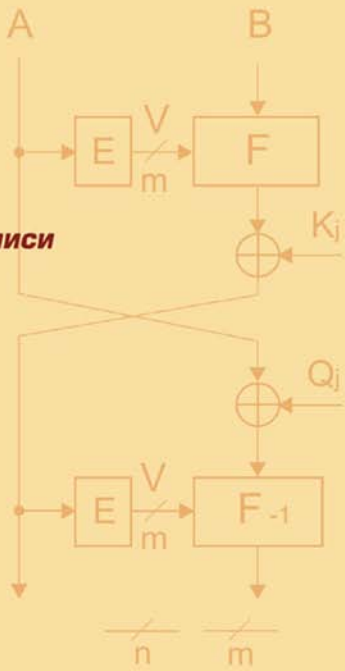
Н. А. Молдовян
А. А. Молдовян

ВВЕДЕНИЕ В КРИПТОСИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

- *Проблематика криптографии*
- *Элементы теории чисел*
- *Двухключевые криптосистемы*
- *Системы электронной цифровой подписи с составным модулем*
- *Открытое распределение ключей и открытое шифрование*
- *Управление ключами и протоколы*



УЧЕБНОЕ ПОСОБИЕ

bhv

Н. А. Молдовян, А. А. Молдовян

Введение в криптосистемы с открытым ключом

Санкт-Петербург
«БХВ-Петербург»
2005

Содержание

Введение **7**

Глоссарий **9**

Глава 1. Проблематика криптографии и симметричные шифры **15**

1.1. Проблемы защиты информации в компьютерных системах 15
1.2. Задачи криптографии 17
1.2.1. Обеспечение секретности 17
1.2.2. Современные приложения 24
1.3. Элементы симметричного шифрования 31
1.3.1. Условная и безусловная секретность 31
1.3.2. Общие вопросы разработки шифров 35
1.3.3. Композиционные и итеративные блочные шифры 37
1.3.4. Управляемые операции как криптографический примитив 43
1.4. Особенности проектирования блочных шифров на основе управляемых операций 45
1.4.1. Управляемые операции и отображения 45
1.4.2. Расписание использования ключа 47
1.4.3. Варианты криптосхем 50
1.4.4. Криптосистемы с гибким алгоритмом и требования к алгоритму предвычислений 55
1.5. Вероятностные шифры 57
1.5.1. Гомофонические шифры 57
1.5.2. Шифры с простым вероятностным механизмом 58
1.6. Особенности приложений 60
1.6.1. Длина ключа и стойкость 60
1.6.2. Повышение стойкости шифрования при ограничении длины секретного ключа 63
1.6.3. Применение долговременных ключевых элементов при ограничении длины секретного ключа 64
1.6.4. Варианты реализации шифров 66

Глава 2. Элементы теории чисел 71

2.1. Некоторые определения и утверждения 71
2.2. Функция Эйлера 74
2.3. Алгоритм Евклида 76
2.4. Расширенный алгоритм Евклида 78
2.5. Показатели и первообразные корни 79
2.6. Теоремы о числе классов с заданным показателем 83
2.7. Китайская теорема об остатках 86
2.8. Алгоритм возведения в степень по модулю 88
2.9. Нахождение чисел, относящихся к заданному показателю 90
2.10 Теоремы о числе решений степенных сравнений 92

Глава 3. Двухключевые криптосистемы 99

3.1. Понятие электронной цифровой подписи 99
3.2. Сравнительная характеристика одноключевых и двухключевых шифров 102
3.3. От открытого распределения ключей до электронной цифровой подписи 104
3.3.1 Система распределения ключей Диффи–Хеллмана 104
3.3.2 Открытый шифр Эль-Гамаля 106
3.4. Системы ЭЦП на основе задачи дискретного логарифмирования 107
3.4.1. Общие положения 107
3.4.2. Сокращение длины подписи 113
3.4.3. Примеры анализа слабых ЭЦП 117
3.4.4. Цифровая подпись Эль-Гамаля 120
3.4.5. Системы ЭЦП с дополнительными свойствами 122
3.4.6. Стандарты ЭЦП 127
3.5. Проблема бесключевого шифрования 129
3.6. Криптосистема RSA 136
3.6.1. Криптографические преобразования в RSA 136
3.6.2. Вопросы выбора параметров системы RSA 140
3.6.3. Слепая подпись на основе системы RSA 143

Глава 4. Системы ЭЦП с составным модулем 146

4.1. Цифровая подпись Рабина 146
4.2. Цифровая подпись Фиата–Шамира 148
4.3. Обобщение схемы Фиата–Шамира 150
4.4. Уменьшение размера открытого ключа в схеме Фиата–Шамира 151
4.5. Схема ЭЦП Онга–Шнорра–Шамира 153
4.6. Варианты схемы Эль-Гамаля с составным модулем 154

4.7. Схемы на основе сложности извлечения корней по составному модулю 157
4.8. Расширение криптосистемы RSA 158
4.8.1. Модифицированные версии и совместимость с RSA 159
4.8.2. Ограничения на выбор параметров криптосхемы 162
4.9. Переход к схемам ЭЦП с простым модулем 163
4.9.1. Переход к простому модулю в ЭЦП Фиата–Шамира 163
4.9.2. Сокращение размера подписи 165
4.9.3. Цифровая подпись Шнорра 166

Глава 5. Открытое распределение ключей и открытое шифрование 168

5.1. Схема открытого шифрования Рабина 169
5.2. Схемы на основе сложности задачи извлечения корней по модулю 172
5.2.1. Открытое шифрование 172
5.2.2. Схема с сокращенной длиной открытого ключа 174
5.2.3. Открытое распределение ключей 175
5.2.4. Схема на основе сложности извлечения корней второй степени 178
5.3. Открытое распределение общего ключа между тремя и более пользователями 179
5.4. Вероятностные механизмы в двухключевых шифрах 181

Глава 6. Хэш-функции 184

6.1. Защита от модифицирования данных 184
6.2. Криптографические контрольные суммы 187
6.3. Построение хэш-функций на основе блочных преобразований 190
6.4. Нахождение коллизий в общем случае 196
6.5. Атака «встреча посередине» 201

Глава 7. Управление ключами и протоколы 205

7.1. Вопросы управления ключами 205
7.2. Генерация ключей 209
7.3. Схемы распределения ключей 211
7.3.1. Централизованное распределение ключей 213
7.3.2. Открытое распределение ключей 217
7.3.3. Распределение ключей на основе симметричного шифрования 220
7.3.4. Распределение ключей на основе двухключевых шифров 223
7.4. Разделение секрета 224

7.4.1. Схемы на основе китайской теоремы об остатках 225
7.4.2. Пороговые схемы на основе многочленов 227
7.4.3. Неоднородное ключевое пространство 229
7.5. Криптографические протоколы 230
7.5.1. Понятие криптографического протокола 230
7.5.2. Временной замок 233
7.5.3. Защита материальных объектов от подделки 235
7.5.4. Электронная жеребьевка 235
7.5.5. Игра в покер по телефону (электронное казино) 237
7.6. Понятие слепой подписи 239
7.7. Протоколы с нулевым разглашением 242
7.7.1. Протокол Фиата–Шамира 242
7.7.2. Протоколы на основе системы RSA 244
7.7.3. Протокол с одним раундом проверки 245
7.7.4. Протоколы на основе сложности разложения на множители 247
7.7.5. Протоколы на основе сложности дискретного логарифмирования 249
7.8. Инфраструктура открытых ключей (PKI) 250

Глава 8. Криптографический практикум. 256

8.1. Задания для практических занятий. 256
8.1.1. Открытое распределение ключей 257
8.1.2. Открытое шифрование 259
8.1.3. Схемы ЭЦП 261
8.1.4. Генерация простых чисел 268
8.2. Задачник 275
8.2.1. Арифметические задачи 275
8.2.2. Схемы цифровой подписи 277
8.2.3. Хэш-функции 280

Заключение 282

Список литературы 283

Памяти нашего отца
посвящается

Введение

Криптографические методы нашли широкое применение в практической информатике для решения многочисленных проблем информационной безопасности. В проблематике современной криптографии можно выделить следующие три типа основных задач:

- обеспечение конфиденциальности (секретности);
- обеспечение анонимности (неотслеживаемости);
- обеспечение аутентификации информации и источника сообщений.

Первый тип задач относится к защите информации от несанкционированного доступа с использованием шифрования по секретному ключу. Доступ к информации имеют только обладатели секретного ключа. Такие задачи решаются с помощью одноключевых криптосистем, которые применяются человечеством уже несколько тысячелетий. Второй и третий типы задач обязаны своей постановкой массовому применению электрических способов обработки и передачи информации. Решение задач обеспечения анонимности (при обращении электронных денег или при тайном электронном голосовании) и аутентификации информации связано с открытием двухключевой криптографии, сделанным около 30 лет назад. С того момента проблематика двухключевой криптографии является наиболее интенсивно развивающимся направлением современной криптографии. Появившееся новое направление криптографии востребовало широкого применения теории чисел, комбинаторики, теории сложности, дискретной математики и других ее разделов. Предоставленная двухключевой криптографией возможность эффективного решения новых типов задач привела к возникновению и широкому практическому использованию новых информационных технологий, дающих существенную экономическую отдачу. В последнее время широко обсуждается проблема электронного правительства, под которым понимается максимальное использование компьютерной техники и электронного документооборота по реализации управленческих функций правительства. Основой придания юридической значимости электронной информации, циркулирующей в информацион-

ных системах в банковской сфере, управленческой деятельности правительства и других государственных органов, включая вооруженные силы и спецслужбы, является двухключевая криптография, обеспечивающая эффективные решения по обеспечению аутентификации и неотслеживаемости. Использование криптографии в современных информационных системах позволяет достигнуть не только максимальной оперативности в финансовой, деловой и управленческой деятельности, но и создать практически непреодолимые преграды для многих противоправных и антиобщественных действий преступных элементов, включая решение проблемы фальсификации денег и ценных бумаг.

В настоящее время имеется достаточно обширная русскоязычная литература по всем разделам криптографии, однако сохраняется потребность в учебных пособиях, в которых детально рассматриваются механизмы построения систем электронной цифровой подписи (ЭЦП) и вопросы обоснования выбора параметров ЭЦП в их взаимосвязи с рассмотренными математическими результатами и методическим материалом для выполнения практических заданий.

Предлагаемая вниманию читателей книга рассматривает в основном вопросы построения шифров с открытым ключом, включая системы ЭЦП и слепой подписи. Симметричные криптосистемы характеризуются достаточно кратко, а именно в плане изложения основных сведений. Дается математический минимум для освоения и полного понимания основных систем ЭЦП и идей, лежащих в их основе. Раскрываются криптографические и организационные основы придания юридической силы электронным документам. Книга ориентирована на использование при преподавании дисциплин «Криптографические методы защиты информации», «Теоретические основы криптографии», «Теоретические основы компьютерной безопасности» и «Криптография» в рамках подготовки специалистов по следующим вузовским специальностям: «Компьютерная безопасность», «Организация и технология защиты информации», «Комплексное обеспечение информационной безопасности автоматизированных систем», «Автоматизированные системы обработки информации и управления», «Прикладная математика».

Глоссарий

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, входными текстами для которого являются блоки фиксированного размера.

Вероятностное шифрование — процесс шифрования с использованием случайных параметров.

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. Существенным является то, что для реализации различных протоколов в двухключевых криптосистемах не требуется, чтобы секретный ключ был известен более чем одному поль-

зователю. Это позволяет реализовать криптографические протоколы, предусматривающие взаимодействие сторон в условиях недоверия друг другу. Двухключевые криптосистемы лежат в основе современных систем электронной цифровой подписи.

Зашифрование — процесс преобразования информации с целью предотвращения несанкционированного доступа. Зашифрование преобразует исходный открытый текст в шифртекст (или криптограмму). Шифртекст называют иногда шифрованным текстом. Зашифрование есть процесс преобразования данных, обеспечивающий защиту информации путем сокрытия ее смысла. Стойкость зашифрования (или уровень защищенности) определяется тем, что при зашифровании используется один или несколько секретных параметров, которые предполагаются известными только законным (легальным) пользователям.

Ключ — секретный параметр, управляющий ходом шифрования. Ключ определяет выбор конкретного варианта преобразования для зашифрования и расшифрования из множества преобразований, составляющих шифр. Синонимами являются секретный ключ, ключ шифрования.

Криптоанализ — процесс (алгоритм) получения исходного текста по шифрованному 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 получения оценки стойкости шифра.

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

Криптографический примитив — в широком смысле это операция или процедура, используемая в качестве элемента шифра, в узком — это операции и процедуры, которые определяют 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.

Криптосистема с закрытым (секретным) ключом — традиционная криптосистема, использующая секретный ключ, известный более чем одному пользователю.

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).

Лобовое нападение (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.

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).

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

Противник — субъект (специальная программа, 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 битовые цепочки, являющиеся частью секретного ключа. Представляет собой вариант отказа от сложных процедур преобразования секретного 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.

Разделение секрета — распределение некоторых долей секретного ключа между несколькими хранителями секрета таким образом, что любой из них или коалиция из малого их числа не могут восстановить секретный ключ. При этом восстановление ключа становится возможным, если численность коалиции хранителей секрета будет равна или превысит некоторое пороговое значение.

Расшифрование — процесс восстановления открытого текста по шифрованному тексту и известному ключу. (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».

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.

Симметричная криптосистема — криптосистема с закрытым ключом. Симметричность означает, что ключи, задающие пару взаимно обратных

криптографических преобразований, могут быть получены один из другого с небольшой трудоемкостью.

Система тайного электронного голосования — криптографический протокол, обеспечивающий тайну голосования и возможность проверки каждым голосующим, как учтен его голос («за» или «против»). При этом никто, включая избирательный комитет, не сможет установить, как проголосовал избиратель.

Слепая подпись — протокол формирования электронной цифровой подписи, позволяющий сформировать правильную ЭЦП, соответствующую некоторому цифровому сообщению, которое подписывающая сторона получает в зашифрованном (т. е. недоступном для нее) виде. Принципиальным требованием к системам слепой подписи является *анонимность* правильной подписи. Подписывающий не должен иметь возможности однозначно установить связь между правильной подписью и какой-либо информацией, которую он получал в процессе подписывания «вслепую», если он подписал таким способом два или более сообщения.

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

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

Уравнение вычисления подписи — уравнение, задающее определенное соотношение между секретным ключом, значением хэш-функции от документа и цифровой подписью. Служит для формирования электронной цифровой 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 зна-

чения хэш-функции). Хэш-функции представляют собой специальный класс криптографических контрольных сумм, обычно вычисляемых 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 (двухключевой шифр) — двухключевая криптосистема. Синонимами являются термины «шифр (криптосистема) с публичным (открытым) ключом», «асимметричная криптосистема».

Шифр (криптосистема) — совокупность алгоритмов, используемых при зашифровании и расшифровании.

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.

ГЛАВА 1

Проблематика криптографии и симметричные шифры

1.1. Проблемы защиты информации в компьютерных системах

Криптография, история которой охватывает несколько тысячелетий, зародилась из потребности передачи секретной информации. Длительное время она была связана только с разработкой специальных методов преобразования информации с целью ее представления в форме, недоступной для потенциального злоумышленника. С началом применения электронных способов передачи и обработки информации задачи криптографии начали расширяться. В настоящее время, когда компьютерные информационные технологии нашли массовое применение, проблематика криптографии включает многочисленные задачи, которые не связаны непосредственно с засекречиванием информации. Современные проблемы криптографии включают разработку систем электронной цифровой подписи и тайного электронного голосования, протоколов электронной жеребьевки и аутентификации удаленных пользователей, методов защиты от навязывания ложных сообщений и т. п.

Многие задачи практической информатики эффективно решаются с использованием криптографических методов. В криптографии рассматривается некий злоумышленник (оппонент, криптоаналитик противника, нарушитель, нелегальный пользователь), который осведомлен об используемых криптографических алгоритмах, протоколах, методах и пытается вскрыть их. Вскрытие криптосистемы может заключаться, например, в несанкционированном чтении информации, формировании чужой подписи, изменении результатов голосования, нарушении тайны голосования, модифицировании данных, которое не будет обнаружено законным получателем. Разнообразные действия оппонента в общем случае называются криптографической атакой (нападением). Специфика криптографии состоит в том, что она направлена на разработку методов, обеспечивающих стойкость к любым действиям зло-

умышленника, хотя на момент разработки криптосистемы невозможно предусмотреть все возможные способы атаки, которые могут быть изобретены в будущем на основе новых идей, достижений теории и технологического прогресса. Центральным является вопрос, насколько надежно решается та или иная криптографическая проблема. Ответ на этот вопрос непосредственно связан с оценкой трудоемкости каждой конкретной атаки на криптосистему. Решение такой задачи, как правило, чрезвычайно сложно и составляет самостоятельный предмет исследований, называемый *криптоанализом*. *Криптография и криптоанализ* образуют единую область науки — *криптологию*, которая в настоящее время является разделом математики, имеющим важные приложения в современных информационных технологиях.

Широкое применение компьютерных технологий в системах обработки данных и управления привело к обострению проблемы защиты информации от несанкционированного доступа. Защита информации в компьютерных системах обладает рядом специфических особенностей, связанных с тем, что информация не является жестко связанной с носителем, может легко и быстро копироваться и передаваться по каналам связи. Это создает большое число угроз информации, которые могут быть реализованы как со стороны внешних, так и внутренних нарушителей.

Радикальное решение проблем защиты информации, циркулирующей в высокопроизводительных автоматизированных системах, может быть получено на базе использования криптографических методов. При этом важным является применение скоростных алгоритмов шифрования, которые не приводят к снижению производительности компьютерных и телекоммуникационных систем. Криптографические преобразования данных являются гибким и эффективным средством обеспечения их конфиденциальности, целостности и подлинности. Использование методов криптографии в совокупности с необходимыми техническими и организационными мероприятиями может обеспечить защиту от широкого спектра потенциальных угроз.

Потребности современной практической информатики привели к возникновению нетрадиционных задач защиты электронных данных, одной из которых является аутентификация электронной информации в условиях, когда обменивающиеся этой информацией стороны не доверяют друг другу. Эта проблема связана с созданием систем электронной цифровой подписи. Теоретическая база для решения этой проблемы появилась после открытия *двухключевой криптографии* американскими исследователями Диффи и Хеллманом в середине 1970-х гг. [22], которое составило блестящее достижение многовекового эволюционного развития криптографии. Революционные идеи двухключевой криптографии привели к резкому росту числа открытых исследований в этой области, показали новые пути развития криптографии, ее

далеко не исчерпанные возможности и уникальное значение в современных условиях бурного развития электронных информационных технологий.

1.2. Задачи криптографии

1.2.1. Обеспечение секретности

Слово «криптография» в переводе с греческого языка означает «тайнопись», что вполне отражает ее первоначальное предназначение. Примитивные с позиций сегодняшнего дня криптографические методы известны с древнейших времен и длительное время рассматривались скорее как некоторые ухищрения, чем как строгая научная дисциплина. Классической задачей криптографии является обеспечение обратимости преобразования некоторого *исходного текста (открытого текста)* в кажущуюся случайной последовательность знаков, называемую *шифртекстом (закрытым текстом)*, или *криптограммой*. При этом шифртекст может содержать как новые знаки, так и уже имеющиеся в исходном сообщении. Количество знаков в криптограмме и в исходном тексте в общем случае может различаться. Непременным требованием является возможность однозначного восстановления исходного текста в полном объеме, используя лишь некоторые логические действия с символами шифртекста. В далекие времена надежность сохранения информации в тайне определялась секретностью самого метода преобразования.

Однако секретность алгоритма принципиально не может обеспечить его *безусловную стойкость*, т. е. невозможность чтения криптограммы противником, обладающим бесконечными вычислительными ресурсами. Поскольку секретные алгоритмы недоступны для проведения широкомасштабных криптоаналитических исследований, то по сравнению с открытыми алгоритмами имеется значительно более высокая вероятность того, что впоследствии будут найдены уязвимые места и эффективные способы их взлома. В связи с этими обстоятельствами в настоящее время наиболее широко используются открытые алгоритмы, прошедшие длительное тестирование и обсуждение в открытой криптографической литературе.

Стойкость современных криптосистем основывается не на секретности алгоритма, а на секретности некоторой информации сравнительно малого размера, называемой *секретным ключом*. Ключ используется для управления процессом криптографического преобразования (шифрования) и является легко сменяемым элементом криптосистемы. Ключ может быть заменен пользователями в произвольный момент времени, тогда как сам алгоритм шифрования является долговременным элементом криптосистемы и связан с длительным этапом разработки и тестирования.

При прочих равных условиях отсутствие полных данных об алгоритме шифрования существенно (при адекватной его реализации) затрудняет проведение криптоаналитической атаки. Поэтому были предложены современные шифры, в которых непосредственно действующий алгоритм шифрования является легко сменяемым элементом и выбирается случайно. При этом общая структура криптосистемы открыта для детального обсуждения, что позволяет оценить ее стойкость в целом. Такие шифры реализуются как гибкие криптосистемы, в которых алгоритм, действующий в сеансе шифрования, формируется по специальному *алгоритму инициализации (предвычислений)*. Этот алгоритм является открытым, а сам действующий алгоритм шифрования — неизвестным и зависимым от секретного ключа пользователя.

В течение многих веков криптография была предметом избранных — жрецов, правителей, крупных военачальников и дипломатов. Несмотря на малую распространенность, использование криптографических методов вскрытия шифров противника оказывало существенное воздействие на исход важных исторических событий. Известен не один пример, когда переоценка используемых средств шифрования приводила к военным и дипломатическим поражениям. Несмотря на применение криптографических методов в важных областях, эпизодическое использование криптографии не могло даже близко подвести ее к той роли, которую она имеет в современном обществе. Своим превращением в научную дисциплину криптография обязана потребностям практики и развитию электронных информационных технологий.

Пробуждение значительного интереса к криптографии и ее развитие началось с XIX в., что связано с зарождением электросвязи. В XX столетии секретные службы большинства развитых стран стали относиться к этой дисциплине как к обязательному инструменту своей деятельности.

Говоря об исторических аспектах научных исследований в области криптографии, необходимо отметить тот факт, что весь период с древних времен до 1949 г. можно назвать донаучным периодом, когда средства закрытия письменной информации не имели строгого математического обоснования. Поворотным моментом, придавшим криптографии научность и выделившим ее в отдельное направление математики, явилась публикация в 1949 г. статьи К. Э. Шеннона «Теория связи в секретных системах» [38]. Указанная работа послужила основой для развития *одноключевых симметричных криптосистем*, в которых предполагается обмен секретными ключами между корреспондентами. Впоследствии в силу особенностей построения симметричные шифры были разделены на две криптосистемы: *поточные* и *блочные шифры*. Отличительная особенность первых состоит в преобразовании каждого символа в потоке исходных данных, тогда как вторые осуществляют последовательное преобразование целых блоков данных.

Фундаментальным выводом из работы Шеннона стало определение зависимости надежности алгоритма от размера и качества секретного ключа, а также от *информационной избыточности* исходного текста. Шеннон ввел формальное определение информации и функции ненадежности ключа как его неопределенности при заданном количестве известных битов закрытого текста. Кроме того, им было введено важное понятие *расстояния единственности* как минимального размера текста, на котором еще возможно однозначное раскрытие исходного текста. Было показано, что расстояние единственности прямо пропорционально длине ключа и обратно пропорционально избыточности исходного текста. Следствием работы Шеннона стало доказательство наличия теоретически стойких шифров, как например шифр Вернама.

Наиболее мощным толчком развития криптографии явилась публикация в 1976 г. фундаментальной статьи У. Диффи и М. Е. Хеллмана «Новые направления в криптографии» [22]. В этой работе впервые было показано, что секретность передачи информации может обеспечиваться без обмена секретными ключами. Тем самым была открыта эпоха *двухключевых (асимметричных) криптосистем*, разновидностью которых являются системы электронной цифровой подписи. Данный вид криптосистем называется также шифрами с открытым ключом. Они используются в системах тайного электронного голосования, защиты от навязывания ложных сообщений, электронной жеребьевки, идентификации и аутентификации удаленных пользователей и ряде других систем.

В последние годы на базе совершенствования электронных технологий появились новые теоретические разработки в области *квантовой криптографии* [2], основанной на принципах неопределенности Гейзенберга.

Наряду с развитием криптографических систем совершенствовались и методы, позволяющие восстанавливать исходное сообщение, исходя из шифртекста и другой известной информации, получившие название *криптоанализа*. Успехи криптоанализа приводили к ужесточению требований к криптографическим алгоритмам. Принципиально важным вопросом криптографии всегда была надежность криптосистем. Эта проблема допускала различное трактование на протяжении всей истории криптографии.

Голландский криптограф Керкхофф (1835–1903) впервые сформулировал *правило оценки стойкости шифра*, в соответствии с которым:

1. Весь механизм преобразований считается известным злоумышленнику.
2. Надежность алгоритма должна определяться только неизвестным значением секретного ключа.

Второе требование можно интерпретировать как направленность на разработку таких алгоритмов, для которых оппонент не сможет разработать методы, позволяющие снять защиту или определить истинный ключ за время существенно меньшее, чем время *полного* (*тотального*) *перебора* всего множества возможных секретных ключей.

Принятие оценки стойкости шифра, по Керкхоффу, отражает осознание необходимости испытания криптосхем в условиях, более благоприятных для атаки, по сравнению с условиями, в которых, как правило, может действовать потенциальный нарушитель. Правило Керкхоффа стимулировало появление более качественных шифрующих алгоритмов. Можно сказать, что в нем содержится первый элемент стандартизации в области криптографии, поскольку предполагается разработка открытых способов преобразований. В настоящее время это правило интерпретируется более широко: *все долговременные элементы системы защиты должны предполагаться известными потенциальному злоумышленнику*. В последнюю формулировку криптосистемы входят как частный случай систем защиты. В расширенном понимании правила Керкхоффа предполагается, что все элементы криптосистем защиты подразделяются на две категории — долговременные и легко сменяемые элементы. К *долговременным* относятся те элементы, которые связаны со структурой криптосистем защиты и заменяются только специалистами. К *легко сменяемым* относятся элементы криптосистемы, которые предназначены для частого модифицирования в соответствии с заданным порядком. Легко сменяемыми элементам шифра являются, например, секретный ключ, пароль, идентификатор и т. п. Правило Керкхоффа отражает тот факт, что надлежащий уровень секретности зашифрованной информации должен быть обеспечен только за счет неизвестных легко сменяемых элементов шифра. Действительно, долговременные элементы системы защиты трудно сохранить в тайне, поэтому система должна быть стойкой в случае, когда они являются известными атакующему.

Согласно современным требованиям криптосистемы с секретным ключом, включая шифры с ключом ограниченного размера (128–256 бит), должны быть стойкими к криптоанализу на основе известного алгоритма, большого объема известного открытого текста и соответствующего ему шифртекста. Несмотря на эти общие требования, шифры, используемые специальными службами, сохраняются, как правило, в секрете. Это обусловлено необходимостью иметь дополнительный запас прочности защиты секретной информации, поскольку в настоящее время создание криптосистем с доказуемой стойкостью является предметом развивающейся теории и представляет собой достаточно сложную проблему. Чтобы избежать возможных слабостей, алгоритм шифрования может быть построен на основе хорошо изученных и проверенных на практике принципов и способов преобразования. Ни один серь-

езный современный пользователь не будет полагаться только на надежность сохранения в секрете своего алгоритма, поскольку крайне сложно гарантировать то, что информация об алгоритме не станет известной злоумышленнику до окончания срока его использования.

Обоснование надежности используемых систем осуществляется как теоретически, так и экспериментально при моделировании криптоаналитических атак с привлечением группы опытных специалистов, которым предоставляются значительно более благоприятные условия по сравнению с условиями, возникающими на практике в предполагаемых областях применения криптоалгоритма. Например, кроме шифртекста и алгоритма преобразований криптоаналитикам предоставляется исходный текст, шифртексты, полученные из данного открытого текста с помощью различных ключей, и т. п. Оценивается стойкость испытываемой криптосистемы ко всем известным методам криптоанализа и, по возможности, разрабатываются новые подходы к ее вскрытию. Если в этих условиях криптосистема оказывается стойкой, то она рекомендуется для того или иного конкретного применения.

В современном криптоанализе рассматриваются атаки на засекречивающие системы на основе следующих известных данных:

- шифртекста;
- открытого текста и соответствующего ему шифртекста;
- выбранного открытого текста;
- выбранного шифртекста;
- адаптированного открытого текста;
- адаптированного шифртекста.

Кроме того, рассматриваются следующие методы инженерного (технического) криптоанализа, дополнительно использующие:

- преднамеренно генерируемые аппаратные ошибки;
- замеры потребляемой мощности;
- замеры времени вычислений;
- некоторую информацию, перехватываемую по каналу побочных электромагнитных излучений.

Мы детально перечислили типы атак на криптосистемы, предназначенные для шифрования данных с целью защиты от несанкционированного чтения. По отношению к иным видам криптосистем существует ряд других атак, которые будут рассмотрены ниже. А сейчас рассмотрим перечисленные типы атак подробнее.

- В случае *криптоанализа на основе известного шифртекста* считается, что противник знает механизм шифрования и ему доступен только шифртекст. Это соответствует модели внешнего нарушителя, который имеет физический доступ к линии связи, но не имеет доступа к аппаратуре зашифрования и расшифрования.
- При *криптоанализе на основе известного открытого текста* предполагается, что криптоаналитику известен шифртекст и некоторая часть исходного текста, а в частных случаях и соответствие между шифртекстом и исходным текстом. Возможность проведения такой атаки складывается при зашифровании документов, подготовленных с использованием стандартных форм, в условиях, когда определенные блоки данных известны и повторяются. В ряде современных средств защиты компьютерной информации используется режим глобального шифрования, в котором вся информация на встроенном магнитном носителе записывается в виде шифртекста, включая главную корневую запись, загрузочный сектор, системные программы и пр. При хищении такого носителя (или компьютера) легко установить, какая часть криптограммы соответствует стандартной системной информации, и получить большой объем известного исходного текста для выполнения криптоанализа.
- В *нападениях на основе выбранного открытого текста* предполагается, что криптоаналитик противника может ввести специально подобранный им текст в шифрующее устройство и получить криптограмму, образованную под управлением секретного ключа. Это соответствует модели внутреннего нарушителя. На практике такая ситуация может возникнуть, когда в атаку на шифр вовлекаются лица, которые не знают секретного ключа, но в соответствии со своими служебными полномочиями имеют возможность использовать шифратор для закрытия передаваемых сообщений. Для осуществления такой атаки могут быть использованы также технические работники, готовящие формы документов, электронные бланки и др.
- *Криптоанализ на основе выбранного шифртекста* предполагает, что противник (оппонент) имеет возможность использовать для расшифрования сформированные им самим шифртексты, которые выбираются специальным образом, чтобы по полученным на выходе дешифратора текстам он мог вычислить секретный ключ шифрования с минимальной трудоемкостью.
- *Атака на основе адаптированных текстов* соответствует случаю, когда атакующий многократно подставляет тексты для зашифрования (или расшифрования), причем каждую новую порцию данных выби-

рает в зависимости от полученного ранее результата преобразования. Этот вид атаки является наиболее благоприятным для нападающего.

В настоящее время к наиболее мощным типам криптоаналитических атак на основе выбранных или адаптированных текстов относится *дифференциальный* (*разностный*) *криптоанализ* [12, 31] и *линейный криптоанализ* [31], а также производные от них методы.

При тестировании новых криптосистем особый интерес представляют нападения на основе известного *секретного* ключа или *расширенного (рабочего)* ключа шифрования. Имеется различие между секретным ключом и расширенным ключом, поскольку секретный ключ не всегда непосредственно используется в преобразованиях шифруемого текста, а часто служит только для выработки расширенного ключа, который и используется в процессе шифрования. Существуют шифры (например блочный шифр ГОСТ 28147–89), в которых секретный ключ непосредственно используется при шифровании данных, т. е. секретный ключ служит и рабочим ключом шифрования. Очевидно, что расширенный ключ является секретным элементом шифра. При проведении криптоанализа на основе известных элементов ключа (секретного или расширенного) предполагается, что криптоаналитик имеет информацию о некоторой части ключа. Чем больше известная доля ключа, при которой шифр оказывается стойким, тем меньше опасений будет вызывать шифр в реальных условиях атаки, когда ключ атакующему неизвестен, но осуществляется попытка восстановить его элементы. При сравнении двух шифров предпочтение можно отдать тому шифру, который имеет лучшие показатели по указанному критерию.

Одним из направлений построения скоростных программных шифров является использование зависимости алгоритма шифрования от секретного ключа. В таких криптосистемах конкретная модификация алгоритма шифрования сменяется одновременно с заменой секретного ключа и атакующему неизвестна. Они называются *недетерминированными* или *гибкими шифрами*. При тестировании гибких шифров представляется разумным проводить анализ их стойкости с использованием атаки на основе *выбранной модификации алгоритма* шифрования. В этом варианте криптоаналитику предоставляется возможность анализировать самую слабую, по его мнению, модификацию из числа потенциально реализуемых вариантов криптоалгоритма. Затем для выбранной модификации проводится криптоанализ на основе специально подобранных текстов, в том числе может рассматриваться и вариант *атаки с частично известным ключом* шифрования. Если не удается найти слабую модификацию криптоалгоритма, то анализируемый гибкий шифр можно признать криптографически стойким.

1.2.2. Современные приложения

Значение криптографии выходит далеко за рамки обеспечения секретности данных. По мере все большей автоматизации процессов передачи и обработки информации и интенсификации информационных потоков криптографические методы приобретают уникальное значение. Новые информационные технологии в своей основе имеют двухключевую криптографию, которая позволяет реализовать протоколы, предполагающие, что секретный ключ известен только одному пользователю, т. е. протоколы, ориентированные на взаимное недоверие взаимодействующих сторон. Отметим основные приложения современной криптографии.

- Защита от несанкционированного чтения (или обеспечение конфиденциальности информации).
- Защита от навязывания ложных сообщений (имитозащита).
- Аутентификация законных пользователей.
- Контроль целостности информации.
- Электронная цифровая подпись.
- Разработка систем тайного электронного голосования.
- Обеспечение хождения электронных денег.
- Проведение электронной жеребьевки.
- Создание электронного казино.
- Построение пороговых схем разделения секрета.
- Защита от отказа по факту приема сообщения.
- Одновременное подписание контракта.
- Защита документов и ценных бумаг от подделки.

Первое направление приложений было раскрыто выше. Кратко поясним некоторые из остальных областей применения криптографии.

Имитозащита — это понятие связано с защитой от навязывания ложных сообщений путем формирования в зависимости от секретного ключа специальной дополнительной информации, называемой *имитовставкой*, которая передается вместе с криптограммой. Дело тут в том, что самого по себе шифрования данных недостаточно для защиты от навязывания ложного сообщения, хотя во многих случаях законный получатель, анализируя семантику сообщения, может легко определить, что криптограмма была модифицирована или подменена, например, при передаче по линии связи. Однако при искажении цифровых данных и в некоторых других случаях обнаружить факт изме-

нения данных по семантике крайне сложно. Одним из способов защиты от навязывания ложного сообщения путем целенаправленного или случайного искажения шифртекста является имитозащита. Для вычисления имитовставки используется алгоритм, задающий зависимость имитовставки от каждого бита сообщения. При этом могут быть использованы следующие два варианта:

- вычисление имитовставки по открытому тексту;
- вычисление имитовставки по шифртексту.

Чем больше длина имитовставки, тем меньше вероятность того, что искажение шифртекста не будет обнаружено санкционированным (законным) получателем. Нарушитель может модифицировать шифртекст, но поскольку он не знает секретного ключа, то не может вычислить новое значение имитовставки, соответствующее модифицированному сообщению. Нарушитель либо не меняет имитовставку, либо подменяет ее на случайное значение. Если используется алгоритм вычисления имитовставки с хорошими криптографическими свойствами, то вероятность того, что факт изменений не будет обнаружен законным получателем, составляет $P = 2^{-n}$, где n — длина имитовставки в битах.

Аутентификация законных пользователей заключается в распознавании пользователей, после чего им предоставляются определенные права доступа к ресурсам вычислительных и автоматизированных информационных систем. Аутентификация основана на том, что законные пользователи обладают некоторой информацией, которая является неизвестной для посторонних. Частным вариантом использования процедуры аутентификации является парольная защита входа в компьютерную систему. Например, пользователь формирует некоторую случайную информацию и, сохраняя ее в секрете, использует как пароль. Пароль в явном виде не хранится в памяти ЭВМ или другого устройства, применяемого для выполнения аутентификации. Это требование направлено на то, чтобы потенциальный внутренний нарушитель не имел возможности считать чужой пароль и присвоить себе полномочия другого пользователя. Для того чтобы система защиты могла идентифицировать легальных (санкционированных) пользователей, в памяти ЭВМ хранятся образы их паролей, вычисленные по специальному криптографическому алгоритму, реализующему так называемую одностороннюю функцию $y = F(x)$. Основное требование к односторонней функции состоит в том, чтобы сложность вычисления значения функции по аргументу была низкой, а сложность определения аргумента по значению функции была высокой (например практически неосуществимой за 10 лет при использовании всех вычислительных ресурсов человечества).

Аутентификация пользователя на рабочей станции может быть осуществлена путем выполнения следующей последовательности шагов:

1. Запрос на ввод идентификатора со стороны системы защиты.
2. Ввод пользователем своего идентификатора (имени) NAME.
3. Запрос на ввод пароля со стороны системы защиты.
4. Ввод пользователем пароля P.
5. Вычисление системой защиты значения односторонней функции y, соответствующего значению аргумента $x = P$.
6. Сравнение системой защиты значения $F(P)$ со значением образа (S) пароля, соответствующего пользователю с идентификатором NAME.

Если $F(P) = S$, то система защиты предоставляет пользователю права доступа (полномочия), соответствующие идентификатору NAME. В противном случае в журнале учета работы пользователей регистрируется событие попытки несанкционированного доступа. Для того чтобы выдать себя за санкционированного пользователя, нарушитель должен ввести правильный пароль. Зная образ S, вычислительно невозможно определить пароль P. Если в системе защиты предусмотрены механизмы противодействия перехвату пароля с помощью программных или аппаратных закладок, а также через побочные электромагнитные излучения и наводки или акустический и оптический каналы, то данный способ аутентификации пользователей обеспечивает высокую надежность защиты от захвата чужих полномочий.

Рассмотренный пример относится к аутентификации пользователей на рабочих станциях, т. е. к задаче защиты входа в ЭВМ. Для взаимной аутентификации удаленных рабочих станций принципиальным является предположение, что потенциальный злоумышленник прослушивает канал связи, а следовательно, описанный выше способ аутентификации неприемлем, поскольку недопустима передача пароля по незащищенному каналу. Аутентификация удаленных рабочих станций может быть выполнена по следующей схеме, основанной на использовании алгоритма шифрования E и общего секретного ключа K для удаленных станций A и B.

1. Станция A посылает запрос на соединение со станцией B.
2. Станция B посылает станции A случайное число R.
3. Станция A зашифровывает R по секретному ключу K, получая шифртекст $C_a = E_K(R)$, и направляет станции B значение C_a.
4. Станция B вычисляет $C_b = E_K(R)$ и сравнивает значения C_b и C_a. Если $C_b = C_a$, то принимается решение, что запрос на установление сеанса связи отправлен станцией A, в противном случае связь прерывается.

Правильно зашифровать случайный текст может только тот, кто знает секретный ключ. Если нарушитель будет перехватывать правильные криптограммы от случайных чисел, то при длине числа R не менее 64 бит он за ра-

зумное время не встретит двух одинаковых чисел, а следовательно, не будет иметь возможности подставить ранее перехваченную правильную криптограмму. В этой схеме роль станции В может играть сервер локальной вычислительной сети. Отметим, что данная схема позволяет станции В удостовериться в том, что связь устанавливается со станцией А. Однако для станции А может существовать аналогичная проблема аутентификации станции В. В этом случае дополнительно осуществляется аналогичная процедура аутентификации, а именно — аутентификация станции В станцией А. Такая схема взаимного распознавания удаленных абонентов (станций) называется *протоколом рукопожатия*.

Контроль целостности информации — это обнаружение любых несанкционированных изменений информации, например, данных или программ, хранимых в компьютере. Имитозащита, в сущности, является важным частным случаем контроля целостности информации, передаваемой в виде шифртекста. В практических приложениях часто требуется удостовериться, что некоторые программы, исходные данные, базы данных не были изменены каким-либо несанкционированным способом, хотя сами данные не являются секретными и хранятся в открытом виде. Контроль целостности информации основан на выработке по некоторому криптографическому закону *кода обнаружения модификаций* (КОМ), имеющего значительно меньший объем, чем охраняемая от модифицирования информация. Основным требованием к алгоритму вычисления КОМ является задание зависимости значения КОМ от каждого бита двоичного представления всех символов исходного текста.

Проверка соответствия информации своему эталонному состоянию (контроль целостности информации) выполняется следующим образом. При фиксации эталонного состояния, например, программного модуля File.exe вычисляется значение КОМ, соответствующее этому файлу. Полученное значение КОМ записывается в таблицу, которая будет использоваться при каждой процедуре проверки целостности информации. Пусть программа File.exe управляет сложным и ответственным технологическим процессом, а ее сбои могут привести к простоям, материальным и финансовым потерям. В таком случае перед каждым запуском программы целесообразно удостовериться в ее целостности. С этой целью каждый раз вычисляется КОМ и сравнивается с соответствующим значением, хранимым в таблице кодов. Этот способ эффективен для обнаружения непреднамеренных искажений данных, которые имеют случайный характер.

При преднамеренном модифицировании информации такая схема контроля целостности данных неприемлема, поскольку злоумышленник может обойти ее следующим образом. Он изменит по своему усмотрению данные, вычислит новое значение КОМ, соответствующее измененным данным, и

внесет его в таблицу кодов взамен эталонного значения КОМ (соответствующего эталонному состоянию данных). Чтобы предотвратить такое нападение, можно дополнительно использовать один из следующих механизмов:

- секретный алгоритм вычисления КОМ;
- алгоритм вычисления КОМ с применением секретного ключа, от которого зависит значение КОМ;
- хранение таблицы кодов в защищенной области памяти или хранение таблицы кодов на переносных носителях, доступ к которым контролируется организационными мерами.

В первом случае сложно обеспечить секретность алгоритма, поскольку он является долговременным элементом криптосистемы. Третий случай требует существенных трудозатрат на выполнение дополнительных организационных мер. Наиболее приемлемым является использование второго механизма. Однако во всех трех случаях необходимо предусмотреть защиту от программных закладок.

Методы, используемые для контроля целостности, должны обеспечить чрезвычайно малую вероятность какого-либо умышленного или неумышленного изменения данных, при котором их кодовое представление осталось бы неизменным. В этой области задача криптоанализа заключается в том, чтобы на основе изучения слабых мест алгоритма генерации КОМ осуществить изменение исходной информации таким образом, чтобы значение контрольного кода не изменилось. Алгоритмы вычисления КОМ называются алгоритмами защитного *контрольного суммирования*, а вырабатываемое ими проверочное значение — *контрольной суммой*. Большую роль в современных криптографических протоколах и системах играют *хэш-функции*, которые представляют собой частный случай алгоритмов вычисления защитных контрольных сумм.

Аутентификация информации — это установление санкционированным получателем (приемником) того факта, что полученное сообщение послано санкционированным отправителем (передатчиком). Соблюдение заранее оговоренного протокола (набора правил и процедур) должно обеспечить максимальную вероятность этого факта. Очевидно, что при этом контролируется и целостность сообщения на возможность подмены или искажения. Принятый протокол должен обеспечить противодействие использованию потенциальным противником ранее переданных сообщений. В симметричных криптосистемах аутентификация осуществляется с применением одного или нескольких секретных ключей и защитных контрольных сумм. В асимметричных криптосистемах аутентификация осуществляется с использованием открытых ключей. Чтобы это было возможным, при распределении открытых ключей

осуществляется их аутентификация с использованием организационно-технических мероприятий.

Проблема аутентификации открытых ключей как одна из фундаментальных проблем криптографии предстала в явном виде с момента открытия *криптографии с открытым ключом* (двухключевой или асимметричной криптографии) в середине 70-х гг. прошлого века. Криптография с открытым ключом дала чрезвычайно удобное решение проблемы распределения секретных ключей, но ее использование требует осуществления процедуры аутентификации открытых ключей. Следует отметить, что проблема аутентификации ключей не является порождением двухключевой криптографии. Она в неявном виде всегда присутствовала в криптографии с секретным ключом. Действительно, при распределении секретных ключей по защищенному каналу одновременно осуществляется и их аутентификация. Например, при получении опечатанного пакета с секретным ключом получатель проверяет целостность пакета и печатей.

Если работа Шеннона «Теория связи в секретных системах» [38] заложила фундамент формирования криптологии как науки, то открытие двухключевой криптографии ознаменовало собой ее переход в качественно новую фазу развития. Это послужило основой для наиболее полного решения проблем аутентификации информации и разработки систем электронной цифровой подписи, которые призваны придать юридическую силу документам и другим сообщениям, переданным в электронном виде.

Электронная цифровая подпись (ЭЦП) основывается на двухключевых криптографических алгоритмах, в которых предусмотрено использование *двух* ключей — *открытого* и *секретного*. Идея использования открытого (т. е. известного всем пользователям криптосистемы и потенциальному злоумышленнику) ключа является фундаментальной, поэтому двухключевые криптосистемы еще называют открытыми шифрами, а выполняемые преобразования — открытым шифрованием. Двухключевые криптоалгоритмы позволяют обеспечить строгую доказательность факта составления того или иного сообщения конкретными абонентами (пользователями) криптосистемы. Доказательство основано на том, что двухключевые криптосистемы функционируют в условиях, когда пользователю нет необходимости сообщать свой секретный ключ какому-либо второму субъекту. Факт использования секретного ключа при выработке подписи к тому или иному электронному документу проверяется с использованием *открытого* ключа. При этом знание открытого ключа не дает возможности выработать правильную цифровую подпись. Таким образом, ответственность за сохранность *секретного* ключа и соблюдение правил его использования лежит на самом владельце этого ключа. Секретный ключ позволяет составить сообщение со специфической внут-

ренней структурой, связанной с подписываемым документом и открытым ключом. Тот факт, что сообщение имеет структуру, сформированную с помощью секретного ключа, проверяется с помощью открытого ключа. Эта процедура называется проверкой электронной цифровой подписи. Вероятность того, что некоторое сообщение, составленное нарушителем, может быть принято за сообщение, подписанное каким-либо абонентом системы ЭЦП, является чрезвычайно низкой, порядка 10^{-30}.

Таким образом, процедура проверки ЭЦП с помощью открытого ключа позволяет с высокой степенью гарантии удостовериться в том, что полученное сообщение было составлено владельцем секретного ключа. Общедоступный открытый ключ формируется на основе секретного ключа, или оба вырабатываются одновременно по специальным процедурам, причем определение секретного ключа по открытому ключу является вычислительно сложной математической задачей.

Под *вычислительно сложными* (трудно решаемыми) задачами понимаются задачи, заведомо имеющие решение, но требующие для его нахождения выполнения чрезвычайно большого числа операций вычислителя (ЭВМ или другого вычислительного устройства). Это число должно быть таким большим, что использование всех вычислительных устройств, которые могут быть вовлечены в единый вычислительный процесс, не позволит найти решение с существенной вероятностью (например вероятностью $> 10^{-9}$) за обозримое время (десятилетия, столетия, тысячелетия и т. д.). Среднее число операций, которые необходимо выполнить для того, чтобы найти решение с помощью лучшего алгоритма, принимается за количественную меру сложности трудно решаемой задачи. Проблема оценки сложности заключается в ее зависимости от алгоритма поиска решения. Для разных алгоритмов в общем случае получаются различные значения сложности. Для конкретной трудно решаемой задачи нелегко доказать, что найден алгоритм с минимальной трудоемкостью (т. е. лучший алгоритм). Применение двухключевых шифров основано на предположении о существовании трудно решаемых задач, т. е. таких задач, для которых не существует решения со сравнительно низкой трудоемкостью.

На базе двухключевых криптографических алгоритмов предложены системы *тайного электронного голосования*, которые используют механизм слепой подписи, т. е. возможность подписать сообщение без ознакомления с его содержанием. Различные способы тайного электронного голосования имеют большие перспективы для совершенствования политической системы управления современным обществом с развитой информационной инфраструктурой.

Протокол слепой подписи позволяет построить различного рода системы *электронных денег*. Отличие электронных денег от перечислений денежных средств с использованием ЭЦП состоит в том, что они обеспечивают тайну покупки, осуществленной между покупателем и продавцом. Общественный интерес представляют также системы электронной жеребьевки, вариантом которых является игра в покер по телефону. В более широком плане могут быть созданы электронные игорные дома, которые будут гарантировать игрокам защиту от обмана на гораздо более высоком уровне, чем это делается в обычных игорных домах.

1.3. Элементы симметричного шифрования

1.3.1. Условная и безусловная секретность

Фундаментальная работа К. Шеннона [38], посвященная теоретическому анализу секретных систем (шифров), положила начало формированию современной криптологии и явилась базой для разработки новых криптосистем. К. Шеннон рассматривает шифрование как отображение исходного сообщения в зашифрованное (криптограмму): $C = F_i(M)$, где C — криптограмма, F_i — отображение, M — исходное сообщение, индекс i соответствует конкретному используемому ключу. Для того чтобы была возможность однозначного расшифрования сообщения, отображение F_i должно иметь единственное обратное отображение, такое что $F_i F_i^{-1} = I$, где I — тождественное отображение: $M = F_i^{-1}(C)$.

Предполагается, что источник ключей является статистическим процессом или устройством, которое задает отображения $F_1, F_2, \ldots, F_{N_1}$ с вероятностями $p_1, p_2, \ldots, p_{N_1}$, а число возможных сообщений N_2 конечно и сообщения $M_1, M_2, \ldots, M_{N_2}$ имеют априорные вероятности $q_1, q_2, \ldots, q_{N_2}$ (q_i – это вероятность, которая может быть приписана тому событию, что перехваченная криптограмма содержит сообщение M_i, без проведения криптоанализа).

Рассмотрим простейший шифр, в котором исходный алфавит сообщения совпадает со множеством знаков ключа и множеством знаков криптограммы, а шифрование выполняется путем последовательной замены знаков исходного сообщения на знаки криптограммы, в зависимости от очередного значения знака ключа. В этом случае сообщение, ключ и криптограмма представляются в виде последовательности букв одного и того же алфавита: $M = (m_1, m_2, \ldots, m_n)$, $K = (k_1, k_2, \ldots, k_n)$, $C = (c_1, c_2, \ldots, c_n)$.

Текущий шаг шифрования описывается соотношением $c_i = f(m_i, k_i)$. В практических криптосистемах длина ключа обычно значительно меньше длины шифруемого сообщения, поэтому часто последовательность (называе-

мая ключевой гаммой) k_1, k_2, …, k_n вычисляется на основе некоторого первичного ключа меньшего размера или может быть даже периодической.

Задачей криптоаналитика является вычисление исходного сообщения по криптограмме при знании множества отображений $F_1, F_2, \ldots, F_{N_1}$. Существуют криптосистемы, для которых любой объем перехваченной информации недостаточен для того, чтобы найти шифрующие отображения, причем данная ситуация не зависит от вычислительной мощности, доступной криптоаналитику. Шифры такого типа называются *безусловно стойкими* (по К. Шеннону — совершенно секретными). В строгом смысле безусловно стойкими являются такие шифры, для которых криптоаналитик (даже если он обладает бесконечными вычислительными ресурсами) не может улучшить оценку исходного сообщения M на основе знания криптограммы C по сравнению с оценкой при неизвестной криптограмме. Это возможно только в случае, когда M и C являются статистически независимыми, т. е. когда выполняется условие $P(M = M_i / C = C_i) = P(M = M_i)$ для всех возможных сообщений M. Последнее соотношение означает, что вероятность того, что в анализируемой криптограмме содержится сообщение M, не зависит от вида криптограммы, т. е. от конкретной последовательности знаков в шифртексте.

Безусловно стойкие криптосистемы существуют, что легко показать. Пусть в рассмотренном выше простейшем шифре используется алфавит из L букв, а текущие знаки криптограммы выбираются по формуле $c_i = f(m_i, k_i) = (m_i + k_i) \bmod L$, где каждому знаку c_i, m_i и k_i поставлен в соответствие его порядковый номер в алфавите. Выберем в качестве ключевой гаммы последовательность, состоящую из n случайных знаков $k_1 k_2 \ldots k_n$, т. е. выберем случайный ключ, размер которого равен длине сообщения. Для генерации ключа воспользуемся некоторым физическим генератором случайных чисел, обеспечивающим на выходе равную вероятность появления любого элемента из множества чисел $\{1, 2, ..., L\}$. Вырабатываемое таким генератором число будем брать в качестве индекса выбираемого знака ключа. Выбранный нами источник обеспечивает равновероятность выбора любого ключа длины n. В этом случае вероятность выбора произвольного заданного ключа длины n составляет $P(K = K_i) = L^{-n}$.

Используемый нами способ зашифрования может преобразовать любое сообщение M_i в криптограмму C_i при использовании некоторого ключа K_i, значение которого зависит от M_i и C_i. Поскольку для всех i $P(K = K_i) = \text{const}$, то произвольное сообщение M_i может быть преобразовано с одинаковой вероятностью в любую криптограмму C_i и M_i, т. е. выполняется условие $P(M = M_i / C = C_i) = L^{-n}$.

