А. И. Музыкантский, В. В. Фурин

Лекции по криптографии

Издание второе, стереотипное

Москва
Издательство МЦНМО
2013

УДК 511+519.719.2
ББК 32.81в6
М89

Музыкантский А. И., Фурин В. В.
М89 Лекции по криптографии. — М.: МЦНМО, 2013. — 2-е изд., стереотип. — 68 с.

Брошюра издана по материалам лекций по криптографии, прочитанных на факультете мировой политики МГУ им. М. В. Ломоносова. Основное внимание уделяется прикладным задачам, решаемым с помощью математических методов криптографии. Доступно рассказывается о том, что такое шифрование, криптографические протоколы, о роли криптографии в массовых информационных коммуникациях.

Первое издание было опубликовано в 2011 году.

ISBN 978-5-4439-0086-5

Оглавление

1. Основные понятия и математическая формализация 4
1.1. Основные понятия криптографии 4
1.2. Из истории криптографии 5
1.2.1. От шифра Цезаря до машины «Enigma» 5
1.2.2. Шифровальная машина «Enigma». Принцип действия и битва за шифры 7
1.3. Математическая формализация 11
1.3.1. Односторонние функции и функции с секретом 11
1.3.2. Алгоритм шифрования RSA 17
1.3.3. Открытый и закрытый ключи 22
1.3.4. Криптография и «трудные» математические задачи 23
2. Криптографические протоколы 26
2.1. Что такое криптографические протоколы 26
2.2. Протокол подбрасывания монеты по телефону 27
2.3. Протокол аутентификации 29
2.4. Доказательство с нулевым разглашением 31
2.5. Электронная подпись 31
2.6. Электронные торги 33
2.7. Протокол электронного голосования 36
2.8. Задачи, решаемые только с использованием криптографических протоколов. Закрытый информационный обмен между двумя партнерами 42
2.9. Криптографические протоколы и «честное слово» 43
3. Криптография и массовые информационные коммуникации . . . 46
3.1. Какие задачи хотелось бы уметь решать. Массовые информационные коммуникации 46
3.2. Инфраструктура открытых ключей и удостоверяющие центры . . . 47
3.2.1. Предпосылки создания 47
3.2.2. Обеспечивающие алгоритмы 48
3.2.3. Обмен между клиентами одного удостоверяющего центра . . 50
3.2.4. Система взаимодействующих удостоверяющих центров 54
3.2.5. Иерархическая система удостоверяющих центров 55
3.2.6. Общий случай 63
3.3. Промежуточные итоги 64
Литература 66

1. Основные понятия и математическая формализация

1.1. Основные понятия криптографии

В этой лекции мы познакомимся с одной из замечательных наук — криптографией. С незапамятных времен человечество обращалось к ней, когда возникала необходимость сохранить в тайне информацию. Можно постараться скрыть сам факт передачи информации, например, применить тайнопись при обычной переписке или проложить секретный канал между отправителем и получателем информации. Но сделать канал передачи информации полностью скрытым от недоброжелателей оказалось практически невозможно. Использовать только способ сокрытия передачи информации для защиты, например, важной государственной информации было ненадежно или очень дорого. Возникла задача использовать общедоступный канал связи, но при этом передавать нужную информацию в таком шифрованном виде, чтобы прочитать ее мог только адресат.

Для разработки методов преобразования информации (или шифрования) с целью ее защиты от незаконных пользователей и возникла наука ***криптография***.

Криптография на современном этапе — это область научных, прикладных, инженерно-технических исследований, основанная на фундаментальных понятиях математики, физики, теории информации и сложности вычислений. Криптография необходима в основном для сохранения государственной тайны, военной тайны, а также коммерческой, юридической, врачебной и т. д. При этом криптография, исторически возникшая именно как наука о методах шифровки и дешифровки, в дальнейшем, особенно с появлением компьютеров, включила в себя и многие другие задачи, возникающие в процессе организации обмена информацией.

Обеспечение конфиденциальности информации, уверенность в отсутствии изменений в передаваемой информации, установление подлинности источника передаваемых сообщений, невозможность отказа от факта совершения определенных действий — вот пример задач, решаемых современной криптографией.

