новых типов универсальных криптосхем и обосновании итеративных шифров с простым расписанием ключа, включая криптосистемы, не требующие изменения расписания ключа при смене режима шифрования. Важным является то, что некоторые математические свойства сложных, на первый взгляд, новых криптографических операций, проекти-

руемых на битовом уровне, достаточно просто и эффективно определяются аналитически для сравнительно большого размера входного блока данных.

Книга имеет также важное методическое значение, которое заключается в получении большого числа новых примеров криптографических примитивов, новых криптосхем, возможности выполнения расчетных работ по анализу свойств практически важных криптосхем и примитивов. Она дает наглядный пример проектирования современных шифров на битовом уровне, включающего теоретический анализ и обоснование выбираемых элементов криптосхем. Синтезируемые блочные шифры на базе УППС могут использоваться в качестве учебных примеров для вычисления дифференциальных и линейных характеристик не только отдельных элементов шифра, но и криптосхемы в целом.

Несколько искусственным выглядит включение в книгу *глав 1, 2, 3 и 10*. Однако в плане изложения введения в проблематику криптографии *главы 1, 2 и 3* существенно дополняют *главу 1* книги [14], продолжением которой является настоящий труд. *Глава 10* ориентирована на методические вопросы и может оказаться весьма полезной студентам, аспирантам и преподавателям. Она включает в себя достаточное количество задач, которые могут быть использованы для закрепления оригинального материала из монографии [14] и настоящей книги.

Таким образом, данная книга представляется интересной широкому кругу читателей: математикам, ведущим теоретические исследования в области одноключевой криптографии, специалистам-практикам, а также преподавателям высших учебных заведений, студентам и аспирантам. Несомненно, что стержнем данной книги является развитие подхода к синтезу блочных шифров, основанного на использовании переменных операций с большим числом реализуемых модификаций и заключающегося в обосновании новых структурных схем итеративных шифров, новых типов управляемых элементов, классификации последних как криптографических примитивов и синтезе управляемых операций различных типов, включая переключаемые.

Авторы осознают, что впервые отражают одно из важных направлений прикладной криптографии и поэтому изложение материала книги может оказаться далеко не оптимальным и требующим улучшения. Несомненно, что в ближайшей перспективе будут получены и новые результаты по развитию предлагаемого в книге подхода к построению высокоскоростных блочных шифров, что приведет к необходимости расширения последующих изданий книги, в которых авторы надеются исправить все обнаруженные неточности и опечатки.

Список литературы

1. Айерлэнд К., Роузен М. Классическое введение в современную теорию чисел. Пер. с англ. – М.: Мир, 1987. – 416 с.
2. Белкин Т.Г., Гуц Н.Д., Молдовян А.А., Молдовян Н.А. Линейный криптоанализ и управляемые перестановки. // Безопасность информационных технологий. М. МИФИ. 2000. № 2. – с. 70 – 80.
3. Бодров А.В., Молдовян Н.А., Молдовяну П.А. Перспективные программные шифры на основе управляемых перестановок // Вопросы защиты информации. 2002. № 1. – с. 44 – 50.
4. Бухштаб А.А. Теория чисел. – М.: Просвещение, 1966. – 384 с.
5. Вентцель Е.С. Теория вероятностей: Учебник для вузов – 6-е изд. стер. – М.: Высшая. школа., 1999. – 576 с.
6. Виноградов И.М. Основы теории чисел. – М.: Наука, 1972. – 167 с.
7. Винокуров А. Ю., Применко Э. А. Анализ тенденций подходов к синтезу современных блочных шифров // Безопасность информационных технологий. 2001 г. № 2. – с. 5 – 14.
8. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М.: Изд-во стандартов, 1989. – 20 с.
9. Гуц Н.Д., Изотов Б.В., Молдовян А.А., Молдовян Н.А. Проектирование двухместных управляемых операций для скоростных гибких криптосистем // Безопасность информационных технологий. М. МИФИ. 2001. № 2. – с. 14 – 23.
10. Гуц Н.Д., Молдовян А.А., Молдовян Н.А. Построение слоистых блоков управляемых перестановок различного порядка // Вопросы защиты информации. 2002. № 4. – с. 19 – 25.
11. Гуц Н.Д., Молдовян А.А., Молдовян Н.А. Построение управляемых блоков перестановок с заданными свойствами // Вопросы защиты информации. 1999. № 4. – с. 39 – 49.
12. Гуц Н.Д., Молдовян А.А., Молдовян Н.А. Гибкие аппаратно-ориентированные шифры на базе управляемых сумматоров // Вопросы защиты информации. 2000. № 1. – с. 8 – 15.
13. Гуц Н.Д., Еремеев М.А., Молдовян А.А. Алгоритм формирования расширенного ключа на основе блоков управляемых перестановок // Вопросы защиты информации. 2001. № 3. – с. 41 – 46.