Последнее означает, что данной криптограмме длины n с вероятностью L^{-n} может соответствовать любое исходное сообщение длины n. При шифро-

вании нового сообщения будем брать новый случайный ключ. Описанные процедуры шифрования обеспечивают безусловную стойкость. Криптосистемы, использующие равновероятный случайный ключ, имеющий длину, равную длине сообщения, называются шифрами с *лентой однократного использования* или шифрами с *бесконечной ключевой гаммой.* На практике такие криптосистемы получили ограниченное применение, поскольку требуют передачи ключа большого объема.

Можно строго доказать, что для достижения безусловной стойкости достаточно использовать равновероятный случайный ключ, имеющий длину, равную длине сообщения, независимо от вида процедур шифрования. Это означает, что для шифров этого типа сами процедуры криптографического преобразования играют второстепенную роль, а принципиальным является использование бесконечного случайного ключа.

Криптосистемы второго типа характеризуются тем, что (по мере того как объем доступной криптоаналитику криптограммы возрастает) при определенном значении $n = n_0$ существует единственное решение криптоаналитической задачи. Минимальный объем криптограммы, для которого существует единственное решение, называется *расстоянием единственности*. В случае ленты однократного использования $n_0 \to \infty$. При конечной длине секретного ключа значение n_0 конечно. Заранее известно, что по криптограмме, имеющей размер больше расстояния единственности, можно найти единственное решение криптоаналитической задачи. Однако для криптоаналитика, обладающего ограниченными вычислительными ресурсами, вероятность найти это решение за время, в течение которого информация представляет ценность, чрезвычайно мала (10^{-30} и менее).

Шифры такого типа называются *условно стойкими*. Их стойкость основана на высокой вычислительной сложности решения криптоаналитической задачи.

Цель разработчика условно стойких криптосистем состоит в том, чтобы уменьшить затраты на процедуры шифрования и расшифрования и одновременно задать такой уровень сложности успешного решения криптоаналитической задачи, при котором для ее осуществления требуются настолько большие затраты, что делает нахождение решения экономически невыгодным. Задачи такого объема вычислений называются трудными или вычислительно сложными, а об их решении говорится, что оно является *вычислительно нереализуемым* (или вычислительно неосуществимым). Шифры, основанные на задачах, нахождение решения которых является вычислительно нереализуемым, называются также *вычислительно стойкими*. Наибольшее практическое применение получили именно вычислительно стойкие криптосистемы.

Под стойкостью криптосистем такого типа понимается сложность решения задачи криптоанализа в определенных условиях. К. Шеннон ввел понятие *рабочей характеристики* $W(n)$ шифра как среднего количества работы по нахождению ключа по известным n знакам криптограммы при использовании лучшего алгоритма криптоанализа. Количество работы может быть измерено, например, количеством операций, которые необходимо выполнить для вычисления ключа. Этот параметр непосредственно связан с алгоритмом вычисления ключа. Сложность определения $W(n)$ связана со сложностью нахождения лучшего способа раскрытия. Особый интерес представляет предельное значение $W(n)$ при $n \to \infty$.

В настоящее время неизвестны вычислительно стойкие криптосистемы, для которых строго получена нижняя граница $W(\infty)$. Ввиду сложности таких оценок, практические шифры характеризуют достигнутой оценкой рабочей характеристики $W'(\infty)$, которую получают для наилучшего из известных методов вычисления ключа.

К. Шеннон предложил модель для оценки расстояния единственности, из которой получено соотношение $n_0 = H(K)/D$, где $H(K)$ — *энтропия ключа* (для случайного ключа это длина ключа в битах), D — *избыточность языка*, измеренная в битах на 1 знак. Данное соотношение можно переписать в виде $H(K) \le nD$, где $H(K)$ характеризует число неизвестных в двоичном представлении ключа, а nD — число уравнений, имеющихся для нахождения ключа. Если число уравнений меньше числа неизвестных, то решение системы уравнений неоднозначно и криптосистема в этих условиях является безусловно стойкой. Если число уравнений больше числа неизвестных, то существует единственное решение и криптосистема не является безусловно стойкой. Однако она может оставаться вычислительно стойкой при $n >> n_0$. Уровень стойкости вычислительно стойких криптосистем в определяющей степени зависит от конкретного типа шифрующих процедур (здесь мы исключаем случай выбора секретного ключа очень малого размера, при котором сложность перебора возможных ключей является низкой). Конкретные процедуры преобразования также определяют ход рабочей характеристики, т. е. конкретный вид зависимости $W(n)$.

В следующих разделах будут рассмотрены двухключевые шифры, которые определяют содержание современных направлений развития криптографии. Они по своей природе являются вычислительно стойкими, но не безусловно стойкими криптосистемами. Предположение о существовании сложных вычислительных проблем является фундаментальным для современной криптографии.

1.3.2. Общие вопросы разработки шифров

В предыдущем разделе было показано, что шифр, обладающий безусловной стойкостью, может быть построен только при использовании случайного равновероятного ключа, длина которого равна длине исходного текста, причем для каждого нового сообщения используется новый ключ. Поскольку ключ используется однократно и выбирается случайно, то можно говорить о бесконечном случайном ключе. Заметим, что при использовании бесконечного ключа нет необходимости использовать какие-либо сложные процедуры преобразования знаков исходного текста в знаки шифртекста, поскольку достаточно применить простейшую операцию наложения знаков ключа на соответствующие знаки открытого текста (например операцию побитового сложения по модулю два).

Под практической секретностью часто понимается трудоемкость решения задачи криптоанализа шифров с конечным ключом. Это относится к теоретической модели криптосистемы, которая абстрагируется от конкретных условий использования шифров, т. е. связано с теоретической оценкой вычислительной сложности решения задачи криптоанализа. Представляется предпочтительным характеризовать шифры с ключом конечного размера как вычислительно стойкие криптосистемы, поскольку под практической секретностью можно понимать также секретность, которая зависит не только от теоретического уровня стойкости, но и от организационно-технических условий использования шифров. Например, при использовании шифров с бесконечным ключом практическая секретность определяется различными каналами утечки, связанными с обработкой исходного сообщения в криптосистеме. Через эти отводные каналы может произойти перехват части сообщения или ключа. В сущности даже при применении таких шифров имеется определенная вероятность того, что передаваемая информация станет известной нарушителю.

Под практической стойкостью в широком смысле можно понимать стойкость шифра с учетом большого числа факторов, имеющих место в условиях реального применения криптосистемы и связанных с обеспечением целостности всех составных частей реальной криптосистемы: криптографического протокола, шифратора, защищенного канала, по которому передается секретный ключ, процедур управления ключами и элементов окружения (защищенных помещений, обслуживающего персонала, физических и технических средств защиты и т. п.). В теоретической модели криптосистемы рассматривается только стойкость, связанная с решением задачи криптоанализа, которая определяет максимально достижимый уровень секретности. (В конкретных условиях практической эксплуатации криптосистем их секретность может быть значительно ниже этого предельного значения.)

Поскольку реальная практическая секретность даже при применении шифров с бесконечной ключевой гаммой не может гарантировать абсолютную защиту информации, то вполне оправданно использовать удобные в практической эксплуатации криптосистемы с конечным ключом, основанные на высокой вычислительной сложности криптоаналитической задачи. Сам факт возможности раскрытия вычислительно стойких шифров ничего не говорит о трудоемкости такого раскрытия. Для осуществления раскрытия шифра на практике трудоемкость задачи криптоанализа является важнейшей характеристикой, поскольку ресурсы нападающего реально являются ограниченными.

При использовании шифров с конечным размером ключа конкретный выбор алгоритма шифрования является решающим для обеспечения практической стойкости. Именно процедуры преобразования определяют сложность криптоанализа. При этом необходимо иметь в виду, что размер ключа должен быть выбран достаточно большим, чтобы предотвратить тотальный перебор по пространству ключей (т. е. сделать вычислительно невозможным опробование всех возможных ключей при использовании современных вычислительных систем). При длине ключа 128 бит и более это условие выполняется.

Если вычислительно стойкий шифр не позволяет раскрыть ключ с вероятностью, больше вероятности утечки информации через каналы, связанные с практическими условиями функционирования криптосистемы, то его применение является предпочтительным. Кроме того, существует ряд задач, в которых шифры с бесконечным ключом неприменимы. (Например, в средствах защиты информации, циркулирующей в компьютерных системах, где используется шифрование всех данных, хранимых на встроенном магнитном носителе.) В таких случаях предпочтительными для использования являются шифры с конечным ключом (при условии, что такие криптосистемы обладают достаточно высокой вычислительной стойкостью).

При разработке вычислительно стойких шифров используют два общих принципа: *рассеивание и перемешивание*.

Перемешиванием называется распространение влияния одного знака открытого текста на много символов шифртекста, что обусловливает лавинный эффект (в случае блочных шифров требуется обеспечить распространение влияния каждого бита входного текста на все биты выходного текста). *Рассеиванием* называется шифрующее преобразование, нарушающее взаимосвязи статистических характеристик входного и выходного текста, т. е. обеспечивающее маскировку статистических свойств исходного сообщения. Примером процедур, обеспечивающих рассеивание, является такая перестановка знаков открытого текста, которая приводит к выравниванию распределения избыточности по всему тексту. (Нужно отметить, что избыточность исходно-

го текста играет большую роль при осуществлении криптоанализа на основе шифртекста, однако при криптоанализе на основе известного или специально подобранного текста ее рассмотрение теряет смысл.)

Для предотвращения возможности вычисления ключа по частям реализуют принцип распространения влияния одного знака ключа на большое число знаков криптограммы. Криптосистемы, в которых выполняется несколько последовательных простых процедур шифрования, последний принцип воплощают автоматически при реализации принципа перемешивания.

1.3.3. Композиционные и итеративные блочные шифры

В современных автоматизированных системах обработки информации во многих случаях предпочтительно применение блочных шифров. *Блочные шифры* представляют собой криптосистему, которая осуществляет шифрование информации блоками фиксированной длины, например, равной n бит. Этот вид криптографических преобразований называется блочным шифрованием. Для осуществления блочного шифрования данные представляются в виде последовательности n-битовых блоков. В реальных ситуациях файлы, определенные поля в электронных документах и другие виды электронных сообщений имеют произвольную длину и, как правило, не кратную длине блока, поэтому используется некоторый способ дополнения последнего блока данных.

Последний блок принято дополнять двоичным вектором (1, 0, 0, ... , 0), в котором количество нулей может быть от 0 до $(n-2)$. Если длина последнего блока равна n, то к сообщению присоединяется дополнительный n-битовый блок, имеющий структуру (1, 0, 0, ..., 0). Этот способ позволяет однозначно распознать присоединенный двоичный вектор и отбросить его при необходимости. Используя такой способ дополнения сообщения до длины кратной n, любое сообщение M можно представить в виде последовательности (конкатенации) n-битовых подблоков M_i, а именно $M = M_1|M_2| \ldots |M_i| \ldots |M_m$.

Каждый из блоков исходного сообщения может быть преобразован независимо от других блоков, поэтому при применении блочных шифров возможен произвольный доступ к зашифрованным массивам данных. Наиболее общим механизмом блочного шифрования является такой, в котором допустимо преобразование любого входного блока в любой выходной блок, причем размер выходного блока равен или больше размера входного блока. Блок шифртекста не может быть меньше блока открытого текста, т. к. в этом случае одному и тому же блоку шифртекста будут соответствовать несколько различных блоков открытого текста. Это означало бы неоднозначность расшифрования. Если длина выходного блока больше n, то одному и тому же

блоку открытого текста могут соответствовать несколько различных блоков шифртекста. В этом случае расшифрование возможно и однозначно. (Примером таких криптосистем являются вероятностные шифры.) Поскольку увеличение длины зашифрованных данных вносит определенные ограничения на области применения, то в наиболее широко применяемых шифрах размер выходных блоков равен размеру входных. Блочные шифры задают взаимно однозначное соответствие между возможными входными и выходными блоками. Поскольку входное и выходное множество блоков совпадает, то шифрование задает некоторую подстановку на множестве чисел 0, 1, … , $2^n - 1$, которую можно представить в виде следующей таблицы:

$$\begin{pmatrix} 0 & 1 & 2 \ldots & 2^n - 1 \\ \mathrm{E}_k(0)\, \mathrm{E}_k(1)\, \mathrm{E}_k(2) \ldots \mathrm{E}_k\left(2^n - 1\right) \end{pmatrix},$$

где $\mathbf{E}_K(M)$ — функция шифрования по ключу K, т. е. функция, задаваемая процедурами шифрования при использовании ключа шифрования K.

Функция шифрования ставит в соответствие блоку открытого текста M блок криптограммы C, что записывается в виде $C = \mathbf{E}_K(M)$. Для данного ключа реализуется одна подстановка. В общем случае различным ключам соответствуют различные подстановки. Если в шифре используется ключ длиной k бит, то этот шифр задает не более 2^k различных подстановок, что составляет обычно чрезвычайно малую долю от числа возможных подстановок, которое равно $2^n!$. Для того чтобы реализовать все возможные подстановки, необходимо использовать ключ длиной порядка

$$k = \log_2(2^n!) \approx n2^n \text{ бит}.$$

Одним из статистических методов раскрытия шифров является *частотный криптоанализ*. Этот метод основан на исследовании частоты появления знаков в криптограмме и сопоставлении ее с частотой появления знаков в исходном тексте. Частотный метод позволяет раскрывать шифры, основанные на одноалфавитной подстановке, которые соответствуют блочному шифрованию при использовании входных блоков малого размера (например при $n = 8$). С увеличением размера входного блока частотные свойства языка, на котором составлен открытый текст, становятся менее выраженными, но даже при $n = 16$ неравномерность частотных свойств исходного текста может быть эффективно использована для раскрытия шифра. При $n = 32$ частотный криптоанализ становится крайне сложным, и в некоторых случаях могут применяться блочные шифры с таким размером входа. Минимальной безопасной длиной блока принято считать значение $n = 64$. Чем больше размер входного блока, тем более высокая стойкость может быть достигнута. Однако для блоков большого размера усложняется производство устройств шифрования.

При разработке американского стандарта DES определенным компромиссом между стойкостью и удобством реализации был выбор значения $n = 64$. На протяжении 25 лет этот размер был общепринятым. В настоящее время возможности микроэлектроники значительно выросли и стандартным размером входного блока стало значение $n = 128$.

Сам по себе большой размер входного блока является только необходимым условием достижения высокой стойкости разрабатываемого алгоритма. Проектирование стойких блочных шифров связано с использованием нелинейных преобразований с хорошими свойствами рассеивания и перемешивания или комбинированием линейных и нелинейных преобразований. Достоинством линейных преобразований является простота реализации, малое время их выполнения и удобство использования секретного ключа в качестве параметра преобразования. Однако использования только линейных преобразований недостаточно для построения стойких шифров.

Один из способов достижения хорошего рассеивания и перемешивания состоит в построении составного (*композиционного*) шифра, включающего ряд последовательно используемых простых шифров, каждый из которых вносит небольшой вклад в рассеивание или перемешивание. Данная идея построения композиционных шифров была предложена и обоснована К. Шенноном. В композиционных шифрах шифрующие процедуры одного типа чередуются с процедурами другого типа. В качестве простых шифров могут быть использованы, например, подстановки (S), перестановки (T) и линейные преобразования (L). В этом случае результирующий шифр представляется в виде $F = T_n L_m S_n \ldots T_2 L_2 S_2 T_1 L_1 S_1$.

Секретный ключ может использоваться при выполнении процедур какого-то одного типа (T, L или S). Ключ может использоваться также при осуществлении процедур всех (или нескольких) типов.

Простейший композиционный шифр показан на рис. 1.1, где каскад блоков подстановок S представлен блоками подстановок сравнительно малого размера (например в виде совокупности операций подстановки над 4-битовыми или 8-битовыми подблоками входного сообщения), T — операция перестановки битов и L — операция линейного преобразования, с помощью которой осуществляется смешивание шифруемых данных с секретным ключом, представленным в виде совокупности подключей $K_1, K_2, \ldots, K_R$. В данной итеративной криптосхеме процедура шифрования состоит в последовательном выполнении R раундов преобразования с использованием различных раундовых ключей. Выполнение операции подстановки состоит в замене, например, входного 4-битового двоичного вектора на 4-битовый выходной вектор в соответствии с некоторой таблицей, называемой таблицей подстановки.

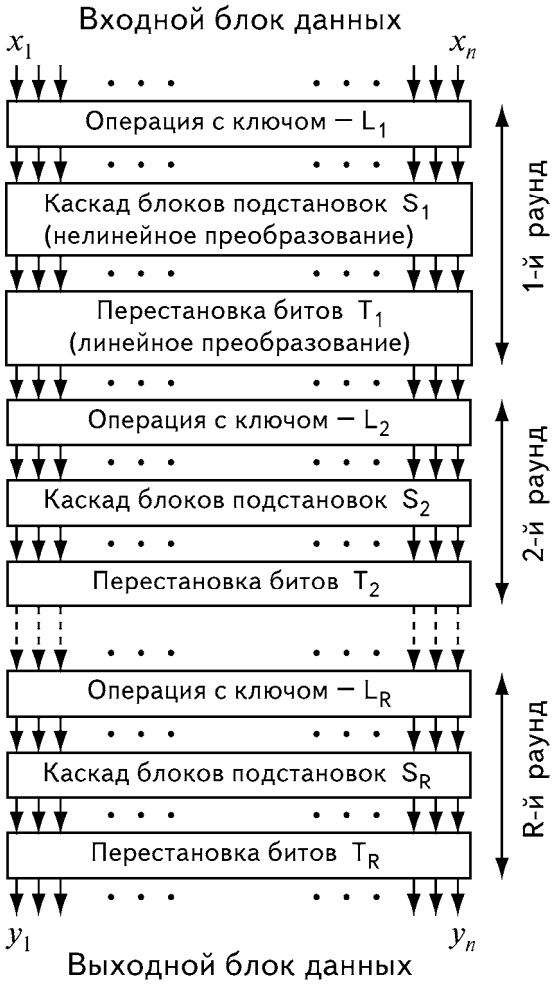


Рис. 1.1. Преобразования в композиционном шифре

Представляя 4-битовые двоичные векторы их численными значениями (т. е. интерпретируя двоичный вектор двоичным числом), таблицу подстановки можно записать в виде:

$$\begin{pmatrix} 0 & 1 & 2 & \dots & 15 \\ \alpha_0 & \alpha_1 & \alpha_2 & & \alpha_{15} \end{pmatrix},$$

где $\forall i \ \alpha_i \in \{0, 1, \dots, 15\}$, а столбцы задают соответствие между входным 4-битовым значением (верхняя строка) и выходным 4-битовым значением (нижняя строка). Аналогично описывается подстановка произвольного размера.

Для некоторой заданной таблицы подстановки легко записать таблицу, определяющую подстановку, которая является обратной по отношению к заданной. Аналогичным образом может быть задана и операция перестановки битов T и соответствующая ей обратная перестановка T^{-1}. Для линейного преобразования L также легко задать соответствующее ему обратное преобразование L^{-1}. Процедура расшифрования выполняется по схеме, приведенной на рис. 1.2.

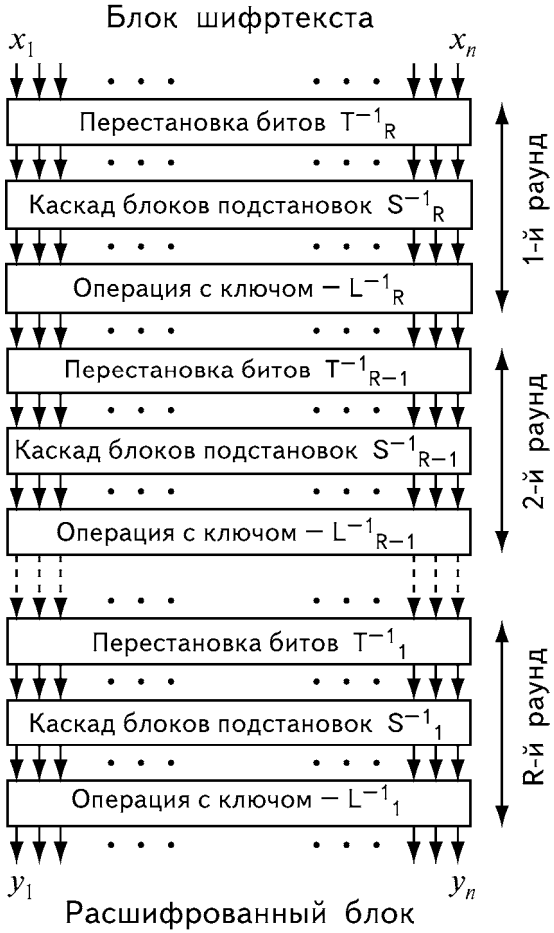


Рис. 1.2. Схема процедуры расшифрования в композиционном шифре

Рассмотренный выше шифр относится к так называемым *итеративным шифрам*, в которых шифрование осуществляется как многократное выполнение типовой процедуры преобразования (называемой *раундом шифрования* или *раундовой функцией* шифрования), представляющей собой композицию

трех разнотипных простых преобразований. При этом в разных раундах используются разные ключи, называемые *раундовыми* ключами.

При реализации в виде быстродействующих устройств процедуры зашифрования или расшифрования в композиционном шифре будут выполняться с помощью различных электронных схем. В дальнейшем мы рассмотрим специальные криптосхемы, которые обеспечивают возможность использования одной и той же электронной схемы для выполнения зашифрования и расшифрования, что делает аппаратную реализацию более экономичной. В таких шифрах смена режима шифрования осуществляется путем изменения очередности использования раундовых ключей.

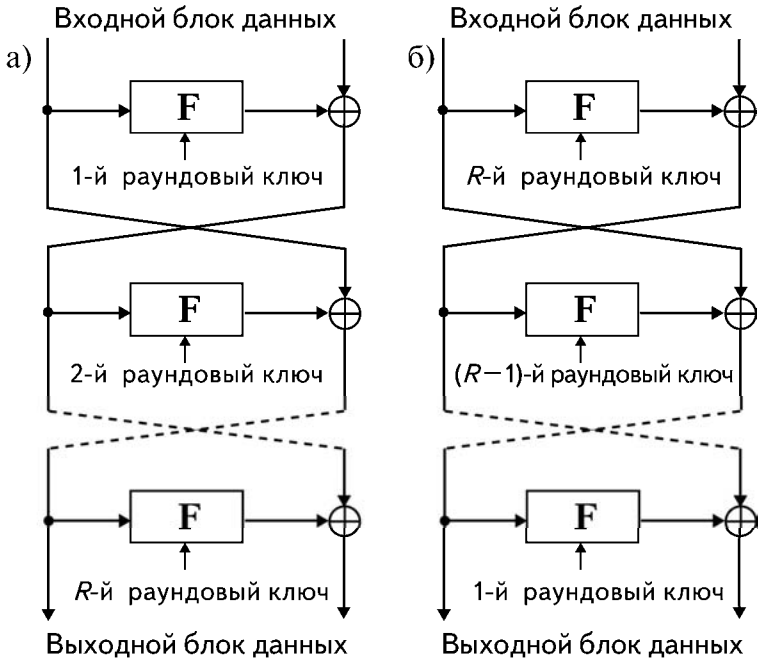


Рис. 1.3. Криптосхема Фейстеля: а — зашифрование; б — расшифрование

Криптосхема Фейстеля (рис. 1.3) представляет собой общую схему построения n-битового блочного шифра на основе произвольной функции F с размером входа $n/2$. Важным достоинством данной структуры является то, что она задает один и тот же алгоритм для выполнения зашифрования и расшифрования. Задание конкретного режима шифрования определяется очередностью использования раундовых ключей. Смена режима шифрования обеспечивается изменением очередности использования раундовых ключей на обратную. Стойкость шифров, построенных по этой схеме, определяется свойствами раундовой функции F. Известно большое число различных шиф-

ров, основанных на этой схеме и отличающихся только числом раундов и устройством раундовой функции.

Замечательным является то обстоятельство, что для произвольной раундовой функции обеспечивается возможность расшифрования, т. е. криптосхема Фейстеля есть некоторый универсальный способ превращения произвольного отображения над двоичными векторами длины $n/2$ в подстановку (*биективное* отображение) над векторами длины n. Благодаря этому свойству разработчик криптоалгоритма свободен от необходимости выбора только таких преобразований, которые допускают обратные преобразования невысокой вычислительной сложности. При построении раундовой функции достаточно найти быстрые прямые преобразования, удовлетворяющие необходимым криптографическим свойствам. Эти же преобразования при смене расписания использования подключей будут выполнять корректную процедуру расшифровывания. Основной задачей при разработке шифров на основе криптосхемы Фейстеля является построение раундовой функции, обладающей хорошими лавинными свойствами (влияние одного входного бита на многие выходные биты) и обеспечивающей стойкость ко всем известным видам криптоатак.

Определенным недостатком криптосхемы Фейстеля является то, что при выполнении шифрования (т. е. вычислении раундовой функции) используется только половина преобразуемого блока данных. Этот недостаток устраняется в универсальных криптосхемах другого типа, основанных на использовании так называемых *управляемых* операций.

1.3.4. 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 [27], в которой рассмотрено использование управляемой подстановочно-перестановочной сети. Другие попытки [33] 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 вариан-

тов криптосхем на основе управляемых операций, зависящих от секретного ключа, показали, 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 [30, 37], RC5 [35], RC6 [36].

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, выбираемое в зависимости от преобразуемых данных. Хотя 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 [9, 12]. Эта идея наиболее детально проработана по отношению к управляемым битовым перестановкам, выполняемым над подблоками данных размером 32 и 64 бит [9, 42, 43]. 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 объяснить 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 [9] 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 нейтрализует линейный криптоанализ [9, 28]. 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 [5], представляющие собой частный случай управляемых двухместных операций [9]. 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 книге [12] дано обобщение управляемых перестановок и построен класс управляемых операционных подстановок, реализуемых на основе управляемых подстановочно-перестановочных сетей с использованием управляемых элементов минимального размера. 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 операций, зависящих от преобразуемых данных и реализуемых с помощью перестановочных и подстановочно-перестановочных сетей, показана в работах [41–43]. При использовании сравнительно малых аппаратных ресурсов обеспечивается высокая производительность при реализации как в заказных, так и в программируемых СБИС [12, 42].

1.4. Особенности проектирования блочных шифров на основе управляемых операций

1.4.1. 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 дво-

ичных 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 однозначно определить 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 управляемой, 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 V 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 информа-

ционный $2n$-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 осуществляется простым разветвлением проводников и практически не требует затрат схемотехнических ресурсов. Блок такого разветвления будем называть блоком расширения E.

1.4.2. 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. Последнего можно избежать, если ПУСК выполнять параллельно с осуществлением шифрующих процедур.

Это может быть сделано следующим образом. На первом раунде шифрования непосредственно используется 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-м 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. Недостаток такого подхода к формированию раундовых ключей состоит в том, 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-м раунде расшифрования и $(R - i + 1)$-м 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 некоторых сетевых приложениях.

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 взаимно обратные 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.

1.4.3. 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

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

Криптосхема, представленная на рис. 1.4, *а*, 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-й 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:

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

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

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 $(R-j+1)$-й раундовый ключ процедуры расшифрования, шифрование является корректным, т. е. процедура расшифрования будет обратной процедуре зашифрования.

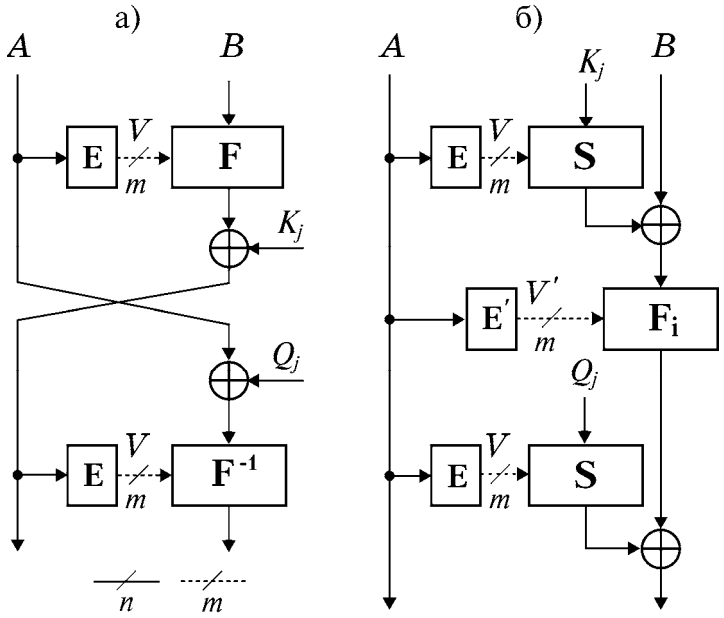


Рис. 1.4

Один раунд шифрования, представленный на рис. 1.4, *б*, 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 преобразова-

ния. 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. 1.4, *а* и 1.4, *б*, представлена на рис. 1.5, *а*.

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 чему становится возможным осуществить смену режима шифрования путем простого изменения расписания ключа.

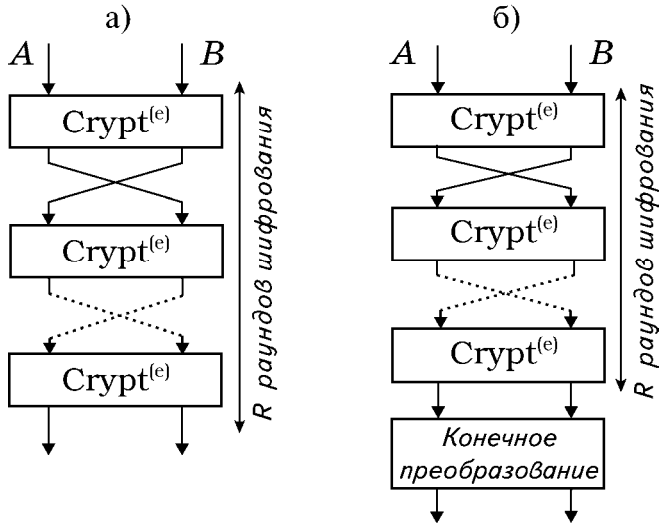


Рис. 1.5

На рис. 1.6, *а* и 1.6, *б* показаны структуры раундового преобразования, которые требуют применения симметрирующего конечного преобразования.

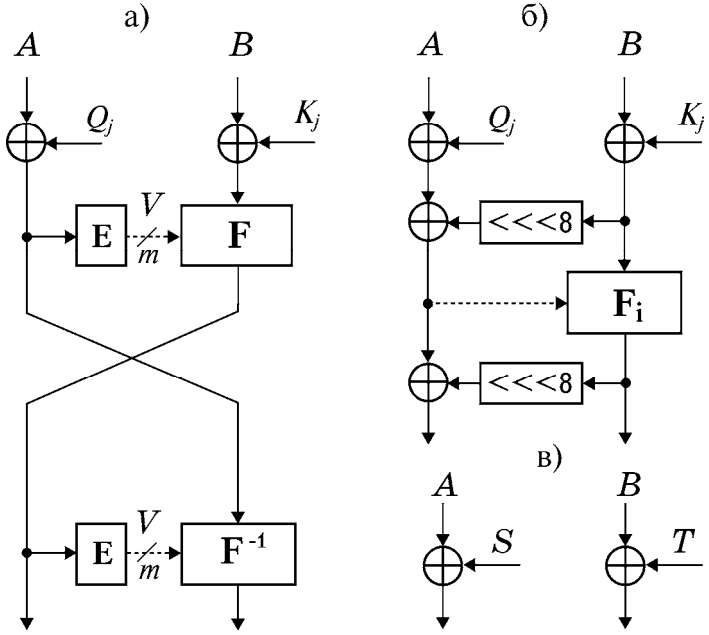


Рис. 1.6

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. 1.6, *в*) 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. 1.6, *б*, 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.

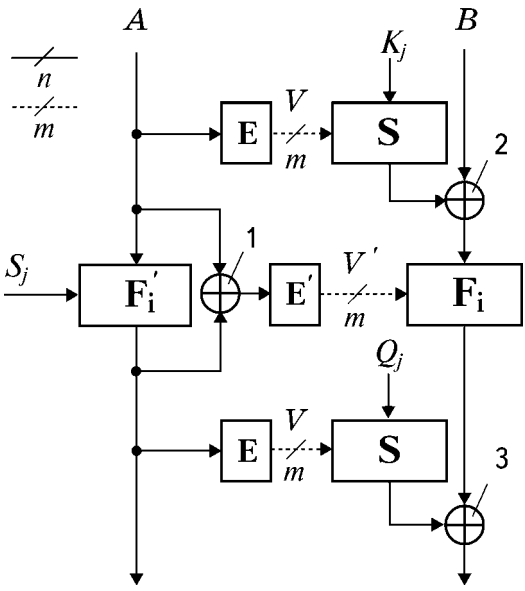


Рис. 1.7

Представляет интерес криптосхема, показанная на рис. 1.7, где левый и правый подблоки данных преобразуются путем выполнения над ними управляемых инволюций $\mathbf{F_i'}$ и $\mathbf{F_i}$, соответственно. Если временные задержки, вносимые управляемыми операциями **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 монографиях [9, 12].

1.4.4. 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, 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 [8, 9]. 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 значительные 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 [8, 11]. Смысл использования сложных процедур настройки состоит в том, чтобы заставить нападающего отка-

заться от их рассмотрения и принять предположение о случайности ключа шифрования.

1.5. Вероятностные шифры

Одним из перспективных способов повышения стойкости известных шифров является задание неопределенности хода шифрования информации. Эта идея может быть реализована путем введения в преобразуемое сообщение случайных данных. Если в механизме шифрования используются операции или процедуры, зависящие от преобразуемых данных (как это имеет место в шифре RC5), тогда сами операции будут изменяться случайно. Идея введения элементов случайности в процедуру шифрования преследует цель затруднить использование общего принципа криптоанализа блочных шифров, основанного на попытках выявления статистических свойств алгоритма шифрования, например, путем подбора специальных исходных текстов или криптограмм.

1.5.1. Гомофонические шифры

Криптограммы, полученные методом одноалфавитной или многоалфавитной подстановки, легко раскрываются частотным криптоанализом. Чтобы скрыть частотные свойства источника сообщений и тем самым затруднить криптоанализ, может быть применен метод *гомофонического* (монофонического) шифрования, который заключается в выравнивании частотности знаков криптограммы, т. е. использовании такого механизма криптографических преобразований, который бы вырабатывал криптограммы с одинаковой частотой появления каждого знака, используемого для записи шифртекста. Наиболее простым механизмом гомофонического шифрования является следующий способ. Пусть дан источник сообщений с известными статистическими свойствами. Выразим частоты появления каждой буквы исходного алфавита целым числом f_i, где i — номер буквы в алфавите: $f_1, f_2, ..., f_L$, где L — число букв в исходном алфавите. Каждой букве T_i, где $i = 1, 2, ..., L$, исходного алфавита поставим в соответствие подмножество Ψ_i знаков выходного алфавита (т. е. алфавита, с помощью которого записывается криптограмма). Причем зададим эти множества в соответствии с двумя условиями:

- никакая пара подмножеств не содержит одинаковых элементов;
- количество различных знаков в подмножестве Ψ_i равно (или пропорционально) f_i.

Шифрование будем осуществлять путем замены каждой буквы исходного текста T_i на случайно выбираемый знак из подмножества Ψ_i. В этом случае при многократной замене заданной буквы T_i исходного текста знаками подмножества Ψ_i каждый знак выходного алфавита будет использоваться в среднем одинаковое число раз. Это число обратно пропорционально числу элементов в подмножестве Ψ_i, т. е. пропорционально $1/f_i$. Частота обращения к подмножеству Ψ_i равна частоте появления буквы T_i в исходном тексте, т. е. пропорциональна f_i. Из этих соотношений приходим к выводу, что в среднем частоты появления в криптограмме всех знаков выходного алфавита равны. Расшифрование не представляет трудностей — по знаку криптограммы определяем подмножество, к которому он относится, а по конкретному подмножеству — соответствующую ему букву исходного алфавита. Описанный способ шифрования требует использования $f_1 + f_2 + ... + f_L$ знаков в выходном алфавите. Наиболее существенным элементом в этом способе шифрования является то, что в преобразования вводится вероятностный процесс — случайный выбор элемента из заданного подмножества.

Рассмотренный способ в настоящее время не представляет большого интереса для практических приложений, однако основная идея введения случайности в процесс шифрования может быть использована для построения современных вероятностных блочных шифров.

1.5.2. Шифры с простым вероятностным механизмом

В предыдущих разделах был рассмотрен ряд шифров, в которых используются операции преобразования, зависящие от преобразуемых данных. Такие операции не являются заранее предопределенными и изменяются от одного входного блока к другому. Если такие шифрующие механизмы применяются для преобразования случайных данных, то операции будут изменяться случайным образом. «Подмешивание» случайных данных к шифруемому сообщению позволяет задать вероятностный характер операций преобразования и тем самым повысить вычислительную стойкость криптосистемы [9]. Пусть **E** есть b-битовая функция шифрования, P — p-битовый блок открытого текста и R — r-битовый случайный блок, где $b = r + p$. Подадим на вход шифрующей функции блок $B = R \,||\, P$, где знак «||» обозначает конкатенацию двоичных векторов R и P: $P \to B = R||P \to C = \mathbf{E}(B, K)$, где K — ключ шифрования. Поскольку при шифровании размер входного блока увеличивается, то такое шифрование отображает данный блок открытого текста P на большое множество блоков шифртекста $\{C_1, C_2, ... , C_n\}$, где $n = 2^r$. Общая схема вероятностного шифра с простым механизмом присоединения случайных данных показана на рис. 1.8. Датчик случайных чисел (ДСЧ) является внутренней

частью шифратора так же, как и алгоритм шифрования, реализующий функцию шифрования **E**. Предполагается, что он расположен в защищенной части шифрующей аппаратуры и не может быть подменен злоумышленником (т. е. злоумышленник не имеет доступа к значению R). Это предположение является приемлемым, поскольку шифраторы проектируются таким образом, чтобы обеспечить защиту от подмены алгоритма шифрования так же, как и от считывания и копирования ключа. При дешифровании блока шифртекста законный пользователь, владеющий секретным ключом, восстанавливает блок $B = R||P$, после чего значение R отбрасывается и выделяется исходное сообщение P.

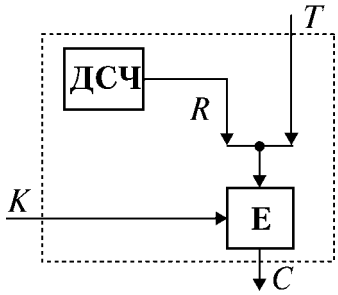


Рис. 1.8. Базовая схема вероятностного шифрования

Выбирая различные значения отношения b/p, можно регулировать степень усиления шифрования. Чем больше указанное отношение, тем больше усиление. Отличие вероятностного шифрования от криптосхемы с частой сменой сеансовых ключей состоит в том, что она не приводит к существенному снижению скорости шифрования в случае использования шифров с этапом предвычислений, на котором формируется ключ шифрования под управлением сеансового ключа. Схема вероятностного шифрования позволяет регулировать степень уменьшения скорости преобразования. Если функция **E** имеет исходное значение скорости преобразования s_0, то при использовании вероятностного шифрования она составляет $s = s_0(b - r)/b$.

Известен ряд успешных атак на криптосистемы DES, RC5, Blowfish при использовании пониженного числа раундов шифрования. Очевидно, что можно выбрать такую длину случайного блока R, чтобы сделать редуцированные версии этих шифров стойкими к известным атакам. Для этого можно выбрать значение $r = b - 1$. Шифры с простым вероятностным механизмом предоставляют следующие преимущества:

- стойкость известных блочных шифров может быть существенно повышена;

- в определенном смысле можно управлять стойкостью шифра путем выбора различных значений отношения r/b;
- вероятностная криптосхема позволяет использовать новые механизмы задания зависимости процедур шифрования от секретного ключа.

Плата за эти достоинства состоит в следующих недостатках:

- скорость уменьшается в r/b раз;
- блоки шифртекста имеют длину больше, чем блоки исходного текста.

Последний недостаток накладывает существенные ограничения на применение вероятностных шифров в компьютерных системах. Для компенсации эффекта расширения можно использовать предварительное сжатие исходного сообщения. Этот способ в ряде случаев позволяет разработать вероятностные шифры, для которых длины шифртекста равны длине входного сообщения. При этом сжатие данных перед их зашифрованием существенно повышает стойкость шифрования. Для многих применений в телекоммуникационных системах рассмотренный вариант вероятностного шифрования применим без существенных ограничений.

Замечание. Интересно, что сжатие данных перед их шифрованием существенно повышает стойкость шифрования к атакам на основе шифртекста. Однако по отношению к атакам на основе известных исходных текстов или подобранных текстов предварительное сжатие информации не повышает стойкость шифрования, поскольку в соответствии с принципом Керкхоффа мы должны полагать, что криптоаналитик знает используемый нами алгоритм сжатия.

Рассмотренные способы вероятностного шифрования представляются весьма эффективными для страхования от непредвиденных слабостей используемого алгоритма шифрования и от встроенных потайных «дверей».

1.6. Особенности приложений

1.6.1. Длина ключа и стойкость

Длина ключа определяет верхнюю границу стойкости криптосистемы. Нападающий всегда может воспользоваться силовой атакой, которая состоит в тотальном переборе по всему пространству возможных ключей. Однако размер этого пространства при увеличении длины ключа растет по экспоненциальному закону. Если длина ключа в битах $l = 64$, то число возможных ключей составляет более 10^{19}. При $l = 128$ их число уже составляет более 10^{38}. В настоящее время вычислительные технологии находятся близко к решению задачи перебора 10^{20} вариантов за разумный интервал времени. Перебор 10^{38}

вариантов представляется недостижимым не только для современных технологий, но и в обозримом будущем. В настоящее время происходит массовый переход от 56-битового ключа в криптосистеме DES к 80-битовому или 128-битовому ключу в новых симметричных шифрах.

К доказуемо стойким блочным криптосистемам можно отнести шифры, для которых теоретически было бы доказано, что лучший способ криптоанализа имеет трудоемкость не меньше некоторого значения, которое гарантирует, что в ближайшее время этот способ будет вычислительно неосуществимым. В настоящее время для многочисленных шифров предложены различные методы криптоанализа и дана оценка трудоемкости решения криптоаналитической задачи для каждого из них. Однако существует принципиальная теоретическая трудность нахождения лучшего алгоритма криптоанализа, что и определяет сложность общего доказательства стойкости.

Для того чтобы новый шифр мог быть принят на вооружение, необходимо выполнить следующие условия:

- он должен быть составлен в соответствии с требованиями конкретного приложения и включать механизмы, реализующие современные принципы задания криптографической стойкости;
- в течение длительного времени он должен тестироваться опытными экспертами, и должна быть доказана его стойкость ко всем известным методам криптоанализа.

Тестирование шифров относится к наиболее сложному и дорогостоящему этапу разработки криптосистем. Для повышения уверенности в стойкости новых шифров их подвергают испытаниям в условиях, благоприятствующих успеху решения криптоаналитической задачи. Например, исследуются модификации с уменьшенным числом раундов шифрования, предполагается возможность генерации аппаратных ошибок шифратора, считается, что часть ключа шифрования известна нападающему. В случае тестирования недетерминированных шифров рассматриваются нападения на основе известной модификации алгоритма шифрования и на основе выбора наиболее слабой модификации. Могут быть использованы варианты атак на модификации шифра с уменьшенным размером ключа шифрования и с уменьшенным размером входного блока данных. Чем строже процедура тестирования, тем больше гарантий, что не будут найдены новые специфические способы нападений. Теоретические гарантии стойкости могут дать только шифры с бесконечным случайным ключом. Однако для защиты компьютерной информации они представляются крайне неудобными. Кроме того, в таких приложениях возникает проблема защиты ключа, объем которого равен объему информации.

Близким к понятию секретного ключа является понятие *пароля*. Пароль также представляет собой секретный элемент. Обычно под ключом понимают элемент, управляющий процессом шифрования, а под паролем — элемент, служащий для аутентификации субъекта (например пользователя или рабочей станции). Во многих случаях пароль служит в качестве информации, управляющей процессом шифрования, а секретный ключ — для аутентификации (подтверждения подлинности) пользователей (в некоторых случаях используется термин *идентификация* пользователя, однако обычно под идентификацией также понимается отождествление пользователя с конкретной записью в списке легальных пользователей, осуществляемое системой контроля доступа, с целью определения совокупности предоставляемых полномочий по использованию ресурсов компьютерной системы). Случайный ключ или пароль труден для запоминания, поэтому его часто хранят на съемных носителях, которые, в свою очередь, хранятся в условиях, обеспечивающих защиту от несанкционированного доступа. Выбор пароля, удобного для запоминания, существенно сужает число возможных его вариантов, поэтому требуется выбирать более длинные пароли или даже целые фразы.

При использовании двухключевых шифров применяются секретные ключи, имеющие длину существенно большую, чем длина ключей в одноключевых криптосистемах. Это связано с особенностями криптоаналитической задачи в случае асимметричных шифров. В некоторых двухключевых криптосистемах секретные ключи выбираются случайно, а затем генерируются соответствующие им открытые ключи (например в методе Диффи–Хеллмана и цифровой подписи Эль-Гамаля). В других криптосистемах формируются секретные ключи, удовлетворяющие специальным требованиям (например множители p и q в системе цифровой подписи RSA). Во втором случае секретные ключи также выбираются случайно, однако после случайного выбора секретного параметра осуществляется его проверка на удовлетворение определенным условиям (например требуется использование простых чисел, чисел, относящихся к заданным показателям по модулю, или чисел, взаимно простых с некоторым уже выбранным параметром).

Случайные секретные ключи удобно хранить на отчуждаемых электронных устройствах (электронных ключах, интеллектуальных электронных картах). Возможно комбинированное использование запоминаемых паролей и секретных ключей, хранимых на отчуждаемых носителях. В настоящее время для пользователей широко доступны миниатюрные и надежные носители ключевой информации.

1.6.2. 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 (настройки) служит определенным барьером против силовой атаки. Предвычисления, которые необходимо выполнить для каждого испытываемого варианта ключа, существенно затрудняют 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 с, 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 значения однонаправленной функции по значению аргумента, задаваемому паролем. 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. Если время проверки одного пароля составит около 1 с и более для процессоров широкого применения, то атака на основе перебора паролей существенно затруднится, а для законного пользователя вносимая задержка не будет существенной.

1.6.3. 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. Приме-

ром криптосистемы с долговременным ключом является российский стандарт шифрования ГОСТ 28147–89 [16, 17], в котором таблицы подстановок являются секретными. В соответствии с общепризнанным принципом Керкхоффа 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 Керкхоффа, но в случае ограничения длины секретного ключа этот прием является обоснованным тем, что этот элемент усиления не имеет цели 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 стан-

дарт 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.

1.6.4. Варианты реализации шифров

В настоящее время алгоритмы, лежавшие в основе механических и электромеханических шифраторов, практически не применяются. Алгоритмы, представляющие практический интерес, являются для ручного шифрования достаточно сложными. Современные шифры 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 книгах [9, 12].

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 подстановки можно генерировать. Соответствующий генератор и представляет собой алгоритм шифрования. Если бы мы могли выбирать непосредственным образом некоторую секретную подстановку, то в принципе 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 состоит в решении задачи генерации псевдослучайной (практически неотличимой от случайной) подстановки для каждого возможного значения ключа. 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 [9].

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 цик-

лического 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};
- умножение;
- 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 использова-

ны 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 [8, 9, 11].

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 [8, 9]. 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 множество 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 с) 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.

ГЛАВА 2

Элементы теории чисел

2.1. Некоторые определения и утверждения

Детальное рассмотрение приводимых здесь фрагментов теории чисел можно найти в работах [1, 3, 4, 7].

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 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 1. 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

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 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 1, то это означает, что не существует числа b, при котором $ab \equiv 1 \bmod n$. Что мы и хотели показать.

О делимости остатка

Утверждение 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 1, легко показать, что в результате получим некоторую перестановку элементов $\boldsymbol{Z}_p$. Иными словами, в наборе $\{a \bmod p, 2a \bmod p, ..., (p-1)a \bmod p\}$ содержится $(p-1)$ 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 от перестановки мест сомножителей произведение не изменяется, 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 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. 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 $\varphi(n)$ и определяется как число положительных целых чисел, которые меньше n и являются взаимно простыми с n.

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

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

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

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

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 $\varphi(ab) = \varphi(a) \times \varphi(b)$.

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 заметить, 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

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

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

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

Доказательство

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

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

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

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

Покажем, что последние два множества включают одни и те же элементы. Действительно, для всех значений $i = 1, 2, \ldots, \varphi(n)$ числа a и x_i являются взаимно простыми с n, поэтому число ax_i тоже является взаимно простым с n. Следовательно, все элементы Φ' являются целыми числами, меньшими n (умножение мы выполняли по модулю n) и взаимно простыми с n. В 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. Таким образом, множества Φ и Φ' содержат одни и те же элементы. Перемножая все элементы множества Φ', а затем все элементы множества Φ, получаем равенство

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

из которого следуют сравнения

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

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

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Из последнего сравнения вытекают следующие соотношения:

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

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

Обобщенная теорема Эйлера

Обобщенной функцией Эйлера называется функция $L(n)$, определенная для всех натуральных чисел следующим образом: $L(1) = 1$, а при $n > 1$

$$L(n) = HOK\left[p_1^{\alpha_1 - 1}(p_1 - 1); p_2^{\alpha_2 - 1}(p_2 - 1); \ldots; p_k^{\alpha_k - 1}(p_k - 1) \right],$$

где $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ и НОК — наименьшее общее кратное.

Обобщенная теорема Эйлера гласит: Если НОД$(a, n) = 1$, то

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

Доказательство

Пусть $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ — каноническое разложение числа n. По теореме Эйлера $\forall\, i \in \{1, 2, \ldots, k\}$ имеем: $a^{p_i^{\alpha_i - 1}(p_i - 1)} \equiv 1 \bmod p_i^{\alpha_i}$. Возводя обе части последнего сравнения в целочисленную степень $\dfrac{L(n)}{p_i^{\alpha_i - 1}(p_i - 1)}$ (последнее число является целым, так как по определению $p_i^{\alpha_i - 1}(p_i - 1) \mid L(n)$), получаем: $\forall\, i \in \{1, 2, \ldots, k\}$ имеет место $a^{L(n)} \equiv 1 \bmod p_i^{\alpha_i}$. Так как $\forall\, i, j \in \{1, 2, \ldots, k\}$ ($i \neq j$) числа $p_i^{\alpha_i}$ и $p_j^{\alpha_j}$ являются взаимно простыми, то по теореме 12 (*см. раздел 2.7*) имеем $a^{L(n)} \equiv 1 \bmod n$.

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 $d = \text{НОД}(a, b)$. Отсюда следует, что $\text{НОД}(a,b) = \text{НОД}\,[b,$ **MOD** $(a, b)]$.

Для всякого a имеем также $\text{НОД}(a, 0) = |a|$. Пусть задано 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: $d = \text{НОД}(a, b)$.

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:

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

откуда получается следующая индуктивная процедура вычисления x_i и y_i:

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

$$y_i = y_{i-2} - y_{i-1}q_{i-1}.$$

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) = $ax + by$.

2.5. Показатели и первообразные корни

Для взаимно простого с модулем n числа a, согласно теореме Эйлера, существует некоторое положительное $\gamma = \varphi(n)$, 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 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 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

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 6

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

Доказательство

Действительно, пусть $a^{\varepsilon'}$ 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 $(a^{\varepsilon'})^{\varepsilon} \equiv 1 \pmod{n}$, отку-

да, по утверждению 5, следует, что ε' делится на δ. Таким образом, $\varepsilon'|\delta$ и $\delta|\varepsilon'$, поэтому $\delta = b$.

Утверждение 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 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 $\delta_j = z q_s^{\alpha_s}$. Пусть a_j — одно из чисел ряда $1, 2, \ldots, p-1$, принадлежащих показателю δ_j. Согласно утверждению 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 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 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 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 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 10

Пусть $\alpha \ge 1$ 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 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 12

Пусть p — простое нечетное число, $\alpha \geq 1$; 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 понятия 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 $\gamma = \operatorname{ind}_g a$ (или просто $\gamma = \operatorname{ind} a$). Из утверждения 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 $\gamma' = \operatorname{ind}_g a$, 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 $c \mid (\gamma - \gamma')$, т. е. $\gamma \equiv \gamma' \pmod c$.

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

Рассмотрим следующие свойства индексов:

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

и, в частности,

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

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

2.6. Теоремы о числе классов с заданным показателем

Рассмотрим вопрос о числе классов с заданным показателем. Иными словами, речь идет о количестве чисел, меньших модуля и относящихся к заданному показателю (каждое из таких чисел задает класс сравнимых с ним чисел по заданному модулю). Если взять классы, взаимно простые с модулем n, то каждый такой класс относится к некоторому показателю δ, причем $\delta \mid n$. Обо-

значим число классов, относящихся к показателю δ через $\psi(\delta)$. Если δ не делит $\varphi(n)$, то такое δ не может быть показателем ни для какого числа, поэтому в этом случае имеем $\psi(\delta) = 0$. Если понятно о каком модуле идет речь, то показатель числа a будем обозначать как $P(a)$.

Теорема 1

Сравнение степени k по простому модулю p с коэффициентом при старшем члене, не делящемся на p, может иметь не больше чем k решений.