14. Гуц Н.Д., Изотов Б.В., Молдовян А.А., Молдовян Н.А. Криптография: скоростные шифры // СПб, БХВ- Петербург. 2002. – 495 с.

15. Гуц Н.Д., Изотов Б.В., Молдовян Н.А. Управляемые перестановки с симметричной структурой в блочных шифрах // Вопросы защиты информации. 2000. № 4, – с. 57 – 65.

16. Гуц Н.Д., Молдовян Н.А. Рекурсивная модель построения блоков перестановок удвоенного порядка // Вопросы защиты информации, М. ВИМИ. 2003. № 2. – с. 2 – 8.

17. Денисов О.В. Асимптотическая формула для числа корреляционно-иммунных порядка *k* булевых функций // Дискретная математика. 1991. Т.3. 2. – с. 25 – 46.

18. Еремеев М.А. Управляемые операционные подстановки: синтез, свойства и применение // Безопасность информационных технологий, Москва, МИФИ. 2002. № 2. – с. 17 – 26.

19. Еремеев М.А., Изотов Б.В., Молдовян А.А. Способы скоростного шифрования в случае частой смены ключей // Труды всеармейской научно-практической конференции «Инновационная деятельность в вооруженных силах Российской федерации», СПб, ВУС, 2001. – с. 55 – 59.

20. Еремеев М.А., Коркишко Т.А., Мельник А.А., Молдовян А.А., Молдовяну П.А. Аппаратная поддержка программных шифров на базе управляемых перестановок // Вопросы защиты информации. 2002. № 2. – с. 26 – 37.

21. Еремеев М.А., Корниенко А.А., Максимов Ю.Н. Криптопреобразования и помехоустойчивое кодирование информации на основе свойств эллиптических кривых // Проблемы информационной безопасности, СПб. 2000. № 1. – с. 46 – 51.

22. Еремеев М.А., Максимов Ю.Н. Построение криптосистем на основе свойств эллиптических кривых // Безопасность информационных технологий. 1995. № 2. – с. 52 – 55.

23. Еремеев М.А., Молдовян А.А., Молдовян Н.А. Защитные преобразования информации в АСУ на основе нового примитива // Автоматика и телемеханика. 2002. № 12. – с. 35 – 47.

24. Еремеев М.А., Молдовян А.А., Молдовян Н.А. О реализации шифров на основе управляемых операционных подстановок // Приборостроение. Известия ВУЗов. СПб. 2003. № 3. – с. 13 – 19.

25. Еремеев М.А., Молдовян Н.А. Синтез аппаратно-ориентированных управляемых подстановок над векторами большой длины // Вопросы защиты информации, Москва. 2001. № 4. – с. 46 – 52.

26. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях. – М: Кудиц-Образ, 2001. – 368 с.

27. Изотов Б.В., Молдовян А.А., Молдовян Н.А. Скоростные методы защиты информации в АСУ на базе управляемых операций // Автоматика и телемеханика. 2001. № 6. – с. 168 – 184.

28. Кнут Д. Искусство программирования. Т.2 Получисленные алгоритмы. Третье издание. – М: МЦНМО, 1999.

29. Коблиц Н. Введение в эллиптические кривые и модулярные формы: Пер. с англ. - М.: Мир, 1988. – 320 с.

30. Крамер Г. Математические методы статистики. –М.: Изд-во «Мир». 1975. – 648 с.

31. Кузнецов Ю.В., Шкарин С.А. Коды Рида-Маллера (обзор публикаций) // Математические вопросы кибернетики. М.: Наука. 6. 1996. – с. 5 – 50.

32. Ланкастер П. Теория матриц: Пер. с англ. – М.: Наука, 1982. – 272 с.

33. Лидл Р., Нидеррайтер Г. Конечные поля / Пер. с англ. –М.: Мир, 1988. – 820 с. (т. 1, 2).

34. Логачев О.В., Сальников А.А., Ященко В.В. Бент-функции на конечной абелевой группе // Дискретная математика. 1997. Т.9.4. – с. 3 – 20.

35. Логачев О.В., Сальников А.А., Ященко В.В. Криптографические свойства дискретных функций // Материалы конференции «Московский университет и развитие криптографии в России». М.: МЦНМО. 2003. – с. 174 – 199.

36. Логачев О.В., Сальников А.А., Ященко В.В. Невырожденная нормальная форма булевых функций // Доклады Академии наук. 2000. Т.373. 2. – с. 164 – 167.

37. Логачев О.В., Сальников А.А., Ященко В.В. Некоторые характеристики «нелинейности» групповых отображений // Дискретный анализ и исследование операций. Серий 1. 2001. Т.8. 1. – с. 40 – 54.

38. Мельник А.А., Молдовян Н.А., Гуц Н.Д., Изотов Б.В., Коркишко Т.А. Реализация скоростного шифра на базе управляемых перестановок // Вопросы защиты информации. 2001. № 2. – с. 44 – 53.

39. Минаева Е.В. Современные подходы к конструированию оптимальных алгоритмов генерации расширенного ключа в итеративных блочных шифрах // Безопасность информационных технологий. 2000. № 2. – с. 24 – 26.

40. Молдовян А.А., Молдовян Н.А. Гибкие алгоритмы защиты информации в АСУ // Автоматика и телемеханика, 1998. № 8. – с. 166 – 174.

41. Молдовян А.А., Советов Б.Я. Криптография. СПб., Лань, 2000. – 218 с.

42. Молдовян Н.А. Скоростные блочные шифры. СПб, СПбГУ, 1998. – 230 с.

43. Ростовцев А. Г., Маховенко Е.Б. Введение в криптографию с открытым ключом. – СПб.: «Мир и Семья», 2001. – 336 с.

44. Рязанов Б.В. О распределении спектральной сложности булевых функций // Дискретная математика. 1994. Т.6.2. – с. 111 – 119.

45. Семаев И.А. О вычислении логарифмов на эллиптических кривых // Дискретная математика. 1996. Т. 8, № 1. – с. 65 – 71.

46. Степанов С.А. Арифметика алгебраических кривых. –М.: Наука. Гл. ред. физ.-мат. лит., 1991. –388 с.

47. Столлингс В. Криптография и защита сетей: принципы и практика., 2-е изд. : Пер. с англ. – Изд. Дом «Вильямс», 2001. – 672с.: ил.

48. Харин Ю.С., Берник В.И., Матвеев Г. В. Математические основы криптологии. – Минск.: БГУ. 1999. –319 с.

49. Шафаревич И.Р. Основы алгебраической геометрии: В 2-х т. –М.: Наука. Гл. ред. физ.-мат. лит., 1988.

50. Ященко В.В. О двух характеристиках нелинейности булевых отображений // Дискретный анализ и исследование операций, серий 1. 1998. Т. 5. 2. – с. 90 – 96.