Теорема 2

В последовательности $a, a^2, a^3, \ldots$ все числа принадлежат $P(a)$ классам, представителями которых являются числа $a, a^2, a^3, \ldots, a^{P(a)}$, где $P(a)$ есть показатель числа a по некоторому модулю n.

Доказательство

Числа $a, a^2, a^3, \ldots, a^{P(a)}$ попарно несравнимы по модулю n. Действительно, сравнение $a^i \equiv a^j \bmod n$ имеет место только тогда, когда $i \equiv j \bmod P(a)$. Но среди показателей степеней a в рассматриваемой последовательности нет сравнимых по модулю $P(a)$. С другой стороны, из сравнения $i \equiv j \bmod P(a)$ следует $a^i \equiv a^j \bmod n$, т. е. если показатели сравнимы по модулю $P(a)$, то степени сравнимы по модулю n. Поэтому в последовательности $a, a^2, a^3, \ldots$ не может быть больше чем $P(a)$ несравнимых между собой чисел, т. е. каждое число вида a^N сравнимо с некоторым числом из последовательности $a, a^2, a^3, \ldots, a^{P(a)}$.

Теорема 3

$P(a^j) = P(a)$ тогда и только тогда, когда НОД$(i, P(a)) = 1$.

Доказательство

Достаточность. Пусть НОД$(i, P(a)) = 1$. Найдем наименьшее положительное целое число δ, такое что $(a^j)^\delta \equiv 1 \bmod n$. Имеем $a^{i\delta} \equiv 1 \bmod n$, следовательно, $P(a) | i\delta$. Из условий НОД$(i, P(a)) = 1$ и $P(a) | i\delta$ следует, что $P(a) | \delta$ и наименьшее натуральное число, делящееся нацело на $P(a)$, равно $\delta = P(a)$.

Необходимость. Предположим противное: НОД$(i, P(a)) = d > 1$. Тогда $\frac{P(a)}{d}$ и $\frac{i}{d}$ являются целыми числами и $(a^i)^{\frac{P(a)}{d}} = (a^{P(a)})^{\frac{i}{d}} \equiv 1 \bmod n$, т. е. $P(a^i) \le \frac{P(a)}{d} < P(a)$.

Эта теорема показывает, что среди степеней последовательности $a, a^2, a^3, \ldots, a^{P(a)}$ все степени a^i, у которых i взаимно простое с $P(a)$, имеют тот же показатель по модулю n, что и само число a.

Теорема 4

Если по модулю n $P(a) = k$, то классы $\overline{a}, \overline{a^2}, \ldots, \overline{a^k}$ представляют собой различные решения сравнения $x^k \equiv 1 \bmod n$. (В общем случае указанные классы не охватывают *всех* решений данного сравнения.)

Доказательство

Если $P(a) = k$, то $a^k \equiv 1 \bmod n$, поэтому при произвольном значении целого числа $i \geq 0$ имеем $(a^i)^k = (a^k)^i \equiv 1 \bmod n$. Поэтому все числа a^i удовлетворяют сравнению $x^k \equiv 1 \bmod n$, т. е. классы $\overline{a}, \overline{a^2}, \ldots, \overline{a^k}$ есть решения этого сравнения. Согласно теореме 2, все эти решения различны. Но в общем случае существуют и другие решения, например, для такого составного модуля n, при котором $P(a) = L(n)$, решениями сравнения $x^{P(a)} \equiv 1 \bmod n$ являются классы $\overline{1}, \overline{2}, \ldots, \overline{n-1}$, тогда как теорема указывает на $L(n) < n - 1$ решений.

Теорема 5

Если по простому модулю p $P(a) = k$, то классы $\overline{a}, \overline{a^2}, \ldots, \overline{a^k}$ представляют собой *все* решения сравнения $x^k \equiv 1 \bmod p$.

Доказательство

В теореме 4 установлено, что классы $\overline{a}, \overline{a^2}, \ldots, \overline{a^k}$ образуют различные решения сравнения $x^k \equiv 1 \bmod n$. Поскольку p есть простой модуль, то это сравнение не может иметь больше чем k решений (теорема 1), т. е. классы $\overline{a}, \overline{a^2}, \ldots, \overline{a^k}$ исчерпывают все решения.

Теорема 6

Количество классов, относящихся к какому-либо показателю по модулю n, равно функции Эйлера $\varphi(n)$ от модуля:

$$\sum_{\delta | \varphi(n)} \psi(\delta) = \varphi(n).$$

Доказательство

Доказательство непосредственно следует из того, что каждый из взаимно простых с модулем классов относится к какому-либо показателю.

Теорема 7

По простому модулю p для любого целого $\delta \geq 1$ имеет место неравенство

$$\psi(\delta) \leq \varphi(p).$$

Доказательство

Если для данного значения показателя δ не существует классов, относящихся к нему, то $\psi(\delta) = 0$, т. е. $\psi(\delta) < \varphi(\delta)$. Если существует хоть один такой класс, то классы, к которым относятся числа a, $a^2 \bmod p$, $a^3 \bmod p$, …, $a^\delta \bmod p$, образуют все решения сравнения $x^\delta \equiv 1 \bmod p$. По теореме 3 $P(a^i) = P(a)$, $1 \leq i \leq \delta$ тогда и только тогда, когда НОД$(P(a), i) = 1$. Число таких степеней i равно $\varphi(\delta)$. Если имеется какой-либо класс, показатель которого по модулю p равен δ, то он также должен удовлетворять сравнению $x^\delta \equiv 1 \bmod p$. Поэтому класс должен находиться среди $\varphi(\delta)$ классов, представителями которых являются числа a, $a^2 \bmod p$, $a^3 \bmod p$, …, $a^\delta \bmod p$. Отсюда следует, что $\psi(\delta) = \varphi(p)$, если $\psi(\delta) \neq 0$. Учитывая случай $\psi(\delta) = 0$, имеем $\psi(\delta) \leq \varphi(p)$.

Теорема 8

По простому модулю p при $\delta | p - 1$ имеет место равенство $\psi(\delta) = \varphi(\delta)$.

Теорема 9

По любому простому модулю p существует $\varphi(p - 1)$ первообразных корней.

Доказательство

Доказательство вытекает непосредственно из теоремы 8 при $\delta = p - 1$.

Теорема 10

Первообразные корни по модулю n существуют тогда и только тогда, когда либо 1) $n = p^\alpha$ или $n = 2p^\alpha$, где p — любое нечетное простое число, α — любое целое положительное число либо 2) $n = 2^\alpha$, где $0 \leq \alpha \leq 2$.

Пояснение. Если n имеет хотя бы два нечетных простых делителя p_1 и p_2, то числа $p_1 - 1$ и $p_2 - 1$ не являются взаимно простыми, поэтому $L(n) < \varphi(n)$. Из обобщенной теоремы Эйлера следует, что для любого a, взаимно простого с n, имеем $P(a) \leq L(n) < \varphi(n)$, т. е. первообразных корней не существует.

2.7. Китайская теорема об остатках

Эта теорема является одним из весьма полезных и часто используемых в криптографии результатов теории чисел. Она фактически утверждает, что

любое значение из множества минимальных положительных представителей ($\mathbf{Z}/N$) классов вычетов по модулю $N = n_1 n_2 \ldots n_g$, где $\forall\, i, j \in \{1, 2, \ldots, g\}$ НОД(n_i, n_j) = 1, может быть представлено в виде набора остатков от деления этого значения на каждый из сомножителей n_i; т. е. имеется взаимно однозначное соответствие

$$R \leftrightarrow (r_1, r_2, \ldots, r_g), \text{ где } R \in \mathbf{Z}/N,\ r_1 \in \mathbf{Z}/n_1,\ r_2 \in \mathbf{Z}/n_2,\ \ldots,\ r_g \in \mathbf{Z}/n_g.$$

Китайская теорема об остатках гласит следующее.

Теорема 11

Пусть $n_1, n_2, \ldots, n_g$ — набор попарно взаимно простых чисел, $N = n_1 n_2 \ldots n_g$; числа $c_1, c_2, \ldots, c_g$ удовлетворяют условиям $c_1 N/n_1 \equiv 1 \bmod n_1$, $c_2 N/n_2 \equiv 1 \bmod n_2$, …, $c_g N/n_g \equiv 1 \bmod n_g$. Тогда решение системы

$$\begin{cases} x \equiv r_1 \bmod n_1 \\ x \equiv r_2 \bmod n_2 \\ \quad \ldots\ldots \\ x \equiv r_g \bmod n_g \end{cases}$$

имеет вид $R' \equiv r_1 c_1 N/n_1 + r_2 c_2 N/n_2 + \ldots + r_g c_g N/n_g \pmod N$.

Доказательство

Поскольку $\forall\, i, j \in \{1, 2, \ldots, g\}$ и $i \neq j$ $n_i \left| \dfrac{N}{n_j} \right.$ (например $n_1 \left| \dfrac{N}{n_2} \right.$, $n_3 \left| \dfrac{N}{n_2} \right.$, …, $n_g \left| \dfrac{N}{n_2} \right.$), то $R' \bmod n_i = (r_1 c_1 N/n_1 + r_2 c_2 N/n_2 + \ldots + r_g c_g N/n_g) \bmod n_i = (r_i c_i N/n_i) \bmod n_i$.

То есть $R' \equiv r_i c_i N/n_i \equiv r_i \pmod{n_i}$ для $i = 1, 2, \ldots, g$.

Минимальное положительное число R из класса вычетов по модулю N, представляющего собой решение системы сравнений, и является тем элементом кольца $\mathbf{Z}/N$, который ставится в соответствие набору значений $r_1, r_2, \ldots, r_g$. Подобрать необходимые значения $c_i N/n_i \equiv 1 \bmod n_i$ достаточно просто. Их можно вычислить по формуле $c_i = N/n_i[(n_i/N) \bmod n_i]$. Если дано некоторое $R \in \mathbf{Z}/N$, то, выполнив деление R на каждое из чисел n_i, определим однозначно набор остатков, который ставится в соответствие заданному числу R. Фактически китайская теорема об остатках указывает способ решения сравнений указанного типа.

В дальнейшем будет использована также следующая теорема:

Теорема 12

Если $a \equiv b \pmod{n_1}$, $a \equiv b \pmod{n_2}$,…, $a \equiv b \pmod{n_g}$, то $a \equiv b \pmod{N}$, где $N = \text{НОК}[n_1, n_2, \ldots, n_g]$.

Доказательство

Из условия непосредственно следует, что $n_1 | a - b$, $n_2 | a - b$, …, $n_g | a - b$, откуда получаем $N | a - b$. Последнее означает, что выполняется $a \equiv b \pmod{N}$.

2.8. Алгоритм возведения в степень по модулю

Пусть требуется вычислить значение $S = a^W \bmod n$, где числа W, n и a достаточно большие. Если модуль может быть разложен на сравнительно небольшие простые делители, т. е. $n = p_1 p_2 \ldots p_g$, то возведение в степень по такому модулю может быть сильно упрощено, если использовать китайскую теорему об остатках. Сначала вычисляются вычеты $a^W \bmod p_i$ по каждому из простых делителей p_i, $i = 1, 2, \ldots, g$. Используя теорему Ферма, эти вычисления можно сделать достаточно быстрыми. В результате получим следующую систему сравнений:

$$\begin{cases} a^W \equiv r_1 \bmod p_1, & 0 \le r_1 < p_1 \\ a^W \equiv r_2 \bmod p_2, & 0 \le r_1 < p_2 \\ \qquad\qquad \ldots\ldots & \\ a^W \equiv r_g \bmod p_g, & 0 \le r_1 < p_g \end{cases}.$$

Полагая $x = a^W$ и решая по китайской теореме об остатках систему сравнений, приведенную выше, найдем значение $R \in \mathbf{Z}/n$, удовлетворяющее системе. Поскольку a^W также удовлетворяет рассматриваемой системе сравнений и все ее решения сравнимы между собой по модулю n, то $R \equiv a^W \bmod n$, т. е. $S = R$.

Однако в криптографических задачах возведение в степень обычно осуществляется по простому модулю или по модулю, представляющему собой произведение двух больших простых чисел, поэтому рассмотренный выше способ редко применяется. Кроме того, даже для сомножителей p_i сравнительно малого размера (например 32-битовых чисел в двоичном представлении) прямолинейное вычисление значения $a^{W \bmod p_i - 1} \bmod p_i$ путем перемножения $W \bmod (p_i - 1)$ одинаковых сомножителей, имеющих значение a, окажется достаточно трудоемким. Обычно используется следующий способ бы-

строго возведения в большую степень по большому модулю с произвольной структурой. Однако комбинирование первого способа с последним позволяет обеспечить дополнительное снижение сложности возведения в степень в случае составного модуля.

Алгоритм быстрого возведения в степень по модулю

Пусть требуется вычислить значение $S = a^W \bmod n$. Представим степень W в виде разложения по степеням числа 2:

$$W = w_{g-1}2^{g-1} + w_{g-2}2^{g-2} + \ldots + w_2 2^2 + w_1 2^1 + w_0,$$

где w_i есть цифра 0 или 1.

Преобразуем $S = a^W \bmod n$ следующим образом:

$$S = a^W \bmod n = a^{w_{g-1}2^{g-1} + w_{g-2}2^{g-2} + \ldots + w_2 2^2 + w_1 2^1 + w_0} \bmod n =$$

$$= (a^2)^{w_{g-1}2^{g-2} + w_{g-2}2^{g-3} + \ldots + w_2 2^1 + w_1 2^0} \cdot a^{w_0} \bmod n =$$

$$= ((a^2)^2)^{w_{g-1}2^{g-3} + w_{g-2}2^{g-4} + \ldots + w_2 2^0} \cdot (a^2)^{w_1} \cdot a^{w_0} \bmod n =$$

$$= (\ldots((a^2)^2\ldots)^2)^{w_{g-1}} \cdot \ldots \cdot (a^8)^{w_3} \cdot (a^4)^{w_2} \cdot (a^2)^{w_1} \cdot a^{w_0} \bmod n.$$

Из последней формулы нетрудно вывести следующий псевдокод, описывающий алгоритм быстрого возведения в степень:

ВХОД:

1. S:=1 и c:=a {Присвоить начальные значения переменным S и c}
2. For i:= 0 to g – 1 do {Основной цикл}

   2.1. If w_i = 1, then S:= Sc mod n. Otherwise go to step 2.2.

   2.2. $c := c^2 \bmod n$.

ВЫХОД: $S = a^W \bmod n$.

В более понятном и удобном для программирования виде алгоритм быстрого возведения в степень записывается следующим образом.

ВХОД: Целочисленные значения n, $a > 0$ и $W \geq 0$.

1. Инициализируем переменные S:= 1; V:= W и c := a.
2. Если V = 0, то СТОП .

3. Если $V \bmod 2 = 1$ (т. е. текущее значение V является нечетным), то присваиваем новые значения переменным $S := Sc \bmod n$ и $V := (V - 1)/2$ и переходим к шагу 5. В противном случае переходим к шагу 4.
4. Выполняем операцию присваивания $V := V/2$.
5. Выполняем операцию присваивания $c := c^2 \bmod n$.
6. Переходим к шагу 2.

ВЫХОД: Значение $S = a^W \bmod n$.

Нетрудно подсчитать, что средняя сложность данного алгоритма составляет $1{,}5g$ операций умножения двух g-битовых чисел плюс $1{,}5g$ операций деления $2g$-битовых чисел на g-битовое число. Для 1000-битовых и более длинных чисел данный алгоритм выполняется на ЭВМ достаточно быстро.

2.9. Нахождение чисел, относящихся к заданному простому показателю

Нахождение первообразных корней

Для любого числа a, взаимно простого с модулем m, существуют числа δ, такие что $a^\delta = 1 \bmod m$. 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 функции Эйлера от модуля, т. е. делители числа $\varphi(m)$, которое является наибольшим возможным показателем по модулю m. Если модуль является простым числом p, то число $\psi(\gamma)$ чисел, относящихся к показателю γ, равно функции Эйлера от числа γ, т. е. $\psi(\gamma) = \varphi(\gamma)$. Чем меньше показатель, тем меньше существует чисел, относящихся к нему. Наибольшее число чисел относится к показателю $p - 1$, т. е. имеется достаточно большое число первообразных корней и при случайном выборе чисел мы с большой вероятностью попадаем на них. Как можно проверить, что выбранное число действительно является первообразным корнем по модулю p?

Пусть имеем разложение $p-1 = d_1^{\alpha_1} d_2^{\alpha_2} \ldots d_h^{\alpha_h}$. *Если для каждого простого делителя* d_i *числа* $p-1$ *выполняется условие* $\alpha^{\frac{p-1}{d_i}} \neq 1 \bmod p$, *то число* α *является первообразным корнем (примитивным элементом) по модулю* p.

Доказательство

Допустим, что α не является первообразным корнем и относится к показателю $\delta < p-1$. Поскольку $\delta \mid p-1$, то $k = (p-1)/\delta$ есть простой или составной делитель числа $p-1$, т. е., по крайней мере, для одного из делителей d_i имеем $d_i \mid k$. Следовательно, число $\frac{p-1}{d_i\delta}$ является целым. По предположению имеем: $\alpha^{\delta} = 1 \bmod p \Rightarrow (\alpha^{\delta})^{\frac{p-1}{d_i\delta}} = 1 \bmod p \Rightarrow \alpha^{\frac{p-1}{d_i}} = 1 \bmod p$, что противоречит исходному условию.

Нахождение чисел, относящихся к заданному простому показателю

Если длина простого делителя γ (случай составного делителя будет рассмотрен отдельно) существенно меньше длины простого модуля, то нахождение чисел, относящихся к γ как к показателю, путем случайного выбора чисел a и проверки соотношения $a^{\gamma} = 1 \bmod p$ является вычислительно неэффективным. В реальных системах ЭЦП с сокращенной длиной подписи простой модуль p имеет размер 1024 или 2048 бит, а размер показателя γ составляет около 160–256 бит. Для нахождения числа α, относящегося к простому 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. Действительно, мы рассматриваем простое число γ, поэтому оно не имеет делителей, значение которых меньше его самого, а при выполнении условия $\alpha^{\gamma} = 1 \bmod p$ показатель α делит γ.

Нахождение чисел, относящихся к заданному составному показателю

Если требуется найти число α, относящееся к составному показателю γ, каноническое разложение которого имеет вид $\gamma = q_1^{\omega_1} q_2^{\omega_2} \ldots q_z^{\omega_z}$, то можно воспользоваться следующим алгоритмом:

1. Выбирается случайное число b, превосходящее 1 и меньшее числа p.
2. Вычисляется значение $\gamma' = (p-1)/\gamma$ и число $g = b^{\gamma'} \bmod p$.
3. Если $g = 1$, то перейти к шагу 1.
4. Если для каждого простого делителя q_i числа γ выполняется условие $g^{\frac{\gamma}{q_i}} \neq 1 \bmod p$, то в качестве числа α взять число g. В противном случае повторить шаги 1–4.

Докажем, что этот алгоритм действительно находит число, относящееся к показателю γ. Для полученного числа $\alpha \neq 1$ имеем $\alpha = b^{(p-1)/\gamma} \bmod p$. Согласно теореме Ферма, имеем $\alpha^{\gamma} = a^{(p-1)} = 1 \bmod p$. Допустим, что α относится к показателю $\delta < \gamma$. Тогда $\delta \mid \gamma$ и $k = \gamma/\delta$ есть простой или составной делитель числа γ, т. е., по крайней мере, для одного из делителей q_i имеем $q_i \mid k$. Следовательно, число $\frac{\gamma}{q_i \delta}$ является целым. По предположению имеем: $\alpha^{\delta} = 1 \bmod p \Rightarrow (\alpha^{\delta})^{\frac{\gamma}{q_i \delta}} = 1 \bmod p \Rightarrow \alpha^{\frac{\gamma}{q_i}} = 1 \bmod p$, что противоречит условию, проверяемому на шаге 4 алгоритма поиска числа α.

2.10. Теоремы о числе решений степенных сравнений

Вначале рассмотрим несколько утверждений, используемых далее при доказательстве теорем о числе решений степенных сравнений.

Утверждение 13

Для произвольных натуральных чисел k и m из выполнимости сравнения $a \equiv b \pmod m$ следует сравнимость чисел ka и kb по модулю km, т. е. $ka \equiv kb \pmod{km}$.

Доказательство

По условию имеем $k > 0$ и $m \mid a - b$, откуда следует $km \mid ka - kb \Rightarrow ka \equiv kb \pmod{km}$.

Утверждение 14

Для произвольных натуральных чисел k и m из выполнимости сравнения $ka \equiv kb \pmod{km}$ следует сравнимость чисел a и b по модулю m, т. е. $a \equiv b \pmod m$.

Доказательство

По условию имеем $k \neq 0$ и $km|ka - kb$, откуда следует $km|k(a-b) \Rightarrow m|a-b \Rightarrow a \equiv b \pmod{km}$.

Утверждение 15

Если обе части сравнения $f(x) \equiv g(x) \pmod{m}$ *и модуль умножим на одно и то же число* $k > 0$, *то получим сравнение* $kf(x) \equiv kg(x) \pmod{km}$, *эквивалентное первоначальному.*

Доказательство

Если при некотором $x = x_0$ имеем $f(x_0) \equiv g(x_0) \pmod{m}$, то из утверждения 13 имеем $kf(x_0) \equiv kg(x_0) \pmod{m}$. Из $kf(x_0) \equiv kg(x_0) \pmod{m}$, согласно утверждению 14, имеем $f(x_0) \equiv g(x_0) \pmod{m}$.

Утверждение 16

Если НОД$(a, m) = d$ *и* $d \nmid b$, *то сравнение* $ax \equiv b \pmod{m}$ *не имеет решений.*

Доказательство

Допустим, что существует число x_0, такое что $ax_0 \equiv b \pmod{m}$, т. е. $ax_0 = Qm + b$, где Q есть некоторое целое число. Поскольку $d \mid ax_0$ и $d \mid m$, то из последнего равенства следует $d \nmid b$. Мы пришли к противоречию, следовательно, наше допущение неверно, что доказывает утверждение.

Утверждение 17

Если НОД$(a, m) = 1$, *то сравнение* $ax \equiv b \pmod{m}$ *имеет единственное решение.*

Доказательство

Если НОД$(a, m) = 1$, то в приведенной системе наименьших неотрицательных вычетов по модулю m существует единственный элемент a^{-1}, такой что $a^{-1}a = 1$. Для этого элемента имеем НОД$(a^{-1}, m) = 1$. Умножая обе части сравнения $ax \equiv b \pmod{m}$ на a^{-1}, получим $a^{-1}ax \equiv a^{-1}b \pmod{m} \Rightarrow x \equiv a^{-1}b \pmod{m}$.

Утверждение 18

Числа класса $\overline{a}$ *по модулю m образуют следующие k классов по модулю km:* $\overline{a}, \overline{a+m}, \overline{a+2m}, \ldots, \overline{a+(k-1)m}$.

Утверждение 19

Если НОД$(a, m) = d$ *и* $d \mid b$, *то сравнение* $ax \equiv b \pmod{m}$ *имеет d решений. Все эти решения принадлежат одному классу по модулю m/d.*

Доказательство

Из утверждения 15 следует, что при НОД(a, m) = d и $d \mid b$ сравнение $ax \equiv b \pmod m$ эквивалентно сравнению вида $a_1 x \equiv b_1 \pmod{m_1}$, где $m_1 = m/d$, НОД(a_1, m_1) = 1.

Согласно утверждению 17, последнее сравнение имеет одно решение, представляющее собой один класс по модулю m/d, т. е. этому сравнению удовлетворяют числа вида $x \equiv \alpha \pmod{m/d}$, где α может быть найдено, например, по формуле $x \equiv a_1^{-1} b_1 \equiv a_1^{\varphi(m/d)-1} b_1 \pmod{m/d}$. Числа этого класса по модулю m/d образуют d классов по модулю m (*см. утверждение 18*), и решения сравнения $ax \equiv b \pmod m$ могут быть записаны в виде $x \equiv \alpha \pmod m$, $x \equiv \alpha + m/d \pmod m$, …, $x \equiv \alpha + (d-1)m/d \pmod m$.

Утверждение 20

Если НОД(a, m) = 1 *и* x *пробегает значения, образующие приведенную систему вычетов, то* ax *также принимает значения, образующие приведенную систему вычетов по модулю* m.

Теорема 13

1. *При* $p \nmid a$ *сравнение* $x^n \equiv a \pmod p$ *по простому модулю* $p > 2$ *либо совсем не имеет решений, либо число решений равно наибольшему общему делителю* n *и* $p - 1$.

2. *Сравнение* (1) *не имеет решений, если для* δ = НОД(n, $p-1$) $\delta \nmid \operatorname{ind} a$, *и имеет* δ *решений, если* $\delta \mid \operatorname{ind} a$.

Доказательство

Пусть НОД(n, $p-1$) = δ. По простому модулю p существуют первообразные корни (*см. утверждение 8*). Пусть g есть некоторый первообразный корень. Индексируя сравнение

$$x^n \equiv a \pmod p \tag{1}$$

по основанию g, получаем сравнение

$$n \operatorname{ind} x \equiv \operatorname{ind} a \pmod{p-1}, \tag{2}$$

которое эквивалентно исходному. Если обозначить $\operatorname{ind} x = z$, $\operatorname{ind} a = b$, то для неизвестной z получим сравнение 1-й степени:

$$nz \equiv b \pmod{p-1}. \tag{3}$$

При $\delta \nmid \operatorname{ind} a$, т. е. $\delta \nmid b$, сравнение (3) не имеет решений (*см. утверждение 16*), но тогда не существует и значений x, удовлетворяющих сравнениям (2) и (1). При $\delta \mid \operatorname{ind} a$, т. е. $\delta \mid b$, согласно утверждению 19, сравнение (3) имеет δ решений. Значения $\operatorname{ind} x$, удовлетворяющие сравнению (2), принадлежат

δ классам по модулю $p-1$, а следовательно, и для x существует δ классов по модулю p, удовлетворяющих сравнению (1).

Замечание

1. При $p \mid a$ сравнение (1) может быть записано в виде $x^n \equiv 0 \pmod p$, и в этом случае оно имеет одно решение $x \equiv 0 \pmod p$.
2. Пусть g и h — произвольные первообразные корни по модулю p. Тогда имеем
   $h \equiv g^{\operatorname{ind}_g h} \equiv (h^{\operatorname{ind}_h g})^{\operatorname{ind}_g h} \equiv h^{\operatorname{ind}_h g \cdot \operatorname{ind}_g h} \pmod p \Rightarrow \operatorname{ind}_h g \cdot \operatorname{ind}_g h \equiv 1 \pmod{p-1} \Rightarrow \operatorname{ind}_h g \equiv (\operatorname{ind}_g h)^{-1} \pmod{p-1}$, т. е. индексы первообразных корней являются взаимно простыми с $p-1$.
   Действительно, пусть НОД$(\operatorname{ind}_g a, p-1) = \delta > 1$. Тогда
   $a^{\frac{p-1}{\delta}} \equiv (g^{\operatorname{ind}_g a})^{\frac{p-1}{\delta}} \equiv (g^{p-1})^{\frac{\operatorname{ind}_g a}{\delta}} \equiv 1^z \equiv 1 \pmod p$, где z — целое число, т. е. показатель, к которому относится число a, меньше чем $p-1$. Поскольку НОД$(\operatorname{ind}_h g, p-1) = 1$ и $\operatorname{ind}_h a \equiv \operatorname{ind}_h g \cdot \operatorname{ind}_g a$, то из $\delta \nmid \operatorname{ind}_g a$ (или из $\delta \mid \operatorname{ind}_g a$) следует $\delta \nmid \operatorname{ind}_h a$ (или $\delta \mid \operatorname{ind}_h a$).
3. Заметим, что $\operatorname{ind} 1 = p-1$, т. е. $\operatorname{ind} 1 \equiv 0 \pmod{p-1}$. Следовательно, $\delta \mid \operatorname{ind} 1$ и сравнение $x^n \equiv 1 \pmod p$ имеет δ решений.

Определение

1. *Число a называется вычетом n-й степени по простому модулю p, если $p \nmid a$ и сравнение $x^n \equiv a \pmod p$ имеет решения.*

2. *Число a называется невычетом n-й степени по простому модулю p, если сравнение $x^n \equiv a \pmod p$ не имеет решений.*

Теорема 14

По простому модулю $p > 2$ число классов вычетов n-й степени равно $(p-1)/\delta$, где $\delta = (n, p-1)$.

Доказательство

Всего по модулю p имеется $p-1$ классов, взаимно простых с модулем (класс $\overline{0}$, состоящий из чисел, делящихся на p, согласно указанному выше определению, не относится к вычетам n-й степени). Индексы этих классов образуют полную систему вычетов по модулю $p-1$, т. е. каждому такому классу (если взять $\operatorname{ind} 1 = p-1$) можно сопоставить индекс, равный одному из чисел: $1, 2, \ldots, p-1$.

Пусть НОД(n, $p-1$) = δ. Среди чисел 1, 2, …, $p-1$ имеется $\frac{p-1}{\delta}$ чисел, делящихся на δ, которыми являются числа δ, 2δ, …, $(\frac{p-1}{\delta}-1)\delta$, $(p-1)$. Эти числа являются индексами $\frac{p-1}{\delta}$ различных чисел из множества $\{1, 2, \ldots, p-1\}$. Если a равно любому из этих чисел, то, согласно теореме 13, сравнение $x^n \equiv a \pmod p$ имеет решения, т. е. существует $\frac{p-1}{\delta}$ классов вычетов n-й степени по простому модулю p.

Теорема 15

Если δ = (n, $p-1$)*, то вычеты n-й степени по простому модулю* $p>2$ *совпадают с вычетами степени d по этому модулю.*

Доказательство

Если НОД(n, $p-1$) = δ, то будем иметь также НОД(δ, $p-1$) = δ. При $p \nmid a$ сравнения $x^n \equiv a \pmod p$ и $x^\delta \equiv a \pmod p$, согласно теореме 13, имеют решения при одних и тех же числах a, таких что $\delta \mid \operatorname{ind} a$.

В силу теоремы 15 можно рассматривать вычеты и невычеты n-й степени только для таких n, которые являются делителями числа $p-1$, т. е. рассматривать сравнения вида $x^n \equiv a \pmod p$, где $n \mid p-1$. Действительно, сравнение $x^n \equiv a \pmod p$ можно записать в виде:

$$(x^\delta)^{n/\delta} \equiv a \pmod p, \qquad (4)$$

где НОД((n/δ), $p-1$) = 1, т. е. сравнение (4) имеет единственное решение относительно неизвестной $z = x^\delta$. Пусть $x^\delta \equiv b \pmod p$. Последнее сравнение имеет вид сравнения $x^n \equiv a \pmod p$, где $n \mid p-1$. Для случая $n \mid p-1$ теоремы 13 и 14 имеют следующую формулировку.

Теорема 13*

При $p \nmid a$ *и* $n \mid p-1$ *сравнение* $x^n \equiv a \pmod p$ *по простому модулю* $p>2$ *либо совсем не имеет решений, либо имеет n решений. По такому модулю a является вычетом n-й степени тогда и только тогда, когда* $n \mid \operatorname{ind} a$.

Теорема 14*

По простому модулю $p>2$ *и* $n \mid p-1$ *число классов вычетов n-й степени равно* $(p-1)/n$.

Теорема 16

При $n \mid p-1$ a является вычетом n-й степени по простому модулю $p>2$ тогда и только тогда, когда $a^{(p-1)/n} \equiv 1 \pmod p$.

Доказательство

Индексируя сравнение $a^{(p-1)/n} \equiv 1 \pmod p$, получаем, что это сравнение имеет место тогда и только тогда, когда $\frac{p-1}{n} \operatorname{ind} a \equiv 0 \pmod{p-1}$, т. е. когда $\frac{p-1}{n} \operatorname{ind} a = Q(p-1)$ или $\operatorname{ind} a = nQ$ (Q — целое число), т. е. сравнение $a^{(p-1)/n} \equiv 1 \pmod p$ имеет место тогда и только тогда, когда $n \mid \operatorname{ind} a$.

Теорема 17

При $p \nmid a$ и $n \mid p-1$ все решения сравнения $x^n \equiv a \pmod p$ по простому модулю $p>2$ можно получить, умножая одно решение этого сравнения на различные решения сравнения $x^n \equiv 1 \pmod p$.

Другими словами, все корни n-й степени из $\bar{a}$ по модулю p можно получить, умножая один из этих корней на различные корни n-й степени из $\bar{1}$.

Доказательство

Поскольку $\operatorname{ind} 1 = 0$ и $n \mid 0$, сравнение $x^n \equiv 1 \pmod p$ имеет n решений. Обозначим эти решения через $\bar{t}_1, \ldots, \bar{t}_n$, где $t_i \not\equiv t_j \pmod p$ и $p \nmid t_i$. Пусть $\bar{x}_0$ удовлетворяет сравнению (1), тогда $(x_0 t_i)^n = x_0^n t_i^n \equiv a \cdot 1 \equiv a \pmod p$ при $i = 1, 2, \ldots, n$, т. е. классы $\overline{x_0 t_1}, \ldots, \overline{x_0 t_n}$ представляют собой решения сравнения $x^n \equiv a \pmod p$.

Поскольку $p \nmid a$, то $p \nmid x_0$ и НОД$(x_0, p) = 1$, следовательно, числа $x_0 t_1, \ldots, x_0 t_n$ являются (утверждение 20) частью приведенной системы вычетов по модулю p и $x_0 t_1, \ldots, x_0 t_n$ представляют собой n различных решений сравнения (1). Степень этого сравнения равна n, следовательно, $x_0 t_1, \ldots, x_0 t_n$ охватывают все решения.

Важным частным случаем степенных сравнений являются сравнения второй степени. Из доказанных выше теорем вытекают следующие следствия, относящиеся к случаю $n = 2$.

Следствие 3

Необходимым и достаточным условием того, чтобы число a было квадратичным вычетом по простому модулю $p>2$ ($p \nmid a$), является выполнимость сравнения

$$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

Следствие 4

Если a является квадратичным вычетом по простому модулю $p > 2$ ($p \nmid a$), то сравнение $x^2 \equiv a \pmod{p}$ имеет два решения.

Следствие 5

По любому простому модулю $p > 2$ ($p \nmid a$) число классов квадратичных вычетов и число классов квадратичных невычетов равно $\frac{p-1}{2}$.

Для сравнений $x^2 \equiv a \pmod{p}$ имеет место следующее утверждение.

Теорема 18

Необходимым и достаточным условием того, чтобы число a было квадратичным невычетом по простому модулю $p > 2$ ($p \nmid a$), является выполнимость сравнения

$$a^{\frac{p-1}{2}} \equiv -1 \pmod{p}.$$

Доказательство

В силу малой теоремы Ферма имеем $a^{p-1} - 1 \equiv 0 \pmod{p}$. Поскольку $2 \mid p - 1$, то можем записать $(a^{\frac{p-1}{2}} - 1)(a^{\frac{p-1}{2}} + 1) \equiv 0 \pmod{p}$. Последнее сравнение выполняется, когда имеет место либо $a^{\frac{p-1}{2}} - 1 \equiv 0 \pmod{p}$, либо $a^{\frac{p-1}{2}} + 1 \equiv 0 \pmod{p}$. Первое из последних двух сравнений выполняется, когда a является квадратичным вычетом. Если a является квадратичным невычетом, то имеем $a^{\frac{p-1}{2}} + 1 \equiv 0 \pmod{p}$.

ГЛАВА 3

Двухключевые криптосистемы

3.1. Понятие электронной цифровой подписи

Основываясь на идее использования односторонней функции с потайным ходом, Диффи и Хеллман предложили структуру криптосистемы с открытыми ключами для сети с многими абонентами. Каждый абонент, например *i*-й, случайным образом выбирает некоторое значение параметра z_i и держит его в секрете. Далее он формирует алгоритм $\mathbf{E}_{z_i}$ и предоставляет его для опубликования в открытом справочнике. Он также формирует алгоритм $\mathbf{D}_{z_i}$ и держит его в секрете. Любой другой абонент, например *j*-й, может послать *i*-му абоненту секретное сообщение *M*. Для этого он использует открытый алгоритм шифрования $\mathbf{E}_{z_i}$ и вычисляет криптограмму $C = f_{z_i}(M)$, которую посылает *i*-му абоненту. Используя секретный алгоритм $\mathbf{D}_{z_i}$, абонент *i* вычисляет исходный открытый текст: $f_{z_i}^{-1}(C) = M$.

Эта обобщенная схема открытого шифрования может быть применена для получения *цифровых подписей*. В общем случае цифровая подпись представляет собой некоторое число со специфической структурой, которое допускает проверку с помощью открытого ключа того факта, что оно было выработано для некоторого сообщения с использованием секретного ключа. Для реализации цифровой подписи необходимо выбрать такую одностороннюю функцию с потайным ходом (с секретом) f_z, для которой при всех значениях параметра *z* область определения функции f_z совпадает с областью ее значений. При этом условии для любого сообщения, которое может быть представлено в виде числа из области определения функции $f_z(x)$, абонент *i* может сформировать с помощью секретного алгоритма число $S = f_{z_i}^{-1}(M)$. Если сообщение слишком велико, то его можно разбить на части необходимого размера и каждую из них подписать независимо.

Каждый пользователь криптосистемы может по значению *S* восстановить сообщение *M*, используя открытый алгоритм шифрования $\mathbf{E}_{z_i}$. Если *M* пред-

ставляет собой осмысленное сообщение или может быть сопоставлено с таковым по некоторому заранее оговоренному правилу, то значение S может рассматриваться как цифровая подпись абонента i под сообщением M. Реально только владелец секретного алгоритма $\mathbf{D}_{z_i}$ мог выработать «открытый» текст S, который с помощью алгоритма $\mathbf{E}_{z_i}$ зашифровывается в осмысленную криптограмму M, поскольку лишь абонент i знает способ вычисления $f_{z_i}^{-1}$.

Абонент i может послать абоненту j также секретное сообщение с подписью. Для этого он зашифровывает S с помощью открытого алгоритма $\mathbf{E}_{z_j}$, получая криптограмму $C = \mathbf{E}_{z_j}(S)$. Получив зашифрованное сообщение, j-й абонент расшифровывает его своим секретным алгоритмом $\mathbf{D}_{z_j}(C) = S$, затем число S зашифровывает открытым алгоритмом i-го абонента $\mathbf{E}_{z_i}(S) = M$. Таким образом, по полученной криптограмме C абонент j восстанавливает подпись абонента i и исходное сообщение.

Использование протоколов на основе симметричных криптографических методов предполагает, что две стороны *доверяют друг другу*. Криптосистемы с открытым ключом (асимметричные криптосистемы) позволяют реализовать протоколы взаимодействия сторон, которые *не доверяют друг другу*. Системы электронной цифровой подписи являются одним из важнейших примеров такого типа. Для того чтобы цифровая подпись могла бы эффективно использоваться в реальных деловых взаимоотношениях абонентов, следует придать ей юридическую силу. Для этого необходимо принять соответствующие правовые законы на государственном (или международном) уровне, а обмен открытыми ключами подкреплять обычной юридической процедурой, чтобы обеспечить защиту от отказов от открытых ключей.

Виды нападений на электронную цифровую подпись

В системах электронной цифровой подписи (ЭЦП) используются три криптографических алгоритма:

- алгоритм формирования подписи по секретному ключу;
- алгоритм проверки подписи по открытому ключу;
- алгоритм вычисления хэш-функции от подписываемого сообщения.

К математическим основам цифровой подписи можно отнести также алгоритмы формирования секретного и открытого ключей. Функционирование реальных систем требует также правового, организационного и программно-технического обеспечения. Правовое обеспечение включает принятие законов, придающих юридическую силу цифровой подписи. Организационное

обеспечение включает регистрацию пользователей в некотором доверительном центре, оформление документов между пользователем и доверительным центром (или между двумя пользователями) об ответственности за переданные друг другу открытые ключи. К программно-техническому обеспечению относится вся совокупность программных и аппаратных средств, позволяющих осуществить сложные вычисления и обеспечивающих сохранность базы данных, содержащей подписанные документы и образцы подписей к ним.

Возможные виды нападений на цифровую подпись можно классифицировать, разбивая их на следующие группы:

- нападения на криптографические алгоритмы;
- нападения, связанные с нарушением протокола;
- нападения, основанные на нарушении целостности механизмов системы цифровой подписи.

Нападающим может оказаться внешний субъект, а также подписывающая сторона (отказ от подписи) или сторона, проверяющая подпись (навязывание ложной подписи).

Нападения на криптографические алгоритмы связаны с решением сложных математических задач, например дискретного логарифмирования по модулю большого простого числа. Шансы на успех у нападающего крайне низки. Такой вид нападений может быть предпринят против двухключевого криптографического алгоритма или против хэш-функции. В первом случае подделывается подпись, во втором — документ. К нападениям, связанным с нарушением протокола, относятся, например, повторения подписанных сообщений, задерживание сообщений. Чтобы предотвратить такие действия, в документе предусматриваются специальные поля, в которых указывается дата и номер документа. Необходимо использовать также механизмы, обеспечивающие защиту от фактов отказа о получении сообщений.

Атаки, основанные на нарушении целостности механизмов системы цифровой подписи, являются наиболее многообразными. К ним относятся удаление из базы данных подписанного сообщения, перехват секретного ключа программными или аппаратными средствами, навязывание ложного открытого ключа, подмена открытого ключа в базе данных. Эти примеры показывают, что широкий спектр нападений связан с несанкционированным доступом к данным, циркулирующим в системе цифровой подписи. Безопасное функционирование системы цифровой подписи требует использования защищенного окружения.

Нападения на криптосистему могут основываться также на навязывании пользователям системы цифровой подписи или средств защиты от несанкционированного доступа с замаскированными слабостями, а также на навя-

зывании другого системного и прикладного программного обеспечения с встроенными недокументированными закладками. Чтобы не допустить этого, криптографические средства и средства защиты от несанкционированного доступа подлежат сертификированию специальными организациями.

Имеется еще один вариант атак, связанный с используемыми программами для создания документов. По умолчанию предполагается, что подписывающий и проверяющий пользуются одинаковыми программами для создания и чтения документов. Если эти программы отличаются, то подписывающий и проверяющий «видят» разные документы. Подписывающий считает, что он подписал один документ, а проверяющий убежден, что подписан другой документ. Нарушитель может попытаться модифицировать у подписывающего или у проверяющего (или у них обоих) указанную программу и реализовать те или иные незаконные цели. Этот аспект явно показывает, что для интерпретации содержимого документа должна использоваться стандартная программа, оговоренная в протоколе ЭЦП, а при формировании подписи подписывающий должен быть убежден, что он ознакомился с документом, раскрытым правильной программой. Если возникают малейшие сомнения, то подписывающий должен проверить контрольную сумму файлов, содержащих программу редактирования, таблицы кодировок и другие данные, с которыми работает программа редактирования документов.

3.2. 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 сущность одноключевой криптографии.

Главным требованием, предъявляемым к симметричным шифрам, является их стойкость. Абсолютно (безусловно) стойкими криптосистемами являются шифры с ключом однократного использования [38]. 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 [2, 6]. 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 электронной цифровой 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 бы хотя бы еще один участник протокола знал 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 электронной цифровой 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.

3.3. От открытого распределения ключей до электронной цифровой подписи

3.3.1. Система распределения 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 [22] была показана возможность построения практически стойких секретных систем, которые *не требуют передачи секретного ключа*. Предложенная ими система получила название метода открытого распределения ключей. В ней каждый абонент выбирает случайным образом секретный ключ x и вырабатывает от-

крытый 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 1024 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 известным им 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, 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 нарушителю удастся подменить открытый ключ какого-либо абонента, 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.

3.3.2. Открытый шифр Эль-Гамаля

В рассмотренной выше двухключевой криптосистеме осуществляется открытое распределение 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 [23]. Чтобы послать пользователю i секретное сообщение T, отправитель может действовать 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 $Z = y^{x'} \pmod p$, 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: $C = Z\,T \pmod p$.

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-й 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 два раза по сравнению с исходным текстом. Легко показать, что i-й пользователь, получив криптограмму (y', C) и используя свой секретный ключ x, 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.

3.4. Системы ЭЦП на основе задачи дискретного логарифмирования

3.4.1. 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 удалось бы для данного значения h вычислить подпись s, то мы бы уже имели 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 подписи потребуется решить за-

дачу 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:

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

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 станет известным 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 $s = (h-k)/x \pmod{p-1}$. 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*.

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 возможности сформировать подпись к другому. 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 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 использование 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, связанной с решением проблемы дискретного логарифмирования.

Данная пара уравнений может использоваться в качестве системы ЭЦП, хотя некоторым недостатком последней является указанное ограничение (НОД(x, $p-1$) = 1) на выбор секретного ключа.

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

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

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 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 параметр r вычисляется по формуле $r = \alpha^k y^{-s} \pmod{p}$, где значения k и s выбираются произвольно, поэтому 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 обстоятельст-

во 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$(k, p-1) = 1$. Тогда существует и можно найти число, обратное к k, а затем вычислить s. Уравнение поверки подписи следует составлять так, чтобы в соответствующем ему уравнении формирования подписи значение хэш-функции не находилось в знаменателе, поскольку оно является случайным и в общем случае может содержать отличные от 1 общие множители с $p-1$.

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:

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

Однако это не снимает перечисленных замечаний 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 использовании 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 обеих 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 односторонней 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 $F''(h)$ не вносит принципиальных изменений, 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 автоматически не приведет 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:

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

3.4.2. 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.

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 $p-1$, 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.

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)**
$\alpha^a = y^b r^c \pmod p$	$a = bx + ck \pmod{p-1}$	$a = bx + ck \bmod q$
$y^a = \alpha^b r^c \pmod p$	$ax = b + ck \pmod{p-1}$	$ax = b + ck \bmod q$
$r^a = \alpha^b y^c \pmod p$	$ak = b + cx \pmod{p-1}$	$ak = b + cx \bmod q$
$r^a \alpha^b = y^c \pmod p$	$ak + b = cx \pmod{p-1}$	$ak + b = cx \bmod q$
$r^a \alpha^b y^c = 1 \pmod p$	$ak + b + cx = 1 \pmod{p-1}$	$ak + b + cx = 1 \bmod q$

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 повышает сложность задачи дискретного логарифмирования по модулю p. Существуют различные способы формирования простых чисел, удовлетворяющих этим условиям.

В табл. 3.2, составленной по данным [10], 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 может опираться на тот факт, что если правая и левая части уравнения проверки подписи сравнимы по модулю p, то остатки от деления левой и правой частей на p будут сравнимы по любому другому модулю. Осуществляется это следующим образом. Выбирается уравнение проверки подписи типа $R^a = \alpha^b y^c \pmod{p}$ и преобразуется к эквивалентному виду

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

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

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

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

В последней формуле следует задать зависимость пары чисел u и v от хэш-функции подписываемого документа h и подписи (r, s), где $r = ((R \bmod p) \bmod q)$. 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 1990-х гг.

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 варианты 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 системы цифровой подписи RSA увеличение размера модуля приводит к увеличению размера подписи.

3.4.3. 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 $r = \alpha^{sr} y^{h} \pmod p$, видим, что правая и левая части совпадают, т. е. сформированная подпись является правильной.

Пример 3.2. Уравнение подписи вида $\alpha^{r} = y^{hs} r^{s} \pmod p$. 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 $r = \alpha^{z} y^{-h} \pmod p$.
3. Вычислить $s = r/z \pmod{p-1}$.
4. Предъявить в качестве подписи к хэш-функции *h* пару чисел (r, s).

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

Пример 3.3. Уравнение подписи вида $\alpha^{hs} = y^{r} r^{s} \pmod p$. 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 $r = \alpha^{h} y^{-w} \pmod p$.
3. Вычислить $s = r/w \pmod{p-1}$, т. е. $r = sw \pmod{p-1}$.
4. Предъявить в качестве подписи к хэш-функции *h* пару чисел (r, s).

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

Пример 3.4. Уравнение подписи вида $\alpha^{hs} = y^{rf(h)} r^{s} \pmod p$, 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 $r = \alpha^{h} y^{-w} \pmod p$.
3. Вычислить $s = rf(h)/w \pmod{p-1}$.
4. Предъявить в качестве подписи к хэш-функции *h* пару чисел (r, s).

Подставляя в правую часть уравнения проверки подписи значение произведения $rf(h) = ws \pmod{p-1}$ и $r = \alpha^{h} y^{-w} \pmod p$, получаем $y^{ws} \alpha^{hs} y^{-ws} =$

$\alpha^{hs} \pmod p$. Совпадение левой и правой частей уравнения проверки показывает, что сформированная подпись (r, s) является правильной.

Пример 3.5. Уравнение подписи вида $r^{f(h)} = \alpha^h y^{rs} \pmod p$, 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 $z/w = h/f(h) \pmod{p-1}$.
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

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

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

3.4.4. Цифровая подпись Эль-Гамаля

Рассмотрим систему цифровой подписи [23], носящую имя ее изобретателя Эль-Гамаля и явившуюся отправной точкой развития многочисленных систем ЭЦП, основанных на сложности задачи дискретного логарифмирования. Пусть мы имеем большое простое число p, такое что разложение числа $p-1$ содержит, по крайней мере, один большой простой множитель, и первообразный корень α по модулю p.

Механизм подписывания состоит в следующем. Некоторый абонент **A** выбирает секретный ключ x_A, по которому формирует открытый ключ $y_A = \alpha^{x_A}$. Подписью абонента **A** под документом M (подписываемое сообщение должно иметь длину меньше простого модуля p, т. е. $M<p$) 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 служит для проверки того факта, что документ подписан абонентом A. (Значение $y_A = \alpha^{x_A}$ является открытым ключом абонента **A** и доступно для всех пользователей, что дает каждому возможность убедиться в том, что данное сообщение действительно подписано абонентом **A**.)

Данная система электронной цифровой подписи основана на том, что только действительный владелец секретного ключа x_A может выработать пару чисел (r, s), удовлетворяющую уравнению проверки подписи. Используя значение x_A, абонент **A** вырабатывает подпись 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 $r = \alpha^k \pmod p$, а именно: $\alpha^M = \alpha^{x_A r} \alpha^{ks} = y_A^r r^s \pmod p$.

Из двух последних формул видно, что владелец секретного ключа в состоянии подписать документ, а его подпись может быть проверена по его открытому ключу. Нахождение пары чисел (r, s) без знания секретного ключа является вычислительно сложным. Различных подписей, соответствующих данному документу, может быть чрезвычайно много (заметим, что k может иметь разные значения), но выработать правильную подпись способен только владелец секретного ключа. Множество возможных подписей отличаются значением r, но для данного r найти соответствующее значение s без знания секретного ключа практически невозможно. Для вычисления секретного ключа по открытому ключу необходимо решить задачу дискретного логарифмирования, которая является вычислительно сложной.

Особенностью системы цифровой подписи Эль-Гамаля является генерация случайного числа k. В этой криптосистеме не допускается использовать одно и то же значение k для формирования подписи к двум разным сообщениям. Это связано с тем, что на основе двух разных подписей, сформированных при одном и том же значении k, имеется возможность вычислить секретный ключ. Кроме того, использованные при выработке значения k подлежат гарантированному уничтожению. Если нарушитель сможет получить значение k, то он также сможет вычислить секретный ключ. В реально используемых системах реализуется случайная генерация числа k большого размера и некоторый механизм его уничтожения после выработки подписи. При программной реализации обеспечивается такая схема формирования подписи, что число k появляется только в регистрах микропроцессора и оперативной памяти, а механизм уничтожения состоит в записи нового случайного значения в ячейки памяти, которые до этого содержали значение k.

Выше мы рассмотрели механизмы двухключевой криптографии, которые дают возможность подписывать сообщения ограниченной длины (порядка 10^3 бит). Если сообщение имеет большой размер, то при прямолинейном использовании таких механизмов потребуется разбить исходное сообщение на большое число блоков меньшего размера и выработать столько подписей, сколько блоков будет содержаться в сообщении. Это сильно усложняет проблему хранения подписей и самих подписанных сообщений, а также организации баз данных, содержащих большое число подписанных документов. Для упрощения этой проблемы подписывается не сам документ, а некоторый его цифровой образ небольшого размера, полученный по специальным криптографическим процедурам, называемым *хэшированием*.

Алгоритм хэширования должен быть таким, чтобы обеспечить вычислительную неосуществимость нахождения двух сообщений с одинаковым значением цифрового образа (значением хэш-функции от сообщения). В настоящее время разработаны алгоритмы, которые удовлетворяют этому условию и позволяют легко вычислить значение хэш-функции от данного документа. Вместо подписывания большого числа отдельных частей электронного документа в реальных системах цифровой подписи осуществляется вычисление хэш-функции от документа и подписывание хэш-функции. Если хэш-функция подписана, то документ считается подписанным.

3.4.5. Системы ЭЦП с дополнительными свойствами

Представляет интерес построить такую систему ЭЦП, в который было бы вычислительно сложно 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 (вместо h), 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: $M = m \mid v$.
- 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' подпи-

сано 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 s 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 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.

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 варианты сокращения размера элементов подписи r и s и уравнение проверки подписи с восстановлением сообщения могут быть также реализованы.

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***
$s + xr'f(h) + k = 0 \pmod q$	$m = \alpha^{sy} r'^{f(h)r} \pmod p$	(r, s, h)
$s + xf(h, r) + k = 0 \pmod q$	$m = \alpha^{sy} f(h,r)^{r} \pmod p$	(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)
$r'f(h) + xs + k = 0 \pmod q$	$m = \alpha^{r'f(h)y} s^{r} \pmod p$	(r, s, h)

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 стойкость 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(α, p) $\neq 1$ или НОД(α, 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, представленным 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)

3.4.6. Стандарты ЭЦП

Американский стандарт DSS

Стандарт DSS (Digital Signature Scheme) относится к схемам ЭЦП, основанным на сложности задачи дискретного логарифмирования, с сокращенной длиной подписи. В этом стандарте используются следующие параметры: p — простое число, $512 \le p \le 1024$; q — простой делитель числа $p-1$, $2^{159} \le q \le 2^{160}$; α — число, относящееся к показателю q по модулю p.

Секретный ключ представляет собой случайно генерируемое число x, $1 < x < q$.

Формирование открытого ключа осуществляется путем возведения числа α в степень x по модулю p: $y = \alpha^x \bmod p$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k, $1 < k < q$.
2. Вычисляется значение $r = (\alpha^k \bmod p) \bmod q$, являющееся первой частью подписи.
3. Вычисляется хэш-функция H от подписываемого сообщения: $E = H(m)$. (Хэш-функция также специфицируется стандартом, который предписывает использование алгоритма хэширования SHA-1.)
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$s = \frac{E + xr}{k} \bmod q .$$

Таким образом, формирование подписи (r, s) завершено. Значение $s = 0$ не допускается стандартом. В этом случае выполняется повторное формирование подписи при новых значениях k до тех пор, пока не будет получено $s \neq 0$.

Проверка подлинности подписи осуществляется следующим образом.

1. Поверяются соотношения $r < q$ и $s < q$. Если они не выполняются, то подпись признается недействительной.
2. Вычисляется значение $w = s^{-1} \bmod q$.
3. В соответствии с алгоритмом SHA-1 вычисляется значение хэш-функции $E = H(m)$.
4. Вычисляются значения $u = Ew \bmod q$ и $v = rw \bmod q$.
5. Вычисляется значение $r' = (\alpha^u y^v \bmod p) \bmod q$.

6. Сравниваются значения r и r'. Если $r = r'$, то результат проверки подлинности подписи признается положительным.

Российский стандарт ГОСТ Р 34.10–94

Российский стандарт ГОСТ Р 34.10–94 во многом сходен со схемой DSS. В нем используются следующие параметры: p — простое число, причем $510 \le p \le 512$ либо $1022 \le q \le 1024$ для более ответственных случаев; q — простой делитель числа $p-1$, $2^{255} \le q \le 2^{256}$; α — число, относящееся к показателю q по модулю p.

Секретный ключ представляет собой случайно генерируемое число x, $1 < x < q$.

Формирование открытого ключа осуществляется путем возведения числа α в степень x по модулю p: $y = \alpha^x \bmod p$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k, $1 < k < q$.
2. Вычисляется значение $r = (\alpha^k \bmod p) \bmod q$, являющееся первой частью подписи.
3. Вычисляется хэш-функция H от подписываемого сообщения: $E = H(m)$. (Хэш-функция специфицируется стандартом ГОСТ Р 34.11–94.)
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$s = kE + xr \bmod q.$$

Значение параметра $s = 0$ в подписи (r, s) не допускается стандартом.

Проверка подлинности подписи осуществляется следующим образом.

1. Проверяются соотношения $r < q$ и $s < q$. Если они не выполняются, то подпись признается недействительной.
2. В соответствии с алгоритмом ГОСТ Р 34.11–94 вычисляется значение хэш-функции $E = H(m)$.
3. Вычисляется значение $w = E^{-1} \bmod q$, если $E \ne 0$, и $w = 1$, если $E = 0$.
4. Вычисляются значения $u = sw \bmod q$ и $v = -rw \bmod q$.
5. Вычисляется значение $r' = (\alpha^u y^v \bmod p) \bmod q$.
6. Сравниваются значения r и r'. Если $r = r'$, то результат проверки подлинности подписи признается положительным.

В стандарте ГОСТ Р 34.10–94 специфицирована процедура генерации простых чисел p и q.

3.5. 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 со 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 вооб-

ще 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», названный по имени своего изобретателя трехпроходным протоколом Шамира. На самом деле применяются даже два ключа шифрования, но ни один из них не передается ни по какому каналу, т. е. они используются локально каждым из взаимодействующих абонентов. В этом протоколе используется коммутативный 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 применяются 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. Не представляет никакого труда 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 возведение в различную степень. Пусть зашифрование сообщения $m < p$ состоит в возведении 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 упомянутому выше 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. Например, нарушитель по значениям c_2 и $c_3 = c_2^{d_A} \pmod p$ может попытаться вычислить d_A и восстановить сообщение $m = c_1^{d_A} \pmod p$. Однако для этого ему придется решить задачу дискретного логарифмирования, что является вычислительно неосуществимым при правильном выборе простого числа p (разложение числа $p-1$ должно содержать, по крайней мере, один большой простой множитель).

Таким образом, 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 $c_2 = c_1^{e_B} = m^{e_A e_B} \pmod p$ и отправляет c_2 передатчику. (Этот шаг соответствует дополнительному запиранию кейса на второй замок.)

3. Передатчик, получив значение c_2, вычисляет $c_3 = c_2^{d_A} = m^{e_A e_B d_A} = m^{e_B} \pmod p$ и отправляет c_3 приемнику. (Этот шаг соответствует открытию первого замка. Кейс с сообщением остался закрытым только на замок, установленный приемником сообщения.)

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–Хеллмана. 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 ключей — открытого и секретного, является 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, взаимно обратное значению первого ключа по модулю $p-1$. По-

следнее 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, по-

скольку 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 взаимно обратных 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) [7]. 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.

3.6. Криптосистема RSA

3.6.1. Криптографические преобразования в RSA

Система RSA [7, 30, 31] является наиболее широко известной системой цифровой подписи. При этом она является и наиболее простой для понимания. Рассмотрим эту криптосистему. Согласно известной из теории чисел теореме Эйлера для любых взаимно простых целых чисел M и n, где $M<n$, выполняется соотношение $M^{\varphi(n)} = 1 (\mathrm{mod}\ n)$.

В качестве числа M будем брать исходное сообщение, которое необходимо подписать или зашифровать. Условие взаимной простоты чисел M и n обеспечим тем, что будем выбирать число n, равное произведению двух больших простых множителей p и q. В этом случае вероятность того, что случайное сообщение не будет взаимно простым с модулем, является пренебрежимо малой. (Ниже будет показано, что даже в случае, когда либо p, либо q делят число M, система RSA также работает корректно.) В качестве одностороннего преобразования будем использовать возведение в степень по модулю n. При некотором значении этой степени e будем иметь функцию шифрования **E**, которая преобразует исходное сообщение M в криптограмму $C = \mathbf{E}(M) = M^e (\mathrm{mod}\ n)$.

Параметр e будем полагать общедоступным (открытым, публичным). По известному значению S при известных n и e вычислительно сложно найти M. В качестве потайного хода соответствующей односторонней функции зашифрования $M^e (\mathrm{mod}\ n)$ будем использовать также возведение в степень, но с другим значением степени. Новое значение степени d необходимо выбрать таким, чтобы функция расшифрования $\mathbf{D}(C) = C^d (\mathrm{mod}\ n)$ была обратной по отношению к $\mathbf{E}(M) = M^e (\mathrm{mod}\ n)$, т. е. должно выполняться условие $M = \mathbf{D}[\mathbf{E}(M)] = (M^e)^d = M^{ed} (\mathrm{mod}\ n)$.

Из этого соотношения следует: $ed = 1 (\mathrm{mod}\ \varphi(n))$. Таким образом, две процедуры возведения в степень по модулю n будут взаимно обратными, если произведение степеней равно 1 по модулю функции Эйлера от числа n. Параметр d является ключом к потайному ходу, поэтому он считается секретным. Теперь задача состоит в выборе необходимых значений степеней e и d. Очевидно, что первоначально необходимо установить значение функции

Эйлера от модуля n. Легко заметить, что для любого простого числа p имеем $\varphi(p) = p - 1$. Поскольку мы выбрали $n = pq$, где оба множителя являются простыми числами, то, используя свойство мультипликативности функции Эйлера, получаем: $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$.

Еще Евклиду было известно, что если целые числа e и m удовлетворяют условиям $0 < e < m$ и НОД(m, e) = 1, то существует единственное d, удовлетворяющее условиям $0 < d < m$ и $de = 1 \pmod m$, и, кроме того, d может быть вычислено с помощью расширенного алгоритма Евклида.

Приходим к следующей схеме функционирования криптосистемы RSA.

1. Каждый пользователь выбирает два больших не равных между собой простых числа p и q, находит их произведение $n = pq$ и вычисляет значение $\varphi(n) = (p-1)(q-1)$. Одним из требований к выбору p и q является то, что, по крайней мере, одно из чисел $(p-1)$ или $(q-1)$ должно иметь один большой простой множитель. Размер модуля n должен быть не менее 512 бит. Для ответственных применений системы RSA рекомендуемая длина модуля составляет от 1024 бит.
2. Затем выбирается целое число e, такое что $e < \varphi(n)$ и НОД(e, $\varphi(n)$) = 1, и вычисляется d, удовлетворяющее условию $ed = 1 \pmod{\varphi(n)}$.
3. *Секретным ключом* является тройка чисел p, q, d, которая держится в секрете. На самом деле хранить в секрете достаточно только число d, поскольку простые числа p и q нужны только на этапе формирования модуля n и вычисления d. После этого числа p и q могут быть уничтожены.
4. Пара чисел n, e является *открытым ключом*, который предоставляется всем абонентам криптосистемы RSA.
5. *Процедура подписывания* сообщения M — это возведение числа M в степень d по mod n, а именно: $S = M^d \bmod n$. Число S есть *цифровая подпись*, которую может выработать только владелец секретного ключа.
6. *Процедура проверки* подписи S, соответствующей сообщению M, — это возведение числа S в целую степень e по mod n, а именно: $M' = S^e \bmod n$.

Если $M' = M$, то сообщение M признается подписанным пользователем, который предоставил ранее открытый ключ e. Видно, что

$$S^e = (M^d)^e = M^{de} = M^{Q\varphi(n)+1} = M^{Q\varphi(n)}M = (M^{\varphi(n)})^Q M = 1^Q M \pmod n,$$

т. е. сформировать подпись, соответствующую данному открытому ключу и данному сообщению, можно только по известному секретному ключу d.

В RSA открытый ключ используется для шифрования сообщений в соответствии с формулой: $C = M^e \bmod n$. Криптограмму C может расшифровать только владелец секретного ключа, выполнив преобразование в соответствии с выражением $M = C^d \bmod n$. (Правда, у получателя криптограммы может возникнуть вопрос о подлинности отправителя, хотя он уверен, что криптограмму прочитать никто другой не может.) Если сообщение имеет большой размер, то оно разбивается на блоки, длина которых меньше длины модуля n, и каждый из этих блоков подписывается отдельно. Процедура расшифрования совпадает с процедурой генерации подписи, а процедура зашифрования — с процедурой проверки подписи.

Стойкость криптосистемы RSA основана на сложности разложения модуля на два больших простых множителя. Если задачу такого разложения удалось бы решить, то тогда можно было бы легко вычислить функцию Эйлера от модуля и затем, используя расширенный алгоритм Евклида, определить секретный ключ по открытому.

До настоящего времени не найдены практически реализуемые общие способы решения этой задачи при длине модуля 512 бит и более. Однако для частных видов простых чисел p и q сложность этой задачи резко понижается, поэтому в криптосистеме RSA необходимо выполнить ряд специальных тестов при генерации секретного ключа. Другой особенностью системы RSA является ее мультипликативность: $\mathbf{E}(M_1, M_2) = \mathbf{E}(M_1)\mathbf{E}(M_2) \pmod n$, что позволяет нарушителю на основе двух подписанных сообщений сформировать подпись третьего сообщения, которое равно $M_3 = M_1 M_2 \pmod n$. Поскольку M_3 в подавляющем большинстве случаев не будет представлять собой какого-либо осмысленного текста, то наличие этой особенности не является существенным недостатком. В системе RSA необходимо учитывать также следующую возможность. Выбрав произвольное значение S, можно вычислить значение $M' = S^e$, т. е. произвольное значение можно представить как подпись некоторого сообщения. Такие фиктивные сообщения, естественно, являются случайными. Однако в ряде применений имеют место случаи, когда требуется подписывать случайные сообщения. В таких случаях можно использовать следующую схему.

1. К сообщению T, которое необходимо подписать и передать по открытому каналу, присоединяется заранее условленный двоичный вектор V длиной $v = 64$ бит:

$$M \rightarrow T \mid V.$$

2. Вырабатывается подпись для сообщения M:

$$S = M^d \pmod n.$$

3. Значение S направляется приемной стороне.
4. Приемная сторона по значению S вычисляет значения:

$$M' = S^e \pmod n,\ V' = M' \pmod{2^v} \text{ и } T' = M' \operatorname{div} 2^v.$$

5. Если V' равно заранее условленному значению V, т. е. если выполняется условие $V' = V$, то принимается решение, что сообщение T' подписано владельцем секретного ключа, соответствующего открытому ключу, с помощью которого выполнялась проверка подписи. (Вероятность того, что случайное сообщение может быть принято за подписанное, равна 2^{-v}.)

Более практичным представляется формирование подписи по значению хэш-функции от сообщения (даже если сообщение имеет малую длину). Если нарушитель попытается представить некоторое значение хэш-функции, которое якобы подписано, то он должен будет представить также и сообщение, от которого получено данное значение хэш-функции. Для криптографически стойкой хэш-функции нарушитель не сможет подобрать такое сообщение. Практичность этого способа состоит в том, что для сообщений большого размера именно таким способом и вычисляется подпись. Здесь только расширяется применение хэш-функций и на случай подписывания сообщений малого размера.

Таким образом, секретный ключ служит для подписывания сообщений, открытый ключ — для проверки подписи. Для того чтобы послать секретное сообщение некоторому абоненту А, любой пользователь может воспользоваться открытым ключом абонента А и сформировать криптограмму $C = \mathbf{E}_A(M)$. По значению C восстановить сообщение M может только абонент А, который знает секретный ключ, соответствующий открытому ключу, с помощью которого была сформирована криптограмма. В криптосистеме RSA генерация подписи совпадает с процедурой расшифрования, а проверка подписи — с процедурой зашифрования.

Рассмотрим вопрос о корректности RSA-преобразования в случае, когда либо p, либо q делят число M (это случаи, когда M не является взаимно простым с модулем, т. е. когда теорему Эйлера применить нельзя). Для определенности допустим, что число p делит M. Это означает, что $M \equiv 0 \bmod p$, т. е. имеем:

$$M^{ed} \equiv M \bmod p.$$

Поскольку p делит M, то НОД(q, M) = 1, поэтому, по теореме Ферма, имеем:

$$M^{q-1} \equiv 1 \bmod q \Rightarrow M^{(q-1)(p-1)} \equiv 1 \bmod q \Rightarrow M^{k\varphi(n)+1} = M^{ed} \equiv M \bmod q.$$

Согласно теореме 12 (*см. раздел 2.7*), имеем $M^{ed} \equiv M \bmod n$. Таким образом, с учетом изложенного выше корректность RSA-преобразования доказана для всех возможных значений $M < n$.

3.6.2. Вопросы выбора параметров системы RSA

Выбор множителей p и q

Для формирования модуля n рекомендуется генерировать простые множители p и q, удовлетворяющие следующим требованиям:

1. Длина чисел p и q должна быть примерно одинаковой, т. е. $|p| \approx |q|$.
2. Числа p и q должны быть существенно различными, т. е. абсолютная величина разности $p - q$ должна быть большой. Это предотвращает атаки, основанные на разложении модуля методом Ферма, в котором используется перебор делителей, начиная с целого числа, ближайшего к значению $n^{1/2}$.
3. Числа p и q должны быть сильно простыми.

Число p называется сильно простым, если разложение чисел $p-1$ и $p+1$ содержит большой простой множитель r и s, соответственно. Причем разложение числа $r-1$ также содержит большой делитель.

Выбор длины секретных ключей e и d

Длина открытого e и закрытого d ключа влияет на сложность операции проверки и формирования подписи, соответственно. Однако в силу условия $ed = 1 \bmod \varphi(n)$ суммарная длина этих ключей не может быть меньше $|\varphi(n)| \approx |n|$. Для обеспечения высокой стойкости разумно выбрать достаточно большую длину этих ключей. Тогда нарушитель за разумное время не сможет с существенной вероятностью найти значение секретного ключа, используя атаку, основанную на переборе значений секретного ключа. Это могло бы случиться, если бы длина секретного ключа составляла 64 бит или менее. Выбор секретного ключа менее четверти длины модуля n считается небезопасным ввиду некоторых других атак.

Выбор сравнительно малой длины открытого ключа не является критичным. Более того, выбор малой длины открытого ключа является разумным в плане уменьшения объема справочника открытых ключей, копии которого хранятся у большого числа пользователей. При этом существенно уменьшается время проверки подписи (в 5 раз и более по сравнению со случаем большой длины ключа e). Причем это практически не влияет на время подпи-

сывания, поскольку вычисляемая по e секретная экспонента d имеет обычно размер $|d| \approx \varphi(n)$ независимо от выбранного размера открытого ключа.

Следует иметь в виду, что при реализации RSA может быть встроен потайной люк, использующий следующую особенность этой криптосистемы. Осуществим многократное шифрование некоторого сообщения M:

$$(\ldots((M^e)^e)^e\ldots)^e = M^{e^k} \bmod n.$$

Для любого значения e существует такое k, что будем иметь $M^{e^k} \bmod n = M$. Если k невелико, то можно расшифровывать зашифрованные сообщения или подписывать документы без знания секретного ключа. Предварительно без детального рассмотрения проблемы можно предположить, что при случайном формировании секретного и открытого ключей значение k с большой вероятностью настолько велико, что рассматриваемая атака вычислительно нереализуема. Однако в отдельных случаях ключ e может относиться к достаточно малому значению k, и рассматриваемая атака окажется вполне эффективной. Чтобы получить более полную картину по данной ситуации, учтем, что фактически рассматривается сравнение $e^k \equiv 1 \bmod \varphi(n)$. Это означает, что минимальное k является показателем числа e по модулю $\varphi(\varphi(n))$. Из этого замечания, учитывая, что показателями могут быть только делители $\varphi(\varphi(n))$, можно сделать следующие выводы:

1. Если в разложении $\varphi(\varphi(n))$ имеются множители небольшой длины, то имеется существенная вероятность случайного выбора «плохого» открытого ключа. При использовании случайного выбора открытого ключа e модуль n желательно выбрать таким, чтобы в разложении $\varphi(\varphi(n))$ имелось сравнительно малое число делителей небольшой длины. Таким способом можно задать пренебрежимо малую вероятность случайного попадания на плохой открытый ключ.
2. Попадание на плохой открытый ключ можно устранить полностью, если после случайного выбора e осуществить проверку, к какому показателю по модулю $\varphi(n)$ относится число e. Для этого нужно знать разложение $\varphi(\varphi(n))$. О последнем нужно позаботиться при формировании модуля n, чтобы устранить задачу разложения $\varphi(n)$ на простые множители.
3. Если открытый ключ e выбран «хорошим», то «хорошим» является и секретный ключ d, так как два взаимно обратных числа относятся к одному и тому же показателю: $(1)^k = (ed)^k = e^k d^k = d^k = 1 \bmod \varphi(n)$, где в качестве k берется меньший из показателей чисел e и d (вначале предполагается, что эти показатели не равны).

4. При программной реализации шифра RSA существует возможность встроить потайной ход, задавая случайную генерацию модуля n «от значения» $\varphi(\varphi(n))$, разложение которого содержит один или несколько делителей заданной длины (например 7–8 десятичных знаков и менее).

Атака на основе вычисления $\varphi(n)$

Один из способов, которым потенциальный нарушитель может пытаться атаковать систему RSA, состоит в нахождении $\varphi(n)$, поскольку в этом случае d может быть легко вычислен из соотношения $ed \equiv 1 \pmod{\varphi(n)}$. Однако следующие соображения показывают, что этот подход не проще, чем попытки разложить n. Если известно значение $\varphi(n)$, то p и q могут быть найдены следующим образом. Из соотношений

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1 = n - (p+q) + 1$$

видно, что по $\varphi(n)$ легко находится $p + q$. Более того, из соотношения

$$(p-q)^2 = (p+q)^2 - 4pq = (p+q)^2 - 4n$$

видно, что, зная n и $p + q$, легко получить $p - q$ (извлечение квадратного корня не является трудно решаемой задачей). После этого, зная $p - q$ и $p + q$, легко получить p и q по следующим формулам:

$$p = (1/2)\,[(p+q) + (p-q)],$$

$$q = (1/2)\,[(p+q) - (p-q)].$$

Это показывает, что любая попытка найти функцию Эйлера от модуля n эквивалентна решению сложной проблемы разложения n на множители. Иными словами, стойкость системы RSA к рассмотренной атаке не ниже сложности разложения модуля на простые множители.

Система RSA на основе одного модуля n

Можно ли реализовать надежную систему ЭЦП типа RSA, в которой используется единый составной модуль для всех пользователей? Очевидно, что в такой постановке задачи предполагается наличие доверительного центра, который генерирует модуль и знает его разложение. Одной из функций доверительного центра является снабжение каждого пользователя уникальными открытым e и закрытым d ключами. Будет ли стойкой такая система? В отличие от классического варианта использования RSA, в котором предполагается, что каждому пользователю соответствует уникальный модуль и последний генерируется самостоятельно самим пользователем, в схеме с доверительным центром возникает новая возможность атаковать систему, зная пару

соответствующих друг другу ключей e и d. Покажем, что эта атака может оказаться эффективной и, следовательно, система RSA с одним модулем не является стойкой.

Действительно, по открытому e и секретному d ключам можно определить $ed = 1 \bmod \varphi(n)$. Следовательно, имеем $ed - 1 = Q\varphi(n)$. Поскольку $\varphi(n) = n - (p + q) + 1$, то оценить Q можно из разложения числа $ed - 1$. Если атакующему удастся разложить последнее число, то, перемножая соответствующий набор делителей, можно найти частичное произведение П, которое примерно равно n. С достаточно большой вероятностью $(ed - 1)/\Pi$ равно Q. Перебирая различные значения, найдем Q, а затем $\varphi(n) = (ed - 1)/Q$. Эту атаку может опробовать любой пользователь.

Еще больше возможностей имеет коалиция из нескольких пользователей, которые раскрывают друг другу свои секретные ключи. Например, в случае двух пользователей имеем такие варианты:

$$e_1d_1 - 1 = Q_1\varphi(n),\ e_2d_2 - 1 = Q_2\varphi(n) \text{ и } e_1d_1 - e_2d_2 = Q_3\varphi(n).$$

Теперь можно выбрать, какое из чисел $e_1d_1 - 1$, $e_2d_2 - 1$ или $e_1d_1 - e_2d_2$ может быть легко разложено на простые множители. Чем больше коалиция, тем больше различных чисел может быть опробовано на предмет разложения на множители. Это общая идея нахождения $\varphi(n)$. Рассмотрим более конкретный подход.

Пусть $p' = p - 1$ и $q' = q - 1$, тогда имеем

$$e_1d_1 - 1 = Q'_1 \cdot 2^{s_1} p'q';\quad e_2d_2 - 1 = Q'_2 \cdot 2^{s_2} p'q',$$

где Q'_1 и Q'_2 — некоторые нечетные целые числа. Если НОД(Q'_1, Q'_2)=1, то НОД$(e_1d_1 - 1, e_2d_2 - 1) = 2^{|s_1 - s_2|} p'q'$. Последнее значение легко находится с помощью расширенного алгоритма Евклида. По найденному значению $2^{|s_1 - s_2|} p'q'$ последовательным делением на 2 легко найдем произведение $\varphi(n) = p'q'$. Например, в качестве критерия правильного значения $p'q'$ можно использовать вычисленные значения p и q, такие что $pq = n$ (p и q вычисляются по предположительно известному значению функции Эйлера от n).

3.6.3. Слепая подпись на основе системы RSA

Понятие *слепой подписи* впервые было введено Д. Чаумом (D. Chaum), который также предложил первые варианты ее реализации [19–21]. Под слепой подписью понимается двухключевая криптосистема, которая позволяет осуществить подписывание электронных сообщений таким образом, чтобы подписывающая сторона не имела доступа к информации, содержащейся в

подписываемом сообщении. Такое надуманное и нелепое, на первый взгляд, требование в ряде криптографических протоколов имеет очень большое значение. Слепая подпись используется, например, в системах тайного электронного голосования и электронных денег, т. е. в таких криптографических протоколах, где требуется обеспечить решение задачи неотслеживаемости. Сама процедура слепой подписи требует от одного из участников согласия на то, что он может подвергнуться некоторым посягательствам путем приписывания ему определенных обязательств, которые он не хотел бы брать на себя.

В реальных протоколах, где используется слепая подпись, предусматриваются также процедуры, которые дают определенные гарантии подписывающей стороне, что ее не обманут. Эти гарантии основаны на некоторых дополнительных процедурах и соглашениях, вносящих ограничения и налагающих определенную ответственность на сторону, представляющую документ для слепого подписывания. Естественно, что каждая из сторон должна согласиться на тот или иной риск и на определенные гарантии, в противном случае нельзя было бы решить ту задачу, которую хотят решить обе стороны. Так, при использовании систем ЭЦП пользователи принимают риск, что кто-то может вычислить их секретный ключ. Гарантией защиты их интересов является высокая сложность задачи вычисления секретного ключа. Другим типом гарантий, используемых в криптографических протоколах, является низкая вероятность повторения случайных чисел, если они являются достаточно длинными. Используя протокол слепой подписи, можно решить некоторые важные для практики проблемы (такими являются, например, построение систем тайного электронного голосования, электронных денег).

Рассмотрим протокол слепой подписи Чаума, основанный на уже известной читателю криптосистеме RSA. Пусть субъект **A** желает подписать некоторое сообщение M у субъекта **B**. Для этого необходимо осуществить следующие шаги.

1. Пользователь **A** (субъекты являются пользователями данной криптосистемы) генерирует случайное простое число k, такое что НОД(k, N) = 1, где N — часть открытого ключа пользователя **B**, т. е. модуль, по которому осуществляются вычисления.

2. Далее он вычисляет значение $M' = k^e M \pmod N$, которое представляет для подписывания. Подписывающий не может получить доступ к сообщению M, поскольку оно зашифровано путем наложения на него «разового» ключа k^e и использования операции модульного умножения.

3. Пользователь **B** подписывает сообщение M' в соответствии с процедурой подписывания в криптосистеме RSA: $S' = (k^e M)^d = kM^d \pmod N$. Сформировав подпись S', подписывающий также не имеет возможно-

сти получить доступ к значению M^d, поскольку оно зашифровано путем наложения на него «разового» ключа k. Если бы подписывающий смог узнать значение M^d, то он тогда смог бы легко вычислить M: $(M^d)^e = M \pmod N$. Это означает, что, получив значение $M^d \pmod N$, пользователь **A** должен держать его в секрете от подписавшего.

4. Теперь, используя расширенный алгоритм Евклида, пользователь **B** вычисляет для числа k мультипликативно обратный элемент (k^{-1}) в поле вычетов по модулю N и восстанавливает подпись для сообщения M, а именно: $S = k^{-1}S' = k^{-1}kM^d = M^d \pmod N$.

Таким образом, цель достигнута — пользователь **A** сформировал правильную подпись пользователя **B**, соответствующую сообщению M, причем он уверен, что 1) подписывающий не знает содержания сообщения M и 2) в дальнейшем, когда подписавшему предъявят сообщение M и подпись к нему, он удостоверится в подлинности своей подписи, но не сможет выяснить, кто представлял ему этот документ на подпись (предполагается, что подписывающий формирует вслепую подписи к двум и более различным документам).

ГЛАВА 4

Системы ЭЦП с составным модулем

4.1. Цифровая подпись Рабина

Система ЭЦП, предложенная М. О. Рабином [34], основана на сложности вычисления квадратных корней по модулю n, представляющему собой произведение двух больших простых чисел p и q. В этой схеме используется тот факт, что при определенном выборе чисел p и q специального вида владелец секретного ключа может легко вычислять квадратный корень по модулю n из чисел, являющихся квадратичными вычетами. Модуль n представляет собой открытый ключ, который используется при проверке подлинности подписи. Делители модуля p и q составляют секретный ключ и используются при формировании подписи. Особенностью чисел p и q является то, что они должны быть сравнимы с числом 3 по модулю 4, т. е. удовлетворять соотношениям:

$$p \equiv 3 \bmod 4 \quad \text{и} \quad q \equiv 3 \bmod 4.$$

При выполнении этих условий владелец секретного ключа может легко вычислять корни второй степени из квадратичных вычетов по модулю n. Пусть a — квадратичный вычет, причем НОД(a, n) = 1. В соответствии с нашим допущением существует такое z, что $z^2 \equiv a \bmod n$. В соответствии с китайской теоремой об остатках выполняются также сравнения $z^2 \equiv a \bmod p$ и $z^2 \equiv a \bmod q$, т. е. имеем $z_p \equiv a^{1/2} \bmod p$ и $z_q \equiv a^{1/2} \bmod q$. Известно, что для квадратичных вычетов по простому модулю p выполняется сравнение $a^{\frac{p-1}{2}} \equiv 1 \bmod p$, откуда для случая $p \equiv 3 \bmod 4$ легко получить:

$$a \equiv a \cdot a^{\frac{p-1}{2}} \equiv a^{\frac{p+1}{2}} \bmod p \Rightarrow a^{1/2} \equiv a^{\frac{p+1}{4}} \bmod p .$$

Теперь учтем, что если $z_p{}^2 \equiv a \bmod p$, то также имеем $(-z_p)^2 \equiv a \bmod p$. Таким образом, имеем следующие два корня из a по модулю p:

$$z_{p_1} \equiv a^{\frac{p+1}{4}} \bmod p \quad \text{и} \quad z_{p_2} \equiv p - z_{p_1} \equiv p - a^{\frac{p+1}{4}} \bmod p$$

и следующие два корня по модулю q:

$$z_{q_1} = a^{\frac{q+1}{4}} \bmod q \quad \text{и} \quad z_{q_2} = q - z_{q_1} = q - a^{\frac{q+1}{4}} \bmod q .$$

Используя китайскую теорему об остатках и следующие пары чисел (z_{p_1}, z_{q_1}), (z_{p_1}, z_{q_2}), (z_{p_2}, z_{q_1}) и (z_{p_2}, z_{q_2}), легко найти четыре корня z_1, z_2, z_3 и z_4 второй степени из a по модулю n. Например, z_1 вычисляется по следующей формуле:

$$z_1 = \left(z_{p_1} q(q^{-1} \bmod p) + z_{q_1} p(p^{-1} \bmod q) \right) \bmod n .$$

Рассмотрим процедуру формирования подписи к сообщению m в ЭЦП М. Рабина.

Формирование секретного ключа заключается в формировании больших простых чисел p и q, сравнимых с числом 3 по модулю 4.

Формирование открытого ключа n осуществляется путем перемножения чисел p и q: $n = pq$.

Вычисление подписи s к сообщению $m = (m_0, m_1, \ldots, m_i, \ldots, m_{h-1})$, где $\forall i \in \{0, 1, \ldots, h-1\}$ $m_i \in \{0, 1\}$, включает следующие шаги.

1. Генерируется случайное число $r = (r_0, r_1, \ldots, r_i, \ldots, r_{t-1})$, где $\forall i \in \{0, 1, \ldots, t-1\}$ $r_i \in \{0, 1\}$, которое объединяется с сообщением:

   $$m||r = (m_0, m_1, \ldots, m_{h-1}, r_0, r_1, \ldots, r_{t-1}).$$

   Двоичному вектору $m||r$ сопоставляется число $a < n$. Проверяется выполнение условий $a^{\frac{p-1}{2}} \equiv 1 \bmod p$ и $a^{\frac{q-1}{2}} \equiv 1 \bmod q$. Если одно из соотношений не выполняется, то генерируется новое случайное число r. Процедура повторяется до тех пор, пока не будет найдено такое значение r, для которого оба условия выполняются, т. е. получено a, являющееся квадратичным вычетом по mod n.

2. Вычисляются значения $z_{p_1} \equiv a^{(p+1)/4} \bmod p$, $z_{q_1} \equiv a^{(q+1)/4} \bmod q$. (Могут быть использованы также пары значений (z_{p_1}, z_{q_2}), (z_{p_2}, z_{q_1}) или (z_{p_2}, z_{q_2}).)

3. По китайской теореме об остатках вычисляется квадратный корень z_1 из a по модулю n.

4. В качестве подписи к m берется пара чисел (r, z_1).

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение a':

$$a' = z_1^2 \bmod n.$$

2. Число a' представляется в двоичном виде и интерпретируется как двоичный вектор $m' \| r'$:

$$a' = (a'_0, a'_1, \ldots, a'_{u-1}) = m' \| r'.$$

3. Сравниваются значения a и a'. Если $m = m'$ и $r = r'$, то подпись признается подлинной.

Нетрудно видеть, что для специально выбираемых сообщений малой длины подделка подписи не представляет труда и без знания секретного ключа. Например, для 1024-битового открытого ключа n выбираем произвольное 200-битовое число $v = (v_0, v_1, \ldots, v_{199})$, вычисляем $w = v^2 \bmod n = (w_0, w_1, \ldots, w_{399})$. Имеем подпись (r, z_1), где $r = (w_{300}, w_{301}, \ldots, w_{399})$ и $z_1 = (v_0, v_1, \ldots, v_{199})$, к сообщению $m = (w_0, w_1, \ldots, w_{299})$. Для предотвращения атак на основе выбранных сообщений в схему подписи можно включить шаг, предусматривающий вычисление хэш-функции от сообщения, после чего формируется подпись к хэш-функции. В этом случае нарушитель не сможет представить сообщение, хэш-функция от которого равна случайному значению, полученному указанным выше путем.

4.2. Цифровая подпись Фиата–Шамира

Схема ЭЦП Фиата–Шамира основана на сложности вычисления квадратных корней по модулю, представляющему собой произведение двух больших простых чисел [25]. Возможны два варианта данной ЭЦП: 1) каждый абонент формирует свой индивидуальный модуль $n = pq$, где p и q — большие простые числа, и 2) все абоненты используют одно и то же значение составного модуля n, которое вырабатывается доверительным центром, причем исходные делители p и q уничтожаются (для формирования открытых и закрытых ключей не требуется знать разложение модуля). Первый случай является предпочтительным, поскольку может обеспечить абонентам более полные гарантии того, что разложение модуля неизвестно потенциальному нарушителю.

Формирование секретного ключа осуществляется следующим образом. Абонент генерирует h случайных чисел $x_0, x_1, \ldots, x_i, \ldots, x_{h-1}$, таких что $n^{1/2} + 1 \le x_i \le n - 1$ и НОД$(x_i, n) = 1$ для $i = 0, 1, \ldots, h - 1$.

Формирование открытого ключа осуществляется путем вычисления значений обратных квадратам чисел x_i по модулю n. Открытым ключом является набор чисел $y_0, y_1, \ldots, y_i, \ldots, y_{h-1}$, для которых имеем $y_i = x_i^{-2} \bmod n$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k, $k < n-1$. В определенном смысле можно сказать, что k играет роль разового секретного ключа, обеспечивающего маскирование основного (долговременного) секретного ключа.
2. Вычисляется значение $r = k^2 \bmod n$.
3. К сообщению m присоединяется число r, и от полученного нового сообщения вычисляется h-битовое значение некоторой специфицированной хэш-функции H: $E = H(m \| r)$, которое является первой частью подписи. Значение E представляется в виде последовательности битов e_i: $E = (e_0, e_1, \ldots, e_i, \ldots, e_{h-1})$. На данном шаге обеспечивается задание зависимости значения E от случайного числа k и от сообщения m. Этим обеспечивается привязка подписи к значениям k и m. Кроме того, устраняются потенциальные атаки, основанные на предварительно найденных коллизиях, т. е. использующие пары документов m_1 и m_2, которые обладают одинаковыми значениями хэш-функции $H(m_1) = H(m_2)$.
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$S = k\prod_{i=0}^{h-1} x_i^{e_i} \bmod n .$$

На этом завершается формирование подписи (E, S) к сообщению m. Нетрудно увидеть, что к фиксированному сообщению могут быть выработаны различные значения подписи, поскольку при ее формировании используется случайное число k.

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r':

$$r' = S^2\prod_{i=0}^{h-1} y_i^{e_i} \bmod n .$$

2. К сообщению m присоединяется число r', и вычисляется значение хэш-функции $E' = H(m \| r')$.
3. Сравниваются значения E и E'. Если $E = E'$, то подпись признается подлинной.

Достоинством рассмотренной схемы ЭЦП является сравнительно малая сложность процедуры формирования и проверки подписи, что обеспечивает

достаточно высокое быстродействие. Недостатком является большой размер открытого и закрытого ключей. Для 1024-битового модуля и 160-битовой хэш-функции размер каждого из ключей составит около 164 Кбит. Этот недостаток приводит к тому, что размер справочника открытых ключей может оказаться весьма объемным при достаточно большом числе абонентов. Кроме того, существенно усложняется аппаратная реализация.

4.3. Обобщение схемы Фиата–Шамира

Схема подписи Фиата–Шамира может быть обобщена практически для произвольного значения степени корня, которое связывает открытые и закрытые ключи. В обобщенной схеме каждый абонент формирует свой индивидуальный модуль $n = pq$, где p и q — большие простые числа, генерируемые таким образом, чтобы числа $q-1$ и $p-1$ содержали одинаковый простой делитель q' достаточно большой разрядности (например 256-битовый).

Формирование секретного ключа осуществляется следующим образом. Абонент генерирует число α, относящееся к показателю q', и h случайных чисел $v_0, v_1, \ldots, v_{h-1}$. Затем вычисляет секретный ключ в виде набора чисел $x_0, x_1, \ldots, x_i, \ldots, x_{h-1}$, таких что для $i = 0, 1, \ldots, h-1$ имеет место $x_i = \alpha^{v_i} \bmod n$. Числа p и q также являются секретными, но они необходимы только на этапе генерации чисел α и q'.

Формирование открытого ключа осуществляется путем вычисления значений обратных t-й степени чисел x_i по модулю n. Открытым ключом является набор чисел $y_0, y_1, \ldots, y_i, \ldots, y_{h-1}$, для которых имеем $y_i = \alpha^{-tv_i} \bmod n = x_i^{-t} \bmod n$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k', $1 \le k' \le q' - 1$, и вычисляется $k = \alpha^{k'} \bmod n$ (разовый секретный ключ).
2. Вычисляется значение $r = \alpha^{k't} \bmod n = k^t \bmod n$.
3. К сообщению m присоединяется число r, и от полученного нового сообщения вычисляется h-битовое значение некоторой специфицированной хэш-функции H: $E = H(m \| r)$, которое является первой частью подписи. Значение E представляется в виде последовательности битов e_i: $E = (e_0, e_1, \ldots, e_i, \ldots, e_{h-1})$. На данном шаге обеспечивается задание зависимости значения E от случайного числа k' и от сообщения m. Этим обеспечивается привязка подписи к значениям k' и m, что уст-

раняет потенциальные атаки, основанные на предварительно найденных коллизиях.

4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$S = k\prod_{i=0}^{h-1} x_i^{e_i} \bmod n.$$

На этом завершается формирование подписи (E, S) к сообщению m.

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r':

$$r' = S^t \prod_{i=0}^{h-1} y_i^{e_i} \bmod n.$$

2. К сообщению m присоединяется число r', и вычисляется значение хэш-функции $E' = H(m \| r')$.

3. Сравниваются значения E и E'. Если $E = E'$, то подпись признается подлинной.

Время формирования и время проверки подписи в обобщенной схеме имеют примерно те же значения, что и в схеме Фиата–Шамира. В обобщенном варианте модуль n является индивидуальным для каждого пользователя, т. е. является частью его открытого ключа. Отметим, что связь между элементами секретного ключа $x_0, x_1, \ldots, x_{h-1}$ и элементами открытого ключа $y_0, y_1, \ldots, y_i, \ldots, y_{h-1}$ можно задать непосредственно по формуле $y_i = x_i^{-t} \bmod n$, где НОД$(x_i, n) = 1$. Значение k можно выбрать случайно, а параметр r рассчитать по формуле $r = k^t \bmod n$. Тогда не потребуется находить число α, относящееся к простому показателю, и снимаются дополнительные ограничения на выбор чисел p и q (требование наличия одинакового делителя q' в разложении чисел $p-1$ и $q-1$).

4.4. Уменьшение размера открытого ключа в схеме Фиата–Шамира

Недостатком схемы ЭЦП Фиата–Шамира является большой размер открытого и секретного ключей. Он превышает размер составного модуля n в h раз. Это существенно больше, чем в системе RSA и ЭЦП, основанных на сложности дискретного логарифмирования. Фактически в схеме Фиата–Шамира используется h открытых и h секретных ключей обычного размера.

Этот недостаток может быть устранен заменой операций умножения ключей, выбираемых в зависимости от одного из двух параметров цифровой подписи (параметра E), на одну операцию возведения ключа в степень E по модулю n. В этом случае можно обойтись одним открытым и одним секретным ключом длины $|n|$. Рассмотрим получающуюся после такого модифицирования схему ЭЦП. Возможны два варианта модифицированной схемы: 1) каждый абонент формирует свой индивидуальный модуль $n = pq$, где p и q — большие простые числа, и 2) все абоненты используют одно и то же значение составного модуля n, которое вырабатывается доверительным центром (исходные делители p и q уничтожаются).

Формирование секретного ключа осуществляется следующим образом. Абонент генерирует случайное число x, для которого выполняются условия $n^{1/2} + 1 \le x \le n - 1$ и НОД$(x_i, n) = 1$.

Формирование открытого ключа осуществляется путем вычисления значения, обратного квадрату числа x по модулю n. Открытым ключом является число $y = x_i^{-2} \bmod n$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k, $k < n - 1$.
2. Вычисляется значение $r = k^2 \bmod n$.
3. К сообщению m присоединяется число r, и от полученного нового сообщения вычисляется h-битовое значение некоторой специфицированной хэш-функции H: $E = H(m\|r)$, которое является первой частью подписи. Значение E представляется в виде последовательности битов e_i: $E = (e_0, e_1, \ldots, e_i, \ldots, e_{h-1})$.
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$S = k \cdot x^E \bmod n .$$

Подписью к сообщению m является пара чисел (E, S). Нетрудно увидеть, что к фиксированному сообщению могут быть выработаны различные значения подписи, поскольку при ее формировании используется случайное число k.

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r':

$$r' = S^2 y^E \bmod n .$$

2. К сообщению m присоединяется число r' и вычисляется значение хэш-функции $E' = H(m\|r')$.

3. Сравниваются значения E и E'. Если $E = E'$, то подпись признается подлинной. (Действительно, имеем:
$r' = S^2 y^E \bmod n = k^2 (x^E)^2 (x^{-2})^E \bmod n = k^2 \bmod n = r$.)

4.5. Схема ЭЦП Онга–Шнорра–Шамира

Данная остроумная схема построения системы ЭЦП является наиболее быстродействующей, однако было показано, что она не является безопасной, в результате чего она не нашла применения на практике. Тем не менее, ее конструкция является оригинальной и полезной для ознакомления. Основывается схема ЭЦП Онга–Шнорра–Шамира на сложности разложения RSA-модуля n и на сложности извлечения квадратных корней по составному модулю.

Открытый ключ h генерируется следующим образом. Формируется RSA-модуль n и выбирается случайное k, взаимно простое с модулем n. Затем вычисляется h:

$$h = -(k^{-1})^2 \bmod n = -k^{-2} \bmod n .$$

Открытым ключом является пара чисел (n, h). Секретным ключом является число k. Заметим, что для нахождения секретного ключа и для формирования подписи нет необходимости знать разложение модуля на множители. Однако формирование модуля должно быть выполнено с соблюдением всех рекомендаций, что и в случае криптосистемы RSA, так как факторизация модуля означает раскрытие (взлом) рассматриваемой криптосистемы.

Формирование подписи к сообщению M осуществляется следующим путем. Выбирается случайное число r, такое что r и n являются взаимно простыми. Затем вычисляются два числа S_1 и S_2, образующие подпись к сообщению M:

$$S_1 = \frac{1}{2}\left[\frac{M}{r} + r\right] \bmod n ;$$

$$S_2 = \frac{k}{2}\left[\frac{M}{r} - r\right] \bmod n .$$

Подлинность подписи проверяется по уравнению

$$M' = (S_1^2 + hS_2^2) \bmod n .$$

Если $M' = M$, то подпись принимается как подлинная. К сожалению, как отмечалось выше, эта схема оказалась нестойкой. В последующих своих вер-

сиях эта схема претерпевала несколько изменений, однако авторам не удалось придать ей достаточную стойкость. Подробный анализ безопасности этой схемы приводится в работах [18, 24, 32, 46].

4.6. Варианты схемы Эль-Гамаля с составным модулем

Схема ЭЦП Эль-Гамаля потенциально содержит в себе возможность использования некоторого механизма сокращения длины подписи, однако известные ее варианты требуют использования двух параметров r и s в качестве подписи. Удвоение длины подписи по сравнению с размером простого модуля связано с тем, что для обеспечения невозможности вычисления секретного ключа x на основе известной подписи используется разовый открытый ключ r, вычисляемый по разовому секретному числу k. Благодаря этому для потенциального нарушителя в уравнении генерации подписи присутствуют две неизвестные величины: x и k. Поэтому он не имеет возможности с большой вероятностью вычислить секретный ключ x (смысл использования разового ключа состоит в маскировке секретного ключа).

Применение в схемах ЭЦП, подобных системе Эль-Гамаля, составного модуля вместо простого [10] связано с тем, что большое число таких схем не обеспечивает стойкости к атакам, основанным на вычислении подписи путем подбора параметра r в виде $r = \alpha^t y^u \bmod p$. В таких атаках требуется знание функции Эйлера от модуля, поэтому они могут быть устранены применением составного модуля, подобного модулю в системе RSA. Это обусловливает интерес к разработкам схем ЭЦП, в которых сочетаются механизмы, используемые в схемах, основанных на сложности факторизации модуля, и в схемах, основанных на задаче дискретного логарифмирования.

Рассмотрим схемы ЭЦП с обобщенным уравнением проверки подписи следующего вида:

$$r^{f_1(H,S)} = y^{f_2(H,S,r)} \alpha^{f_3(H,S,r)} \bmod n,$$

где f_1, f_2 и f_3 — некоторые простые функции, S — подпись, $n = pq$, причем в разложении чисел $p-1$ и $q-1$ содержится одинаковый большой простой множитель q', имеющий длину 160–256 бит, α — число, относящееся к показателю q' по модулю n. Предполагается, что открытым ключом является пара чисел (y, n), где первое значение вычисляется по формуле $y = \alpha^x \bmod n$. Секретными являются значения x, p, q и q'. В схемах ЭЦП этого типа составной модуль применен для устранения атак, связанных с попыткой вычисления подписи, путем представления маскирующего параметра r в виде произведе-

ния некоторых степеней чисел y и α. Благодаря использованию составного модуля, предпосылки для проведения данной атаки устраняются даже в случаях использования упрощенных вариантов приведенного выше обобщенного уравнения, в которых используется только один из параметров y и α:

$$r^{f_1(H,S)} = y^{f_2(H,S,r)} \bmod n;$$

$$r^{f_1(H,S)} = \alpha^{f_2(H,S,r)} \bmod n.$$

Параметр r задает в уравнении генерации подписи неизвестную k, которая изменяется с каждой процедурой генерации подписи. Это предотвращает атаки, связанные с попыткой вычисления секретного ключа из уравнения формирования подписи. Однако благодаря наличию этого маскирующего параметра в случае простого модуля создаются предпосылки вычисления подписи по уравнению проверки без знания секретного ключа. Применение составного модуля позволяет построить стойкие схемы ЭЦП с более простыми уравнениями проверки по сравнению со случаем применения простого модуля. Ряд возможных вариантов представлен в табл. 4.1.

Таблица 4.1

Варианты ЭЦП с составным модулем, где F — некоторая трудно обратимая сжимающая функция, g — простое число длиной 160 – 256 бит ($g \neq q'$), $r' = r \bmod g$.

№ п/п	Уравнение проверки подписи	Уравнение формирования подписи	Подпись
1	$r^H = y^r \alpha^S \bmod n$	$kH = xr + S \bmod q'$	(r, S)
2	$\alpha^H = y^r r^S \bmod n$	$H = xr + kS \bmod q'$	(r, S)
3	$r^H = y^{rS} \bmod n$	$kH = xrS \bmod q'$	(r, S)
4	$r^H = y^{r+S^2} \bmod n$	$kH = xr + S^2 \bmod q'$	(r, S)
5	$r^S = \alpha^{r+H} \bmod n$	$kS = r + H \bmod q'$	(r, S)
6	$F(r) = F(\alpha^{SH \cdot F(r)} \bmod n)$	$k = SH \cdot F(r) \bmod q'$	$(F(r), S)$
7	$F(r) = F(\alpha^{S \cdot H + F(r)} \bmod n)$	$k = SH + F(r) \bmod q'$	$(F(r), S)$
8	$r' = (\alpha^{S \cdot r' + H} \bmod n) \bmod g$	$k = Sr' + H \bmod q'$	(r', S)
9	$F(r) = F(\alpha^{S^2 + H \cdot F(r)} \bmod n)$	$k = S^2 + H \cdot F(r) \bmod q'p'$	$(F(r), S)$
10	$r' = (\alpha^{Hr' + S^2} \bmod n) \bmod g$	$k = Hr' + S^2 \bmod q'p'$	(r', S)

В этих схемах открытым ключом является тройка чисел y, n и α (в случае схем 1–2) или пара чисел y и n (схемы 3–4) или α и n (в случае схем 5–10), где $y = \alpha^x \bmod n$, причем α относится к показателю $q' \mid \varphi(n)$ по модулю n (в схемах 5–8) или к показателю $q'p' \mid \varphi(n)$ по модулю n (в схемах 9–10, где числа q' и p' сравнимы с числом 3 по модулю 4, что используется для извлечения квадратного корня по модулю $q'p'$ при вычислении S, кроме того, $p' \mid p-1$, $p' \nmid q-1$, $q' \mid q-1$ и $q' \nmid p-1$). Параметр r формируется по случайно выбираемому числу в соответствии с формулой $r = \alpha^k \bmod n$. В данных схемах значение y также относится к показателю q' по модулю n, поэтому вычисление значения y как степени числа α теряет смысл.

Следует отметить, что при использовании простого модуля p вместо составного n в схемах 3–10, представленных в табл. 4.1, появляется возможность вычисления подписи без знания секретного ключа путем представления параметра r в виде $r = y^t \bmod p$ или в виде $r = \alpha^t \bmod p$ (выбирается произвольное t и затем находится r) и вычисления параметра S. Например, в схемах 6–10 вычисление осуществляется в соответствии с одним из следующих обобщенных выражений: $t = \Psi(S, H, r)$, $t = \Psi(S, H, F(r))$ или $t = \Psi(S, H, r')$, где Ψ — некоторое достаточно простое выражение (правая часть уравнения формирования подписи), в котором не присутствует слагаемое S. Последнее требование является условием предотвращения рассматриваемой атаки в случае составного модуля. Действительно, если $t = \Psi'(H, F(r)) + S$, то можно выбрать достаточно большое t, после чего S легко вычисляется по формуле $S = t - \Psi'(H, F(r))$, где не используется операция взятия остатка от деления на модуль. Заметим, что стойкими являются схемы с уравнением проверки подписи типа

$$r = y^{\Psi'(H, F(r)) + S^2} \bmod n \quad \text{или} \quad F(r) = F(y^{\Psi'(H, F(r)) + S^2} \bmod n)$$

(второе уравнение относится к схеме с сокращенной подписью, полученной из первого уравнения). В последних схемах ЭЦП рассматриваемая атака предотвращается, поскольку для нахождения S требуется выполнить вычисление квадратного корня из числа $t - \Psi'(H, \mathrm{F}(r))$ по неизвестному составному модулю $q'p'$. Отметим также, что схемы 9 и 10 в табл. 4.1 имеют скорее методическое значение, чем практическое, поскольку для вычисления подписи потребуется осуществить несколько попыток выбора различных значений k (пока значение $t - \Psi'(H, \mathrm{F}(r))$ не окажется квадратичным вычетом по модулю $q'p'$).

Схемы 1 и 2 при простом модуле допускают вычисление без знания секретного ключа x некоторых значений H и (r, S), которые удовлетворяют уравнению проверки подписи. В случае составного модуля эта возможность устранена.

4.7. Схемы на основе сложности извлечения корней по составному модулю

В табл. 4.1 приведены схемы с достаточно простым уравнением проверки подписи. Дальнейшее упрощение формулы проверки подписи связано с отказом от использования параметра S, который играет роль «подгоночного» параметра (владелец секретного ключа может вычислить значение S, которое будет соответствовать заранее вычисленному r). Наличие этого подгоночного параметра создает предпосылки к рассмотренной выше атаке с представлением параметра r в виде $r = y^t \bmod p$. В схемах ЭЦП, в которых подпись задана только параметром r, общая предпосылка для атак указанного типа устраняется. Применение составного модуля обеспечивает стойкость новых схем ЭЦП, приведенных в табл. 4.2. В схемах 1–4 секретным ключом являются числа x и q', а в схемах 5 и 6 — число q'. Формирование подписи осуществляется следующим образом: в зависимости от значения H вычисляется число k, а затем — значение $r = \alpha^k \bmod n$. Заметим также, что в схемах 4 и 6 степень v есть небольшое натуральное число, поскольку проверка подписи осуществляется как v-кратное возведение подписи r в степень H. Последнее приводит к тому, что в этих схемах процедура проверки подлинности подписи является более продолжительной по сравнению с остальными схемами.