51. Ященко В.В. О критерии распространения для булевых функций и о бент-функциях // Проблемы передачи информации. 1997. Т. 33. 1. – с. 75 – 86.

52. Ященко В.В. Свойства булевых функций, сводимые к свойствам их координатных функций // Вестник МГУ, серия «Математика». 1997. 4. – с. 11 – 13.

53. Benes V.E. Algebraic and Topological Properties of Connecting Networks // Bell Systems Technical Journal. 1962. Vol. 41. pp. 1249 – 1274.

54. Benes V.E. Mathematical Theory of Connecting Networks and Telephone Traffic // – New York.: Academic Press, 1965. – 319 p.

55. Bennett C.H., Brassard G. Quantum cryptography: Public key distribution and coin tossing // Proc. of IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, Dec. 1984, pp. 175 – 179.

56. Beth T., Schaefer F. Non-super singular elliptic curves for public key cryptosystems // Advances in Cryptology – EUROCRYPT'91, Springer-Verlag, 1992. pp.155 – 160.

57. Biham E. and Shamir A. Differencial Cryptanalysis of the Data Encryption Standart // Springer-Verlag, Berlin, Heidelberg, New York, 1993.

58. Biham E. New Types of Cryptoanalystic Attacks Using Related Keys // Advances in Cryptology – EUROCRYPT'93, Springer-Verlag, 1994. pp. 398 – 409.

59. Biryukov A., Wagner D. Advanced Slide Attacks // Advances in Cryptology – Proceeding EUROCRYPT'2000. LNCS. Vol.1807. Springer Verlag, 2000. pp. 589 – 606.

60. By Hang-Georg Ruck. A Note on Elliptic Curves Over Finite Fields // Math. Comp. Vol. 49, No. 179, pp. 301 – 304.

61. Carlet C. A Larger Class of Cryptographic Boolean Functions via a Study of the Maiorana-McFarland Construction. Advances in Cryptology - CRYPTO 2002 Proceedings.LNCS. Springer Verlag. 2002, Vol. 2442. pp. 549 – 564.

62. Ch. Lee, D.Hong, Sun. Lee, San. Lee, S. Yang, J. Lim. A chosen plaintext linear attack on block cipher CIKS-1. Springer-Verlag LNCS, Vol. 2513, pp. 456 – 468.

63. Chaum D. Blind Signature Systems. U.S. Patent # 4,759,063, Jul. 19, 1988.

64. Chaum D. Blind Signatures for Untraceable Payments. Advances in Cryptology: Proc. of CRYPTO'82. Plenum Press, 1983, pp. 199 – 203.

65. Chaum D. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. Communication of the ACM, Vol. 28, No. 10, Oct. 1985. pp. 1030 – 1044.

66. Cheung O.Y.H, Tsoi K.H., Leong P.H.W., and Leong M.P., Tradeoffs in Parallel and Serial Implementations of the International Data Encryption Algorithm, proceedings of CHES 2001, LNCS 2162, Springer-Verlag, 2001, pp. 333 – 337.