Таблица 4.2

Схемы с упрощенным уравнением проверки подписи

№ п/п	Уравнение проверки подписи	Формула для вычисления значения k	Открытый ключ
1	$r^y = y^H \bmod n$	$k = xH/y \bmod q'$	(y, n)
2	$r^v = y^H \bmod n$	$k = xH/v \bmod q'$	(y, n)
3	$r^H = y \bmod n$	$k = x/H \bmod q'$	(y, n)
4	$r^{H^v} = y \bmod n$	$k = x \cdot H^{-v} \bmod q'$	(y, n)
5	$r^H = \alpha \bmod n$	$k = H^{-1} \bmod q'$	(α, n)
6	$r^{H^v} = \alpha \bmod n$	$k = H^{-v} \bmod q'$	(α, n)

Стойкость схем основана на сложности разложения модуля n на множители и на сложности извлечения корня большой степени по составному модулю. В этом плане схемы, представленные в табл. 4.2, аналогичны системе

RSA: если нарушителю удастся найти разложение $n = pq$ или узнать секретное число q', то он сможет сформировать правильную подпись к произвольному документу. Отличие состоит в том, что в предлагаемых схемах формирование подписи не является процедурой шифрования, т. е. по подписи невозможно восстановить сообщение. Кроме того, в схемах 3–6 значение степени, в которую возводится подпись при проверке ее подлинности, зависит от значения хэш-функции подписываемого документа, т. е. не является фиксированным. Легко заметить, что нет необходимости осуществлять вычисление подписи r с использованием секретного значения x, поскольку значение r можно определять с использованием формулы $r = y^k \bmod n$. Поэтому y всегонавсего играет роль числа, относящегося к показателю q' по модулю n, также как и α, т. е. параметр x фактически не играет никакой роли секретного элемента.

Таким образом, из схем, представленных в таблице 4.1, при соответствующем упрощении вытекают схемы, основанные на сложности извлечения корней большой степени по составному модулю. В этих схемах подписью служит только один параметр r, однако размер подписи равен размеру модуля. Сокращение длины подписи в этих схемах проблематично.

4.8. Расширение криптосистемы RSA

Рассмотрим подробнее криптосхему 5 из таблицы 4.2. Как уже отмечалось, в ней используется RSA-модуль $n = pq$, особенностью которого является то, что простые числа p и q формируются таким образом, чтобы числа $p - 1$ и $q - 1$ содержали одинаковый простой делитель γ заданного размера, например $161 \le |\gamma| \le 257$ бит. Для этой цели может быть использован алгоритм генерации сильных псевдопростых чисел [26] или алгоритм генерации простых чисел, описанный в российском стандарте ЭЦП ГОСТ Р 34.10–94.

Секретным ключом служит тройка чисел (p, q, γ).

Открытым ключом является пара чисел (n, α), где α генерируется следующим образом. Выбирается случайное число β, такое что НОД$(\beta, n) = 1$, вычисляется значение $t = \varphi(n)/\gamma^2 = (p-1)(q-1)/\gamma^2$, затем число $z = \beta^t \bmod n$. Если $z \neq 1$, то z берется в качестве элемента α открытого ключа, т. е. $\alpha = z$. Поскольку β и n являются взаимно простыми числами, то также имеем НОД$(\alpha, n) = 1$.

Сформированное таким образом число α является генератором группы $\{\alpha, \alpha^2 \bmod n, \ldots, \alpha^{\gamma-1} \bmod n, \alpha^\gamma \bmod n\}$ порядка γ, т. е. имеем $\alpha^\gamma \bmod n = 1$.

Процедура вычисления подписи. Для того чтобы подписать сообщение M, выполняются следующие шаги:

1. Используя некоторую хэш-функцию, генерирующую хэш-значения H длиной $160 \le |H| \le 256$ бит, вычисляется значение хэш-функции от данного сообщения: $H = H(M)$.
2. Проверяется выполнимость условий $H \neq 0$ и $H \neq 1$ (в RSA такая проверка также необходима). Если $H = 0$ или $H = 1$, то сообщение модифицируется без изменения смыслового содержания и возвращается к шагу 1.
3. С помощью расширенного алгоритма Евклида вычисляется значение $U = H^{-1} \bmod \gamma$.
4. Вычисляется подпись S: $S = \alpha^{U} \bmod n$.

Процедура проверки подписи. Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение хэш-функции от сообщения M: $H = H(M)$.
2. Вычисляется значение $\alpha' = S^{H} \bmod n$. Если $\alpha' = \alpha$, то подпись признается подлинной. В противном случае подпись признается недействительной.

При правильно сформированной подписи условие $\alpha' = \alpha$ выполняется, так как имеем:

$$S^{H} \bmod n = (\alpha^{K})^{H} \bmod n = (\alpha^{H^{-1}})^{H} \bmod n = \alpha^{H^{-1}H} \bmod n =$$
$$= \alpha^{H^{-1}H \bmod \gamma} \bmod n = \alpha^{1} \bmod n = \alpha.$$

Определить значение числа U, соответствующего правильной ЭЦП, можно только по известному значению γ. Нахождение значения γ по открытому ключу (n, α) вычислительно неосуществимо, поскольку для этого требуется решить задачу разложения многоразрядного числа n на два больших простых множителя.

4.8.1. Модифицированные версии и совместимость с RSA

Схема подписи, описанная выше, может быть подвергнута следующей теоретической атаке. Атакующий предоставляет подписывающему значение хэш-функции $H = H_1H_2$. Если подписывающий самостоятельно не осуществит вычисление значения хэш-функции от документа, который он желает подписать, а подпишет представленное ему значение H, то в распоряжении ата-

кующего окажется подпись $S = \alpha^{H^{-1}} = \alpha^{H_1^{-1}H_2^{-1}} (\mathrm{mod}\, n)$, по которой он вычислит подписи $S_1 = S^{H_2} \bmod n$ и $S_2 = S^{H_1} \bmod n$. Значения подписей S_1 и S_2 соответствуют значениям хэш-функций H_1 и H_2 как подлинные подписи. Действительно, данные подписи удовлетворяют уравнению проверки подписи, например: $S_1^{H_1} = S^{H_2 H_1} = S^H = \alpha (\mathrm{mod}\, n)$. Из этого вытекает стандартная рекомендация для пользователей систем ЭЦП: подписывающий должен придерживаться общего правила — не подписывать случайные значения хэш-функций, предоставляемые на подпись посторонними лицами. Рассмотренная атака аналогична нападениям на криптосистему RSA, использующим свойство мультипликативности RSA-преобразования [37].

Кардинальное устранение различных вариантов подобных атак связано со следующим изменением уравнения проверки подписи. Например, можно использовать следующее уравнение проверки подписи: $S^{H^2+H} = \alpha \bmod n$. Однако в этом случае время проверки подписи увеличивается примерно в два раза. Решение указанной проблемы практически без увеличения времени проверки и генерации подписи обеспечивается модификацией схемы ЭЦП, в которой используется уравнение проверки подписи следующего вида $S^{H+H^{<7<}} = \alpha \bmod n$, где $H^{<7<}$ – операция, которая заключается в том, что число H представляется в двоичном виде, интерпретируется как конкатенация 16-битовых двоичных векторов, и над каждым из последних выполняется циклический сдвиг влево на 7 битов.

Более простым вариантом представляется схема с уравнением проверки вида $(S + H)^H = \alpha \bmod n$, которому соответствует уравнение генерации подписи $S = (\alpha^{H^{-1} \bmod \gamma} - H) \bmod n$. Эта схема предпочтительна тем, что не вводятся дополнительные нестандартные операции, а сложность проверки и генерации подписи возрастает только на одну операцию сложения (вычитания).

На выбор элемента α открытого ключа можно наложить требование выполнения следующих двух условий: $\alpha^{\gamma} \bmod n = 1$ (исходное требование) и НОД(α, $\varphi(n)$) = 1 (дополнительное требование). Такое значение α может быть легко найдено, выбирая несколько различных значений β (*см. раздел 2.1*), по которым вычисляются различные значения α, удовлетворяющие исходному требованию. Затем с помощью алгоритма Евклида выбирается такое значение $\alpha = \varepsilon$, которое удовлетворяет также и дополнительному требованию.

Используя открытый ключ (n, ε), легко можно выполнить зашифрование по открытому ключу в соответствии с RSA-шифрованием: $C = M^{\varepsilon} \bmod n$. Для

обеспечения возможности выполнения расшифрования криптограммы дополнительно вычисляется секретная экспонента $\delta = \varepsilon^{-1} \bmod \varphi(n)$. Расшифрование выполняется по формуле $M = C^{\delta} \bmod n$. Очевидно, что при указанном выборе открытого ключа имеется также возможность осуществления процедур генерации и проверки подписи в соответствии со схемой RSA.

По сравнению с RSA процедура генерации подписи в новой схеме является более быстрой, а проверка подписи — более медленной. Новая схема ЭЦП обеспечивает существенное уменьшение суммарного времени генерации и проверки подписи (в 2–4 раза). Сравнение производительности новой схемы ЭЦП с RSA для случая длины модуля $|n|$ = 1024 и 2048 бит приведено в табл. 4.3.

Таблица 4.3

Среднее время, необходимое для выполнения процедур генерации и проверки подписи (в относительных единицах), при длине модуля |n| = 1024 (2048) бит

RSA ($\|e\| = 16$)		**Новая схема ЭЦП**	
		$\|\gamma\| = 161$	$\|\gamma\| = 257$
Генерация подписи	1024 (2048)	161 (161)	257 (257)
Проверка подписи	16 (16)	161 (161)	257 (257)
Суммарное время	1040 (2064)	321(321)	514 (514)

Таким образом, рассмотренная двухключевая криптосистема основана на сложности решения следующих трех задач: факторизации модуля на два больших простых числа, дискретного логарифмирования и извлечения корней по RSA-модулю. Она может быть использована для реализации цифровой подписи и открытого распределения ключей. Так же как и в случае RSA, нахождение эффективных методов решения одной из перечисленных задач для общего случая означает взлом криптосистемы. В настоящее время при длине модуля n более 1024 бит указанные задачи признаются сложными [30, 31], т. е. RSA и новая криптосистема имеют основания для практического применения при решении задач защиты и аутентификации информации.

В новой схеме элемент α открытого ключа (n, α) является генератором циклической группы порядка γ, а само значение γ является секретным, тогда как в RSA в качестве элемента e открытого ключа (n, e) может быть выбрано произвольное число, удовлетворяющее условиям $e < \varphi(n)$ и НОД$(n, e) = 1$, а секретный ключ d вычисляется как число обратное к e по модулю $\varphi(n)$.

Новая криптосхема с открытым ключом является в достаточно полном смысле совместимой с широко используемой криптосистемой RSA. Это позволяет применить различные варианты комбинирования процедур шифрования, открытого распределения ключей и генерации/проверки подписи, относящихся к сравниваемым криптосистемам. Это обеспечивает дополнительную гибкость при практическом использовании этих криптосистем. Совместимость новой криптосхемы с RSA достигается выбором значения α, являющегося взаимно простым с функцией Эйлера от модуля.

4.8.2. Ограничения на выбор параметров криптосхемы

По сравнению с системой RSA в криптосхеме, в которой элементом открытого ключа является число α, относящееся к простому показателю, появляются дополнительные потенциальные возможности найти разложение модуля, используя определенную связь между α и n. Действительно, для некоторого числа β, такого что НОД(β, n) = 1, по теореме Эйлера, имеем $\beta^{\varphi(n)} \equiv 1 \bmod n$, где $\varphi(n) = (p-1)(q-1)$. Допустим, что в каноническом разложении чисел $p-1$ и $q-1$ нет одинаковых множителей. Если мы сформируем параметр α в соответствии с формулой $\alpha = \beta^{\varphi(n)/\gamma} \bmod n \neq 1$, где γ делит $p-1$, но не делит $q-1$, то будем иметь:

$$\alpha = \beta^{\varphi(n)/\gamma} = (\beta^{(q-1)})^{(p-1)/\gamma} \bmod n \Rightarrow \alpha \equiv (\beta^{(q-1)})^{(p-1)/\gamma} \equiv 1^{(p-1)/\gamma} \equiv 1 \bmod q \Rightarrow$$
$$\Rightarrow \alpha - 1 \equiv 0 \bmod q \Rightarrow q \mid \alpha - 1 \Rightarrow \text{НОД}(\alpha - 1, n) = q\ .$$

Таким образом, при таком выборе параметра α разложение модуля представляет простую задачу, решаемую с помощью расширенного алгоритма Евклида.

Для генерации «стойкого» параметра α могут быть использованы следующие два подхода. Первый из них встречался выше. Здесь мы его рассмотрим несколько подробнее. Он состоит в выборе таких чисел p и q, для которых каноническое разложение чисел $p-1$ и $q-1$ содержит единственный одинаковый простой множитель γ, отличный от числа 2, причем γ^2 не делит ни одно из чисел $p-1$ и $q-1$. В этом случае параметр α может формироваться в соответствии с формулой $\alpha = \beta^{L(n)/\gamma} \bmod n \neq 1$, где $L(n)$ = НОК[$p-1$, $q-1$] — обобщенная функция Эйлера, или по формуле

$$\alpha = \beta^{\frac{(p-1)(q-1)}{\gamma^2}} \bmod n \neq 1.$$

В последнем случае имеем:

$$\alpha = \beta^{uv} \bmod n \neq 1,$$

где $u = (p-1)/\gamma$ и $v = (q-1)/\gamma$. Из последней формулы видно, что при выборе в качестве β числа, являющегося первообразным корнем одновременно по mod p и по mod q, выполняются неравенства

$$\alpha \neq 1 \bmod q \quad \text{и} \quad \alpha \neq 1 \bmod p.$$

Действительно, сравнение $\alpha \equiv \beta^{uv} \equiv 1 \bmod q$ или $\alpha \equiv \beta^{uv} \equiv 1 \bmod p$ может выполняться только в случае, если $uv \equiv 0 \bmod (q-1)$ или $uv \equiv 0 \bmod (p-1)$, соответственно. Однако последние соотношения не выполняются, поскольку по построению чисел p и q значение uv не делится ни на $q-1$, ни на $p-1$.

Во втором подходе формируются такие простые числа p и q, для которых каноническое разложение чисел $p-1$ и $q-1$ содержит единственный одинаковый простой множитель — число 2, а также заданные простые множители γ' и γ'' (γ'' | q–1 и γ' | p – 1). В качестве параметра α берется число, относящееся к показателю $\gamma = \gamma'\gamma''$. Оно может формироваться в соответствии с формулой $\alpha = \beta^{\varphi(n)/\gamma} \bmod n \neq 1$. Аналогично первому случаю имеем:

$$\alpha = \beta^{uv} \bmod n \neq 1,$$

где $u = (p-1)/\gamma'$ и $v = (q-1)/\gamma''$, а также $\alpha \neq 1 \bmod q$ и $\alpha \neq 1 \bmod p$ при выборе числа β, являющегося первообразным корнем одновременно по mod p и по mod q. Отметим также, что в рассмотренных двух подходах к формированию параметра α при произвольном выборе числа β сравнение $\alpha \equiv 1 \bmod p$ или $\alpha \equiv 1 \bmod q$ может выполняться только с пренебрежимо малой вероятностью.

4.9. Переход к схемам ЭЦП с простым модулем

4.9.1. Переход к простому модулю в ЭЦП Фиата–Шамира

Рассмотрим еще один вариант ЭЦП, который «выводится» из схемы Фиата–Шамира. В новом варианте используется большой простой модуль p (вместо составного RSA-модуля) и первообразный корень α по модулю p.

Формирование секретного ключа осуществляется следующим образом. Абонент генерирует h случайных чисел $x_0, x_1, \ldots, x_i, \ldots, x_{h-1}$, таких что для $i = 0, 1, \ldots, h-1$ имеет место $1 \le x_i \le p-1$.

Формирование открытого ключа осуществляется путем вычисления значений $y_i = \alpha^{-x_i} \bmod p$. Открытым ключом является набор чисел $y_0, y_1, \ldots, y_i, \ldots, y_{h-1}$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k ($2 \le k \le p-1$).
2. Вычисляется значение $r = \alpha^k \bmod p$.
3. К сообщению m присоединяется число r, и от полученного нового сообщения вычисляется h-битовое значение некоторой специфицированной хэш-функции H: $E = H(m||r)$, которое является первой частью подписи. Значение E представляется в виде последовательности битов e_i: $E = (e_0, e_1, \ldots, e_i, \ldots, e_{h-1})$.
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$S = k + \sum_{i=0}^{h-1} e_i x_i \bmod (p-1).$$

Подписью к сообщению m является пара чисел (E, S).

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r':

$$r' = \alpha^S \cdot \prod_{i=0}^{h-1} y_i^{e_i} \bmod p.$$

2. К сообщению m присоединяется число r', и вычисляется значение хэш-функции $E' = H(m||r')$.
3. Сравниваются значения E и E'. Если $E = E'$, то подпись признается подлинной.

При предварительном вычислении параметра r для многих сообщений сложность процедуры формирования подписи в этом варианте ЭЦП меньше, чем в предыдущих схемах, однако процедура проверки подписи является более сложной, поскольку включает одну операцию возведения в большую степень по модулю p. При указанном условии эта схема, так же как и цифровая подпись Эль-Гамаля (и некоторые другие схемы ЭЦП, основанные на сложности дискретного логарифмирования), обеспечивает наиболее быстрое подписывание сообщений, что может представлять практический интерес для систем скоростной передачи аутентифицированных данных в центр сбора информации.

Длина подписи $|S| + |E|$ превышает $|p|$ на h (h — длина хэш-функции, обычно 128–256 бит), однако $h << |p|$. При выборе в качестве α числа, относящегося к делителю $p-1$ длины 160–256 бит как к показателю по модулю p, длина подписи оказывается существенно меньше длины модуля.

4.9.2. Сокращение размера подписи

Для построения системы ЭЦП с сокращенной длиной подписи в схеме ЭЦП с формированием параметра S по формуле $S = k + \sum_{i=0}^{h-1} e_i x_i \bmod (p-1)$ можно использовать вместо первообразного корня число α, относящееся к некоторому показателю q длины 160–256 бит (очевидно, что в этом случае потребуется сгенерировать такое число p, чтобы в разложении $p-1$ содержался множитель q). Тогда параметр S будет вычисляться по модулю q, что приведет к существенному уменьшению его длины. Кроме того, для уменьшения размера секретного и открытого ключей в формуле $S = k + \sum_{i=0}^{h-1} e_i x_i \bmod q$ вместо суммы $\sum_{i=0}^{h-1} e_i x_i \bmod q$ можно использовать произведение $E \cdot x \bmod q$, где x — секретный ключ. В результате мы приходим к следующей системе ЭЦП.

Формирование секретного ключа осуществляется как генерирование случайного числа x, $1 < x \le p-1$.

Формирование открытого ключа осуществляется по формуле $y = \alpha^{-x} \bmod p$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k ($k < q$).
2. Вычисляется значение $r = \alpha^k \bmod p$.
3. К сообщению m присоединяется число r, и от полученного нового сообщения вычисляется h-битовое (h = 128–256) значение некоторой специфицированной хэш-функции H: $E = H(m||r)$, которое является первой частью подписи. Значение E интерпретируется как двоичное число.
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:

$$S = k + Ex \bmod q .$$

Подписью к сообщению m является пара чисел (E, S). Длина подписи в этой схеме составляет от 256 до 512 бит, причем не зависит от размера числа p, которое определяет стойкость ЭЦП.

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r':

$$r' = \alpha^S \cdot y^E \bmod p .$$

2. К сообщению m присоединяется число r', и вычисляется значение хэш-функции $E' = H(m||r')$.
3. Сравниваются значения E и E'. Если $E = E'$, то подпись признается подлинной. (Действительно, имеем:
$$r' = \alpha^S \cdot y^E \bmod p = \alpha^{k+Ex \bmod q} \cdot (\alpha^{-x})^E \bmod p = \alpha^k \bmod p = r .)$$

Достоинством этой схемы является малый размер подписи при сохранении высокого уровня стойкости. Фактически мы пришли к схеме подписи Шнорра.

4.9.3. Цифровая подпись Шнорра

Схема ЭЦП Шнорра [39, 40] аналогична рассмотренным ранее системам, основанным на сложности задачи дискретного логарифмирования. Общими для абонентов являются следующие параметры: p и q — большие простые числа, такие что q делит $p - 1$; α — число, относящееся к показателю q по модулю p.

Секретный ключ представляет собой случайно генерируемое число x, $1 < x < q$.

Формирование открытого ключа осуществляется путем возведения числа α в степень x по модулю p: $y = \alpha^x \bmod p$.

Вычисление подписи к сообщению m включает следующие шаги.

1. Генерируется случайное число k, $1 < k < q$, играющее роль разового секретного ключа, обеспечивающего маскирование ключа x.
2. Вычисляется значение $r = \alpha^k \bmod p$.
3. К сообщению m присоединяется число r, и вычисляется хэш-функция H от значения $m||r$: $E = H(m||r)$. Значение E является первой частью подписи.
4. Вычисляется вторая часть подписи в соответствии со следующей формулой:
$$S = k - xE \bmod q .$$

Таким образом, формирование подписи (E, S) завершено.

Проверка подлинности подписи осуществляется следующим образом.

1. Вычисляется значение r': $r' = \alpha^S y^E \bmod p$.

2. К сообщению m присоединяется число r', и вычисляется значение хэш-функции $E' = H(m \| r')$.
3. Сравниваются значения E и E'. Если $E = E'$, то результат проверки подлинности подписи признается положительным.

Достоинством схемы подписи Шнорра является возможность выбора такого конкретного варианта реализации, при котором обеспечивается сравнительно малая длина подписи. Например, при 160-битовом простом числе q и 160-битовой хэш-функции размер подписи составит 320 бит. Поскольку перед хэшированием подписываемого сообщения к нему присоединяется случайное значение r, то предварительно найденные коллизии окажутся практически бесполезными для нарушителя, если при вычислении хэш-функции в качестве начальных блоков данных использовать фрагменты двоичного вектора, представляющего значение r. Это замечание имеет больше теоретический характер, поскольку используемые на практике хэш-функции являются стойкими к поиску коллизий.

ГЛАВА 5

Открытое распределение ключей и открытое шифрование

Наиболее привлекательной системой открытого распределения ключей является система Диффи–Хеллмана. Она часто используется как двухключевой компонент гибридных криптосистем. Достоинство этой системы состоит в том, что сразу же после распределения подлинных открытых ключей (т. е. для которых выполнена процедура аутентификации) для каждой пары абонентов задан общий секрет (который может быть использован как ключ симметричного шифрования). Для вычисления этого секрета им не требуется направлять друг другу какие-либо сообщения. Каждый из них может вычислить секрет, разделяемый с любым другим пользователем, используя свой секретный ключ и справочник открытых ключей. В криптосистеме RSA решение задачи распределения открытых ключей приводит только к появлению возможности распределения секретного ключа симметричной криптосистемы по открытому каналу связи. В системе RSA и других двухключевых шифрах задача открытого распределения ключей симметричного шифрования решается путем зашифрования последних на открытом ключе пользователя, причем возможность проверки подлинности отправителя обеспечивается тем, что отправитель подписывает передаваемый ключ (подпись к ключу также шифруется на открытом ключе получателя).

Двухключевая криптосистема, которая позволяет осуществлять шифрование по открытому ключу и подписывание сообщений по секретному, может быть использована для открытого распределения ключей. При этом для того, чтобы пара абонентов стала обладать общим секретом, достаточно передачи одного сообщения от одного из абонентов другому.

Если двухключевая криптосистема позволяет осуществлять открытое шифрование, то для формирования общего секрета нужно передать два сообщения. В противном случае получатель зашифрованного ключа симметричной криптосистемы не будет иметь возможности проверить подлинность отправителя. При передаче еще одного встречного сообщения взаимная проверка подлинности абонентов A и B осуществляется следующим образом:

1. Абонент A выбирает случайный секретный ключ K_A, зашифровывает его на открытом ключе абонента B и направляет криптограмму $C_1 = E(K_A, y_B)$ абоненту B.
2. Абонент B расшифровывает криптограмму C_1 по своему секретному ключу x_B: $K_A = D(C_1, x_B)$, выбирает случайный секретный ключ K_B, вычисляет общий секрет
$$K_{AB} = K_A K_B,$$
зашифровывает K_B на открытом ключе абонента A и направляет криптограмму $C_2 = E(K_B, y_A)$ абоненту A.
3. Абонент A расшифровывает криптограмму C_2 по своему секретному ключу x_A: $K_B = D(C_2, x_A)$ и вычисляет общий секрет
$$K_{AB} = K_A K_B.$$
4. Абоненты признают друг друга подлинными, если протокол рукопожатия дает положительную проверку подлинности при использовании секрета K_{AB}.

Ниже рассматриваются две системы открытого шифрования, которые могут быть применены в рамках этой схемы формирования общего секрета у двух удаленных абонентов.

***Замечание* 1.** Для того чтобы нарушитель получил значение K_{AB}, он должен вычислить x_A либо x_B по соответствующему открытому ключу. Но мы предполагаем, что такая задача практически невыполнима.

***Замечание* 2.** В двухключевых криптосистемах реализация функций формирования/проверки цифровой подписи и открытого шифрования может основываться на отличающихся механизмах. Если реализована одна функция, то не всегда очевидно, как можно реализовать другую.

5.1. Схема открытого шифрования Рабина

Алгоритм открытого шифрования, предложенный М. О. Рабином [34], основан на возведении блоков открытого текста M длины $|M| < |n|$ в квадрат по составному модулю n. Стойкость данного метода шифрования определяется сложностью разложения модуля n на простые множители. С этой целью используется RSA-модуль $n = pq$ большого размера (1024–2048 бит). Особенностью формирования модуля является то, что простые множители p и q имеют специальный вид, а именно сравнимы с числом 3 по модулю 4, т. е. для них выполняются следующие соотношения:

$$p = 3 \bmod 4,$$
$$q = 3 \bmod 4.$$

Это позволяет владельцу секретного ключа, который знает разложение модуля, достаточно просто вычислять корни второй степени из квадратичных вычетов по модулю n. Действительно, пусть a — квадратичный вычет, причем НОД(a, n) = 1. В соответствии с нашим допущением существует такое M, что $M^2 = a \bmod n$. В соответствии с теоремой, обратной теореме 11 (*глава 2*), выполняются также сравнения $M^2 = a \bmod p$ и $M^2 = a \bmod q$, т. е. имеем $m_p = a^{1/2} \bmod p$ и $m_q = a^{1/2} \bmod q$. Известно, что для квадратичных вычетов по простому модулю p выполняется сравнение $a^{\frac{p-1}{2}} = 1 \bmod p$, откуда для случая $p = 3 \bmod 4$ легко получить:

$$a = a \cdot a^{\frac{p-1}{2}} = a^{\frac{p+1}{2}} \bmod p \Rightarrow a^{1/2} = a^{\frac{p+1}{4}} \bmod p .$$

Теперь учтем, что если $m_p^2 = a \bmod p$, то также имеем $(-m_p)^2 = a \bmod p$. Таким образом, имеем следующие два корня из a по модулю p:

$$m_{p_1} = a^{\frac{p+1}{4}} \bmod p \quad \text{и}$$

$$m_{p_2} = p - m_{p_1} = p - a^{\frac{p+1}{4}} \bmod p$$

и следующие два корня по модулю q:

$$m_{q_1} = a^{\frac{q+1}{4}} \bmod q \quad \text{и}$$

$$m_{q_2} = q - m_{q_1} = q - a^{\frac{q+1}{4}} \bmod q .$$

Используя китайскую теорему об остатках и следующие пары чисел (m_{p_1}, m_{q_1}), (m_{p_1}, m_{q_2}), (m_{p_2}, m_{q_1}) и (m_{p_2}, m_{q_2}) легко найти четыре корня m_1, m_2, m_3 и m_4 второй степени из a по модулю n. Например, m_1 вычисляется по следующей формуле:

$$m_1 = \left(m_{p_1} q (q^{-1} \bmod p) + m_{q_1} p (p^{-1} \bmod q) \right) \bmod n .$$

Система шифрования М. Рабина включает следующие элементы.

Формирование секретного ключа заключается в формировании больших простых чисел p и q, сравнимых с числом 3 по модулю 4.

Формирование открытого ключа n осуществляется путем перемножения чисел p и q: $n = pq$.

Процедура зашифрования сообщения M предельно проста, а именно состоит в возведении числа M (сообщение интерпретируется как число) в квадрат по модулю открытого ключа:

$$C = M^2 \bmod n.$$

Процедура расшифрования криптограммы C несколько сложнее. Она состоит в извлечении квадратного корня из криптограммы по модулю n. В силу процедуры зашифрования корректная криптограмма является квадратичным вычетом по модулю открытого ключа. Для вычисления квадратных корней из криптограммы предварительно вычисляют следующие значения, соответствующие корням по модулям p и q:

$$m_{p_1} = C^{\frac{p+1}{4}} \bmod p \;;\quad m_{p_2} = p - m_{p_1} = p - C^{\frac{p+1}{4}} \bmod p \;;$$

$$m_{q_1} = C^{\frac{q+1}{4}} \bmod q \;;\quad m_{q_2} = q - m_{q_1} = q - C^{\frac{q+1}{4}} \bmod q \;.$$

В соответствии с приведенными предварительными сведениями существует четыре различных корня, которые обозначим как M_1, M_2, M_3 и M_4:

$$M_1 = \left(m_{p_1} a + m_{q_1} b \right) \bmod n \;;$$

$$M_2 = \left(m_{p_1} a + m_{q_2} b \right) \bmod n \;;$$

$$M_3 = \left(m_{p_2} a + m_{q_1} b \right) \bmod n \;;$$

$$M_4 = \left(m_{p_2} a + m_{q_2} b \right) \bmod n \;;$$

где $a = q(q^{-1} \bmod p)$ и $b = p(p^{-1} \bmod q)$. Таким образом, расшифрование формально неоднозначно, однако обычно не представляет труда выбрать правильный открытый текст. Для формализации этого выбора перед зашифрованием к исходному открытому сообщению можно присоединять некоторую заранее оговоренную метку, представляющую собой, например, 32-битовое число.

Для устранения этого недостатка Х. Вильямс модифицировал схему шифрования Рабина [46]. Другие варианты модернизации этой схемы представлены в работах [44] и [29].

5.2. Схемы на основе сложности извлечения корней по модулю

5.2.1. Открытое шифрование

Рассмотрим схему вероятностного открытого шифрования, основанную на модульном возведении в степень разового ключа шифрования, передаваемого совместно с каждой порцией зашифрованных данных. Идея передачи разового ключа шифрования K с каждой порцией зашифрованных данных ранее нам встречалась в системе открытого шифрования Эль-Гамаля. Шифры с таким механизмом являются вероятностными, и длина криптограммы превышает в два раза длину блока исходного текста.

Чтобы ключ шифрования блока данных оказался доступен только владельцу открытого ключа, получателю информации передается не сам разовый ключ K, а некоторый его образ в виде числа R, являющегося целой степенью от K: $R = K^t \bmod n$, где $t = 2, 3, \ldots$ — натуральное число. Причем, чтобы сделать практически невозможным извлечение корня степени t из R, в качестве модуля n выбирается составное число $n = pq$, где p и q — большие простые числа, генерируемые таким образом, чтобы оба числа $q-1$ и $p-1$ содержали одинаковый большой (например 160-битовый) простой делитель q'. Числа p, q и q' составляют секретный ключ. Модуль n будет представлять первую часть открытого ключа. Второй частью открытого ключа является число α, относящееся к показателю q' по модулю n. Рассмотрим схему шифрования. Для того чтобы владелец открытого ключа смог легко вычислить ключ шифрования K по его образу R, значение K может выбираться из множества чисел, относящихся к показателю q' по модулю n.

Секретный ключ представляет собой тройку простых чисел p, q и q'.

Формирование открытого ключа (n, α) осуществляется путем перемножения чисел p и q, в результате чего получаем значение модуля $n = pq$, по которому будут осуществляться вычисления, и генерации числа α, относящегося к показателю q'. Двоичное число α находят следующим образом. Генерируют случайное число β и вычисляют дополнительный параметр $g = \varphi(n)/q'^2$, после чего получают число $\alpha = \beta^g \bmod n$. Если вычисленное $\alpha \neq 1$, то оно берется в качестве второй части открытого ключа. (В противном случае генерируется другое случайное число β и вычисляется новое значение $\alpha = \beta^g \bmod n$.)

Процедура зашифрования сообщения m включает следующие шаги.

1. Генерируется случайное число k, $1 \le k \le q' - 1$, и вычисляется $K = \alpha^k \bmod n$ (разовый ключ шифрования блока данных).
2. Зашифровывается сообщение путем умножения m на K по модулю n: $C = Km \bmod n$.
3. Зашифровывается ключ K путем возведения его в степень t: $R = K^t \bmod n$.
4. Формируется и отправляется владельцу открытого ключа криптограмма (R, C).

Процедура расшифрования криптограммы (R, C) выполняется следующим образом.

1. Вычисляется разовый ключ расшифрования блока данных: $K^{-1} = R^{-1/t \bmod q'} \bmod n$.
2. Расшифровывается шифртекст C: $K^{-1}C \bmod n = m$. (Поскольку число K было сформировано по формуле $K = \alpha^k \bmod n$, где НОД$(\alpha, n) = 1$, то K и n — взаимно простые числа, т. е. число $K^{-1} \bmod n$ существует и может быть вычислено с помощью расширенного алгоритма Евклида.)

Может быть применен альтернативный вариант шифрования, включающий следующие процедуры.

Альтернативная процедура зашифрования включает следующие шаги.

1. Генерируется случайное число k, $1 \le k \le q' - 1$, и вычисляется $K = \alpha^k \bmod n$ (разовый ключ шифрования блока данных).
2. Если НОД$(K, \varphi(n)) \ne 1$, то перейти к шагу 1.
3. Зашифровывается сообщение путем возведения числа m в степень K по модулю n: $C = m^K \bmod n$.
4. Зашифровывается ключ K путем возведения его в степень t: $R = K^t \bmod n$.
5. Формируется и отправляется владельцу открытого ключа криптограмма (R, C).

Альтернативная процедура расшифрования включает следующие шаги.

1. Вычисляется разовый ключ зашифрования блока данных: $K = R^{1/t \bmod q'} \bmod n$.
2. Вычисляется ключ расшифрования: $K' = K^{-1} \bmod \varphi(n)$.
3. Расшифровывается шифртекст C: $C^{K'} \bmod n = m$.

Замечание. Поскольку на процесс зашифрования каждого блока данных влияет случайно выбираемое число k, то данный способ открытого шифрования является вероятностным. Его стойкость основана на том, что извлечение корня степени t на множестве чисел

$$\{\alpha^1 \bmod n, \alpha^2 \bmod n, \ldots, \alpha^{q'-1} \bmod n, 1\}$$

является трудной задачей, если неизвестно число q' или разложение модуля. В качестве параметра α можно использовать и число, относящееся к некоторому составному показателю γ, однако в этом случае на выбор значения t накладывается следующее ограничение: число t должно быть взаимно простым с γ. Ограничения на выбор чисел p, q и α рассмотрены в *разделе 4.8.*

5.2.2. Схема с сокращенной длиной открытого ключа

Открытое шифрование, основанное на сложности задачи извлечения корня степени $t \geq 3$ по RSA-модулю, может быть построено по другой схеме, которая в качестве открытого ключа использует только само значение модуля n (не требуется использования дополнительного специального числа α). Пусть задано некоторое значение степени $t \geq 3$. Открытый ключ n формируется следующим образом. Выбираются два больших простых числа p и q, вычисляется значение $n = pq$, после чего проверяется выполнимость условия НОД($\varphi(n)$, t) = 1. Если последнее условие не выполняется, то выбираются новые случайные числа p и q до тех пор, пока не будет выполнено последнее условие. Таким образом, формируется такой открытый ключ n, что для t существует обратный элемент $u = t^{-1}$ по модулю $\varphi(n)$. Благодаря этому можно задать следующую процедуру зашифрования сообщения $m < n$:

$$c = m^t \bmod n.$$

Соответствующая процедура расшифрования имеет вид:

$$m = c^u \bmod n = c^{1/t \bmod \varphi(n)} \bmod n.$$

Значение $u = t^{-1} \bmod \varphi(n)$ является секретным ключом, соответствующим данному открытому ключу n. Поскольку разложение числа n держится в секрете, то вычисление m по t и n является трудной задачей.

Нетрудно установить, что возведение значения хэш-функции $H = H(m)$ от сообщения m в степень u может быть использовано как процедура генерации подписи S к сообщению m:

$$S = m^u \bmod n.$$

Уравнением проверки подписи является следующее:

$$m = S^t \bmod n.$$

Сравнение с криптосистемой RSA показывает, что мы пришли к частному варианту ее реализации, когда для всех пользователей устанавливается одно и то же значение открытой экспоненты $e = t$.

5.2.3. Открытое распределение ключей

В рамках рассмотренной выше схемы вероятностного открытого шифрования (*см. раздел 5.2.1*) использовалась передача разовых ключей, зашифрованных путем их возведения в натуральную степень $t \geq 2$. Фактически как ее составная часть была рассмотрена также и система открытого распределения ключей, основанная на возведении ключа шифрования в степень $t > 1$ по составному модулю, являющемуся частью открытого ключа получателя (n, α), который соответствует секретному ключу, представляющему собой тройку простых чисел p, q и q', где $q' | \varphi(n)$ и $n = pq$. Таким образом, имеем следующую схему распределения открытых ключей, которая может быть использована в гибридных криптосистемах.

Процедура зашифрования ключа, распределяемого по открытому каналу, включает следующие шаги.

- Генерируется случайное число k, $1 \leq k \leq q' - 1$, и вычисляется ключ $K = \alpha^k \bmod n$, предназначенный для шифрования данных с использованием некоторой симметричной криптосистемы.
- Зашифровывается ключ K путем возведения его в степень t: $R = K^t \bmod n$.
- Зашифрованный ключ в виде числа R направляется получателю информации, который является владельцем открытого ключа.

Процедура расшифрования ключа симметричного шифрования выполняется по формуле:

$$K = R^{1/t \bmod q'} \bmod n.$$

Теперь у получателя имеется секретный ключ, однако подлинность отправителя еще требует подтверждения. Следующим шагом формирования общего секрета является передача еще одного ключа, выбираемого абонентом, который ранее был получателем. При этом второй ключ зашифровывается перед отправкой на открытом ключе абонента, который ранее был отправителем. После распределения двух ключей общий ключ симметричного шифрования вычисляется, например, как сумма или произведение этих ключей (очевидно, существует и ряд других вариантов формирования общего

секрета на основе пары одинаковых ключей, которые имеются у каждого из двух подлинных абонентов). Если один из участников протокола формирования общего секрета является нарушителем, то у них окажутся разные значения общего секрета.

В более конкретном представлении алгоритм открытого распределения ключей имеет следующий вид.

1. Пользователь А генерирует случайное целое число k_A, удовлетворяющее условию $1 \le k_A \le 2^{256}$, вычисляет ключ $K_A = \alpha_B^{k_A} \bmod n_B$, преобразует ключ K_A в соответствии с формулой: $R_A = K_A^t \bmod n_B$, где t — специфицированный параметр преобразования, и отправляет значение R_A пользователю В.
2. Пользователь В генерирует случайное целое число k_B, удовлетворяющее условию $1 \le k_B \le 2^{256}$, вычисляет ключ $K_B = \alpha_A^{k_B} \bmod n_A$, преобразует ключ K_B: $R_B = K_B^t \bmod n_A$ и отправляет значение R_B пользователю А.
3. Пользователь А восстанавливает по R_B ключ K_B: $K_B = R_B^{1/t \bmod q'_A} \bmod n_A$ и вычисляет общий секрет K_{AB}: $K_{AB} = K_A K_B$.
4. Пользователь В восстанавливает по R_A ключ K_A: $K_A = R_A^{1/t \bmod q'_B} \bmod n_B$ и вычисляет общий секрет: $K_{AB} = K_A K_B$.

В результате выполненного протокола оба абонента стали обладателями общего секрета. Общий секрет может быть использован в качестве ключа симметричной криптосистемы для шифрования данных, передаваемых по открытому каналу. Если нарушитель попытается выдать себя за одного из легальных абонентов (например за абонента В), то это легко обнаружится, так как направляемые им сообщения будут преобразовываться в случайный текст на стороне легального абонента. Действительно, нарушитель не имеет возможности восстановить ключ K_A по значению R_A, поскольку он не владеет секретным ключом абонента В. Сообщения, направленные нарушителю, последний не сможет расшифровать по той же причине. С целью взаимной проверки подлинности абонентов можно непосредственно после шага 4 выполнить протокол «Запрос-ответ» [2] с использованием секрета K_{AB} в качестве ключа аутентификации.

Открытое распределение ключей может быть выполнено также и с использованием процедуры подписывания ключа:

1. Пользователь А генерирует случайное целое число k, удовлетворяющее условию $1 \le k \le 2^{256}$, вычисляет ключ $K = \alpha_B^{k} \bmod n_B$, преоб-

разует ключ K в соответствии с формулой $R = K^t \bmod n_B$, генерирует свою подпись S_K, соответствующую K (или S_R, соответствующую R), и отправляет значения R и S_K (или R и S_R) пользователю B.

2. Пользователь B восстанавливает по R ключ K: $K = R^{1/t \bmod q'_B} \bmod n_B$ и по открытому ключу (n_A, α_A) пользователя A проверяет подлинность подписи S_K (или S_R).

В результате этого протокола пользователь B получает секретный ключ пользователя A, в подлинности которого он уверен. Отметим, что пересылка подписи S_K по открытому каналу требует использования такой схемы ЭЦП, которая не позволяет восстанавливать значение подписанного сообщения по открытому ключу при проверке подписи.

Система открытого распределения ключей Диффи–Хеллмана имеет то преимущество, что для каждой пары абонентов общий секрет может быть сформирован без передачи каких-либо дополнительных сообщений сразу же, как только абоненты получат подлинные открытые ключи. Однако разовый общий секрет является фиксированным, поэтому он используется как мастер ключ для шифрования сеансовых ключей или других вспомогательных ключей, обеспечивающих работу симметричной криптосистемы. Вспомогательные ключи зашифровываются по мастер ключу и пересылаются по открытым каналам, т. е. в гибридной криптосистеме, использующей алгоритм Диффи–Хеллмана, также присутствует шаг пересылки ключа шифрования по открытому каналу.

В некоторых случаях руководство компании может захотеть обеспечить секретность сообщений, которыми обмениваются подразделения компании или ее сотрудники, но при этом иметь возможность получить доступ к зашифрованной информации. Для таких случаев может оказаться интересным применение гибридных криптосистем, в которых открытое распределение ключей осуществляется с помощью схемы, основанной на сложности извлечения корней по составному модулю.

Руководство компании формирует составной модуль $n = pq$, где p и q — большие простые числа, такие что обеспечивается высокая сложность задачи факторизации. Зная разложение модуля, которое держится в секрете, ответственное лицо может при необходимости восстановить секретные ключи пользователей и затем расшифровать переданные сообщения. Рассмотрим следующие два варианта реализации такой системы открытого распределения ключей.

5.2.4. Схема на основе сложности извлечения корней второй степени

Формирование секретного ключа. В качестве секретного ключа каждый пользователь выбирает некоторое случайное число x, не превосходящее $n-1$.

Формирование открытого ключа y осуществляется путем вычисления квадрата числа x по модулю n: $y = x^2 \bmod n$.

Формирование общего секретного ключа. Пусть i-й и j-й пользователи хотят сформировать общий секретный ключ симметричного шифрования. Для этого i-й пользователь отправляет j-му пользователю значение $R_i = (y_i y_j)^{x_i} \bmod n$, где индексы указывают на принадлежность соответствующих ключей i-му и j-му пользователям. Со своей стороны, j-й пользователь отправляет i-му пользователю значение $R_j = (y_i y_j)^{x_j} \bmod n$. Получив значение R_i, j-й пользователь формирует число $R_{ij} = (R_i)^{x_j} = (y_i y_j)^{x_i x_j} \bmod n$, а i-й — число $R_{ji} = (R_j)^{x_i} = (y_i y_j)^{x_j x_i} \bmod n$. Нетрудно заметить, что значения R_{ij} и R_{ji} одинаковы и могут быть использованы в качестве общего секрета в симметричной криптосистеме. Общий секрет R_{ji} определяется парой секретных ключей x_i и x_j. Его значение предопределено выбором пары секретных ключей (аналогично схеме открытого распределения ключей Диффи–Хеллмана). Однако для формирования общего секрета пользователи должны обменяться числами R_i и R_j. В системе Диффи–Хеллмана этого не требуется. Необходимость предварительного обмена вспомогательными сообщениями может быть использована для того, чтобы обеспечить формирование случайного общего секрета. Это может быть обеспечено, например, формированием общего числа R_i i-м пользователем по формуле $R_i = (y_i y_j r)^{x_i} \bmod n$ и передачей j-му пользователю пары чисел R_i и r, который возвращает значение $R_j = (y_i y_j r)^{x_j}$. При формировании случайных значений общего секрета можно реже осуществлять смену открытых ключей с целью обновления общего секретного ключа.

Добавляя шаг обмена некоторыми числами, в схеме Диффи–Хеллмана также нетрудно обеспечить формирование случайного общего секрета для каждого сеанса связи. Например, это можно осуществить следующим путем.

1. Пользователь B генерирует случайные числа $x_B \le p-1$ и $k \le p-1$, вычисляет числа $r = \alpha^k \bmod p$ и $R_B = r^{x_B} \bmod p$ и направляет значения R_B и r пользователю A.

2. Пользователь A генерирует случайное число $x_A \le p-1$, вычисляет число $R_A = r^{x_A} \bmod p$ и направляет значение R_A пользователю B.

3. Пользователи A и B вычисляют общий секрет по следующим формулам:

$$\text{A}: (R_B Y_B)^{x_A} = \alpha^{kx_B x_A} \alpha^{x_B x_A} = \alpha^{kx_B x_A + x_B x_A} \bmod p;$$

$$\text{B}: (R_A Y_A)^{x_B} = \alpha^{kx_A x_B} \alpha^{x_A x_B} = \alpha^{kx_A x_B + x_A x_B} \bmod p.$$

Отметим, что числа R_A и R_B, которыми обмениваются пользователи, фактически являются вспомогательными открытыми ключами, подлинность которых подтверждается тем, что значения случайного общего секрета, сформированные пользователями A и B, одинаковы. Последнее легко проверяется пробным шифрованием некоторого контрольного сообщения. Подлинность общего секрета основана на подлинности открытых ключей $Y_A = \alpha^{x_A} \bmod p$ и $Y_B = \alpha^{x_B} \bmod p$.

5.3. Открытое распределение общего ключа между тремя и более пользователями

Система открытого распределения ключей Диффи–Хеллмана может быть использована для формирования общего секрета не только для каждой пары пользователей, но и для троек, четверок и так далее пользователей. Рассмотрим случай формирования общего ключа для трех пользователей.

Пользователи *A, B* и *C*, используя общее простое число p и первообразный корень по модулю p, формируют свои открытые ключи: $Y_A = \alpha^{x_A} \bmod p$, $Y_B = \alpha^{x_B} \bmod p$ и $Y_C = \alpha^{x_C} \bmod p$, соответственно, передают их в удостоверяющий центр, который подтверждает их подлинность своей цифровой подписью и распределяет их между пользователями.

Пользователь *A* посылает пользователю *B* число $Y_{AC} = Y_C^{x_A} \bmod p$ и пользователю *C* — число $Y_{AB} = Y_B^{x_A} \bmod p$. Пользователи *B* и *C* формируют общий секретный ключ: $Z_{ABC} = (Y_{AC})^{x_B} = Y_C^{x_A x_B} = \alpha^{x_C x_A x_B} (\bmod p)$ и $Z_{ABC} = (Y_{AB})^{x_C} = Y_B^{x_A x_C} = \alpha^{x_B x_A x_C} (\bmod p)$, соответственно. Пользователь *B* посылает пользователю *A* число $Y_{CB} = Y_C^{x_B} \bmod p$, после чего пользователь *A*

также получает возможность сформировать общий секрет: $Z_{ABC} = (Y_{CB})^{x_A} = Y_C^{x_B x_A} = \alpha^{x_C x_B x_A} \pmod p$.

Протокол обмена ключами Диффи–Хеллмана, расширенный на случай с тремя участниками, представлен на рис. 5.1. Аналогичным образом можно сформировать общий секрет для четырех, пяти и более участников секретной телеконференции.

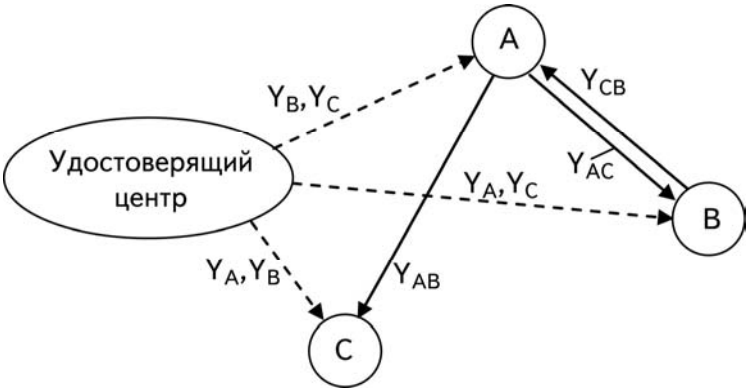


Рис. 5.1. Схема формирования общего секретного ключа

В рассмотренной схеме имеется следующий недостаток. Формирование общего секрета для трех пользователей требует раскрытия общего секрета для двух пользователей третьему, например, пользователю А предоставляется число $Y_{CB} = Y_C^{x_B} \bmod p$, которое является общим секретом для пользователей В и С в исходном варианте алгоритма открытого распределения ключей Диффи–Хеллмана. Этот недостаток устраняется в следующей схеме формирования общего секрета для трех пользователей.

Формируется общий открытый ключ Y для трех пользователей: $Y = Y_A Y_B Y_C \bmod p$. Пользователь А посылает пользователям В и С число $Z_A = Y^{x_A} \bmod p$. Пользователь В отправляет пользователю С числа $Z_B = Y^{x_B} \bmod p$ и $Z_{AB} = Z_A^{x_B} \bmod p$. Пользователь С посылает пользователю A число $Z_{BC} = Z_B^{x_C} \bmod p$ и пользователю В — число $Z_{AC} = Z_A^{x_C} \bmod p$. После этого пользователи А, В и С вычисляют общий для них секретный ключ по следующим формулам:

$$A: \quad Z_{ABC} = Z_{BC}^{x_A} \bmod p = Y^{x_B x_C x_A} \bmod p;$$

$$B: \ Z_{ABC} = Z_{AC}^{x_B} \bmod p = Y^{x_A x_C x_B} \bmod p;$$

$$C: \ Z_{ABC} = Z_{AB}^{x_C} \bmod p = Y^{x_A x_B x_C} \bmod p.$$

В этой схеме общий секрет каждой пары пользователей остался секретным для третьего пользователя.

5.4. Вероятностные механизмы в двухключевых шифрах

В двухключевых криптосистемах шифрование выполняется по общеизвестному алгоритму шифрования $\mathbf{E}_z$ (открытому ключу), поэтому принципиально возможна следующая атака. При наличии шифртекста $C = \mathbf{E}_z(T)$, для того чтобы найти исходный текст T, криптоаналитик может выбирать различные варианты возможных открытых текстов T_i' и вычислять соответствующие им криптограммы $C_1' = \mathbf{E}_z(T_1')$, $C_2' = \mathbf{E}_z(T_2')$, ... , $C_m' = \mathbf{E}_z(T_m')$. Если он угадает истинный исходный текст, то этот факт он установит по равенству $C' = C$. В такой схеме криптоанализа каждая неудачная попытка дает криптоаналитику определенную информацию, поскольку она исключает количество оставшихся вариантов возможных исходных текстов.

Естественно, что вероятность выбора правильного исходного текста крайне низка для обычно используемых размеров входного сообщения, поэтому такая атака на практике не может привести к успеху с существенной вероятностью. Сейчас нас интересует только сам факт утечки определенной информации об исходном сообщении. Можно, однако, представить следующий случай, когда подобная атака имеет высокую результативность. Допустим, что распоряжения руководящего центра (открытые тексты) рассылаются исполнительным органам в виде документов определенного формата и стиля. В этом случае различные варианты открытого текста отличаются только в определенных полях электронного бланка. Кроме того, в определенных случаях стандартный стиль распоряжений может привести к тому, что криптоаналитику будет неизвестна только малая часть исходного сообщения. Например, дата, объем перечисляемой суммы денег, имя исполнителя и т. п.