67. Ciet M., Lange T., Sica F, Quisquater Jean-J. Improved Algorithms for Efficient Arithmetic on Elliptic Curves Using Fast Endomorphisms. Advances in Cryptology. – EUROCRYPT 2003 Proceedings.LNCS. Springer-Verlag. 2003, Vol. 2656. pp. 388 – 400.
68. Couvegnes J.-M., Dewaghe L., Morain F. Isogeny cycles and the SEA algorithm. LIX/RR/96/03.
69. Diffie W., Hellman M.E. New Directions in Cryptography // IEEE Transactions on Information Theory.. 1976, Vol. IT-22. pp. 644 – 654.
70. ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE Transactions on Information Theory. 1985, Vol. IT-31, No. 4. pp.469 – 472.
71. Forre R. The Strict Avalanche Criterion: Spectral Properties of Boolean Functions and an Extended Definition // Advances in Cryptology – CRYPTO'88 Proceedings. LNCS. Springer Verlag. 1989. pp. 450 – 468.
72. Goots N.D., Moldovyan A.A., Moldovyan N.A. Fast encryption algorithm SPECTR-H64 // Proceedings of the International workshop, Methods, Models, and Architectures for Network Security / LNCS. Springer-Verlag. 2001. Vol. 2052. pp. 275 – 286.
73. Goots N.D., Moldovyan A.A., Moldovyan N.A., «Variable Bit Permutations: Linear Characteristics and Pure VBP-Based Cipher», Computer Science Journal of Moldova , 2004, Vol. 12.
74. Izotov B, Moldovyan A., Moldovyan N. Controlled Operations as a Cryptographic Primitive // Proceedings of the International workshop, Methods, Models, and Architectures for Network Security, Lecture Notes in Computer Science, Berlin.: Springer-Verlag, Vol. 2052. 2001. pp. 230 – 241.
75. Izotov B.V., Goots N.D., Moldovyan A.A. and Moldovyan N.A., Fast Ciphers for Cheap Hardware: Differential Analysis of SPECTR-H64, Proceedings of the International workshop, Methods, Models,and Architectures for Network Security (MMM-ANCS'03). LNCS, Springer-Verlag, Vol. 2776. 2003. pp. 449 – 452.
76. Joye M. and Quisquater J.-J. Efficient computation of full Lucas sequences // Electron. Lett., 1996, Vol. 32(6), pp.537 – 538.
77. Kaliski B.S. One-wey Permutations on Elliptic Curves // Journal of Cryptology, Vol. 3, No. 3, 1991. pp. 187 – 199.
78. Kam J.B., Davida G.I. Structured Design of Substitution-Permutation Encryption Networks. // IEEE Transactions on Computers. 1979. Vol. 28, No. 10. pp. 747 – 753.
79. Kaps J. and Paar Chr., «Fast DES Implementations for FPGAs and its Application to a Universal Key-Search Machine», proceedings of 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), August 17 – 18, Ontario, Canada.
80. Kasuya T., Ichikawa T., Matsui M., Hardvware Evaluation of the AES Finalists, 3rd AES Confer. Proc. April 13-14, 2000. New York, NY, USA (http:/www.nist.gov/aes)
81. Kelsey J., Schneier B., Wagner D., Related-Key Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES, Advances in Cryptology – CRYPTO'96 Proceedings. LNCS 1109. Springer Verlag. 1996. pp. 237 – 251.

82. Ko Y., Hong D., Hong S., Lee S., Lim J.. Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property. Proceedings of the International workshop, Methods, Models, and Architectures for Network Security, Lecture Notes in Computer Science , Springer-Verlag, Berlin: 2003. Vol. 2776. pp. 298 – 307.

83. Koblitz N. Constructing Elliptic Curve Cryptosystems in Characteristic 2 // Advances in Cryptology – CRYPTO'90. 1991. pp. 376 – 385.

84. Koufopavlou O., Sklavos N. and Kitsos P., Cryptography in Wireless Protocols: Hardware and Software Implementations, proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'03), Thailand, May 25 – 28, 2003.

85. Kurosawa K., Satoh T. Design of SAC/PC(l) of Order k Boolean Functions and Three Other Cryptographic Criteria // Advances in Cryptology – EUROCRYPT'97 Proceedings. LNCS. Springer Verlag. 1998. pp. 434 – 449.

86. Lercier R., Lubicz D. Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time. Advances in Cryptology – EUROCRYPT 2003 Proceedings.LNCS. Springer Verlag. 2003, Vol. 2656. pp. 360 – 373.

87. Matsui M., Linear cryptoanalysis method for DES cipher. Advances in Cryptology – Proc. Eurocrypt'93, Springer Verlag, 1994. pp. 386 – 397.

88. Meier W., Staffelbach O. Efficient Multiplication on Certain Nonsupersingular Elliptic Curves // Advances in Cryptology – CRYPTO'92, Springer-Verlag, 1993. pp. 333 – 343.