Вероятностные механизмы позволяют предотвратить утечку информации в приведенном выше способе криптоанализа. Так же как и в случае одноключевых криптосистем, при использовании вероятностных механизмов шифрования в двухключевых шифрах данному исходному тексту соответствует множество возможных криптограмм $\{C_1, C_2, \ldots, C_N\}$, каждая из которых дешифруется с помощью секретного алгоритма дешифрования $\mathbf{D}_z$ (секретного

ключа) в один и тот же исходный текст T: $T = \mathbf{D}_z(C_1) = \mathbf{D}_z(C_2) = ... = \mathbf{D}_z(C_N)$. Это возможно только в том случае, если длина шифртекстов превышает размер исходного текста. Если длина шифртекста больше длины исходного текста на r бит, то можно построить такой вероятностный механизм, для которого число шифртекстов, соответствующих данному входному тексту, составляет $N = 2^r$. При вероятностном шифровании для данного T формируется одна из возможных криптограмм, например C_i, выбираемая из множества $\{C_1, C_2, ... , C_N\}$ по случайному закону.

Криптоаналитик может правильно выбрать исходный текст, но он не сможет проверить этот факт, поскольку при шифровании он получит в общем случае другую криптограмму из множества возможных вариантов. Криптоаналитик, зашифровав T, получит $C_j = \mathbf{E}_z(T)$. Вероятность того, что $C_j = C_i$, составляет $1/N$. Криптоаналитику надо угадать не только исходный текст, но и значение некоторого случайно выбираемого параметра, который управляет процессом вероятностного шифрования.

Рассмотренный ранее способ шифрования Эль-Гамаля (*см. раздел 3.3.2*) использует вероятностный механизм преобразования исходного текста. Этот способ основан на процедурах возведения в дискретную степень и фактически реализует гибридную криптосхему с использованием метода открытого распределения ключей Диффи–Хеллмана, в котором выбирается большое простое число p и соответствующий ему первообразный корень α. Каждый пользователь секретной сети выбирает секретный ключ x, вычисляет свой открытый ключ $y = \alpha^x \bmod p$ и помещает y в заверенный справочник. В рамках открытого шифра Эль-Гамаля осуществляется открытое распределение разового ключа шифрования как разового общего секрета для двух пользователей и симметричное шифрование блока данных путем умножения разового ключа шифрования на значение блока открытого текста по модулю p. Чтобы послать пользователю, который является владельцем открытого ключа y, секретное сообщение T, отправитель выполняет следующие шаги:

1. Выбрать случайное число $R < p - 1$ (использование этого случайного значения предопределяет вероятностный характер преобразования текущего блока данных).
2. Вычислить значение $C' = \alpha^R \pmod p$.
3. По открытому ключу i-го пользователя вычислить $C'' = y^R T \pmod p$.
4. Отправить криптограмму (C', C'') пользователю i.

В этом методе длина шифртекста примерно в два раза больше длины исходного текста и данному открытому тексту соответствуют не менее 2^{k-1} различных криптограмм (k — длина модуля p в битах). Пользователь i, получив

криптограмму (C', C''), может легко вычислить исходный текст: $T = C''/(C')^x \pmod p$. Действительно, $(C')^x = (\alpha^R)^x = \alpha^{Rx} \pmod p$ и $C''/(C')^x = y^R T/\alpha^{Rx} = (\alpha^x)^R T/\alpha^{Rx} = \alpha^{xR} T/\alpha^{Rx} = T \pmod p$.

Схема шифрования по открытому ключу, описанная в *разделе 5.2.1*, также относится к вероятностному типу. Для двухключевых шифров общим методом введения случайности в процесс шифрования может также служить простой механизм присоединения случайных данных к шифруемому сообщению (*см. раздел 1.5.2*). В этом случае оговаривается размер t блока исходного текста, а также то, что случайный двоичный вектор присоединяется со стороны, например, старших разрядов. Структура шифруемого блока B может быть следующей: $B = R|T$, где R — случайное число и T — блок исходного текста. Очевидно, что численное значение блока B не должно превышать максимально допустимое значение для используемой криптосистемы. Поэтому размер b блока B должен быть выбран таким, чтобы при максимальном значении он не выходил за пределы допустимых значений. Например, в шифре RSA значение B должно быть меньше модуля $n = pq$. Отсюда получаем условие $2^b < n$ или $b < \log_2 n$.

Если перечисленные выше условия соблюдаются, то последовательные шаги шифрования исходного текста описываются следующим образом:

$$T \to B = R|T \to C = \mathbf{E}_z(R|T),$$

а расшифрование криптограммы выполняется так:

$$C \to B = \mathbf{D}_z(C) \to R|T \to T.$$

Любое сообщение можно разбить на тексты необходимого размера, и каждый из них преобразовать с использованием вероятностного шифрования.

ГЛАВА 6

Хэш-функции

6.1. 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.

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 при необходимости частых проверок целостности данных большого объема такая процедура 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 суммы 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. В зависимости от конкретной решаемой задачи применяются алгоритмы вычисления ЗКС первого или второго типа. 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 эталонному. 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 значение $m = 64$ или даже $m = 32$ обеспечивает 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.

6.2. Криптографические контрольные суммы

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 основан 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 памяти 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 служить 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 того, 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 $(1-x)^N = 1 - Nx + \frac{N(N-1)}{2!}x^2 - \frac{N(N-1)(N-2)}{3!}x^3 ... \approx 1 - Nx$, если x мало, $p \approx N \cdot 2^{-m}$ и $N \approx p \cdot 2^m$.

При $p = 1/2$ имеем $N \approx 2^{m-1}$. В настоящее время для значения $m = 64$ 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* 6.4), 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 блочного одностороннего преобразования или стойкой шифрующей 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 она очень просто 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.

6.3. 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 6.5.* 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 в составе раундового преобразования дополнительно используется поразрядное суммирование по модулю два. На рис. 6.1 показаны три варианта общей структуры таких раундовых преобразований. Может быть использовано также комбинирование этих схем.

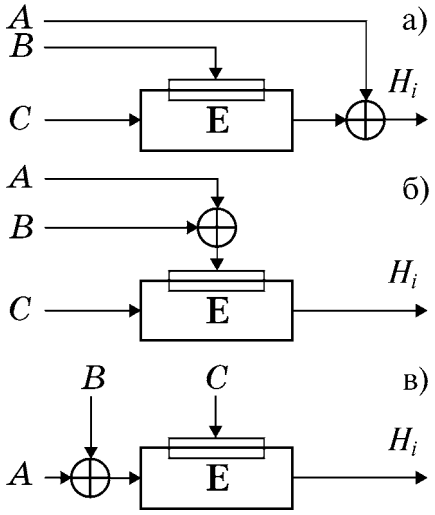


Рис. 6.1. Варианты обобщенной схемы построения раундовой хэш-функции: а) – сложение на выходе функции E, б) – сложение на входе для ключа, в) – сложение на входе функции E

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 [30, 37]. Установлено, что стойкими являются хэш-функции, построенные на основе стойкого m-битового шифра при $m \geq 128$ в соответствии с формулами, представленными в табл. 6.1. Рисунок 6.2 иллюстрирует схемное представление первого, пятого и десятого вариантов данной таблицы. Иногда в схемном представлении показывают только i-й шаг вычислений, обведенный на рис. 6.2 пунктирной рамкой.

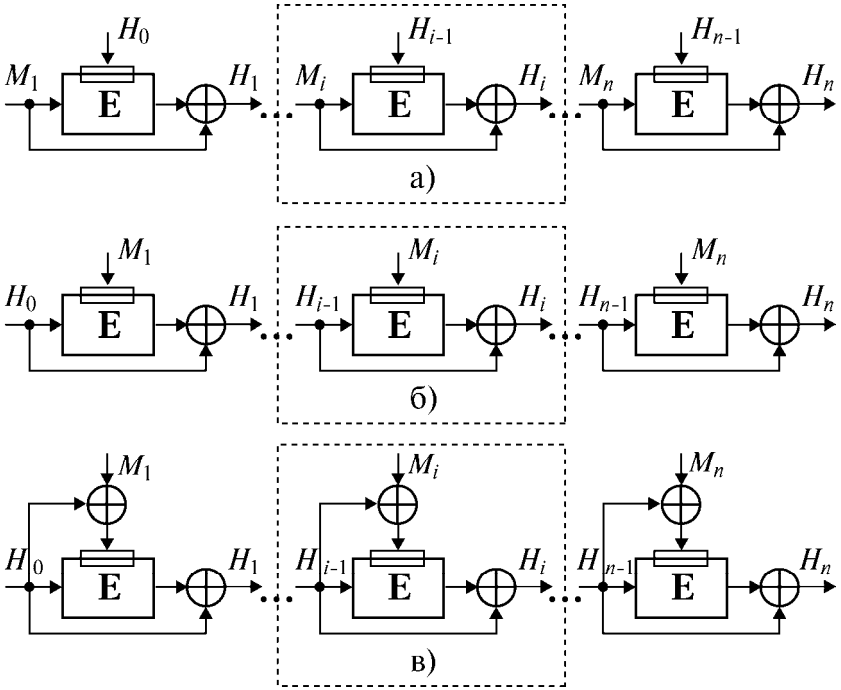


Рис. 6.2. Схематическое представление хэш-функций, построенных в соответствии с вариантами 1 (а), 5 (б) и 10 (в) таблицы 6.1

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. 6.1 двенадцать конструктивных схем.

Более интересным представляется построение хэш-функции на основе однонаправленных функций без дополнительного входа. Такие варианты показаны на рис. 6.3, где предполагается, что размер входа функции F равен $m \geq 128$ бит.

Таблица 6.1

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	$H_i = \mathrm{E}_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
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	$H_i = \mathrm{E}_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_{i-1}$

Однонаправленную функцию можно построить на основе стойкого блочного алгоритма E, например, в соответствии со следующей формулой:

$$\mathrm{F}(X) = X \oplus \mathrm{E}_K(X),$$

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.

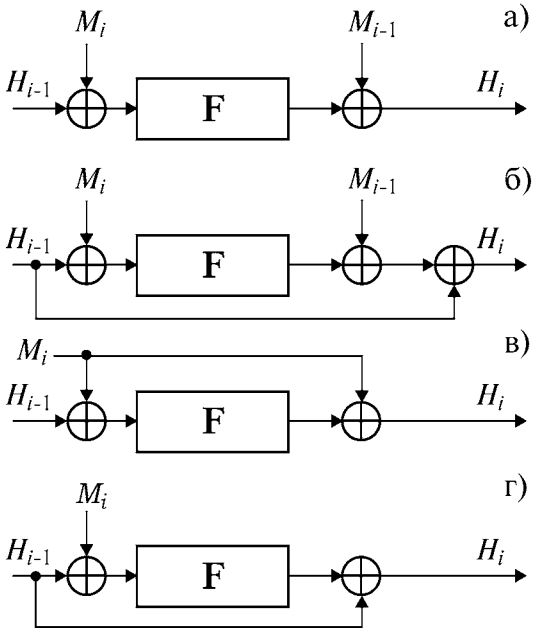


Рис. 6.3. Схемы построения хэш-функций на основе блочных односторонних преобразований

При использовании значения M_i в качестве аргумента и H_{i-1} в качестве ключа мы получаем первую схему из табл. 6.1. 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. 6.3, *г*, преобразуется последовательно к виду, показанному на рис. 6.4, *а* и 6.4, *б*.

Последняя схема напоминает конструкцию, показанную на рис. 6.3, *в*, 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. 6.3, могут быть объединены в различных комбинациях, однако некоторые сочетания могут привести к построению слабой хэш-функции. Например, на рис. 6.5 показан вариант комбинирования элементов, используемых на рис. 6.3, *в* и 6.3, *г*. Такая конструкция однонаправленной функции выглядит несколько искусственной или намеренно выбранной, однако этот пример полезен тем, что показывает необходимость конкретного рассмотрения каж-

дого построения, даже если используются базовые примитивы, являющиеся стойкими как самостоятельные преобразования.

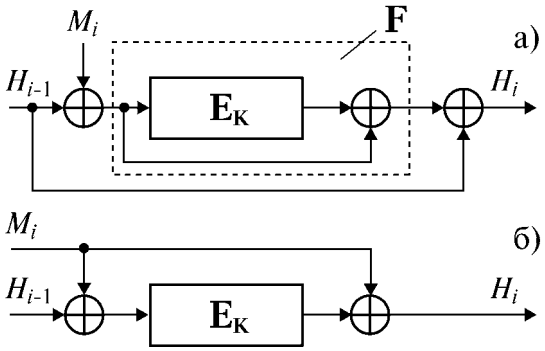


Рис. 6.4. Пример использования шифра E вместо однонаправленного преобразования при построении хэш-функции

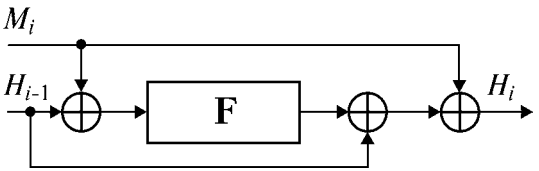


Рис. 6.5. Пример нестойкой хэш-функции с использованием стойкого блочного преобразования F

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 $M_{i+1}, M_{i+2}, \ldots, M_j$ и получаем вместо них новую последовательность блоков $M'_{i+1}, M'_{i+2}, \ldots, M'_j$. 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.

6.4. 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, который 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-м 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

$$p' \approx (1 - 2^{-m})^{N-1} \cdot (1 - 2^{-m})^{N-2} \cdot \ldots \cdot (1 - 2^{-m})^{N-i} \cdot \ldots \cdot (1 - 2^{-m}) =$$
$$= (1 - 2^{-m})^{\Sigma}, \text{ где } \Sigma = 1 + 2 + \ldots + (N-1) = N(N-1)/2.$$

Вероятность нахождения коллизий (т. е. двух одинаковых значений $H^{(i)}$) составляет

$$p \approx 1 - p' = 1 - (1 - 2^{-m})^{\Sigma} \approx 1 - (1 - \Sigma \cdot 2^{-m}) \approx N^2 \cdot 2^{-m-1}.$$

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 значение вероятности наличия коллизий во 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 $p^{(N)} = p^{(2)}\ p^{(3)} \cdot \ldots \cdot p^{(N-1)} \cdot [1-(N-1) \cdot 2^{-m}]$. Таким образом, точное значение вероятности отсутствия коллизий составляет

$$p' = p^{(\mathrm{N})} = (1-2^{-m}) \cdot (1-2 \cdot 2^{-m}) \cdot \ldots \cdot [1-(i-1) \cdot 2^{-m}] \cdot \ldots \cdot$$
$$\cdot [1-(N-1) \cdot 2^{-m}] = \prod_{i=1}^{i=N-1} \left(1 - i \cdot 2^{-m}\right).$$

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

$$p' = \prod_{i=1}^{i=N-1} (1 - i \cdot 2^{-m}) \approx \prod_{i=1}^{i=N-1} \exp(-i \cdot 2^{-m}) =$$

$$= \exp\left(-\sum_{i=1}^{i=N-1} i \cdot 2^{-m}\right) = \exp[-N(N-1) \cdot 2^{-m-1}].$$

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 бит 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 содержание, 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 (рис. 6.6) 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 $p = 0.5$.

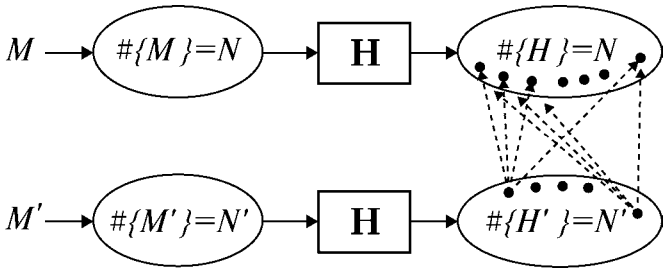


Рис. 6.6. Поиск коллизии, соответствующей осмысленным сообщениям M и M'

Учитывая парадокс о днях 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 о днях рождения.

Рассмотрим еще один вариант атаки с использованием парадокса о днях 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 о днях рождения с вероятностью 0.5 среди этих значений найдутся два равных. Пусть это будут H_i и H_j. Удалив из M блоки $M_{i+1}, M_{i+2}, \ldots, M_j$, 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 (рис. 6.7). 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.

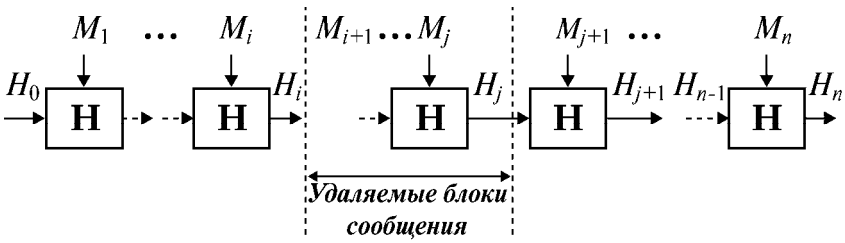


Рис. 6.7. Атака с удалением части сообщения

6.5. 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 результат 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 (рис. 6.8).

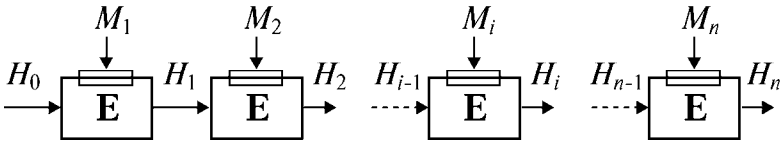


Рис. 6.8. Схема Рабина

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 (рис. 6.9). 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, соответствующего алгоритму шифрования E (рис. 6.10). В качестве начального значения хэш-

функции берется H. Пусть получено N_2 значений $H^{(2)}$ от N_2 различных вариантов второй части.

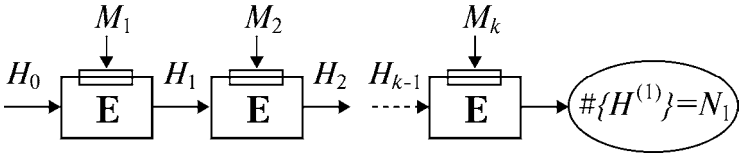


Рис. 6.9. Вычисление промежуточных значений хэш-функции с использованием алгоритма зашифрования

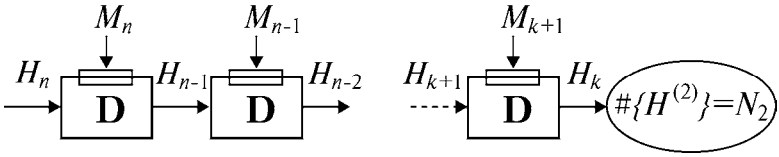


Рис. 6.10. Вычисление промежуточных значений хэш-функции с использованием алгоритма расшифрования

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 о днях 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.

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. Наиболее надежным и общим методом противодействия атаке «встреча посередине» является использование большой разрядности хэш-функций (не менее 128 бит). Тем более, что этот способ устраняет также атаки на основе парадокса о днях рождения, рассмотренные в предыдущем параграфе.

ГЛАВА 7

Управление ключами и протоколы

7.1. Вопросы управления ключами

Механизмы шифрования обеспечивают эффективную защиту информации при условии решения проблемы управления ключами. Как правило, в криптосистемах предполагается использование секретных ключей (исключение составляют криптографические контрольные суммы — хэш-функции). Ключом называется некоторая секретная информация, определяющая ход шифрующего преобразования. Разрядность (размер или длина) ключа должна быть достаточно большой, а сам ключ должен выбираться случайно и равновероятно из пространства возможных ключей. В этом случае предотвращаются наиболее простые атаки, основанные на угадывании ключа или на переборе по всем ключам. При недостаточном размере ключа никакой замысловатый алгоритм не может обеспечить высокую стойкость криптографического преобразования. Использование стойкого алгоритма является необходимым, но недостаточным условием обеспечения высокой стойкости криптографической системы.

Под *ключевой системой* понимают всю совокупность ключей, используемых в рассматриваемой криптосистеме. Часто нарушителю проще осуществить атаку на ключевую систему (или на конкретную реализацию криптосистемы), чем непосредственно на сам алгоритм, лежащий в основе криптосистемы. Управление ключевой системой является важнейшим элементом, определяющим стойкость практически используемых криптосистем.

Управление ключами в криптосистемах включает следующие функции:

- формирование (генерация) криптографических ключей;
- распределение и аутентификация секретных ключей;
- аутентификация открытых ключей;
- использование ключей;

- хранение ключей (поддержание целостности открытых ключей и защиты личных ключей);
- депонирование ключей;
- замена ключей;
- отмена ключей;
- удаление (уничтожение) ключей.

Каждая из этих задач должна быть решена надлежащим образом на всех этапах жизненного цикла ключей. По времени жизни ключи делятся на долговременные и кратковременные. Различные схемы распределения ключей включают ту или иную иерархию ключей. Иерархия ключевой системы может включать два, три и более уровней. Разделение ключей по уровням является одним из механизмов обеспечения секретности и аутентичности ключей. Управление ключами имеет своей целью:

- предотвращение несанкционированного использования секретных и открытых ключей (например осуществление проверки подписи по открытому ключу, срок действия которого истек);
- противодействие угрозам компрометации конфиденциальности секретных ключей и компрометации аутентичности открытых и секретных ключей.

Управление ключами является частью политики безопасности, которая определяет перечень потенциальных угроз информационной безопасности.

Формирование и распределение криптографических ключей основано на использовании специальных правил и схем. При реализации указанных функций используются защищенные каналы, генераторы случайных чисел, алгоритмы криптографической защиты ключевой информации (шифрование ключей) и протоколы пересылки ключей по открытым каналам.

Аутентификация ключей. Первый (базовый) этап аутентификации ключей (открытых или секретных) реализуется некриптографическими методами. После появления аутентифицированных ключей криптосистема может начать функционировать и выполнять *вторичную* аутентификацию ключей, которая заключается в аутентификации новых ключей на основе «базовых» ключей.

Использование ключей состоит в их применении для реализации конкретных криптографических преобразований с целью защиты и/или аутентификации информации, а также подтверждения подлинности пользователей. Ключи могут использоваться непосредственно для выполнения криптографических преобразований или опосредственно. Во втором случае ключи используются для выработки расширенного ключа, а последний — для непосредственного

выполнения шифрования. Необходимость шифрования ключей при хранении и пересылке с помощью других ключей приводит к концепции *иерархии ключей*. Например, в международном стандарте ISO 8532 предусматривается схема, использующая главный ключ (мастер ключ), ключ шифрования ключей и ключ шифрования данных. Обычно нижний иерархический уровень представлен *рабочими* (*сеансовыми*) ключами, которые используются для шифрования передаваемых сообщений, идентификационных номеров, контроля целостности данных. Для их защиты при передаче по открытым каналам используются ключи следующего уровня — ключи шифрования ключей. Последние не должны использоваться в качестве сеансовых. В свою очередь сеансовые ключи не должны использоваться для шифрования других ключей. Верхний уровень представлен главными ключами. Мастер (главные) ключи распределяются по защищенным каналам связи (при личном контакте, с помощью курьера, по квантовому каналу и т. д.).

Хранение ключей предполагает их защиту некриптографическими мерами, хотя криптографические преобразования могут применяться как дополнительные механизмы защиты. Секретные ключи должны записываться на носители в зашифрованном виде, что направлено на устранение их считывания и копирования. При этом желательно хранить в зашифрованном виде не только сами ключи, но и информацию о ключах и их использовании.

При хранении ключей решаются задачи поддержания их целостности и секретности. Сюда включается также задача *архивирования* ключей, которые могут понадобиться в течение достаточно длительного промежутка времени. Например, в системах ЭЦП требуется хранение текущих и всех старых открытых ключей, которые понадобятся при рассмотрении вопросов, связанных с отказами от подписи. При хранении ключей симметричного шифрования должен быть обеспечен учет носителей с ключевой информацией (ключевым материалом). Этот учет может вестись, например, в журнале регистрации, где указывается:

- название носителя, отражающее его содержание;
- учетный номер носителя;
- дата и время записи на носитель ключевой информации;
- когда и кому носитель выдан;
- подпись лица, получившего носитель;
- дата и время возвращения носителя;
- дата и время уничтожения ключевых данных на носителе;
- подпись лиц, осуществивших (или проконтролировавших) процедуру уничтожения ключевой информации и т. п.

Замена ключей осуществляется после завершения времени действия ключа, соответствующего тому или иному иерархическому уровню. Например, сеансовые ключи заменяются после завершения каждого сеанса связи. Замена ключей связана с гарантированным уничтожением старого ключа и шифрованием информации на новом ключе.

Отмена ключей — процедура, прекращающая действие ключей в случае их компрометации до истечения срока их действия. Предусматриваются действия по оповещению всех пользователей, которых затрагивает отмена соответствующего ключа. При отмене открытых ключей оповещаются все абоненты криптосистемы и одновременно прекращается действие сертификатов, ранее подтверждавших отменяемые открытые ключи.

Ключи имеют ограниченный срок действия (*время жизни ключей*). Длительность действия ключей зависит от:

- частоты использования ключей;
- величины потенциального ущерба, связанного с компрометацией ключей;
- объема и характера шифруемой информации;
- особенностей технологического применения шифрования.

Время действия ключей устанавливается с учетом следующих соображений:

- с увеличением времени использования ключа увеличивается вероятность его компрометации;
- с увеличением времени использования ключа увеличивается потенциальный ущерб от его компрометации;
- с увеличением объема перехваченной информации, зашифрованной на одном ключе, потенциально снижается сложность раскрытия ключа;
- с увеличением времени использования ключа у нарушителя усиливается стимул потратить значительные ресурсы на его вскрытие, поскольку возрастает ценность раскрытого ключа.

Депонирование ключей заключается в их предоставлении специальному центру, обеспечивающему целостность хранения ключей и их секретность. Депонирование ключей используется для того, чтобы при наличии правовых санкций специальные государственные службы могли получить доступ к зашифрованной информации с целью решения задач государственной безопасности. Это понятие относится к вопросам практического использования

криптосистем. При утере ключа владелец может обратиться в центр депонирования и получить его копию.

Удаление (уничтожение) ключей предполагает обеспечение таких мер, при которых гарантируется невозможность восстановления ключевой информации с каких-либо носителей и невозможность копирования (перехвата) ключей в процессе их уничтожения.

7.2. Генерация ключей

Секретный ключ является тем элементом криптосистемы, на котором основывается ее стойкость. Если злоумышленнику становится известным секретный ключ, то он получает все привилегии законного пользователя. Ключ может стать известным различными путями, включая установку программных и аппаратных закладок. Принципиальным при теоретическом рассмотрении криптосистем является вопрос о способах раскрытия ключа, основанных на криптоанализе. Барьером для таких атак является надлежащим образом синтезированный алгоритм шифрования. Остальные пути предотвращения перехвата ключевой информации связаны с организационными и техническими мероприятиями. Под раскрытием ключа путем криптоанализа здесь мы будем также понимать определение ключа путем его угадывания (или перебора по пространству всех возможных ключей). Этот тип нападения на криптосистемы является универсальным (все криптосистемы с ключом подвержены такой атаке) и наиболее простым, и в то же время он наиболее просто предотвращается. Чтобы нападающий не смог добиться успеха этим методом, требуется аккуратность в выборе длины ключа и процедур его генерации. В настоящее время длина ключа, равная 128 бит, считается безопасной. (Лобовая атака связана с перебором 2^{128} возможных ключей.) В ряде шифров предусматривается использование ключа длиной 192 и 256 бит. Некоторые криптосистемы не ограничивают пользователя каким-либо фиксированным размером ключа и предоставляют ему право выбора длины ключа в широком диапазоне.

Генерация ключей в асимметричных криптосистемах осуществляется таким образом, что ключи обладают определенными математическими свойствами и в тоже время они выбираются случайно из очень большого множества ключей, обладающих такими свойствами (заметим, что в некоторых двухключевых шифрах секретный ключ может выбираться случайно и равновероятно по множеству допустимых значений, т. е. без учета каких-либо математических свойств, хотя в любой двухключевой системе используются параметры, которые должны обладать определенными математическими свойствами).

Качественный секретный ключ в симметричной криптосистеме представляет собой случайную последовательность битов заданной длины без учета каких-либо специальных математических свойств. Основным принципом генерации ключа является равновероятность выбора по всему ключевому пространству (множеству возможных ключей). При генерации ключей используются электронные устройства, в которых протекает случайный физический процесс. Данные устройства называются датчиком шума (или генератором шума). Показания датчика шума замеряются через определенные интервалы времени и оцифровываются. (Причем желательно использовать такие датчики шума, для которых распределение вероятности цифровых показаний является примерно равномерным.) Полученный ряд чисел является несомненно случайным, однако вероятности появления различных значений могут весьма существенно отличаться.

В некоторых случаях с течением времени или при изменении условий окружающей среды эти отличия вероятностей становятся более выраженными (или появляются, если таковых первоначально не было). Поэтому, чтобы выровнять вероятности выбора ключей, для формирования текущего ключа длины q снимаются показания датчика шума до тех пор, пока длина ряда случайных чисел существенно не превысит величину q. После этого вычисляется хэш-функция длины q от полученной случайной последовательности битов и ее значение берется в качестве ключа. Хэширование обеспечивает «концентрацию» равномерности и случайности, поскольку значение хэш-функции зависит от каждого бита входной последовательности по некоторому «криптографическому» закону.

Периодически механизм генерации ключей подвергается проверке, которая заключается в проведении тестов на случайность и равномерность показаний датчика шума, по которым формируются секретные ключи. В качестве статистических тестов на случайность могут быть использованы, например, следующие: тест хи-квадрат, тест инверсий, тест серий, тесты на сжатие, тест автокорреляции, спектральные и другие тесты.

Достаточно общей практикой является централизованная генерация ключей симметричного шифрования в специальном центре, где имеется возможность осуществления выработки ключей в соответствии с процедурами, гарантирующими случайный характер генерируемых ключевых последовательностей. В основе применяемых аппаратных датчиков случайных чисел лежит некоторый случайный физический процесс (например флуктуации напряжения на выходе шумового диода).

При децентрализованной генерации секретных ключей можно использовать датчики псевдослучайных чисел (ДПСЧ), на вход которых для инициализации датчика подается сформированное пользователем начальное случай-

ное значение (которое является секретным). В таких ДПСЧ возникает проблема генерации случайного начального значения. В качестве источника случайности могут служить временные показатели реакции оператора при диалоге с системой или замеры интервалов между нажатиями клавиш на клавиатуре при наборе случайного текста, результат выбора битов из показаний таймера, коды нажатых клавиш и т. п.

Таким образом, в порядке возрастания качества генерируемых ключей могут использоваться следующие методы:

1. Программная генерация с использованием преобразования (с помощью датчиков псевдослучайных чисел или хэш-функций) отсчетов текущего времени, последовательности введенных пользователем символов и т. п.
2. Программная генерация с вычислением значений хэш-функции от сравнительно длинных участков случайной последовательности, сформированной в соответствии с п. 1.
3. Аппаратная генерация с помощью физических датчиков шума.
4. Аппаратная генерация с помощью физических датчиков шума с контролируемой выходной последовательностью.
5. Аппаратная генерация с помощью физических датчиков шума с контролируемой выходной последовательностью и криптографическим преобразованием выходной последовательности датчика.

В ряде протоколов, требующих генерации случайных чисел, это может решаться с помощью перечисленных выше методов.

7.3. Схемы распределения ключей

Распределение ключей в симметричных криптосистемах основано на использовании защищенных каналов и криптографических протоколов. В двухключевых и гибридных криптосистемах распространение (передача, распределение) ключей основано на предварительной аутентификации открытых ключей, которая осуществляется некриптографическими методами, и последующем использовании криптографических протоколов. Различают схемы и протоколы. Под схемой понимают структурную организацию системы управления ключами, а под протоколом — совокупность действий двух и более сторон, обеспечивающую появление у них общих секретных ключей. Различают два типа протоколов распределения ключей:

- Протоколы передачи (пересылки, доставки) ключей, которые предварительно уже сгенерированы.
- Протоколы совместного формирования (выработки) ключей.

Основное отличие второго типа от первого состоит в том, что вырабатываемый ключ зависит от произвольного выбора двух и более сторон и каждая из взаимодействующих сторон получает этот ключ в результате проведенных вычислений. Различают протоколы и схемы распределения ключей между двумя пользователями (протоколы типа «точка — точка»), в которых передача (или выработка) ключей осуществляется в результате непосредственного взаимодействия двух сторон, и между многими пользователями. При распределении ключей между многими пользователями выделяют схемы централизованного распределения ключей.

Протоколы распределения ключей являются частным случаем криптографических протоколов. В последних можно также выделить два класса протоколов, имеющих принципиальное отличие в плане взаимного доверия пользователей друг другу.

- Протоколы, в которых стороны *доверяют* друг другу (протоколы, основанные на взаимном доверии).
- Протоколы, в которых стороны *не доверяют* друг другу (протоколы, реализующие те или иные задачи в условиях взаимного недоверия).

В первом типе протоколов могут использоваться как симметричные, так и асимметричные шифры, а основу протоколов второго типа составляют двухключевые (асимметричные) шифры. Возможность реализации протоколов со взаимным недоверием базируется на том, что в двухключевых шифрах секретный ключ не должен быть известным более чем одному пользователю, который и является (единственным) владельцем данного конкретного секретного ключа. Примерами систем, использующих протоколы второго типа, являются системы ЭЦП и системы тайного электронного голосования. На основе симметричного шифрования также можно построить системы ЭЦП, однако они основываются на доверии всех пользователей некоторым *доверительным* центрам (одному или нескольким), которые владеют секретными ключами многих пользователей, тогда как в случае использования двухключевой криптографии система ЭЦП строится вокруг *удостоверяющих* центров, которые не владеют секретным ключом ни одного из пользователей.

В многопользовательских *симметричных (одноключевых)* криптосистемах обычно используется централизованное распределение ключей с использованием доверительного *центра распределения ключей* (ЦРК). Применяются также иерархические схемы с доверительными центрами различного уровня: главный, региональный и др. центры.

В многопользовательских *асимметричных (двухключевых) и гибридных* криптосистемах обычно используется децентрализованное распределение секретных ключей, основанное на предварительном распространении открытых ключей, осуществляемом с привлечением удостоверяющих центров (одного или многих с соответствующей иерархической структурой; система распределения открытых ключей обычно называется *инфраструктурой открытых ключей*).

7.3.1. Централизованное распределение ключей

Распределение ключей в криптосистеме является одной из дорогостоящих процедур. Рассмотрим распределение ключей в одноключевых системах. При использовании одноключевой (симметричной) криптографии принципиальным является необходимость наличия в системе связи защищенного канала, по которому секретный ключ доставляется к приемнику и передатчику (в частном случае от передатчика к приемнику).

В качестве передачи по защищенному каналу может служить доставка сообщения через доверенное лицо (курьера), предварительный контакт абонентов или другие способы, в том числе использование охраняемых линий связи и квантовых каналов. Методы двухключевой криптографии позволяют осуществить передачу секретного ключа связи по открытому каналу. Вопрос распределения ключей является очень важным, поскольку обычным требованием является периодическая смена ключей (например от сеанса связи к сеансу, после передачи определенного объема информации или по истечении определенного периода). Нарушение этого правила упрощает задачу раскрытия некоторых типов шифров (которые в настоящее время вряд ли применяются кем-либо). Требование периодической смены ключей диктуется тем, что при компрометации ключа становится возможным несанкционированный доступ ко всей информации, зашифрованной с использованием этого ключа (мы должны считать, что противник хранит все перехваченные криптограммы). Если ключ долго не меняется, то под угрозу ставятся большие объемы информации.

В простейшей схеме двусторонней связи каждый сеанс может осуществляться с использованием одного ключа. Если криптографическая система объединяет N абонентов в сеть, то необходимо распределить между пользователями, по крайней мере, $N(N-1)/2$ ключей. При достаточно большом N это становится проблематичным, т. к. число распределяемых ключей растет по квадратичному закону. Для решения этой задачи используется ряд структурных решений, учитывающих то, что наиболее сложной и дорогостоящей является эксплуатация защищенного канала.

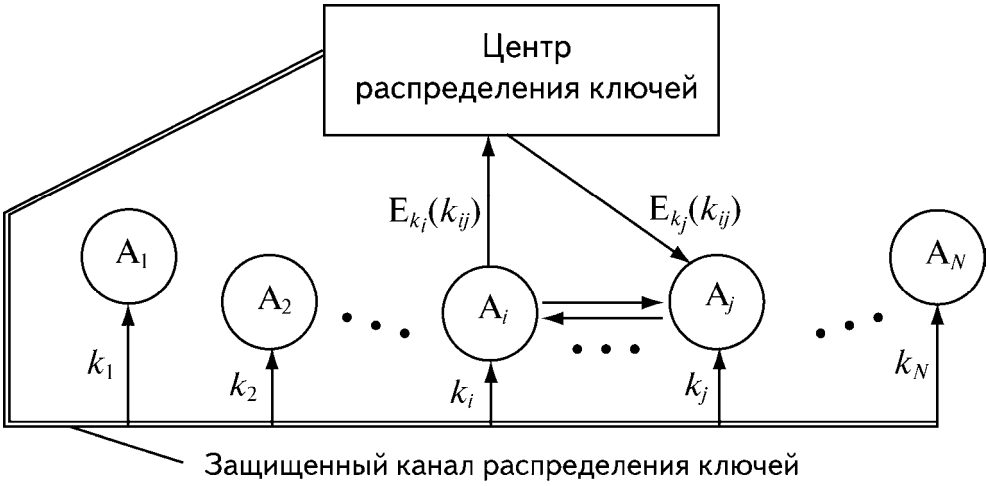


Рис. 7.1. Схема централизованного распределения ключей

В одном из вариантов создается некоторый доверительный центр, называемый центром распределения ключей (ЦРК) и являющийся частью общей сети. ЦРК снабжает всех абонентов различными секретными ключами K_i (i = 1, 2, 3, ..., N), каждый из которых используется только для связи с центром. Секретная связь между любыми двумя абонентами i и j осуществляется следующим образом (рис. 7.1). Абонент A_i, связываясь с абонентом A_j, пересылает в центр выбранный им ключ связи k_{ij}, зашифрованный ключом K_i. ЦРК дешифрует послание, зашифровывает ключ k_{ij} ключом K_j и в таком виде пересылает сеансовый ключ абоненту A_j. После этого секретная связь между данными абонентами может быть осуществлена по открытому каналу. В этой схеме требуется распределить только N ключей.

Поскольку качество выбора сеансового ключа влияет на уровень секретности связи, то эту процедуру можно передать на исполнение ЦРК, в штате которого состоят более опытные специалисты и имеются специальные устройства для генерации ключей. В таком варианте абоненты A_i и A_j запрашивают у ЦРК сеансовый ключ. ЦРК генерирует и направляет им ключ k_{ij}, зашифрованный ключами K_i и K_j, соответственно, для абонентов A_i и A_j.

Если число абонентов криптосистемы велико для обслуживания одним центром, то может быть применена схема централизованного распределения ключей с двумя иерархическими уровнями доверительных центров, в которой имеется главный ЦРК и региональные ЦРК. Каждый из последних обслуживает часть абонентов в соответствии со схемой, представленной на рис. 7.1. В тех случаях, когда секретная связь организуется между пользова-

телями, относящимися к различным региональным ЦРК, доставка сеансового ключа осуществляется через главный ЦРК.

Недостатком централизованного распределения ключей является то, что при компрометации ЦРК имеет место компрометация всей ключевой информации, переданной через этот центр и/или хранимой в нем. Учет ключей ЦРК ведется в журнальном файле, где отражаются следующие сведения:

- дата генерации ключа;
- его тип и название;
- срок действия ключа;
- для кого он сформирован;
- сведения об отправке ключа получателю;
- подтверждение получения ключа получателем и др.

Доступ к ключам имеют только должностные лица ЦРК и ограниченный круг пользователей, ответственных за хранение и использование носителей с ключами. Кроме того, доступ к ключам ограничивается техническими мерами:

- процедура генерации ключей должна быть автоматизирована и выполняться под управлением ЭВМ со встроенной защитой от несанкционированного доступа к обрабатываемой и хранимой информации;
- сеансовый ключ в процессе его формирования должен быть скрыт от пользователя;
- ключи не хранятся, как правило, в открытом виде;
- ключ записывается только на съемные носители, устанавливаемые в ЭВМ на период его использования.

Наиболее удачное решение проблемы распределения сеансовых ключей предоставляют методы двухключевой криптографии, рассмотренные ниже. Для непосредственного шифрования передаваемой информации большого объема они мало пригодны ввиду относительно малой скорости преобразований, которую они могут обеспечить, однако для передачи сеансовых ключей связи по открытому каналу они являются превосходным инструментом. В настоящее время применяются *гибридные криптосистемы*, в которых шифрование осуществляется с помощью одноключевой криптосистемы, а распределение ключей — с помощью двухключевой. В гибридных криптосистемах достигается высокая скорость шифрования, которая характерна для одноключевых шифров, и удобство распределения ключей, предоставляемое двухключевыми шифрами.

Хранение информации в зашифрованном виде предполагает хранение секретного ключа или множества секретных ключей, с помощью которых данные могут быть расшифрованы. Ключи должны храниться на защищенных от несанкционированного доступа носителях. Если информация была зашифрована на одном ключе, то ее можно расшифровать и зашифровать на новом ключе. В определенных случаях это позволяет формировать секретные архивы с использованием одного секретного ключа. Ключ, на котором информация была зашифрована первоначально, подлежит *гарантированному уничтожению*. Процедура уничтожения ключа должна выполняться под контролем владельца секретного ключа, поскольку старые ключи представляют для злоумышленника такой же интерес, как и действующие ключи. Используя старые ключи и копии старых шифртекстов, можно легко восстановить секретную информацию. Очевидно, что сеансовые ключи должны гарантированно уничтожаться как на приемном пункте, так и на передающем. При смене основного ключа (мастер ключа) он должен гарантированно уничтожаться также и в центре распределения ключей. Количество используемых ключей зависит от конкретных особенностей эксплуатации криптосистемы (числа абонентов, объема передаваемой информации, особенностей алгоритма шифрования).

Одной из фундаментальных проблем является аутентификация секретного ключа. Эта проблема известна под названием проблемы первого контакта. Процедуры аутентификации ключа должны гарантировать тот факт, что секретный ключ действительно передан от легального абонента. Простейшим вариантом является передача ключа в опечатанном пакете или обмен ключами при личном контакте лиц, которые планируют в будущем пользоваться секретной связью. Аутентификация главных секретных ключей (*мастер ключей*) осуществляется одновременно с их распределением по защищенному каналу. В связи с этим процедура их аутентификации уходит на второй план.

При использовании двухключевых криптографических алгоритмов проблема аутентификации относится к открытым ключам, поскольку секретные ключи не попадают в поле зрения никого, кроме владельца секретного ключа. Однако остальные абоненты должны быть уверены, что открытый ключ принадлежит именно ему, т. е. соответствует его секретному ключу. Открытые ключи должны быть переданы с использованием процедуры аутентификации владельца секретного ключа. После этого двухключевые шифры могут быть использованы для аутентификации источника электронной информации. Поскольку распределение секретных ключей в асимметричных криптосистемах не осуществляется, то на первый план выдвигается проблема аутентификации (открытых) ключей. Совокупная сложность процедур аутентификации открытых ключей в асимметричных криптосистемах существенно ниже сложности распределения секретных ключей по защищенному каналу в случае

симметричных криптосистем. Даже только в аспекте распределения ключей асимметричные и гибридные криптосистемы являются более экономичными. Возможность реализации систем цифровой подписи и *придания юридической силы электронным документам* ввела в практическую информатику новые перспективные технологии электронного документооборота, что является еще более важным значением двухключевой криптографии.

7.3.2. Открытое распределение ключей

В секретной системе связи распределение ключей между пользователями может осуществляться двумя различными способами.

- Централизованным (с использованием одного или нескольких ЦРК).
- Децентрализованным (пользователи осуществляют прямой обмен ключами, например сеансовыми и долговременными).

Существенным недостатком первого подхода является то, что ЦРК владеет ключами всех пользователей. Компрометация ЦРК приводит к необходимости замены всей ключевой системы. Второй подход опирается на надежное решение проблемы аутентификации (взаимное удостоверение подлинности) абонентов. В случае симметричных криптосистем с большим числом абонентов первый подход является во многих случаях предпочтительным. Поиск новых эффективных решений по распределению ключей, объединяющих достоинства централизованного и децентрализованного способов, привел к открытию двухключевых криптосистем.

В 1976 г. Диффи и Хеллман опубликовали статью, которая ознаменовала собой рождение двухключевой криптографии и привела к сильному росту числа открытых исследований в области криптографии и появлению большого числа новых ее разделов. Она содержала ошеломляющий результат: *возможно построение практически стойких секретных систем, которые не требуют передачи секретного ключа.* Диффи и Хеллман ввели понятие односторонней функции с потайным ходом (с лазейкой). Под односторонней функцией f понимается функция $f(x)$, которая легко вычислима для любого значения аргумента x из области определения, однако для случайно выбранного y из области ее значений вычислительно сложно нахождение значения аргумента x, для которого выполняется $f(x) = y$. Применение таких функций для защиты входа в вычислительную систему путем одностороннего преобразования паролей было известно достаточно давно. Но как применить одностороннюю функцию в криптографических системах, когда даже законный получатель не сможет выполнить процедуру расшифрования? Для шифрования была предложена *односторонняя функция с потайным ходом* (*секретом*).

Под односторонней функцией с потайным ходом понимается семейство обратимых функций f_z с параметром z, таких что для данного z можно найти алгоритмы E_z и D_z, позволяющие легко вычислить значение $f_z(x)$ для всех x из области определения, а также $f_z^{-1}(y)$ для всех y из области значений, однако практически для всех значений параметра z и практически для всех значений y из области значений f_z нахождение $f_z^{-1}(y)$ вычислительно неосуществимо даже при известном E_z. В качестве односторонней функции Диффи и Хеллман предложили функцию дискретного возведения в степень

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

где x — целое число, $1 \le x \le p-1$, p есть k-битовое простое число. Причем выбирается такое число $\alpha < p$, степени которого по модулю p представляют собой упорядоченное множество чисел $\{\alpha^1, \alpha^2, ..., \alpha^{p-1}\}$, являющееся некоторой перестановкой чисел $\{1, 2, ..., p-1\}$. (Такое число α называется первообразным корнем по модулю p, т. е. числом, относящимся к показателю $p-1$ по модулю p.)

Даже для очень больших модулей p (например при $k = 1024$ бит) для данного x легко вычислить значение этой функции. Процедура вычисления этой функции называется дискретным возведением в степень. Для выполнения этой процедуры достаточно выполнения около $2 \cdot \log_2 p$ операций умножения k-битовых чисел (или $\log_2 p$ умножений и $\log_2 p$ делений $2k$-битовых чисел на k-битовые). Процедура быстрого возведения в большую степень основана на предварительном вычислении значений $\{\alpha^1 \bmod p, \alpha^2 \bmod p, \alpha^4 \bmod p, \alpha^8 \bmod p, ..., \alpha^{2^{k-1}} \bmod p\}$ и перемножении (по модулю p) соответствующего набора из этих чисел.

Обратной к функции дискретного возведения в степень является функция $f^{-1}(y)$, которая ставит в соответствие заданному значению y такое значение x, для которого выполняется условие $\alpha^x = y \pmod p$. Задача нахождения такого x называется задачей дискретного логарифмирования (нахождения дискретных логарифмов). Дискретные логарифмы сложно вычисляются, когда число $p-1$ содержит в своем разложении хотя бы один большой простой множитель, например, когда оно представимо в виде $p - 1 = 2p'$, где p' — простое число. При этом условии для больших значений k (например при $k \ge 1024$ бит) трудоемкость задачи нахождения дискретного логарифма настолько велика, что ее решение является вычислительно неосуществимым, т. е. при указанных условиях, накладываемых на выбор чисел p и α, функция дискретного возведения в степень является односторонней.

Системой (методом) *открытого распределения ключей* Диффи–Хеллмана называется следующий способ использования дискретного возведения в сте-

пень 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 1024 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

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

Для любого значения x легко вычислить y, однако при размере числа p более 1000 бит вычислительно неосуществимо выполнение дискретного логарифмирования, а следовательно, и определение числа x, для которого значение $\alpha^x \bmod p$ равно заданному значению y. Все абоненты размещают свои открытые ключи в общедоступном справочнике. Данный справочник должен быть заверен специально созданным *удостоверяющим центром*, чтобы исключить возможные нападения путем подмены открытых ключей или навязывания ложных открытых ключей. Если два абонента A и B хотят установить секретную связь, то они поступают следующим образом. Абонент A берет из справочника открытый ключ абонента B 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 A и B; x_A и x_B — соответствующие секретные ключи. Нет необходимости передавать по сети связи общий секретный ключ Z_{AB}, поскольку абонент B по известному из справочника открытому ключу абонента A аналогичным способом вычисляет значение

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

Нарушителю (потенциальному злоумышленнику) 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 трудную задачу дискретного логарифмирования. Общий секретный ключ может использоваться абонентами для шифрования сеансовых секретных ключей, а последние — для шифрования сообщений с использованием симметричных методов шифрования. Решение задачи дискретного логарифмирования существует, но оно вычислительно неосуществимо. Таким образом, стойкость метода Диффи–Хеллмана основана на сложности задачи дискретного логарифмирования.

В одноключевых криптосистемах существуют две принципиальные проблемы:

- распределение секретных ключей по защищенному каналу;
- аутентификация секретного ключа.

Под аутентификацией понимается проведение процедуры, которая позволяет получателю удостовериться, что секретный ключ принадлежит законному отправителю (например центру распределения ключей).

Система открытого распределения ключей решает первую проблему, т. е. она позволяет обойтись без защищенного канала для распределения секретных ключей. Однако она не устраняет необходимость аутентификации. При этом надо отметить, что в двухключевой криптографии проблема аутентификации не возникает, а передвигается на первый план, так как проблема распределения ключей разрешается ее методами.

Таким образом, в основе открытого распределения ключей лежит принцип отделения процедуры аутентификации от непосредственной процедуры распределения ключей. Организуется централизованное распределение открытых ключей, которое обеспечивает высокую технико-экономическую эффективность осуществления процедуры подтверждения подлинности субъектов. На основе решенной задачи аутентификации абонентов сети осуществляется децентрализованное распределение ключей симметричного шифрования, т. е. взаимодействующие через систему связи субъекты сами формируют и распределяют сеансовые ключи.

7.3.3. Распределение ключей на основе симметричного шифрования

В рамках протоколов распределения ключей на основе симметричного шифрования должна решаться задача обеспечения подлинности сеанса связи, включающая подтверждение подлинности субъектов сеанса связи. Проверку подлинности сеанса связи можно осуществить, используя механизмы запроса-ответа и отметки времени.

Механизм запроса-ответа реализуется с помощью интерактивного взаимодействия пользователей, в котором каждый из пользователей демонстрирует знание некоторого секрета (секретного ключа) K, который распределен между ними заранее, причем таким образом, что секрет не может быть раскрыт нарушителем, прослушивающим канал связи. Одновременно с этим сам процесс интерактивного взаимодействия включает генерацию случайных сообщений, что предотвращает использование нарушителем (с целью выдать себя за законного абонента) копий старых сообщений, переданных между

субъектами в процессе некоторой предыдущей процедуры их взаимной аутентификации. Например, аутентификация абонента А абонентом В осуществляется следующим образом. Абонент В генерирует некоторое случайное число R длиной 64 бит и посылает его субъекту А. Абонент А, используя заранее оговоренный алгоритм блочного шифрования, шифрует полученное случайное число R по ключу K: $C = E_K(R)$. Затем он генерирует свое случайное число R' и пересылает значения R' и C субъекту В. Абонент В самостоятельно осуществляет такое же преобразование числа R: $C^* = E_K(R)$. Если $C^* = C$, то абонент В принимает решение, что субъект А является подлинным. Затем абонент В шифрует полученное случайное число R' по ключу K: $C' = E_K(R')$ и пересылает значение C' субъекту А. После этого абонент А самостоятельно осуществляет преобразование числа R': $C'' = E_K(R')$. Если $C'' = C'$, то абонент А принимает решение, что субъект В является подлинным. Схемы аутентификации такого типа называются протоколами взаимной аутентификации или протоколами рукопожатия.

Замечание. Этот протокол позволяет установить взаимную подлинность двух взаимодействующих субъектов. Но это не означает, что теперь они могут безопасно обменяться открытыми ключами, зашифровав их на секретном ключе. Дело в том, что они могут в дальнейшем отказаться от своего открытого ключа и третья сторона не сможет выявить, кто из этих двух субъектов является нарушителем. Протокол рукопожатия относится к протоколам строгой аутентификации, которые характеризуются использованием криптографических преобразований, позволяющих субъектам продемонстрировать друг другу знание секрета (тем самым подтвердить свою подлинность) без того, чтобы секрет был раскрыт в открытом канале связи. Можно выделить протоколы:

- *простой аутентификации* на основе паролей (наиболее уязвимый тип при взаимодействии через канал связи, поскольку пароли представляются в открытом виде без их преобразования);
- *строгой аутентификации* (обеспечивает аутентификацию удаленных пользователей без разглашения секретного ключа);
- *с нулевым разглашением* (наиболее сильный тип, основанный на предварительном распределении открытых ключей и их аутентификации; каждый из взаимодействующих субъектов убеждается в подлинности другого субъекта с использованием открытого ключа последнего, причем после процедуры аутентификации никакой из субъектов не получает никакой информации о секретном ключе субъекта, подлинность которого подтверждена в ходе осуществления протокола).

Механизм отметки времени включает фиксацию времени для каждого сообщения, благодаря чему можно установить, насколько старо пришедшее сообщение. Если сообщение старо, то может быть принято решение, что оно не является подлинным, а направлено активным нарушителем (например он мог скопировать одно из сообщений, ранее переданных законным абонентом). При этом метка времени или все сообщение целиком (включая метку времени) зашифровывается, что исключает атаки на основе изменения штемпеля метки времени. При использовании механизма отметки времени абоненты договариваются о допустимом интервале временной задержки. Если подлинное сообщение пришло с запозданием, то оно повторяется с установкой нового штемпеля отметки времени.

При использовании механизма отметки времени также следует различать случаи взаимного доверия и недоверия пользователей. В случае взаимного доверия пользователей метка может проставляться отправителем. Что касается второго случая (например в системах ЭЦП), то временная метка должна проставляться третьей стороной — центром установки временных меток (он может входить как составная часть в удостоверяющий центр) — следующим образом. Получив некоторое сообщение M (если требуется обеспечить конфиденциальность сообщения, то представляется хэш-функция от документа H), третья сторона добавляет к нему временную метку t и подписывает документ (M, t) или (H, t), а затем возвращает значение (M, t, S) или (H, t, S) пользователю, представившему сообщение для установки метки времени. Проверка подлинности метки времени включает проверку подлинности подписи третьей стороны.

Протокол распределения ключей должен обеспечить следующее:

- взаимную аутентификацию субъектов сеанса связи;
- подтверждение целостности сеанса связи (например с использованием отметки времени);
- минимальное число сообщений, необходимых для обмена ключами.

В некоторых схемах децентрализованного обмена ключами используется *механизм модификации ключа*. Он заключается в том, что новый сеансовый ключ не пересылается в зашифрованном виде, а формируется из предыдущего значения ключа с помощью некоторой заранее оговоренной односторонней функции преобразования блоков данных. При использовании стойкой однонаправленной функции новый ключ безопасен в той же мере, в какой безопасен прежний ключ. В механизме модификации ключа должна быть предусмотрена синхронизация выработки новых значений ключа обеими сторонами сеанса связи. Недостатком этой системы является то, что компрометация текущего значения сеансового ключа приводит к компрометации

всех последующих сеансов связи, поскольку нарушитель, зная текущее значение ключа, может выполнить его последовательное преобразование с помощью односторонней функции (мы полагаем, что она не является секретной) и получить все последующие сеансовые значения ключа.

7.3.4. Распределение ключей на основе двухключевых шифров

Протокол SKIP (*Simple Key Management for Internet Protocol*, *1994 г.*) является примером использования двухключевой криптографии при распределении ключей. Он предназначен для обеспечения защиты трафика пользователей и прикладных систем при прозрачности средств защиты. В нем используется система открытого распределения ключей Диффи–Хеллмана. Каждый *i*-й узел сети снабжается секретным ключом x_i и аутентифицированными открытыми ключами $y_j = \alpha^{x_j} \bmod p$ остальных узлов. Любая заданная пара узлов (например *i*-й и *j*-й) обладает уникальным разделяемым секретом — общим секретным ключом $K_{ij} = y_i^{x_j} \bmod p = y_j^{x_i} \bmod p$.

Закрытие пакета, направленного из *i*-го узла сети в *j*-й, осуществляется с помощью его шифрования по разовому пакетному ключу K_p. Пакетный ключ зашифровывается с помощью ключа K_{ij}: $C_k = \mathrm{E}(K_p, K_{ij})$. Исходный пакет шифруется полностью, включая служебную информацию (заголовок и концевик). Зашифрованный пакет инкапсулируется в пакет протокола SKIP. К заголовку нового пакета присоединяется значение C_k, которое позволяет *j*-му узлу расшифровать пакетный ключ, а затем и исходный пакет. Заголовок пакета протокола SKIP идентичен заголовку IP-пакета (это обеспечивает прозрачность SKIP-пакета для всего промежуточного оборудования сети) и включает дополнительные поля, в которых содержится информация об используемых алгоритмах и зашифрованный пакетный ключ. На *j*-м узле из заголовка SKIP-пакета извлекается значение C_k, по которому расшифровывается пакетный ключ K_p с помощью общего секретного ключа K_{ij}. Затем с помощью ключа K_p расшифровывается исходный IP-пакет.

Можно отметить следующие моменты, связанные с использованием протокола SKIP.

- Для организации защищенной передачи информации используются сети общего доступа.
- При распределении открытых ключей должен быть решен вопрос об их аутентификации.
- Разделяемый секретный ключ используется только для зашифрования пакетных ключей.

- Исходные IP-пакеты шифруются независимо от того, какие приложения и процессы их породили. Протокол SKIP устанавливается непосредственно над пакетным драйвером, что обеспечивает высокую совместимость с лежащими выше прикладными программами.
- Увеличение размера исходного пакета практически не влияет на скорость передачи данных по сети.
- В протоколе SKIP предусматривается набор различных алгоритмов шифрования, которые пользователь может выбирать самостоятельно. Пользователь может использовать свой собственный алгоритм (свойство независимости от используемых криптографических алгоритмов).

7.4. Разделение секрета

На практике иногда возникает задача доверить секрет нескольким лицам, причем таким образом, чтобы каждый из них владел только частью секрета (*секретного ключа*). Это требуется для того, чтобы один из них или их малая часть не могли восстановить секрет и воспользоваться возможностями, которые предоставляются обладателю секрета. Возможность восстановления секрета должна появляться только тогда, когда объединятся все или достаточно большая часть обладателей секрета. Очевидно, что достаточно просто обеспечить возможность восстановления секрета при объединении всех владельцев секрета. Однако на практике лица, которым доверяется секрет, выполняют и другие функции, кроме того, они могут находиться в отпуске, в командировке или не иметь возможности по состоянию здоровья присутствовать при процедуре восстановления секрета. Поэтому стоит задача дать возможность восстановления секрета для неполной совокупности обладателей секрета, но только в случае, если их число равно или больше некоторого порогового значения. Причем конкретный состав таких неполных групп владельцев секрета может оказаться различным. Так возникает пороговая задача разделения секрета.

В общем виде задача разделения секрета формулируется следующим образом. Требуется разделить секрет S между n хранителями секрета таким образом, чтобы любые t ($t \le n$) из них смогли восстановить секрет, если $t \ge k$, где число k является пороговым значением. Однако если число объединяющихся владельцев секрета будет меньше значения k, то они не должны иметь практической возможности восстановить секрет (даже в случае наличия у них больших вычислительных ресурсов).

Данная задача возникает в случаях необходимости принять коллегиальное решение некоторыми ответственными лицами. Примерами ситуаций, когда может возникнуть такая задача, являются, например:

- запуск ракеты военными;
- открытие хранилища золотовалютных резервов ответственными сотрудниками банка.

Обычно в схемах порогового разделения секрета S каждому владельцу предоставляется часть секрета в виде некоторого числа или нескольких чисел, которые явным образом не связаны с секретом S или его частью. Но в скрытом виде в них содержится необходимая информация о некоторой доле секрета, за счет чего и обеспечивается работа схем разделения секрета. Рассмотрим схему разделения секрета, основанную на китайской теореме об остатках.

7.4.1. Схемы на основе китайской теоремы об остатках

Выберем некоторое множество из попарно взаимно простых чисел $\{m_1, m_2, \ldots, m_n\}$ достаточно большого размера. Значение n соответствует числу владельцев секрета, между которыми необходимо распределить секрет. Вычислим произведение k наименьших из этих чисел. Пусть это произведение равно N. Вычислим произведение $k-1$ наибольших из этих чисел. Пусть это произведение равно M. Таким образом, произведение любых $k-1$ элементов из множества $\{m_1, m_2, \ldots, m_n\}$ всегда меньше произведения любых k чисел из этого же множества. Число k будет называть порогом для конструируемой схемы на основе множества $\{m_1, m_2, \ldots, m_n\}$, если $M<N$. Выберем секретное число S, удовлетворяющее условию $M<S<N$. Между владельцами секрета распределим фрагменты секрета, представляющие собой пары чисел (r_i, m_i), где r_i есть остаток от деления S на m_i.

Если $t \geq k$ владельцев секрета объединят свои секретные фрагменты, то путем решения системы сравнений

$$\begin{cases} x \equiv r_1 \bmod m_1 \\ x \equiv r_2 \bmod m_2 \\ \quad \ldots\ldots \\ x \equiv r_t \bmod m_t \end{cases}$$

они смогут, используя китайскую теорему об остатках, вычислить множество чисел, удовлетворяющих системе, причем одно из чисел x_0 будет удовлетворять условию $x_0 < m_1 m_2 \cdot \ldots \cdot m_t$. Легко показать, что $x_0 = S$. Действительно, по

построению схемы разделения секрета S удовлетворяет рассматриваемой системе сравнений, причем $m_1 m_2 \cdot \ldots \cdot m_t \geq N > S$. Согласно китайской теореме об остатках, система сравнений имеет единственное значение, которое меньше произведения модулей $m_1 m_2 \cdot \ldots \cdot m_t$ и удовлетворяет системе, т. е. $x_0 = S$.

Покажем теперь, что при объединении $t' < k$ владельцев секрета они не смогут восстановить секрет путем решения системы сравнений

$$\begin{cases} x \equiv r_1 \bmod m_1 \\ x \equiv r_2 \bmod m_2 \\ \quad \ldots \ldots \\ x \equiv r_{t'} \bmod m_{t'} . \end{cases}$$

Пусть x'_0 есть наименьшее неотрицательное решение: $0 \leq x'_0 < m_1 m_2 \cdot \ldots \cdot m_{t'}$. Так как $t' < k$, то имеем $x'_0 < M < S$. Хотя известно, что $S = x'_0 + Q \cdot m_1 m_2 \cdot \ldots \cdot m_{t'}$, где Q — некоторое натуральное число (последнее соотношение следует из того, что оно задает все значения, удовлетворяющие рассматриваемой системе, а одним из таких значений является число S), но при соответствующем выборе множества $\{m_1, m_2, \ldots, m_n\}$ число Q настолько велико, что определить его подбором практически невозможно. Мы имеем следующее неравенство $x'_0 < M < S < N$, т. е. имеем

$$M < x'_0 + Q \cdot m_1 m_2 \cdot \ldots \cdot m_{t'} < N$$

$$M - x'_0 < Q \cdot m_1 m_2 \cdot \ldots \cdot m_{t'} < N - x'_0$$

$$\frac{M - x'_0}{m_1 \cdots m_{t'}} < Q < \frac{N - x'_0}{m_1 \cdots m_{t'}} .$$

Для определения Q требуется перебрать все целые значения в интервале от $\dfrac{M - x'_0}{m_1 \cdots m_{t'}}$ до $\dfrac{N - x'_0}{m_1 \cdots m_{t'}}$, т. е. опробовать порядка $\dfrac{N - x'_0}{m_1 \cdots m_{t'}} - \dfrac{M - x'_0}{m_1 \cdots m_{t'}} = \dfrac{N - M}{m_1 \cdots m_{t'}} \geq \dfrac{N - M}{M} = \dfrac{N}{M} - 1$ различных чисел. Если выбрать числа $m_1, m_2, \ldots, m_n$ таким образом, чтобы их разрядность в двоичном представлении была равна от 129 до 130 бит, то такой перебор практически нереализуем, поскольку отношение N/M составит более 2^{100} при $k < 15$.

Очевидно, что кроме выбора безопасного размера чисел $m_1, m_2, \ldots, m_n$ следует иметь в виду выбор безопасного значения k, однако это уже не криптографическая, а организационная проблема. Выбор этого параметра связан с оценкой вероятности преступного сговора k хранителей секрета.

7.4.2. Пороговые схемы на основе многочленов

Впервые построения схем разделения секрета на основе многочленов были предложены А. Шамиром. Рассмотрим некоторый многочлен $(k-1)$-й степени $y = f(x)$ над полем GF(p):

$$f(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \ldots + a_2x^2 + a_1x^1 + a_0,$$

где $a_{k-1}, a_{k-2}, \ldots, a_2, a_1, a_0 \in \mathrm{GF}(p)$. Этот многочлен можно задать непосредственно, указав свободный член и коэффициенты при всех степенях неизвестной. Но в данном случае нас будет интересовать другой способ задания многочлена, поскольку свободный член и/или некоторая совокупность коэффициентов будет рассматриваться как секрет, который разделяется между k хранителями. С учетом нашей цели видно, что значение простого числа p должно быть достаточно большим, в противном случае нарушитель сможет с большой вероятностью найти секрет методом подбора.

Известно, что многочлен $(k-1)$-й степени однозначно задается, если указать k точек, через которые он проходит. Под точкой понимается пара конкретных значений (y_i, x_i), где $y_i = f(x_i)$. Если заданы k точек, через которые проходит многочлен, то можно составить следующую систему из k линейных уравнений, в которой неизвестными будут свободный член и $k-1$ коэффициентов этого многочлена:

$$\begin{aligned}
f(x_1) &= a_{k-1}x_1^{k-1} + a_{k-2}x_1^{k-2} + \ldots + a_2x_1^2 + a_1x_1 + a_0 \\
f(x_2) &= a_{k-1}x_2^{k-1} + a_{k-2}x_2^{k-2} + \ldots + a_2x_2^2 + a_1x_2 + a_0 \\
&\ldots\ldots\ldots\ldots \\
f(x_k) &= a_{k-1}x_k^{k-1} + a_{k-2}x_k^{k-2} + \ldots + a_2x_k^2 + a_1x_k + a_0.
\end{aligned}$$

Решая эту систему, мы найдем неизвестные, а значит, и сам многочлен. Учитывая эти замечания, можно предложить следующую (n, k)-пороговую схему разделения секрета между n хранителями.

1. Выбираем достаточно большое простое число p, например, 512-битовое.
2. Генерируем случайные числа $a_{k-1}, a_{k-2}, \ldots, a_2, a_1, a_0 < p$.
3. Выбираем n различных значений аргумента x: $0 < x_1, x_2, \ldots, x_n < p$.
4. Вычисляем n значений многочлена, соответствующих выбранным значениям аргумента $x_1, x_2, \ldots, x_n$: $f(x_1), f(x_2), \ldots, f(x_n)$.
5. Пары значений $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$ распределяем между n хранителями секрета.

Любая система уравнений, полученная по $t \geq k$ точкам, будет совместной и иметь единственное решение. Действительно, это следует из построения координат точек (y_i, x_i). В то же время система, полученная по любым $t < k$ точкам, будет неопределенной, например при $t = k - 1$ она будет иметь p различных решений. Это означает, что $t \geq k$ хранителей секрета могут восстановить секрет, объединив свои «доли» секрета, а при их числе $t < k$ этого сделать нельзя. Возникает вопрос о частичном раскрытии секрета группой из $t < k$ владельцев «долей» секрета. Наибольшие возможности имеет их группа численностью $t = k - 1$. Если в качестве секрета выбрать свободный член или один из коэффициентов многочлена, то из решения системы из $k - 1$ уравнений эта группа хранителей секрета никакой информации не получит, поскольку секрет является одним из p равноправных решений.

При конкретной практической реализации данной схемы разделения секрета обычно используется интерполяционная формула Лагранжа, которая позволяет сделать вычисления более удобными, в частности, для программной реализации. Допустим, что мы выбрали в качестве секрета свободный член, т. е. $S = a_0 = f(0)$. При наличии t точек, через которые проходит многочлен, формула Лагранжа имеет вид:

$$f(x) = \sum_{i=1}^{i=t} y_i \prod_{j,i=1;\, j \neq i}^{t} \frac{x - x_j}{x_i - x_j}.$$

Подставляя в последнюю формулу значение $x = 0$, получаем

$$S = f(0) = \sum_{i=1}^{i=t} y_i \prod_{j,i=1;\, j \neq i}^{t} \frac{x_j}{x_i - x_j} = \sum_{i=1}^{i=t} y_i c_i \text{, где } c_i = \prod_{j,i=1;\, j \neq i}^{t} \frac{x_j}{x_i - x_j}.$$

Процедура восстановления секрета упрощается, если коэффициенты c_i вычислить предварительно. Действительно, можно заметить, что коэффициенты c_i не зависят от конкретного выбора многочлена (т. е. от значений $a_{k-1}, a_{k-2}, \ldots, a_2, a_1, a_0$), а зависят только от выбора значений аргумента $x_1, x_2, \ldots, x_n$, являющихся одной из координат точек, через которые проходит многочлен. Следовательно, эти значения могут быть вычислены заранее. Более того, если числа $x_1, x_2, \ldots, x_n$ задать как системный параметр, т. е. общими для всех случаев выбора многочлена, то тогда коэффициенты c_i будут общими для любого выбора секрета (и многочлена в целом) и их можно будет вычислять только один раз.

Достоинство схем разделения секрета на основе многочленов состоит в том, что они позволяют легко увеличивать число уже имеющихся владельцев секрета. Достаточно просто вычислить дополнительные точки, через которые проходит многочлен и раздать значения их координат в качестве долей сек-

рета новым хранителям. При этом, естественно, пороговое значение k остается неизменным, поскольку степень многочлена и он сам остались неизменными.

Допустим, один из хранителей допустил компрометацию своей доли секрета. Тогда его долю секрета можно раскрыть всем остальным владельцам секрета, в результате чего исходная (n, k)-пороговая схема превращается в $(n-1, k-1)$-пороговую.

7.4.3. Неоднородное ключевое пространство

Обычными требованиями для выбора секретных ключей симметричных криптосистем являются следующие: 1) длина ключа должна быть достаточно большой и 2) ключ должен выбираться случайно и с равной вероятностью по всему ключевому пространству. В ряде асимметричных криптосистем данное требование остается в силе, однако для некоторых других криптосистем, например RSA, секретный ключ должен удовлетворять определенным требованиям. В этом случае ключевое пространство оказывается в определенном смысле неоднородным.

В случае симметричных криптосистем имеется возможность использования неоднородного ключевого пространства для защиты от несанкционированного использования криптографического оборудования. (Например, нарушителями могут быть сотрудники, должностные обязанности которых не включают зашифрование документов и сообщений.) Несанкционированное использование систем шифрования может быть предотвращено, если в систему включить функции проверки структуры вводимого ключа. Если вводится ключ, обладающий определенными свойствами, то шифрование осуществляется в соответствии с основным алгоритмом, обладающим высокой стойкостью. Если вводимый ключ не обладает такими свойствами, то шифрование осуществляется с помощью ослабленного алгоритма или с помощью алгоритма, имеющего потайной ход.

Приведем возможный вариант реализации упомянутой симметричной криптосистемы. Принимается следующее соглашение о признаке, которым должен обладать ключ для выполнения стойкого криптографического преобразования: младшие (старшие) 64 разряда должны представлять собой значение, которое равно заданному идентификатору K_i, а остальные 192 разряда — случайный секретный ключ K_s, используемый для выполнения шифрующих преобразований. В рассмотренном примере правильный ключ имеет такую структуру $K = K_s \| K_i$ (или $K = K_i \| K_s$), где K_i — оговоренный секретный идентификатор. Можно предложить следующее развитие рассмотренного варианта. Ключ $K = K_s \| K_i$ вводится в устройство шифрования в преобразованном

виде $E_Q(K)$, где E_Q — функция блочного шифрования по ключу Q, который содержится в устройстве и является недоступным потенциальным нарушителям. Центр распределения ключей формирует для законных пользователей ключи вида $K = E_Q(K_s \| K_i)$. Устройство шифрования автоматически выполняет проверку $K_i = E_Q(K) \bmod 2^{64}$. Если проверка положительна, то подключается стойкий алгоритм. Вероятность того, что нарушитель получит доступ к стойкому алгоритму, равна 2^{-64}.

7.5. Криптографические протоколы

7.5.1. Понятие криптографического протокола

В криптографии часто используются термины «алгоритм» и «протокол». Интуитивно смысл этих терминов достаточно понятен. Они широко используются и в других научно-технических областях знаний. Алгоритм — одно из основных понятий в программировании и вычислительной математике, протокол — в связи. В дальнейшем изложении *под алгоритмом* мы будем понимать набор команд, действий, инструкций, вычислений, которые необходимо выполнить для того, чтобы из исходных данных получить тот или иной результат. При этом в процессе выполнения алгоритма могут возникать новые данные как результат преобразования исходных данных, либо как результат осуществления случайного выбора на каком-либо шаге алгоритма, либо как результат выполнения (вычислителем) измерений параметров окружения (внешних объектов). Алгоритм выполняется некоторым субъектом (вычислителем).

Под протоколом мы будем понимать совокупность действий (инструкций, команд, вычислений, алгоритмов), выполняемых в заданной последовательности двумя или более субъектами с целью достижения определенного результата. Корректность выполнения протокола зависит от действий каждого субъекта (пользователя, абонента) криптосистемы. В качестве субъектов могут выступать рабочая станция, программа для ЭВМ, радиопередатчик, космический спутник, оператор, сервер, орган власти и т. д. Обычно участвующие в протоколах той или иной системы субъекты действуют по предписанным алгоритмам, т. е. алгоритм выступает как внутренний элемент протокола. Для того чтобы протокол приводил к желаемой цели, необходимо выполнение следующих условий:

- корректность протокола — совокупность действий, предусмотренных протоколом, должна обеспечить получение требуемого результата при всех возможных ситуациях;

- полнота и однозначность определения — протокол должен специфицировать действия каждого участника протокола для всех возможных ситуаций;
- непротиворечивость — результаты, полученные различными участниками протокола, не должны быть противоречивыми;
- осведомленность и согласие участников протокола — каждый субъект заранее должен знать протокол и все шаги, которые он должен выполнить; все субъекты должны быть согласны играть свою роль.

Криптографические протоколы — это такие протоколы, в которых используются криптографические преобразования данных. Хотя криптографические протоколы часто применяют те или иные алгоритмы шифрования, их целью не обязательно является секретность. Например, стороны криптографического протокола могут желать одновременно подписать какой-либо контракт, провести электронную жеребьевку, идентифицировать участников телеконференции и т. п.

Шифрование данных и вычисление односторонних функций представляют собой выполнение соответствующего алгоритма. Приведенные выше схемы аутентификации пользователя и удаленной рабочей станции, проведения электронной жеребьевки являются примерами протоколов. Если протокол использует некоторую криптографическую функцию, то она должна быть стойкой. Даже если используемые алгоритмы шифрования являются стойкими, то это не гарантирует стойкость протокола. Для того чтобы криптографический протокол был стойким, необходимо, чтобы используемые криптографические алгоритмы были стойкими именно в условиях конкретного применения.

В криптосистемах предполагается существование потенциального нарушителя (в практических приложениях это идеальное понятие вполне материализовано). Разработчики криптографических алгоритмов и протоколов предусматривают по возможности наиболее полный перечень предполагаемых действий нарушителя (или группы нарушителей) и стремятся обеспечить достижение цели протокола с учетом всех возможных атак. Атака на алгоритм, протокол или криптосистему состоит в выполнении нарушителем таких действий, с помощью которых он пытается прочитать криптограмму, взломать односторонние функции (т. е. вычислить значение аргумента по значению функции), выдать себя за другого субъекта, навязать ложные сообщения, расширить свои полномочия и, в общем случае, создать условия, при которых корректность использования алгоритмов и протоколов криптосистемы будет нарушена. Если такие действия нарушителя возможны, то говорят, что криптосистема уязвима по отношению к такой-то атаке. По характеру своих действий можно выделить два типа нарушителей:

- пассивный нарушитель;
- активный нарушитель.

Пассивный нарушитель — это нарушитель, не предпринимающий действий по дезорганизации криптографического протокола. Его целью является перехват сообщений, циркулирующих в криптосистеме, с целью ознакомления с их содержанием, вычисления распределяемых ключей, раскрытия тайны голосования или результатов жеребьевки. Использование радиосвязи для передачи сообщений создает выгодные для пассивного нарушителя условия, в которых только косвенным путем можно установить тот факт, что криптосистема была подвергнута атаке. При использовании проводной связи несанкционированные подключения демаскируют пассивного нарушителя. Однако необходимо учитывать, что для перехвата сообщений он может использовать побочные электромагнитные излучения и наводки.

Активный нарушитель — это нарушитель, который пытается навязать ложные сообщения, перехватить и модифицировать сообщения, получить доступ к базам данных, расширить свои полномочия, навязать ложный открытый ключ, подготовить фиктивные документы, отказаться от подписи и т. д. При использовании проводной телефонной связи создаются условия, благоприятные для активного нарушителя, тогда как при использовании радиосвязи действия последнего могут быть легко обнаружены. Необходимо предусматривать случаи, когда активным нарушителем является легальный пользователь криптосистемы.

По характеру взаимоотношений с организацией использующих криптосистемы (или другие средства защиты) нарушителей можно отнести к следующим двум типам:

- внутренние нарушители;
- внешние нарушители.

Внутренним нарушителем является лицо, имеющее определенные легальные полномочия внутри организации, подвергающейся нападению с его стороны, или участник криптографического протокола, пытающийся нанести определенный ущерб другим участникам криптографического протокола. Внутренние и внешние нарушители могут быть активными или пассивными. Атака с привлечением внутреннего нарушителя называется атакой изнутри, или внутренней атакой.

Атака, в которой участвуют только *внешние* нарушители, называется внешней атакой. Возможен такой вид атак, когда внешние и внутренние нарушители объединяются, что создает наиболее серьезные угрозы безопасной эксплуатации криптосистем. Если нарушитель находится среди разработчиков, то возможны атаки на основе встроенных потайных ходов в алгоритмах

формирования ключевых параметров и трудно обнаруживаемых вредоносных программных закладок.

7.5.2. Временной замок

Можно ли послать некоторое послание M в будущее и быть уверенным, что никто не прочтет его до истечения, например, 50 или 100 лет, и при этом не полагаться на честность каких-либо доверенных лиц? Криптография решает этот вопрос в предположении высокой сложности задачи разложения числа на два больших простых множителя, которая предположительно не может быть решена за разумное время в обозримом будущем. Возможна следующая схема построения временного замка.

1. Сообщение M зашифровывается в соответствии с некоторым стойким алгоритмом симметричного шифрования по ключу K: $C = E_K(M)$. Криптограмма C предоставляется для общего доступа.

2. Ключ K зашифровывается по схеме $Q = K + b \bmod n$, где $b = \alpha^{2^t} \bmod n$, α — первообразный корень по модулю n, n — число, равное произведению двух больших простых чисел p и q, т. е. $n = pq$, t — число, задающее число операций, которые необходимо выполнить для вычисления значения b, т. е. время раскрытия ключа K. Значение Q предоставляется для всеобщего доступа. (При знании разложения числа n легко вычисляются значения функции Эйлера $\varphi(n)$, $T = 2^t \bmod \varphi(n)$ и затем $b = \alpha^T \bmod n$.)

3. Ключ K уничтожается. После чего запускается процесс его раскрытия, рассчитанный на выполнение t опробований различных значений $K' = Q - b' \bmod n$, где b' последовательно увеличивается в соответствии со следующей рекуррентной формулой: $b' \leftarrow (b')^2 \bmod n$ при начальном значении $b' = \alpha$.

Алгоритм раскрытия ключа K:

1. Проверить правильность расшифрования криптограммы C по ключу $K' = Q - b' \bmod p$. Если $M' = D_{K'}(C)$ соответствует исходному сообщению, то СТОП.

2. Изменить значение b' в соответствии с формулой $b' \leftarrow (b')^2 \bmod n$ и перейти к шагу 1.

Для данного уровня техники число попыток определяет время раскрытия ключа. Это позволяет выбором значения t задавать длительность временного интервала, в течение которого необходимо осуществлять вычисления для раскрытия сообщения.

***Замечание* 1**. В рассмотренном выше протоколе имеется требование выбора такого модуля n, чтобы разложение каждого из чисел $p-1$ и $q-1$ содержало, по крайней мере, один большой простой множитель. Если выбрано значение модуля n, то $\varphi(n)=(p-1)(q-1)$ может быть определено (тем, кто закрывает сообщение M) и значение b — вычислено по формуле $b = \alpha^{X^t \bmod \varphi(n)} \bmod n$, где X — число 2, относящееся к достаточно большому показателю по модулю $\varphi(n)$ ($X \geq 2$; выбором значения X можно в определенной степени регулировать сложность раскрытия ключа при фиксированном t).

Фактически мы построили некоторый «временной» замок, который раскрывается просто, но в течение временного интервала заданной продолжительности, причем ускорить задачу расшифрования криптограммы путем распараллеливания практически невозможно. Это можно было бы сделать, если найти разложение модуля n, что, по нашему предположению, является трудно решаемой задачей.

***Замечание* 2**. Имеются определенные проблемы в оценке длительности вычислений, которые необходимы для раскрытия сообщения. Эти проблемы связаны с возможными скачками производительности вычислительной техники. После запуска процесса раскрытия ключа с некоторого текущего значения t могут быть подключены новые более мощные вычислительные машины. Поэтому, несмотря на нераспараллеливаемость задачи раскрытия ключа, длительность ее решения может оказаться существенно меньше, чем предполагалось вначале.

***Замечание* 3**. Для того чтобы упростить распознавание того, что найдено правильное значение ключа K, а значит, и сообщения M, в последнее можно включить некоторый признак (заранее оговоренный идентификатор).

***Замечание* 4**. При неизвестном значении $\varphi(n)$ наиболее эффективная процедура поиска возможных значений параметра b состоит в последовательном возведении в квадрат (или в степень X: $b' \leftarrow (b')^X \bmod n$) и не поддается распараллеливанию. Если значение b искать, просто перебирая все возможные значения этого параметра, то любое разумное распараллеливание этого процесса потребует необозримого интервала времени для решения задачи. Попытка распараллелить поиск значения t связана с выполнением вычислений по формулам $T = 2^t \bmod \varphi(n)$ или $T = X^t \bmod \varphi(n)$, а вычисление $\varphi(n)$ требует знания разложения числа n.

7.5.3. Защита материальных объектов от подделки

Криптографическая защита документов и ценных бумаг от подделки является наиболее надежным современным способом пресечения их фальсификации. Она основана на уникальности любого конкретного материального носителя информации по своей микроструктуре. При наличии соответствующего оборудования (например сканера с высокой разрешающей способностью в случае анализа бумаги) можно выявить уникальные особенности структуры каждого экземпляра носителя из одной и той же заводской партии. Криптографическая защита от подделки осуществляется следующим образом. Считывается информация об уникальных особенностях данного конкретного носителя, формируется цифровой паспорт, включающий содержание документа и информацию о микроструктуре носителя. Затем законный изготовитель документа, используя свой секретный ключ, вычисляет цифровую подпись паспорта и записывает на носителе паспорт и соответствующую ему цифровую подпись.

Проверка истинности документа выполняется путем сканирования микроструктуры материального объекта, на котором сформирован документ, считывания записанной на нем информации и проверки цифровой подписи изготовителя документа по открытому ключу, который является общедоступным и публикуется, например, в ряде официальных изданий или распространяется по официальным каналам. Изготовление фальшивого документа на другом материальном объекте или модифицирование содержания документа (и его цифрового паспорта) практически неосуществимо без знания секретного ключа, с помощью которого формируется подпись. Любая подделка будет обнаружена путем считывания цифрового паспорта и цифровой подписи, сопоставления паспорта с содержанием документа и проверки подписи по открытому ключу (предполагается, что в этом способе защиты документов от подделки используется криптографически стойкая система электронной цифровой подписи).

7.5.4. Электронная жеребьевка

Жеребьевка как способ равноправного случайного выбора определенных вариантов достаточно широко используется в жизни. Например, при проведении футбольных чемпионатов для формирования подгрупп съезжаются представители команд-участниц и проводят жеребьевку, определяющую разбиение команд на группы. Более быстро и экономично это можно сделать «по телефону», а точнее выражаясь — по Интернету. Криптографические прото-

колы позволяют реализовать не только справедливую жеребьевку, но и проведение реальных карточных игр через компьютерные сети, а в общем плане могут служить основой электронного казино.

Для начала рассмотрим простейший вариант проведения *электронной жеребьевки*. Пусть удаленные абоненты **A** и **B** хотят сыграть по телефону партию в шахматы, причем они желают справедливо разыграть цвет фигур, т. е. обеспечить равную вероятность выбора белых фигур для каждого из них. Криптография позволяет реализовать эту жеребьевку по указанной ниже схеме, в которой используется односторонняя функция $y = F(x)$ и оговаривается, что абонент, угадывающий результат опыта с двумя равновероятными событиями, получает право первого хода.

1. Абонент **A** выбирает случайное число x_a, двоичное представление которого имеет, например, 80 разрядов, вычисляет значение $y_a = F(x_a)$ и сообщает величину y_a абоненту **B** (абонент **B** должен угадать четность числа x_a).
2. Поскольку используемая функция является однонаправленной (односторонней), то **B** не может по значению y_a определить x_a, поэтому он вынужден лишь угадать четность x_a. Пусть **B** останавливается на выборе «x_a является четным числом» и сообщает свое предположение абоненту **A**.
3. Абонент **A** сообщает абоненту **B** число x_a.
4. Абонент **B** вычисляет значение $y = F(x_a)$, и если $y = y_a$, то **B** убеждается, что его партнер действительно предоставил для проверки первоначально выбранное число.

Если результаты жеребьевки касаются не розыгрыша цвета фигур в шахматной партии на досуге, а используются для раздачи карт в игре на деньги в покер по телефону (схема случайной раздачи карт по телефону является только технически более сложной), то можно предусмотреть дополнительно использование системы ЭЦП для подписывания всех сообщений, используемых в схеме раздачи карт по телефону и назначении ставок.

В качестве примера экономической выгоды электронной жеребьевки можно привести возможность ее использования на чемпионате мира (Европы и т. д.) по футболу (баскетболу, волейболу и т. д.). Для проведения обычной жеребьевки регулярно в одно и то же место съезжаются представители команд-участниц и международных спортивных организаций. При этом затрачивается время и несутся значительные финансовые затраты. В случае замены этой процедуры электронной жеребьевкой экономится время, а финансовые затраты сводятся к минимуму. Другими примерами ее применения являются проведение розыгрыша лотерей, организация справедливого распреде-

ления путевок в престижные дома отдыха между сотрудниками различных учреждений или других ограниченных ресурсов.

7.5.5. Игра в покер по телефону (электронное казино)

Протокол электронной жеребьевки, рассмотренный в предыдущем разделе, обеспечивает процедуру справедливого угадывания некоторого элементарного события, а следовательно, может быть использован как базовый механизм для проведения различных карточных игр по телефону. Для этого его надо повторить достаточно большое число раз между участниками игр и соответствующим образом интерпретировать полученные результаты. Однако такая организация электронных карточных игр будет включать большое число деталей, которые будут вносить технические сложности при создании соответствующего программного обеспечения. В этом разделе будет рассмотрен протокол, обеспечивающий более эффективное решение вопроса электронной раздачи карт.

В некоторых играх, например электронной рулетке, вопрос фактически состоит в получении цепочки случайных битов, выбор которой определяется всеми участниками игры, а в некоторых случаях — и меньшим их числом. Простейшим вариантом является следующая схема. Каждый участник формирования общего случайного числа выбирает по своему усмотрению некоторое большое число, например, i-й участник выбирает число x_i. Затем каждый из них вычисляет значение односторонней функции $y_i = f(x_i)$ по соответствующему значению аргумента x_i. После этого каждый i-й участник подписывает значение y_i и вместе с подписью направляет число y_i остальным участникам. Затем каждый из них раскрывает свое число x_i перед остальными (теперь никто не может подменить свой выбор, поскольку правильность числа x_i проверяется по значению y_i). Теперь в качестве результата R случайного коллективного выбора можно взять значение некоторой заранее оговоренной хэш-функции H от сообщения, представляющего собой конкатенацию всех чисел x_i: $R = H(x_1, x_2, \ldots, x_n)$, где n — число участников. По этой схеме могут быть построены различные конкретные варианты протоколов совместного формирования случайных чисел. Имея такое многоразрядное случайное число, его можно интерпретировать как некоторое сложное случайное событие в соответствии с правилами той или иной игры.

В случае игры в покер дело несколько сложнее, поскольку эта игра является многошаговой, причем события на последующих шагах зависят от событий на предыдущих шагах, оставаясь в то же время случайными. Кроме того, случайный выбор одних участников не должен до определенного момента стать известным другим участникам, тогда как последующий случайный вы-

бор некоторого другого участника должен зависеть соответствующим образом от случайного выбора на предыдущем шаге. Например, если кто-то случайно выбрал пиковую даму, то на следующем шаге она не может появиться еще у одного игрока, хотя никто, кроме первого игрока, не должен знать, что дама пик уже «на руках». Непосредственное использование полученного выше случайного числа R даже для начального распределения карт между участниками игры проблематично, ввиду того что игроки не должны знать чужих карт.

Рассматривая проблему бесключевого шифрования, мы ознакомились с трехпроходным протоколом Шамира, практическое значение которого пока не было выявлено. В электронных карточных играх этот протокол является весьма востребованным. Точнее говоря, весьма эффективным для рассматриваемого случая является механизм закрытия сообщения (карты) в «сундук» с двумя и более замками, которые могут закрываться и открываться в любой очередности (благодаря коммутативности механизма шифрования). Рассмотрим схему построения протокола игры в покер по телефону с двумя игроками.

Игроки договариваются о выборе общего простого модуля p. Каждый из них формирует себе пару взаимно обратных по модулю $p - 1$ чисел, одно из которых будет ключом зашифрования (e), а другое — ключом расшифрования (d). Договоримся, что индексы переменных, обозначающих ключи, соответствуют номеру игрока. Пусть игра осуществляется с колодой из 52 карт, каждая из которых представлена как некоторое сообщение M_i, где $i = 1, 2, \ldots, 52$. Первый игрок зашифровывает по ключу e_1 все карты: $C_i = M_i^{e_1} \bmod p$, перемешивает полученные шифртексты C_i (случайно изменяет очередность их следований) и направляет их второму игроку. Теперь второй игрок не знает, какую карту представляет каждый из шифртекстов C_i. Второй игрок выбирает произвольные 5 шифртекстов, зашифровывает их на своем ключе:

$$C''_j = C_j^{e_2} \bmod p = M_j^{e_1 e_2} \bmod p \ (j = 1, 2, \ldots, 5).$$

Затем он направляет значения C''_j первому игроку, который их расшифровывает и возвращает значения $C'_j = C''^{d_1}_j \bmod p = M_j^{e_1 e_2 d_1} \bmod p = M_j^{e_2} \bmod p$. Теперь второй игрок раскрывает для себя выбранные им карты: $C'^{d_2}_j \bmod p = M_j^{e_2 d_2} \bmod p = M_j$. Затем он выбирает 5 карт для первого игрока C_k ($k = 1, 2, \ldots, 5$). Первый игрок раскрывает их для себя: $C_k^{d_1} \bmod p = M_k^{e_1 d_1} \bmod p = M_k$. Теперь раздача карт окончена. Остаток карт, зашифрованных на ключе e_1, хранится только у второго игрока. Если потребуется, то игроки сбрасывают карты следующим путем: зашифровывают на своем ключе

и представляют эти карты партнеру по игре (это делается для того, чтобы в дальнейшем нельзя было поменять сброшенные карты). Из колоды игроки получают карты таким же путем, как и при раздаче.

По окончании кона (сеанса) игры партнеры представляют друг другу свои ключи, с помощью которых проверяется корректность их действий, т. е. с их помощью может быть обнаружено жульничество. Следующий кон игры начинается с выбора каждым из игроков новой пары ключей. При наличии компьютера с соответствующим программным обеспечением все это происходит достаточно быстро. При серьезной игре на деньги каждое направляемое сообщение может быть подписано с помощью некоторой системы ЭЦП. Рассмотренная схема отражает основную идею — использование коммутирующих зашифровывающих и расшифровывающих преобразований. Возможны различные частные вариации данной схемы, отличающиеся техническими деталями. Нетрудно также видеть, что рассмотренная схема игры в покер по телефону может быть расширена на произвольное число игроков. При соответствующей проверке используемого программного обеспечения азартные игры по телефону и электронное казино в целом гораздо более надежны в плане защиты от обмана по сравнению с обычными играми. Кроме того, экономится время (если это азартным игрокам нужно), личная безопасность находится на более высоком уровне и практически неограниченно расширяется круг партнеров. Видимо, за электронными игорными домами большое будущее. Единственный недостаток — другой антураж, хотя и в этом аспекте у электронных игр имеется большая гибкость — в конце концов, можно пригласить в свою команду и друзей, совмещая вечеринку с игрой.

Следует отметить, что, несмотря на кажущуюся простоту приведенной выше схемы игры в покер по телефону, при числе участников больше двух требуется рассмотреть различные варианты мошенничества, например случаи сговора нескольких участников против одного. Возможности мошенничества путем сговора можно существенно ограничить, если модифицировать обычные правила игры в покер, согласно которым игроки, сбросившие карты, не раскрывают их и после окончании сеанса игры. Если принять правило раскрывать *все* карты по окончании сеанса (кона) игры, то протокол игры с многими участниками значительно упрощается по сравнению со случаем протокола игры в покер по телефону при обычных правилах.

7.6. Понятие слепой подписи

Ряд важных для практического применения криптографических систем включают в себя как составную часть протокол слепой подписи. К ним отно-

сятся 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 построение схемы слепой подписи на основе использования системы RSA было предложено Чаумом [19–21], оно основано на использовании особенностей преобразований системы RSA. Возможно ли построение некоторого варианта слепой подписи с использованием ЭЦП, основанных на сложности дискретного логарифмирования? На первый взгляд может показаться, что 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 следующий важный момент в полном объеме. Предназначением систем слепой подписи является обеспечение анонимности (неотслеживаемости). Однако в рассмотренной универсальной схеме подписывающий, получив возможность ознакомиться с некоторым подписанным им «вслепую» документом, имеет возможность вычислить значение хэш-функции и найти такое же в списке хэш-функций, который он может вести, с указанием лиц, представивших эти

хэш-функции для подписывания сообщений «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 возможности 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 значение 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. раздел 3.6.3*). Действительно, некоторое лицо формирует значение m', связанное с документом m, который нужно подписать «вслепую», следующим соотношением $m' = k^e m \pmod{N}$, где k — некоторое случайное чис-

ло, 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 в 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.

7.7. Протоколы с нулевым разглашением

Интересным классом протоколов являются протоколы с нулевым разглашением. Такое условное название они получили, ввиду того что в процессе их реализации один пользователь (доказывающий) может убедить другого (проверяющего) в том, что доказывающий владеет некоторым секретом, без раскрытия самого секрета. Вернее сказать, без того, чтобы дать какую-либо дополнительную информацию о секрете к той, что мог бы получить проверяющий из некоторых исходных посылок до начала реализации протокола. Возможное применение протоколов с нулевым разглашением связано с подтверждением подлинности удаленных субъектов (пользователей). В качестве примера ниже рассматриваются несколько протоколов такого типа.

7.7.1. Протокол Фиата–Шамира

Данный протокол основан на сложности извлечения квадратного корня по составному модулю, включающему не менее двух больших простых множителей, при условии, что разложение неизвестно. Доказывающий выбирает два больших простых числа p и q и вычисляет модуль $n = pq$. Затем выбирает случайное число s, такое что $1 \le s \le n - 1$, и вычисляет значение $t = s^2 \bmod n$. (В дальнейшем он будет доказывать проверяющему то, что он знает квадратный корень из t.) Значение t, которое объявляется всем участникам протокола, играет роль открытого ключа в смысле его использования для проверки того, что доказывающий знает s.

Протокол состоит из многократного повторения раунда, включающего следующие три шага.

1. Доказывающий выбирает случайное число k, такое что $1 \le k \le n-1$, вычисляет значение $q = k^2 \bmod n$ и посылает его проверяющему. (Число k играет роль разового ключа для защиты секрета от разглашения в ходе выполнения протокола.)
2. Проверяющий отправляет доказывающему случайный бит $r = 1$ или $r = 0$. (Фактически проверяющий направляет запрос: показать один из двух результатов некоторых вычислений, которые будут пояснены ниже. Если доказывающий покажет оба результата, то тогда проверяющий сможет раскрыть секрет. Протокол не должен этого допускать.)
3. Доказывающий вычисляет значение $x = ks^r \bmod n$ и направляет его проверяющему. (Если $r = 1$, то $x = ks \bmod n$. Если $r = 0$, то $x = k$. Видно, что по этим двум результатам легко вычисляется секрет s.)

Проверяющий считает ответ положительным, если выполняется соотношение $x^2 = qt^r \bmod n$. (Если $r = 1$, то должно выполняться $x^2 = qt \bmod n$. Если $r = 0$, то — $x^2 \bmod n = q$.) В ходе осуществления протокола выполняется z шагов. Вероятность того, что нарушитель (который не знает секрета s) при выполнении одного раунда может дать положительный ответ, равна 2^{-1}, следовательно, вероятность того, что нарушитель может быть принят за пользователя, знающего секрет s, составляет 2^{-z}. Выбирая в протоколе достаточно большое число раундов проверки, можно сделать сколь угодно низкой вероятность обмана.

Рассмотрим две возможные схемы действий нарушителя в одном раунде.

В первой схеме он выбирает произвольное число k и передает проверяющему значение $q = k^2 \bmod n$. Если он получит от проверяющего запрос $r = 0$, то направит правильный ответ $x = k$. Однако правильно ответить на запрос $r = 1$ нарушитель не имеет возможности.

Во второй схеме нарушитель выбирает произвольное число k и направляет проверяющему число $q' = k^2/t$. Если он получит от проверяющего запрос $r = 1$, то направит ответ $x' = k$, который будет принят проверяющим за правильный, поскольку

$$q't = (k^2/t)\,t = k^2 = x'^2 \bmod n.$$

Однако на запрос $r = 0$ нарушитель правильно ответить не сможет.

Таким образом, нарушитель в лучшем случае может правильно ответить только на один вопрос, и в одном раунде с вероятностью 1/2 попытка обмана обнаруживается.

7.7.2. Протоколы на основе системы RSA

Рассмотрим возможную практическую ситуацию с системой RSA, в которой имеется более естественная потребность в протоколе с нулевым разглашением. Пусть некоторый пользователь Штирлиц проявил незаурядность и раздобыл закрытый ключ d Мюллера. Теперь он хочет убедить в этом Шеленберга, но при этом не раскрывать самого секретного ключа. Если он представит Шеленбергу некоторый текст m и соответствующую ему подпись $s = m^d \bmod n$, Шеленберг легко может убедиться в правильности подписи, проверив выполнение соотношения $m = s^e \bmod n$, но он может подозревать, что Штирлиц просто раздобыл образец подписи к указанному сообщению. Направляя на подпись свое случайное число R, Шеленберг был бы уверен в правильности заявления Штирлица, если бы последний представил правильную подпись S, соответствующую указанному числу. Но в этом случае Штирлиц был бы обеспокоен, что это случайное число имеет специальную структуру, такую что, получив подпись к нему, Шеленберг расшифрует некоторое секретное послание C Мюллеру*, что может существенно снизить интерес к секретному ключу d.

> **Замечание*. С целью получения доступа к информации, содержащейся в криптограмме C, выбирается некоторое число t, являющееся взаимно простым с n, вычисляются обратное к t значение $t^{-1} \pmod n$, значения $t^e \bmod n$ и $m' = ct^e \pmod n$. Умножая подпись $s = (m')^d \bmod n$ на t^{-1}, можно восстановить сообщение M:
>
> $$st^{-1} = t^{-1}(m')^d = t^{-1}(ct^e)^d = t^{-1}C^d t^{ed} = t^{-1}C^d t = C^d = M \pmod n.$$

Указанные сомнения двух сторон снимаются следующим протоколом.

1. Штирлиц и Шеленберг, используя протокол бросания монеты, формируют случайное число R. (На этом шаге важно сформировать некоторое значение, которое зависит от выбора обеих сторон.)
2. Штирлиц подписывает совместно сгенерированное простое число R: $S(R) = R^d \bmod n$.
3. Шеленберг проверяет правильность подписи по открытому ключу e, используя стандартное уравнение проверки подписи $S^e \bmod n = R$.

Очевидно, что Шеленберг не имеет сомнений в том, что Штирлиц применил в своих действиях секретный ключ d, поскольку он сам принимал участие в формировании исходного случайного числа R. Однако может показаться, что для Штирлица такой вариант будет неприемлемым, ввиду того что Шеленберг получит возможность убедить Мюллера, что секрет последнего раскрыт (для таких действий могут возникнуть самые разные мотивы). Такая

попытка связана с предоставлением Мюллеру подписи к сообщению R, которое тот никогда не подписывал. Однако данная попытка неубедительна ввиду случайности R. (Действительно, взяв произвольное исходное значение S, по открытому ключу вычисляем $R = S^e \bmod n$, что дает пару значений — случайное сообщение R и подпись S — без использования секретного ключа.)

Рассмотрим теперь случай, когда Штирлицу необходимо продемонстрировать знание разложения модуля n (а следовательно, и владение секретом d) в кабинете у Шеленберга, т. е. с *использованием «ненадежной» вычислительной среды*. В этой ситуации Штирлиц может воспользоваться следующим протоколом с нулевым разглашением.

7.7.3. Протокол с одним раундом проверки

При необходимости доказать знание некоторого результата в обстановке, когда доказывающий не имеет возможности обеспечить защиту информации при выполнении вычислений, требуется выполнить несколько подготовительных шагов.

1. Штирлиц и Шеленберг совместно генерируют случайные числа C_1 и C_2.
2. Штирлиц заблаговременно в доверительной среде формирует разовый открытый ключ t и разовый секретный ключ k, такие что имеет место соотношение $tk = 1 \bmod \varphi(n)$, и записывает их на разных носителях в зашифрованном виде с использованием различных ключей шифрования. (Последнее делается для того, чтобы при раскрытии одного из значений t или k гарантировать секретность второго.) Затем он вычисляет: $C_1^{\,k} \bmod n = X_1$ и $C_2^{\,t} \bmod n = X_2$.
3. Штирлиц предоставляет Шеленбергу пары значений (C_1, X_1) и (C_2, X_2). Последний случайным образом выбирает значение бита r и предоставляет его Штирлицу.
4. Если $r = 0$, то Штирлиц предоставляет значение k. Если $r = 1$, то Штирлиц предоставляет значение t.
5. Если $r = 0$, то Шеленберг проверяет выполнимость следующих соотношений: $C_1^{\,k} \bmod n = X_1$ и $X_2^{\,k} \bmod n = C_2$. Если $r = 1$, то Шеленберг проверяет выполнимость следующих соотношений: $X_1^{\,t} \bmod n = C_1$ и $C_2^{\,t} \bmod n = X_2$.

Если указанные соотношения выполнены, то считается, что Штирлиц действительно знает разложение числа n. Данный протокол можно применить при удаленном доказательстве знания секрета, тогда случайные числа могут

формироваться по телефону с участием обеих сторон. Достоинство протокола состоит в том, что *не требуется выполнения большого числа раундов проверки*, что упрощает его практическое применение. Нетрудно видеть, что доказательство знания разложения модуля n может опираться на умение вычислять корни произвольной степени из произвольных чисел.

Рассмотрим протокол, в котором проверяющий выбирает значение степени корня.

1. Штирлиц и Шеленберг совместно генерируют случайное число C.
2. Шеленберг генерирует и направляет Штирлицу случайное простое число $e > n$. Такое e гарантирует существование обратного значения e^{-1} по модулю $\varphi(n)$.
3. Штирлиц вычисляет значения $d = e^{-1} \bmod \varphi(n)$ и $X = C^d \bmod n$ и направляет число X Шеленбергу.
4. Шеленберг проверяет выполнимость соотношения $X^e \bmod n = C$.

При положительной проверке делается заключение, что Штирлиц знает разложение числа n. Случайное простое число e может генерироваться совместно обеими сторонами протокола. Первый шаг протокола может быть устранен, если на третьем шаге доказывающий вычислит число X как корень e-й степени из некоторого общеизвестного случайного числа, например заданного 100-значным участком десятичной записи приближенного значения числа π.

Следующий однораундовый протокол, включающий два шага, может служить для доказательства знания дискретного логарифма числа y по простому модулю p при основании α.

1. Проверяющий генерирует случайное число $k < p - 1$, вычисляет и направляет доказывающему число $C = \alpha^k \bmod p$.
2. Доказывающий вычисляет и направляет проверяющему число $Z = C^x \bmod p$, где x — значение дискретного логарифма.

Далее проверяется выполнимость соотношения $y^k \bmod p = Z$. При положительном результате проверки делается вывод, что доказывающий знает значение указанного дискретного логарифма. Этот вывод основан на следующих формулах: $y = \alpha^x \bmod p$; $y^k = \alpha^{xk} = (\alpha^k)^x = C^x \bmod p = Z$. Заметим, что в этом протоколе число Z формируется подобно вычислению общего секрета в методе Диффи–Хеллмана.

Аналогичный протокол может быть построен для доказательства знания разложения составного модуля $n = pq$ (где p и q — простые числа, сравнимые

с числом 3 по модулю 4) посредством демонстрации умения извлекать квадратные корни.

1. Проверяющий генерирует случайное число $k < p - 1$, вычисляет и направляет доказывающему число $C = k^2 \bmod n$.
2. Доказывающий вычисляет и направляет проверяющему число $Z = C^{1/2} \bmod n$.

Далее проверяется выполнимость соотношения $Z^2 \bmod n = C$. При положительном результате проверки делается вывод, что доказывающий знает числа p и q.

В качестве упражнения читателю предлагается упростить приводимый ниже протокол.