89. Menezes A. J., Vanstone S.A. Handbook of Applied Cryptography. CRC Press, 1996. – 780 p.

90. Menezes A., Okamoto T. and Vanstone S.A. Reducing elliptic curve logarithms to logarithms in a finite field // University of Waterloo, preliminary version, 1990.

91. Miller S. Uses of elliptic curves in cryptography // Advances in Cryptology - CRYPTO'85. 1986. Vol. 218. pp. 419 – 426.

92. Moldovyan A.A., Moldovyan N.A.. A cipher based on data-dependent permutations // Journal of Cryptology. 2002. Vol. 15. pp. 61 – 72.

93. Moldovyan A.A., Moldovyan N.A. Fast Software Encryption Systems for Secure and Private Communication // 12th International Conf. on Computer Communication. Seoul, Korea August 21-24, 1995. Proceedings, Vol. 1, pp. 415 – 420.

94. Moldovyan A.A., Moldovyan N.A. Software Encryption Algorithms for Transparent Protection Technology // Cryptologia, January,1998, Vol.XXII. No. 1. pp. 56 – 68.

95. Moldovyan A.A., Moldovyan N.A., Moldovyanu P.A. // Computer Science Journal of Moldova, 1994, Vol. 2, No. 3, pp. 269 – 282.

96. Moldovyan A.A., Moldovyan N.A., Sovetov B.Ya. Software-Oriented Ciphers for Computer Communication Protection // Int.Conf. Applications of Computer Systems. ACS'97 Proceedings. November 13-14, 1997. Szczecin, Poland. pp. 443 – 450.

97. Moldovyan A.A.. Fast block ciphers based on controlled permutations // Computer Science Journal of Moldova. 2000, Vol. 8, No. 3, pp. 270 – 283.

98. Moldovyan N.A. Provably Indeterminate 128-bit Cipher // Computer Science Journal of Moldova, 1997, Vol. 5 , No. 2(14), pp. 185 – 197.

99. Moldovyan N.A., «Fast DDP-Based Ciphers: Design and Differential Analysis of Cobra-H64», Computer Science Journal of Moldova, 2003, Vol. 11, No. 3 (33), pp. 292 – 315

100. Nyberg K. On the Construction of Highly Nonlinear Permutations // Advances in cryptology – EUROCRYPT'92 Proceedings. LNCS. Springer Verlag. 1993. pp. 55 – 64.

101. Nyberg K. S-Boxes and Round Functions with Controllable Linearity and Differential Uniformity // Fast Software Encryption. Second international workshop proceedings. LNCS 1008. Springer Verlag. 1994. pp. 111 – 130.

102. Parker S. Notes on Shuffle/Exchange-Type Switching Networks // IEEE Transactions on Computers. 1980. Vol. C-29. No. 5, pp. 213 – 222.

103. Pieprzyk J., Hardjono Th., Seberry J. Fundamentals of Computer Security. Springer-Verlag. Berlin, 2003. – 677 p.

104. Pollard J. Monte Carlo Methods for Index Computation (mod p) // Math. Comp., Vol. 32, No. 143, pp. 918 – 924.

105. Portz M. A. Generalized Description of DES-based and Benes-based Permutation generators // Advanced in Cryptology – AUSCRYPT'92 // Lecture Notes in Computer Science. Springer-Verlag, 1992, Vol. 718, pp. 397 – 409.

106. Preneel B., Bosselaers A., Rijmen V., Van Rompay B., Granboulan L., Stern J., Murphy S., Dichtl M., Serf P., Biham E., Dunkelman O., Furman V., Koeune F., Piret G., Quisquater J-J.,Knudsen L., Raddum H. // Comments by the NESSIE Project on the AES Finalists// 24 may 2000, http://www.nist.gav/aes.

107. Prennel B., Van Leekwijk W., Wan Linden L., Govaerts R., Vandewalle J. Propagation characteristics of boolean functions // Advances in Cryptology – EUROCRYPT'90 Proceedings, LNCS 473, 1991. pp. 161 – 173.