- Штирлиц и Шеленберг, используя протокол бросания монеты по телефону, формируют случайное число r, такое что НОД(r, $\varphi(n)$) = 1, и значение k, для которого выполняется соотношение $rk = e \bmod \varphi(n)$, где e есть открытый ключ, соответствующий секрету d. Поскольку НОД(r, $\varphi(n)$) = 1, то можно вычислить значение $k = e/r \bmod \varphi(n)$. Это вычисление должен делать Штирлиц, так как Мюллер не знает значения $\varphi(n)$.
- Штирлиц и Шеленберг совместно формируют второе случайное число R. (Совместное формирование случайного числа дает каждой стороне гарантию, что это число действительно случайное.)
- Штирлиц, используя секретный ключ Мюллера, вычисляет значения $Y = R^d \bmod n$ и $X = Y^r \bmod n$ и представляет X на проверку Шеленбергу.
- Шеленберг проверяет выполнимость соотношения $X^k \bmod n = R$. Положительная проверка убеждает Шеленберга в том, что Штирлиц действительно знает секретный ключ d.

7.7.4. Протоколы на основе сложности разложения на множители

В протоколах данного типа используются вычисления по модулю составного числа n, представляющего собой произведение двух больших простых чисел. В этом состоит их аналогия с системой RSA, хотя имеются некоторые отличия, связанные с тем, что значение степени, используемой при вычислениях, не используется в качестве части открытого ключа.

***Пример* 1**. Открытым ключом является число n (его разложение $n = pq$ является секретом доказывающего) и число $Y = X^e \bmod n$, где e – специфицированное (т. е. известное доказывающему и проверяющему) простое число, X — секретное число доказывающего. Обозначим передачу сообщения от доказывающего (A) к проверяющему (B) формулой $A \to B$: M. Протокол осуществляется как многократное повторение следующего раунда интерактивного взаимодействия:

1. $A \to B$: Z; $Z = K^e \bmod n$, где K — случайное число (разовый секрет).
2. $B \to A$: r; $r = 0$ или $r = 1$.
3. $A \to B$: W; $W = K$ (если $r = 0$) или $W = KX$ (если $r = 1$).

В качестве проверочного соотношения используется $W^e \bmod n = Z$ (если $r = 0$) или $W^e \bmod n = ZY$ (если $r = 1$). На запрос $r = 1$ ($r = 0$) нарушитель может предпринять такую атаку: на первом шаге он направляет проверяющему «правильное» («неправильное») значение $Z = W^e \bmod n$ ($Z = Y^{-1} \cdot W^e \bmod n$), однако на запрос $r = 1$ ($r = 0$) нарушителю вычислительно сложно найти правильный ответ, поскольку для этого ему надо решить задачу извлечения корня e-й степени по модулю n. Вероятность обмана на текущем шаге составляет 0.5.

***Пример* 2**. Открытым ключом является число n (его разложение $n = pq$ является секретом доказывающего). В этом примере используются случайные числа R и e, причем e — простое число и $e > n$, кроме того, R генерируется с участием обеих сторон.

1. $B \to A$: e; e — случайное простое число, такое что $e > n$.
2. $A \to B$: W; $W = R^d \bmod n$, где d вычисляется доказывающим по формуле $d \equiv e^{-1} \bmod \varphi(n)$.

В качестве проверочного соотношения используется $W^e \bmod n = R$. Для доказательства знания разложения модуля достаточно всего двух шагов протокола.

Для того чтобы выдать себя за абонента A, нарушитель должен извлечь корень e-й степени по модулю n, что является вычислительно сложной задачей, а вероятность угадать правильный ответ пренебрежимо мала для достаточно больших значений R. Доказывающий знает разложение модуля n, поэтому он может вычислить функцию Эйлера от n и вычислить $d = e^{-1} \bmod \varphi(n)$.

7.7.5. Протоколы на основе сложности дискретного логарифмирования

Рассмотрим протокол с нулевым разглашением на основе сложности задачи дискретного логарифмирования по большому простому модулю. Пусть в системе Диффи–Хеллмана доказывающий знает секретный ключ x, соответствующий открытому ключу $y = \alpha^x \bmod p$, т. е. знает дискретный логарифм y по $\bmod p$. Протокол состоит из многократного выполнения следующего раунда:

1. Доказывающий выбирает текущий разовый секрет k, вычисляет значение $z = \alpha^k \bmod p$ (которое играет роль разового открытого ключа) и передает число z проверяющему.
2. Проверяющий случайным образом выбирает значение бита r и предоставляет его доказывающему.
3. Если $r = 0$, то доказывающий раскрывает значение k, а если $r = 1$, то вычисляет значение $w = x + k \bmod (p - 1)$ и направляет его проверяющему.
4. Если $r = 0$, то проверяющий проверяет выполнимость соотношения $\alpha^k \bmod p = z$. Если $r = 1$, то проверяющий проверяет выполнимость соотношения $y \cdot z = \alpha^w \bmod p$.

При положительной проверке делается заключение, что с вероятностью 0.5 доказывающий знает значение x. Этот протокол должен включать много итераций, для того чтобы достичь пренебрежимой вероятности обмана. Необходимо учитывать, что на первом шаге каждой итерации доказывающий должен выбирать новые случайные значения разового секретного ключа k.

Более простое доказательство знания секретного ключа состоит в использовании некоторой системы ЭЦП, основанной на сложности задачи дискретного логарифмирования. Проверяющий и доказывающий совместно генерируют случайное число C, затем доказывающий подписывает это число. Проверив подпись по открытому ключу, проверяющий убеждается в том, что доказывающий знает секретный ключ. Однако после такой проверки у проверяющего оказывается в руках подпись к C, поэтому он может доказать третьей стороне, что доказывающий действительно знает секретный ключ. (Возможность разглашения этого факта.) После предыдущего вероятностного протокола проверяющий этого сделать не может.

Рассмотрим, какие атаки может предпринять нарушитель на вероятностный протокол доказательства знания дискретного логарифма. В первом варианте атаки нарушитель в качестве открытого ключа направляет «правильное» значение $z = \alpha^k \bmod p$, тогда он без труда проходит проверку при $r = 0$, но

при запросе $r = 1$ обман обнаруживается. Во втором варианте нарушитель случайно выбирает число $w \le p - 1$, вычисляет «неправильное» значение $y' = y^{-1} \cdot \alpha^w \bmod p$ и при запросе $r = 1$ имеет возможность обмануть проверяющего, направляя последнему число w для осуществления выполнимости проверочного соотношения. Однако при запросе $r = 0$ обман обнаруживается. Таким образом, в обоих случаях обман обнаруживается с вероятностью 0.5.

Заметим, что последний протокол можно несколько модифицировать, представляя его базовый раунд в таком виде:

1. $A \to B$: $z = y^k \bmod p$.

2. $B \to A$: r.

3. $A \to B$: $w = k$ (если $r = 0$) или $w = kx$ (если $r = 1$).

В качестве проверочного соотношения В использует формулу $y^w \bmod p = z$ (если $r = 0$) или $\alpha^w \bmod p = z$ (если $r = 1$). На запрос $r = 1$ ($r = 0$) нарушитель может предпринять такую атаку: на первом шаге он направляет проверяющему «неправильное» («правильное») значение $z = \alpha^w \bmod p$ ($z = y^k \bmod p$), однако на запрос $r = 0$ ($r = 1$) нарушитель не может найти правильный ответ, поскольку для этого ему надо решить задачу дискретного логарифмирования по модулю p.

Протоколы с нулевым разглашением можно отнести к системам с открытым ключом, которые выполняют задачу аутентификации пользователей. Это значительный шаг в развитии криптографических протоколов при переходе от одноключевых к двухключевым. Системы такого типа еще не выполняют шифрование и не позволяют реализовать цифровую подпись, но они уже относятся к двухключевым. При этом протоколы с нулевым разглашением могут быть построены с использованием более разнообразных типов трудно решаемых задач. Системы цифровой подписи имеют больше ограничений на задачи, которые могут лежать в их основе, но они реализуют больше возможностей, обеспечивая как аутентификацию пользователей, так и придание юридической силы электронной информации. В этом смысле протоколы с нулевым разглашением являются менее «продвинутыми» двухключевыми системами, чем системы ЭЦП. Последние предоставляют более эффективное решение задачи аутентификации пользователей и обеспечивают протоколы с новыми возможностями важного практического значения.

7.8. Инфраструктура открытых ключей (PKI)

Двухключевая криптография предоставляет наиболее удобные механизмы аутентификации, т. е. подтверждения подлинности пользователей и элек-

тронных документов. Этим определяется ее значение для обеспечения информационной безопасности в информационно–вычислительных сетях. Удобство применения двухключевых криптографических алгоритмов для придания юридической силы электронным сообщениям связано с тем, что протоколы на их основе не требуют наличия третьей доверительной стороны (например арбитра) при удаленном взаимодействии через сети ЭВМ не доверяющих друг другу сторон. Однако проблема аутентификации ключей сохраняется и в двухключевой криптографии, хотя она уже решается проще, поскольку аутентификации подлежат открытые ключи, которые не должны держаться в секрете. Аутентификации открытых ключей достаточно, чтобы обеспечить шифрование по открытым ключам и открытое распределение ключей.

Для применения систем ЭЦП требуется еще решить вопрос об отказе от открытых ключей. Этот вопрос решается обычными юридическими процедурами, а именно, владелец открытого ключа (он же является также и единственным владельцем секретного ключа, соответствующего данному открытому ключу) должен взять на себя юридически оформленное обязательство признания того, что данный открытый ключ сформирован и представлен им для выполнения процедуры проверки подлинности подписи в рамках той или иной системы ЭЦП. При этом в системах распределения открытых ключей достаточно, чтобы пользователь один раз оформил такое обязательство традиционным способом. Далее подлинность и юридическая ответственность может распространяться электронным способом с использованием системы ЭЦП. Например, удостоверяющий центр (этот центр не играет роль арбитра и не выступает в роли доверительной стороны) рассылает под свою ответственность справочники открытых ключей, подписанные его цифровой подписью. Подлинность подписи удостоверяющего центра пользователи проверяют с использованием открытого ключа удостоверяющего центра (УЦ). При этом их доверие к ЭЦП основано на наличии у них обязательства удостоверяющего центра, оформленного традиционным способом с указанием объема ответственности УЦ за все документы, подлинность цифровой подписи к которым подтверждается процедурой проверки ЭЦП с использованием открытого ключа, указанного в обязательстве.

При большом числе пользователей система поддержания открытых ключей становится достаточно сложной. Вся совокупность механизмов и правил, связанных с распределением, аутентификацией и юридической поддержкой открытых ключей, называется инфраструктурой открытых ключей (ИОК). Иногда используется термин PKI (пи-ки-ай) — аббревиатура от англоязычного термина *Public Key Infrastructure*. Как составную часть ИОК включает сервисы для распространения справочников открытых ключей и электронных сертификатов с открытыми ключами пользователей, прикладное программ-

ное обеспечение для создания баз открытых ключей с указанием их атрибутов (владельцев, сроков действия и т. п.).

Таким образом, функционирование ИОК основано на программно-технических средствах, криптографических механизмах, обеспечивающих придание юридической силы электронным сообщениям, и определенной совокупности организационно-нормативных мероприятий. Открытые ключи распространяются в виде электронных справочников и в виде цифровых (электронных) сертификатов.

Электронный сертификат фактически представляет собой цифровое сообщение (возможно, расширенное), которое имеет структуру, аналогичную записям, закрепленным в электронном справочнике открытых ключей за конкретными пользователями. При этом цифровой сертификат может дублировать информацию справочника открытых ключей или служить в качестве дополнения к справочнику, т. е. выполнять роль последнего применительно, например, только к одному пользователю (в принципе цифровые сертификаты могут быть выданы и на группу пользователей с указанием соответствующего числа различных открытых ключей и их принадлежности). Главным, что указывается в цифровом сертификате, является открытый ключ и принадлежность конкретному пользователю, а также цифровая подпись УЦ как органа, ответственного за распространение открытых ключей.

Иногда УЦ представляют состоящим из двух компонентов — центра сертификации (ЦС) и центра регистрации (ЦР). Отвлекаясь от совокупности криптографических механизмов и программно-технических и организационно-правовых средств поддержания ИОК, можно выделить следующие компоненты ИОК:

- центр сертификации;
- центр регистрации;
- конечные пользователи;
- справочник открытых ключей и цифровой сертификат.

В развитых ИОК может присутствовать большое число ЦС и ЦР, формирующих определенную иерархическую структуру с выделением различных уровней подчиненности.

Центр сертификации выполняет основные функции УЦ, а именно, формирует электронные справочники и цифровые сертификаты подчиненных центров и конечных пользователей. Отмена старых и введение новых открытых ключей связаны с необходимостью реализации механизмов отзыва сертификатов. Формирование списка отозванных сертификатов также является

функцией ЦС. Периодичность обновления справочника открытых ключей и списка отозванных сертификатов обычно регламентируется.

К основным функциям ЦС можно отнести:

- подготовку традиционных документов о юридической ответственности за свой открытый ключ и свою ЭЦП к распространяемым сообщениям;
- формирование собственного секретного ключа и сертификата ЦС;
- формирование сертификатов открытых ключей конечных пользователей и подчиненных центров;
- формирование списка отозванных сертификатов;
- ведение базы всех изготовленных сертификатов и списков отозванных сертификатов.

Центр регистрации осуществляет регистрацию конечных пользователей и обеспечение их взаимодействия с ЦС. Другими функциями, выполняемыми ЦР, могут являться распространение и публикация справочника открытых ключей (он может быть представлен, например, в виде списка действующих сертификатов), отдельных сертификатов и их списков, а также списков отозванных сертификатов.

В системах ИОК с иерархической структурой УЦ имеется проблема проверки подлинности сертификатов, выданных одним из УЦ, конечными пользователями других УЦ, например в случае принадлежности пользователей различным УЦ одного уровня подчиненности. В таких случаях вопрос доверия любому сертификату решается на основе проверки цепочки сертификатов. Цепочка сертификатов представляет собой последовательность из n сертификатов, в которой:

- для всех $i \in \{1, (n-1)\}$ владелец (обычно один из УЦ или ЦС, хотя в этой роли может выступить также и некоторый пользователь) сертификата S_i является издателем сертификата S_{i+1};
- сертификат S_1 есть самоподписанный сертификат главного (корневого) УЦ («самоподписывание» не несет юридической нагрузки, а служит скорее для сохранения однотипности формата сертификата; юридическая значимость открытого ключа проверки «самоподписи» определяется наличием у проверяющего обычного документа, подтверждающего подлинность этого открытого ключа или подлинность открытого ключа другого УЦ, электронной подписью которого заверен открытый ключ главного УЦ);
- сертификат S_n является сертификатом конечного пользователя.

На рис.7.2 представлена схема ИОК с трехуровневой иерархией УЦ. Для того чтобы все пользователи этой системы могли проверить подлинность сертификатов друг друга, каждый из УЦ, к которому они принадлежат, распределяет между пользователями подписанный этим УЦ справочник сертификатов (или справочник открытых ключей), в котором указаны открытые ключи главного УЦ и всех подчиненных УЦ, через которые проходит путь от данного пользователя к главному УЦ.

Ввиду наличия процедуры обновления сертификатов, возникает необходимость проверки цепочки списков отозванных сертификатов (СОС), которая представляет собой последовательность из n СОС: $R_1, R_2, \ldots, R_n$, где R_1 есть СОС, изданный главным УЦ. Список R_i формируется и распространяется (публикуется) издателем сертификата S_i.

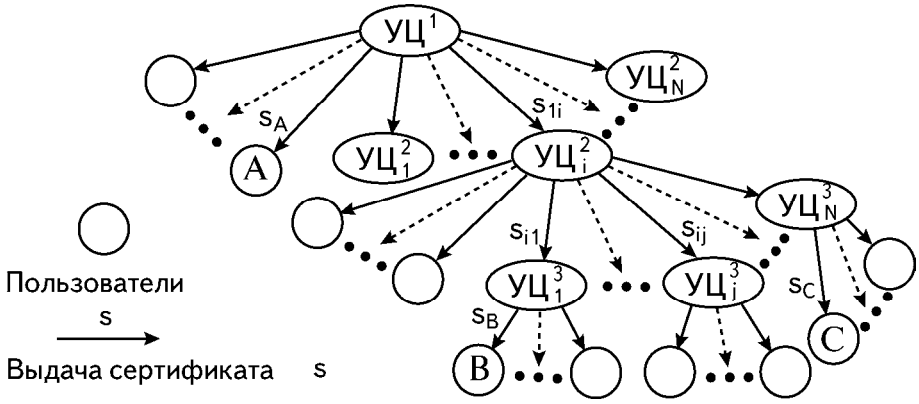


Рис. 7.2. Трехуровневая иерархия удостоверяющих центров

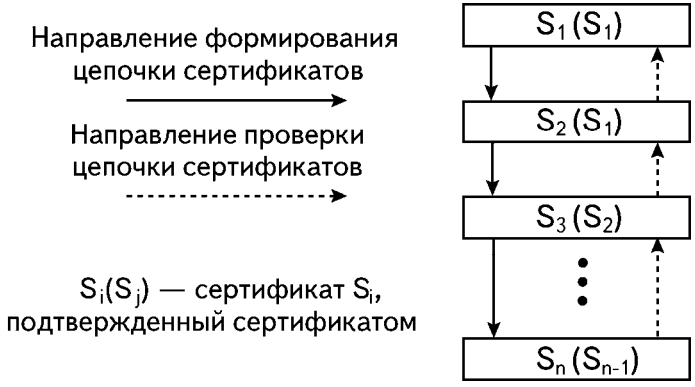


Рис. 7.3. Схема формирования и проверки цепочки сертификатов

Верификация сертификатов в общем случае осуществляется посредством верификации соответствующих цепочек сертификатов. При проверке цепочки сертификатов предполагается проверка также и соответствующей цепочки СОС. Перед верификацией строится цепочка сертификатов и соответствующая ей цепочка СОС, затем выполняется:

- проверка подлинности и сроков действия сертификатов и СОС;
- проверка сертификатов на отзыв (если сертификат S_i подчиненного УЦ был отозван списком R_{i-1} вышестоящего УЦ, все сертификаты, изданные подчиненным УЦ, признаются недействительными).

ГЛАВА 8

Криптографический практикум

8.1. Задания для практических занятий

Предлагаемые темы для выполнения практических работ направлены на закрепление материала, относящегося к двухключевой криптографии — схемам открытого распределения ключей, открытому шифрованию и системам электронной цифровой 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 задания для курсового проектирования 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 используемых в криптосистемах математических результатах. Заслуживают пояснения и некоторые частные аспекты. Например, алгоритм открытого шифрования Эль-Гамаля фактически представляет собой гибридную криптосистему, в которой открытое распределение ключей осуществляется по методу Диффи–Хеллмана, а симметричное шифрование выполняется как модульное умножение блока данных на одноразовый секретный ключ. При выполнении практической работы по ЭЦП Эль-Гамаля целесообразно подчеркнуть важность сохранения в тайне генерируемого случайного числа, поскольку знание этого числа, которое используется при генерации цифровой подписи, позволяет вычислить секретный ключ.

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. Полезным также является пояснение, что слепая подпись используется при решении за-

дач обеспечения анонимности, поэтому, получив от третьих лиц заверенный документ, подписавший не должен иметь возможности однозначно установить, к какому случаю формирования слепой подписи этот документ относится (даже если в каждом процессе формирования слепой подписи подписывающий регистрировал все данные, относящиеся к этому процессу).

8.1.1. Открытое распределение ключей

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 A и B хотят установить секретную связь, то они поступают следующим образом. Абонент A берет из справочника открытый ключ абонента B 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 A и B; x_A и x_B — соответствующие секретные ключи. Нет необходимости передавать по сети связи общий секретный ключ Z_{AB}, поскольку абонент B по известному из справочника открытому ключу абонента A 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 воз-

можных 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: «Открытое распределение 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:

$$C = K^{d} \bmod n.$$

Получатель расшифровывает сеансовый ключ с использованием своего секретного ключа:

$$K = C^{e} \bmod n.$$

Однако получатель должен получить гарантии того, что расшифрованный ключ был действительно отправлен подлинным отправителем. Аутентификация источника сообщения требует использования открытого ключа отправителя, поэтому отправитель должен подписать посланную криптограмму по своему секретному ключу d': $S = H^{d'} \bmod n'$, где H — хэш-функция от криптограммы C. Затем присоединить подпись S к отправляемому значению C. Если модуль отправителя n' больше модуля получателя, то он может отправить только значение $S = C^{d'} \bmod n'$, поскольку получатель, проверяя «подлинность» подписи в этом случае, может восстановить значение C, а затем по своему секретному ключу расшифровать C и получить значение ключа. Однако окончательная подлинность отправителя будет установлена только после того, как отправитель правильно зашифрует или расшифрует некоторое случайное пробное сообщение. Отправитель может подписать и хэш-функцию, полученную от значения ключа. (Подпись, полученная непосредственно по значению ключа, фактически раскрывает ключ, поэтому в открытом виде пересылаться по каналам связи не должна.)

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.

8.1.2. Открытое шифрование

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 $y = \alpha^x \bmod p$ 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 секретный 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 шифрование по Рабину»

Теоретическая часть. В схеме открытого шифрования Рабина используется RSA-модуль $n = pq$, в котором числа p и q сравнимы с числом 3 по модулю 4: $p \equiv 3 \bmod 4$ и $q \equiv 3 \bmod 4$, что обеспечивает при знании разложения модуля (числа p и q являются секретным ключом) возможность выполне-

ния операции извлечения квадратного корня из квадратичных вычетов по модулю n.

Открытым ключом является значение n, с помощью которого сообщение $M<n$ зашифровывается путем возведения числа M в квадрат по модулю n:

$$C = M^2 \bmod n.$$

Процедура расшифрования состоит в извлечении квадратного корня из криптограммы C (очевидно, что она является квадратичным вычетом) по модулю n. Предварительно вычисляют корни из C по модулям p и q:

$$m_{p_1} = C^{\frac{p+1}{4}} \bmod p; \quad m_{p_2} = p - m_{p_1} = p - C^{\frac{p+1}{4}} \bmod p;$$

$$m_{q_1} = C^{\frac{q+1}{4}} \bmod q; \quad m_{q_2} = q - m_{q_1} = q - C^{\frac{q+1}{4}} \bmod q.$$

Из этих четырех значений вычисляются четыре возможных корня из C по модулю n:

$$M_1 = \left(m_{p_1} a + m_{q_1} b \right) \bmod n;$$

$$M_2 = \left(m_{p_1} a + m_{q_2} b \right) \bmod n;$$

$$M_3 = \left(m_{p_2} a + m_{q_1} b \right) \bmod n;$$

$$M_4 = \left(m_{p_2} a + m_{q_2} b \right) \bmod n;$$

где $a = q(q^{-1} \bmod p)$ и $b = p(p^{-1} \bmod q)$. Расшифрование неоднозначно. Для задания однозначности перед зашифрованием к исходному открытому сообщению можно присоединить некоторую заранее оговоренную метку.

Экспериментальная часть. По указанию преподавателя обучающимся индивидуально задаются значения модуля n. Задание состоит в осуществлении зашифрования и расшифрования некоторой совокупности сообщений. Значения m_{p_1}, m_{p_2}, m_{q_1}, m_{q_2}, M_1, M_2, M_3 и M_4 записываются в таблицу результатов вычислений.

Тема: «Вычисление мультипликативно обратных элементов в поле вычетов»

Теоретическая часть. Для вычисления обратных элементов в поле вычетов используется расширенный алгоритм Евклида. Известно, что если чис-

ла 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 вычислений по двум способам.

8.1.3. Схемы ЭЦП

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 x_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 не допускается использовать одно и то же значение k 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, $r = \alpha^k$, s, α^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: «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 простому 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 составлять 160 бит. В этом случае вычисляемое значение s будет иметь размер, не превышающий 160 бит. Благодаря этому достигается сокращение длины подписи почти в два 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, $r = \alpha^{k}, s, \alpha^{M}, y_A^{r}, r^{s}, y_A^{r} r^{s} \pmod p$. 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

$$\alpha^M = y_A^r r^s \pmod p$$

в схеме с сокращенным параметром s ($s < q$) 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. С учетом сокращенной длины параметра s видим, что общая длина подписи составляет примерно 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 исходного 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. Сформировать простое число p, для которого разложение числа $p - 1$ содержит простой делитель q заданного размера. Найти число α, относящееся к показателю q по модулю p. Сформировать секретный ключ x_A, вычислить соответствующий ему открытый ключ $y_A = \alpha^{x_A} \bmod p$. Вычислить значение электронной подписи для 5 различных сообщений, фиксируя получаемые значения следующих параметров 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.

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 малой.

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:

$$S = M^d \bmod n.$$

Процедура проверки подписи:

$$M' = S^e \bmod n.$$

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: «"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 $S = M^d \bmod n$. Результаты оформить в виде таблицы.

Тема: «Схема подписи Онга–Шнорра–Шамира»

Теоретическая часть. Данная схема основана на использовании составного модуля n, что обеспечивает сложность извлечения квадратных корней по модулю n. Открытый ключ состоит из двух частей: RSA-модуля n и числа h, которое генерируется следующим образом. Генерируется секретный ключ в виде случайного числа k, взаимно простого с модулем n. Затем по секретному ключу вычисляется h:

$$h = -(k^{-1})^2 \bmod n = -k^{-2} \bmod n .$$

Для вычисления подписи (S_1, S_2) к сообщению M выбирается случайное число r, такое что r и n являются взаимно простыми. Затем используются соотношения:

$$S_1 = \frac{1}{2}[\frac{M}{r} + r] \bmod n ;$$

$$S_2 = \frac{k}{2}[\frac{M}{r} - r] \bmod n .$$

Уравнение проверки подписи имеет вид

$$M = (S_1^2 + hS_2^2) \bmod n .$$

Следует отметить, что для нахождения секретного ключа и формирования подписи нет необходимости знать разложение модуля на множители, однако оба множителя должны быть большого размера. Также следует заметить, что

данная схема подписи не обладает необходимой криптографической стойкостью.

Экспериментальная часть. Используя заданные значения простых чисел p и q, вычислить модуль n, затем сформировать секретный и открытый ключи k и h. Для сообщений M из заданного множества (число различных сообщений выбирается в зависимости от размера модуля $n = pq$) вычисляется подпись и выполняется проверка правильности подписи. Результаты оформляются в виде сводной таблицы.

Тема: «Схема RSA-подобной ЭЦП»

Теоретическая часть. В данной схеме используется RSA-модуль $n = pq$, причем простые множители p и q таковы, что числа $p-1$ и $q-1$ содержат большой простой делитель γ заданного размера, например, $\gamma \le 160$ бит. Секретным ключом служит тройка чисел (p, q, γ). Открытым ключом является пара чисел (n, α), где α является числом, относящимся к показателю γ по модулю n. Для вычисления подписи используется формула

$$S = \alpha^{H^{-1} \bmod \gamma} \bmod n,$$

где H — значение хэш-функции от подписываемого документа.

Проверка подписи осуществляется по формуле $\alpha = S^H \bmod n$.

Экспериментальная часть. Формируется RSA-модуль с требуемой структурой, причем γ^2 не делит ни одно из чисел $p-1$ и $q-1$. Формируется открытый ключ и вычисляются подписи к заданному числу сообщений (или для заданных значений хэш-функции от сообщений). Затем проверяется правильность подписи к каждому сообщению. Результаты оформляются в виде таблицы. Для больших значений модуля берется меньшее число подписываемых сообщений. Генерация числа α осуществляется следующим образом. Выбирается число β, являющееся первообразным корнем одновременно по mod p и mod q. Затем вычисляются $z = \varphi(n)/\gamma^2$ и $\alpha = \beta^Z \bmod n$.

8.1.4. 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 детерминистических 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 p).

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 табл. 8.1.

- *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 принятия составного числа в качестве псевдопростого. Первый тест используется в качестве предварительной отбраковки чисел. Второму тесту подвергают только числа, прошедшие первый. (Второй тест на самом деле поглощает первый, поскольку проверка условия $b^{\frac{p-1}{2}} \bmod p = 1$ для значений b, являющихся квадратичными вычетами, фактически означает проверку по тесту Ферма.)

Таблица 8.1

Примеры чисел Кармайкла

Число	Разложение	Число	Разложение
561	3·11·17	6 601	7·23·41
1105	5·13·17	8 911	7·19·67
1729	7·13·19	41 041	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

Примеры второго подхода приводятся в следующих двух темах практических работ.

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 достаточно быстрого формирования псевдопростых чисел p, для которых разложение числа $p-1$ содержит заданное число множителей заданного размера.

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 — вероятностному 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. Заметим, что вычислять символ Лежандра $\left(\frac{b}{p}\right)$ каким-либо другим методом для выполнения этого теста не требуется. При простом числе p символ Лежандра $\left(\frac{b}{p}\right) = b^{\frac{p-1}{2}} \bmod p$ будет принимать только два значения, а именно, 1 и $p-1$ для случайно выбираемой последовательности значений основания b. Если p окажется составным числом, то вычисление параметра $b^{\frac{p-1}{2}} \bmod p$ даст значения, отличные от 1 и $p-1$.

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 больше средней длины чисел $q_1, q_2, \ldots, q_k$ (например от $8h$ до $10h$ десятичных знаков). Можно аналогичным образом сформировать 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 атакующий не сможет воспользоваться этим обстоятельством 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 число и $p < (2q+1)^2$. Число p является простым, если выполняются следующие два условия:

1. $2^{qN} = 1 \bmod p$ и
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 $\dfrac{p-1}{q}$. 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:

$$p = mp_i = m(qn + 1) = qN + 1 \Rightarrow m = q(N - mn) + 1.$$

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. $2^{p_i N} = 1 \bmod p_{i-1}$ и
2. $2^{N} \neq 1 \bmod p_{i-1}$.

Если хотя бы одно из условий не выполнено, то значение N увеличивается на 2, 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}.

8.2. Задачник

8.2.1. Арифметические задачи

1. Показать существование чисел, относящихся к простому делителю функции Эйлера от модуля как к показателю.
2. 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 d.
3. Пусть известно разложение числа $p-1$, где p есть большое простое число. Указать вычислительно эффективный способ нахождения числа α, относящегося к показателю $\gamma = d_1 d_2 d_3$ (произведение трех простых делителей числа $p-1$).
4. Пусть известно разложение числа $p-1$, где p есть простое число. Как проверить то, что число α является первообразным корнем?
5. Для числа α, относящегося к простому показателю γ по mod n, имеется γ различных чисел $\{\alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^\gamma = 1\}$. Доказать, что все эти числа относятся к показателю γ.
6. Извлечь квадратный корень из чисел 537, 439, 246 и 238 по модулю 897.
7. Извлечь корень четвертой степени из чисел 23, 26, 51 и 65 по модулю 79. (*Указание*: использовать тот факт, что 23, 26, 51 и 65 являются квадратичными вычетами по модулю 79, причем $79 \equiv 3 \bmod 4$.)
8. Извлечь кубический корень из чисел 5, 21, 31, 32, 36, 37 и 39 по модулю 41.
9. Извлечь кубический корень из чисел 24, 29 и 34 по модулю 41.
10. Вывести формулу для вычисления кубических корней по модулю простого числа p, такого что $p \equiv 5 \bmod 6$. (*Указание*: воспользоваться формулой $a^{(p-1)/2} = 1 \bmod p$ для квадратичных вычетов и $a^{(p-1)/2} = -1 \bmod p$ для квадратичных невычетов.)
11. Определить, какие из чисел 1034, 1234, 1959, 2477, 3074 и 4179 являются квадратичными вычетами по RSA-модулю $n = 5963$.
12. Дано значение модуля. Как найти число, относящееся к составному показателю по этому модулю?
13. Для числа α, относящегося к показателю γ по mod n, имеется γ различных чисел $\{\alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^\gamma = 1\}$, для которых при любом $i < \gamma$

имеем $(\alpha^i)^\gamma \equiv (\alpha^\gamma)^i \equiv 1 \bmod p$. Относятся ли все эти числа к показателю γ? Содержатся ли среди этих чисел все числа, относящиеся к показателю γ?

14. Вычислить число первообразных корней по модулям 67; 47; 97; 131.
15. Определить число чисел, относящихся к показателю 2 по модулям 67; 47; 97; 131.
16. Определить число чисел, относящихся к показателю 11 по модулям 23; 67; 133.
17. Оценить вероятность того, что случайно выбранное число $a<p$ окажется первообразным корнем по простому модулю p.
18. Указать все показатели по модулям 137, 196, 386 и 625.
19. Указать все показатели по модулю $n = 3 \cdot 5 \cdot 129 \cdot 257$.
20. Оценить вероятность того, что случайно выбранное число $a<n$ будет относиться к показателю $\varphi(n)$ для значений $n = 257; 129 \cdot 257^2$.
21. Доказать, что существуют числа, относящиеся к любому простому делителю $\varphi(n)$ как к показателю по модулю n, где n — произвольное составное число.
22. Оценить вероятность случайного выбора числа $a<p$, относящегося к простому делителю $\delta \mid p-1$ как к показателю.
23. Оценить вероятность того, что для случайного числа $\beta<p$ будет выполняться сравнение $\beta^{(p-1)/\delta} \equiv 1 \bmod p$, где p — простое число и $\delta \mid p-1$.
24. Вычислить значение обобщенной функции Эйлера для чисел 72; 100; 110; 1210.
25. Вычислить значения обобщенной функции Эйлера и функции Эйлера для чисел 504; 512; 825.
26. Решить систему сравнений $x = 14 \bmod 29$, $x = 17 \bmod 31$.
27. Решить систему сравнений $x = 21 \bmod 63$, $x = 13 \bmod 37$.
28. Используя китайскую теорему об остатках, решить систему сравнений $x \equiv 23 \bmod 67$, $x \equiv 30 \bmod 49$, $x \equiv 13 \bmod 17$.
29. Задан многочлен третьей степени над полем GF(7), проходящий через точки (1,1); (2,3); (4,7) и (9,9). Вычислить коэффициенты и свободный член этого многочлена.
30. Вычислить функцию Эйлера от чисел N_1=567 и N_2=1024.
31. Вычислить функцию Эйлера от чисел N_1=1280 и N_2=512.
32. Вычислить обобщенную функцию Эйлера от чисел N_1=213 и N_2=2025.

33. Вычислить $a/b \bmod p$ для значений 1) $a = 79$, $b = 11$, $p = 97$ и 2) $a = 5$, $b = 17$, $p = 131$.
34. Вычислить $a^b \bmod p$ для значений 1) $a = 13$, $b = 17$, $p = 131$ и 2) $a = 5$, $b = 17$, $p = 131$.
35. Вычислить $a^b \bmod p$ для значений 1) $a = 7$, $b = 20$, $p = 109$ и 2) $a = 38$, $b = 7$, $p = 47$.
36. Вычислить $a/b \bmod p$ для значений 1) $a = 79$, $b = 11$, $p = 97$ и 2) $a = 5$, $b = 17$, $p = 131$.
37. Вычислить $a/b \bmod p$ для значений 1) $a = 6$, $b = 18$, $p = 123$ и 2) $a = 8$, $b = 24$, $p = 107$.
38. Найти линейное представление с целыми коэффициентами для наибольшего общего делителя чисел $a = 13$; $b = 87$.
39. Найти линейное представление с целыми коэффициентами для наибольшего общего делителя чисел $a = 11$; $b = 97$.
40. Показать, что для значения b, являющегося взаимно простым с n, выполняется соотношение $a/b = ab^{\varphi(n)-1} \bmod n$, где $\varphi(n)$ — функция Эйлера.
41. Вывести формулу $\varphi(p^\alpha) = p^{\alpha-1}(p-1)$ для значения функции Эйлера от числа p^α, где p — простое число.
42. Найти все целочисленные решения уравнения $34x + 289y = 187$.
43. Решить систему сравнений $x^2 \equiv 22 \bmod 29$, $x \equiv 7 \bmod 13$.
44. Решить систему сравнений $x^3 \equiv 18 \bmod 23$, $x \equiv 7 \bmod 11$.
45. Решить систему сравнений $x^2 \equiv 11 \bmod 19$, $x^3 \equiv 10 \bmod 23$.
46. Решить систему сравнений $x^2 \equiv 22 \bmod 29$, $x^2 \equiv 4 \bmod 13$.

8.2.2. Схемы цифровой подписи

1. Вычислить закрытый ключ d криптосистемы RSA, соответствующий открытому ключу $e = 97$, для значений модуля $n_1 = 299$ и $n_2 = 527$.
2. Вычислить открытый ключ e криптосистемы RSA, соответствующий закрытому ключу $d = 91$, для значений модуля $n_1 = 187$ и $n_2 = 319$.
3. Вычислить открытый ключ e криптосистемы RSA, соответствующий закрытому ключу $d = 71$, для значений модуля $n_1 = 229$ и $n_2 = 451$.

4. Сформировать соответствующие друг другу открытый ключ e и закрытый ключ d криптосистемы RSA при значениях модуля $n_1 = 377$ и $n_2 = 451$.
5. Показать, что для модуля n системы RSA выполняется условие $\varphi(n^2) = n\varphi(n)$.
6. В криптосистеме RSA с модулем $n = 5963$ и закрытым $d = 37$ и открытым $e = 157$ ключами пятикратное шифрование сообщения M дает криптограмму, совпадающую с исходным сообщением. Объяснить почему.
7. Предложить вариант слепой подписи с использованием системы ЭЦП Эль-Гамаля.
8. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $m = \alpha^{rs} yr \pmod p$.
9. В схеме ЭЦП с уравнением проверки подписи $\alpha = S^H \bmod n$, где $n = 1541$, выбрать секретный ключ γ и вычислить открытый ключ α.
10. В схеме ЭЦП с уравнением проверки подписи $\alpha = S^H \bmod n$, где $n = 2201$, выбрать секретный ключ γ и вычислить открытый ключ α.
11. В схеме ЭЦП с уравнением проверки подписи $\alpha = S^H \bmod n$, где $n = 4757$, выбрать секретный ключ γ и вычислить открытый ключ α.
12. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $m = \alpha^{f(r)s} yr \pmod p$.
13. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{hs} = y^r r^s \pmod p$.
14. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{hr} = y^s r \pmod p$.
15. Показать способ подделки подписи в системе ЭЦП с уравнением проверки подписи $\alpha^{rs} = y^h r \pmod p$.
16. 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.
17. 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.
18. Дано уравнение проверки подписи $y_1^{h/s} = y_2^s \bmod p$, где $y_1 = \alpha^{x_1} \bmod p$ является открытым ключом, (s, y_2) — подпись к сообщению, хэш-функция которого равна h. Составить уравнение вычисления подписи. Является ли стойкой такая ЭЦП?

19. Предложить атаку на ЭЦП, описанную в предыдущей задаче. Использовать представление $y_2 = y_1^z \bmod p$.
20. 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 $r = \alpha^k \bmod p$.
21. 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 $r = \alpha^k \bmod p$.
22. 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 $r = \alpha^k \bmod p$.
23. Предложить систему ЭЦП с уравнением проверки подписи $S^2 \equiv (H||r) \bmod n$, где (S, r) — подпись, H — хэш-функция от подписываемого документа.
24. Предложить систему ЭЦП с уравнением проверки подписи $S^3 \equiv H \bmod n$, где S — подпись, H — хэш-функция от подписываемого документа.
25. Нарушитель, получив подпись S' к подготовленному им значению хэш-функции H', сформировал «несанкционированную» подпись к значению хэш-функции H. Каким образом он сделал это, если уравнением проверки подписи является $\alpha = S^H \bmod n$, где (n, α) — открытый ключ, n — RSA-модуль?
26. Рассмотреть систему ЭЦП с уравнением проверки подписи $\alpha = S^{H^2+H} \bmod n$, где (n, α) — открытый ключ, n — RSA-модуль. С какой целью в качестве показателя в этом уравнении используется сумма значения хэш-функции и его квадрата? Как изменится время генерации и время проверки подписи?
27. Рассмотреть систему ЭЦП с уравнением проверки подписи $\alpha^{H \operatorname{div} 2^{64}} + (H \bmod 2^{64}) \cdot 2^{64} = S^H \bmod n$, где (n, α) — открытый ключ, n — RSA-модуль, H — 128-битовое значение хэш-функции. С какой целью осуществлено усложнение исходного уравнения проверки подписи $\alpha = S^H \bmod n$? Как изменится время генерации и время проверки подписи?
28. Предложить схему слепой подписи с уравнением проверки подписи $S^3 = H \bmod n$, где S — подпись, H — хэш-функция от подписываемого документа, n — RSA-модуль.

29. Записать процедуру генерации подписи (S, R) в схеме ЭЦП с уравнением проверки подписи $S^2 = H||R \bmod n$, где n — RSA-модуль. С какой целью используется параметр R? Из каких соображений может выбираться размер этого числа?
30. Является ли стойкой схема ЭЦП с уравнением проверки подписи $S^2 + \alpha^R = H \bmod n$, где n — RSA-модуль и (R,S) — подпись?
31. Является ли стойкой схема ЭЦП с уравнением проверки подписи $S^3 + S^2R = H \bmod n$, где n — RSA-модуль и (R,S) — подпись?
32. Является ли стойкой схема ЭЦП с уравнением проверки подписи $S^2 + SR^3 = H \bmod n$, где n — RSA-модуль и (R,S) — подпись? С какой целью используется параметр R? Какие могут быть даны рекомендации по выбору модуля?
33. Тест на простоту числа p состоит в проверке выполнимости соотношения $a^{\varphi(n)} = 1 \bmod n$, где $n = pq$ и q — заведомо простое число. Показать, что этот тест является эквивалентным тесту Ферма по отношению к числам Кармайкла.
34. Составить уравнение генерации подписи в схеме ЭЦП с уравнением проверки подписи $(S + H)^H = \alpha \bmod n$, где n — RSA-модуль. Обосновать переход от исходного уравнения проверки подписи $S^H = \alpha \bmod n$ к указанному выше.
35. Составить уравнение генерации подписи в схеме ЭЦП с уравнением проверки подписи $(SH)^H = \alpha \bmod n$, где n — RSA-модуль. Обосновать переход от исходного уравнения проверки подписи $S^H = \alpha \bmod n$ к указанному выше. Сравнить со схемой ЭЦП из задачи 34. Какая из них предпочтительна?

8.2.3. Хэш-функции

1. 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.
2. Показать слабость итеративной хэш-функции, основанной на раундовой функции $h_i = h_{i-1} * a^{M_i} \bmod p$, где M_i — блоки данных, h_i — раундовое значение хэш-функции, a и p — известные параметры, * — операция сложения или умножения по модулю p.
3. Дана хэш-функция, описываемая раундовым преобразованием $h_i = M_i (M_i \cdot h_{i-1})^a \bmod p$, где M_i — блоки данных, h_i — раундовое значение хэш-

функции, a и p — известные параметры. Показать, что она не удовлетворяет требованию устойчивости к коллизиям в слабом смысле.

4. Дана хэш-функция, описываемая раундовым преобразованием $h_i = h_{i-1}^{M_i} \bmod p$, где M_i — блоки данных, h_i — раундовое значение хэш-функции, p — простой модуль. Показать, что она не удовлетворяет требованию устойчивости к коллизиям в сильном смысле.
5. 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.
6. 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.
7. Предложить эффективную атаку на хэш-функцию $h_i = ((a^{h_{i-1}})^{M_i} \bmod p) \cdot \cdot \bmod q$, где M_i — блоки данных; h_i — раундовое значение хэш-функции; a, q и p — известные параметры.
8. Дана хэш-функция, описываемая раундовым преобразованием $h_i = M_i \cdot \cdot a^{h_{i-1}M_i} \bmod p$, где M_i — блоки данных, h_i — раундовое значение хэш-функции, a и p — известные параметры. Показать, что она не удовлетворяет требованию устойчивости к коллизиям в слабом смысле.
9. Дана хэш-функция с раундовым преобразованием $h_i = h_{i-1} \cdot a^{h_{i-1}M_i} \bmod p$, где M_i — блоки данных, h_i — раундовое значение хэш-функции, a и p — известные параметры. Показать, что она не удовлетворяет требованию устойчивости к коллизиям в слабом смысле.
10. Предложить способ встраивания потайного люка в хэш-функцию с раундовым преобразованием $h_i = ((a^{h_{i-1}} b^{M_i}) \bmod p) \bmod q$, где M_i — блоки данных; h_i — раундовое значение хэш-функции; a, q и p — известные параметры. (*Указание*: выбрать $b = a^x \bmod p$, где x держится в секрете; это ключ к потайному люку).
11. Может ли секрет в хэш-функции из задачи 10 рассматриваться как многоразовый, т. е. для выполнения многократного модифицирования документов с сохранением значения хэш-функции?
12. Является ли однонаправленной хэш-функция $h_i = (h_{i-1} + a^{M_i}) \bmod p$, где M_i — блоки данных; h_i — раундовое значение хэш-функции; a и p — известные параметры? Обладает ли она коллизионной стойкостью в слабом смысле?

Заключение

В данной книге были рассмотрены двухключевые криптосистемы, основанные на результатах теории чисел. Основной акцент был сделан на идеях, используемых в двухключевых алгоритмах и протоколах на их основе. Симметричная криптография рассмотрена весьма кратко в первой главе с целью показать общую картину проблематики и методов современной криптографии, а также место, которое занимает в ней криптография с открытым ключом. Читателям, интересующимся детальным изложением вопросов синтеза симметричных шифров на основе управляемых операций, можно рекомендовать книгу [12]. Вопросы разработки программных и гибких шифров подробно рассмотрены в книгах [8], [9], зарубежные алгоритмы — в [15], [30] и [31], прикладные аспекты — в [8] и [15]. Широкий круг вопросов по симметричной криптографии также представлен в книгах [16], [17] и [37].

Математические результаты в книге приведены только в том минимуме, который необходим для понимания механизмов реализации идеологии двухключевых шифров и выбора параметров конкретных криптосистем. Благодаря этому подходу, изложенный материал представляется достаточно доступным введением в двухключевые шифры не только для студентов технических вузов, но и для учащихся старших классов, интересующихся математикой и информатикой. Более глубокое и всестороннее изучение симметричной и асимметричной криптографии потребует дополнительного изучения теории вероятности, аппарата булевых функций, теории сложности и ряда других разделов дискретной математики. Желающие более широко ознакомиться с математическим аппаратом, используемым в криптографии, могут обратиться к книгам [13], [14], [17] и [31].

Авторы надеются, что ознакомление с настоящей книгой дало читателю общее и достаточно полное предварительное ознакомление с идеями и проблематикой современной двухключевой криптографии и что оно явится хорошей подготовкой для удовлетворения дальнейших интересов в ознакомлении с новыми разделами этой дисциплины и пригодится в практической деятельности.

Литература

1. Айерлэнд К., Роузен М. Классическое введение в современную теорию чисел.: Пер. с англ. — М.: Мир. — 1987. — 416 с.
2. Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. — М.: Гелиос АРВ. — 2002. — 480 с.
3. Бухштаб А. А. Теория чисел. — М.: Просвещение. — 1966. — 384 с.
4. Виноградов И. М. Основы теории чисел. — М.: Наука. — 1972. — 167 с.
5. 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.
6. Иванов М. А. Криптографические методы защиты информации в компьютерных системах и сетях. — М.: Кудиц-Образ. — 2001. — 368 с.
7. Коутинхо С. Введение в теорию чисел. Алгоритм RSA. — М.: Постмаркет. — 2001. — 323 с.
8. Молдовян А. А., Молдовян Н. А., Советов Б. Я. Криптография. — СПб.: Лань. — 2000. — 218 с.
9. Молдовян А. А., Молдовян Н. А., Гуц Н. Д., Изотов Б. В. Криптография: скоростные шифры. — СПб.: БХВ-Петербург. — 2002. — 495 с.
10. Молдовян Д. Н. Схемы цифровой подписи на основе сложности факторизации модуля // Вопросы защиты информации. — 2004. — № 4 (67). — С. 6–11.
11. Молдовян Н. А. Скоростные блочные шифры. — СПб.: СПбГУ. — 1998. — 230 с.
12. Молдовян Н. А., Молдовян А. А., Еремеев М. А. Криптография: от примитивов к синтезу алгоритмов. — СПб.: БХВ-Петербург. — 2004. — 446 с.
13. Ростовцев А. Г. Алгебраические основы криптографии. — СПб.: Мир и Семья. — 2000. — 353 с.
14. Ростовцев А. Г., Маховенко Е. Б. Введение в криптографию с открытым ключом. — СПб.: Мир и Семья. — 2001. — 336 с.

15. Столлингс В. Криптография и защита сетей: принципы и практика. — 2-е изд.: Пер. с англ. — Изд. дом «Вильямс». — 2001. — 672 с.

16. Харин Ю. С., Берник В. И., Матвеев Г. В. Математические основы криптологии. — Минск.: БГУ. — 1999. — 319 с.

17. Харин Ю.С., Берник В.И., Матвеев Г.В., Агиевич С.В. Математические и компьютерные основы криптологии. — Минск: Новое знание. — 2003. — 381 с.

18. Adleman L. M., Estes D., McCurley K. S. Solving bivariate quadratic congruences in random polynomial time // Mathematics of Computation. — 1987. — V. 48. — N. 177. — P. 17–28.

19. Chaum D. Blind Signature Systems. U.S. Patent # 4,759,063. — 19 July 1988.

20. Chaum D. Blind Signatures for Untraceable Payments // Advances in Cryptology: Proc. of CRYPTO'82. Plenum Press. —1983. — P. 199–203.

21. Chaum D. Security Without Identification: Transaction Systems to Make Big Brother Obsolete // Communication of the ACM. — 1985.— V. 28. — N. 10. — P. 1030–1044.

22. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Transactions on Information Theory. —1976. —V. IT–22. — P. 644–654.

23. ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE Transactions on Information Theory. — 1985. —V. IT–31. — N. 4. — P. 469–472.

24. Estes D., Adleman L. M., Konpella K., McCurleyK. S., Miller G. L. Breaking the Ong-Schnorr-Shamir signature schemes for quadratic number fields // Advances in Cryptology — CRYPTO'85 Proceedings. Springer Verlag. — 1986. — P. 3–13.

25. Fiat A., Shamir A. How to prove yourself: Practical solutions to identification and signature problems // Advances in Cryptology — CRYPTO'86. Springer-Verlag. —1987. — V. 263. — P. 186–194.

26. Gordon J. Strong primes are easy to find // Advances in Cryptology — EUROCRYPT'84. Springer-Verlag. —1985. — V. 209. — P. 216–223.

27. Kam J.B., Davida G.I. Structured Design of Substitution-Permutation Encryption Networks. // IEEE Transactions on Computers. —1979. — V. 28. — N. 10. — P. 747–753.

28. 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 Net-

work Security, Lecture Notes in Computer Science. Springer-Verlag. — 2003. — V. 2776. — P. 298–307.

29. Lieberherr K. Uniform Complexity and Digital Signatures // Theoretical Computer Science. — Oct. 1981. — V. 16. — N. 1. — P. 99–110.
30. Menezes A. J., Vanstone S. A. Handbook of Applied Cryptography // CRC Press. — 1996. — 780 p.
31. Pieprzyk J., Hardjono Th., Seberry J. Fundumentals of Computer Security // Springer-Verlag. — 2003. — 677 p.
32. Pollard J. M., Schnorr C. P. An efficient solution of the congruence $x^2+ky^2=m \pmod{n}$ // IEEE Transactions on Information Theory. — 1987. —V. IT–33. — N. 5. — P. 702–709.
33. Portz M. A. Generalized Description of DES-based and Benes-based Permutation generators // Advances in Cryptology — AUSCRYPT'92 // Lecture Notes in Computer Science. Springer-Verlag. —1992. — V. 718. — P. 397–409.
34. Rabin M. O. Digitalized signatures and public key functions as intractable as factorization // Technical report MIT/LCS/TR-212, MIT Laboratory for Computer Science. — 1979.
35. Rivest R. L. The RC5™ Encryption Algorithm // Fast Software Encryption, Second International Workshop // Lecture Notes in Computer Science. Springer-Verlag. —1995. — V. 1008. — P. 86–96.
36. 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).
37. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code (Second Edition) // New York: John Wiley & Sons. — 1996. — 758 p.
38. Shannon C. E. Communication Theory of Secrecy Systems // Bell Systems Technical Journal. —1949. — V. 28. — P. 656–715.
39. Schnorr C.P. Efficient signature generation by smart cards // J. Cryptology. —1991. — V. 4. — P. 161–174.
40. Schnorr C. P. Efficient identification and signatures for smart cards // Advances in Cryptology — CRYPTO'89. Springer-Verlag. —1990. — V. 435. — P. 239–252.
41. Sklavos N., Moldovyan A. A., Koufopavlou O. Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation //

Workshop MMM-ANCS'2003 Proc. Springer-Verlag. — 2003. — V.2776. — P. 343–354.

42. Sklavos N., Moldovyan N. A., Koufopavlou O. High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers // Mobile Networks and Applications. — 2005. — V. 10. — P. 237–249.
43. Sklavos N., Moldovyan N.A., Koufopavlou O. Pure DDP-Base Cipher: Architecture Analysis, Hardware Implementation cost and Performance up to 6.5 gbps // International Arab Journal of Information Technology. — 2005. — V. 2. — N. 1. — P. 24–32.
44. Shimada M. Another Practical Public-key Cryptosystem // Electronics Letters. —1992. —V. 28. — N.23. — P. 2146–2147.
45. 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». Springer-Verlag. — 1998. — V. 1372. — P. 270–283.
46. Williams H. C. A Modification of the RSA Publik-Key Encryption Procedure // IEEE Transactions on Information Theory. — 1980. — V. IT–26. — N. 6. — P. 726–729.