108. Rivest R., Shamir A., Adleman A.. A method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communication of the ACM, Vol. 21, No. 2. Feb 1978, pp. 120 – 126.

109. Rivest R.L. The RC5 Encryption Algorithm // 2d Int. Workshop «Fast Software Encryption». Proc./. Springer-Verlag LNCS. 1995, Vol. 1008. pp. 86 – 96.

110. Rivest R.L., Robshaw M.J.B., Sidney R., Yin Y.L. The RC6 Block Cipher. Proc. of 1st Advanced Encryption Standard Candidate Conference, Venture, California, Aug. 20 – 22, 1998, (http://www.nist.gov/aes).

111. Rueppel R.A. Analysis and design of stream ciphers. – N. Y.: Springer-Verlag, 1986 г.

112. Savas E., Schmidt T. A., Koc C. K. Generating Elliptic Curves of Prime Order. proceedings of CHES 2001, LNCS 2162, Springer-Verlag, 2001, pp. 142 – 158.

113. Schneier B., Applied Cryptography: Protocols, Algorithms, and Source Code (Second Edition) – New York.: John Wiley & Sons. 1996. – 758 p.

114. Seberry J., Zhang X-M., Zheng Y. Nonlinearly Balanced Boolean Functions and Their Propogation Chracteristics // Advances in Cryptology – CRYPTO'93 Proceedings. LNCS. Springer Verlag. 1994. pp. 49 – 60.

115. Seberry J., Zhang X-M., Zheng Y. Relations Among Nonlinearity Criteria // Advances in Cryptology – EUROCRYPT'94 Proceedings. LNCS. Springer Verlag. 1995. pp. 376 – 388.

116. Shannon C. E. Communication Theory of Secrecy Systems, Bell Systems Technical Journal, Vol. 28, 1949, P.656 – 715.

117. Sklavos N. and Koufopavlou O. «Architectures and VLSI Implementations of the AES-Proposal Rijndael», IEEE Transactions on Computers, Vol. 51, Issue 12, 2002, pp. 1454 – 1459.

118. Sklavos N. and Koufopavlou O., «Architectures and FPGA Implementations of the SCO (-1,-2,-3) Ciphers Family», proceedings of the of 12th International Conference on Very Large Scale Integration, (IFIP VLSI SOC '03), Darmstadt, Germany, December 1-3, 2003.

119. Sklavos N., Moldovyan A. A., and Koufopavlou O. Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation. Workshop MMM-ANCS'2003 Proc // LNCS, Springer-Verlag, Berlin, 2003, Vol. 2776, pp. 343 – 354.

120. Sklavos N., Moldovyan A.A., Koufopavlou O., «High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers», Mobile Networks and Applications, Special Issue on : Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks, MONET Journal, Kluwer, 2004.

121. Staffelbach O., Meier W. Nonlinearity Criteria for Cryptographic Functions // Advances in Cryptology – EUROCRYPT'89 Proceedings. LNCS. Springer Verlag. 1990. pp. 549 – 562.

122. Steven M., Merrit M. An Attack on the Interlock Protocol When Used for Autentication // IEEE Transactions on Information Theory, Vol. 40, No. 1, 1994. pp. 273 – 275.

123. Van Rompay B., Knudsen L., Rijmen V. Differential cryptanalysis of the ICE encryption algorithm // Proceedings of the 6th International Workshop, «Fast Software Encryption - FSE'98». LNCS, Springer-Verlag, Vol. 1372, 1998, pp. 270 – 283.

124. Waksman A. A Permutation Network // Journal of the ACM. 1968. Vol. 15. No. 1, pp. 159 – 163.

125. Xiao G.Z., Masser J.L. A Spectral characterization of correlation-immune combining functions // IEEE Trans. Inform. Theory. 1988. No. 34. pp. 569 – 571.