В дальнейшем мы будем применять понятия, аналогичные военной терминологии. При отправлении сообщения обязательно есть *отправитель* и *адресат*, или *получатель*, кому направлено сообщение. Нежелательный для отправителя получатель сообщения будет именоваться *противником*, который пытается создать *угрозу сообщению*. *Атакой на шифр* будем именовать попытки противника разгадать наш шифр (иногда употребляется также термин *взлом шифра*). Возможность нашего шифра противостоять угрозам будем именовать *стойкостью шифра*.

В дальнейшем мы словом *дешифрирование* будем обозначать попытку взлома шифра незаконным получателем (противником); словом *расшифрование* мы обозначаем восстановление исходного текста из шифротекста его законным получателем.

Очень важно правильно понимать проблему соотношения цены информации, затрат на ее защиту и затрат на ее добывание. Если затраты на шифрование сопоставимы со стоимостью самой информации, то стоит задуматься об изменении способа шифрования. Аналогичны действия противника — если затраты на дешифровку выше стоимости информации, то стоит подумать о другом способе добычи информации. Отправитель сообщения должен понимать, что если материальные и людские затраты противника на дешифровку сообщения будут намного выше цены информации, содержащейся в сообщении, то он вряд ли возьмется за задачи дешифровки конкретных сообщений отправителя.

В мире разрабатываются специальные стандарты, которые позволяют отправителям сообщений, применяющих их в информационном обмене, с большой долей уверенности считать, что в течение определенного срока самые современные технические средства противников не смогут дешифровать информацию или, например, подделать зашифрованную электронную подпись отправителя.

Более подробно с основными понятиями криптографии можно ознакомиться в работе [2].

1.2. Из истории криптографии

1.2.1. От шифра Цезаря до машины «Enigma»

Одним из первых наиболее известных криптографических методов преобразования информации был шифр Сцитала, использующий метод перестановки букв. Брался жезл определенного диаметра, на него наматывалась лента и сообщение записывалось вдоль жезла

в несколько «строк». Далее лента разматывалась, и сообщение получалось зашифрованным. Чтобы прочитать сообщение, необходимо было иметь жезл такого же диаметра. Специальный метод дешифровки таких сообщений разработал Аристотель. Он использовал конус, для чего наматывал и двигал по нему ленту с сообщением. Когда появлялась возможность прочитать сообщение, на конусе делалась засечка, это и позволяло определить диаметр жезла, на котором сообщение было зашифровано.

Также интересным представляется шифр Цезаря, использующий метод замены. Отправитель и адресат договариваются о величине сдвига — определенном числе (у Цезаря обычно 3), после этого записывают алфавит по кругу (*а* после *я*, если алфавит русский, или *a* после *z*, если алфавит латинский) и заменяют букву в открытом тексте на букву, получившуюся в результате сдвига по кругу. Шифр можно было вскрыть в результате кропотливой работы с разными величинами сдвигов. Максимально мы можем использовать лишь 26 (в случае латинского алфавита) нетривиальных ключей.

В рассмотренных выше двух примерах появляется очень важное понятие криптографии — *ключ* (в шифре Сцитала это диаметр жезла, в шифре Цезаря — величина сдвига). Чтобы разгадать шифрованное сообщение, требуется не только получить само сообщение, знать метод шифрования, но и разгадать ключ.

Более сложный вариант шифрования был описан в рассказе «Пляшущие человечки» Артура Конан-Дойля. Для дешифровки Шерлоку Холмсу пришлось применять дополнительные методы (помимо перебора), такие, как частота появления тех или иных букв английского алфавита. Будучи знакомым со многими работами по дешифровке посланий, Эдгар По в рассказе «Золотой жук» писал: «...Едва ли разуму человека дано загадать такую загадку, которую разум его собрата, направленный должным образом, не смог бы разгадать».

Одним из первых, кто создал профессиональную службу по шифровке и дешифровке сообщений, был кардинал Ришелье. В 1628 году в почтамте Парижа был открыт «Черный кабинет». Его возглавил Антуан Россиньоль, который являлся автором шифра, используемого в будущем вплоть до наполеоновских войн. Ему принадлежит авторство доктрины: «Стойкость военного шифра должна обеспечить секретность за время, необходимое для выполнения приказа. Стойкость дипломатического шифра должна обеспечивать секретность в течение десятилетий».

С появлением электромеханических устройств появились новые возможности шифрования. Двадцатые и тридцатые годы XX века ста-

ли периодом расцвета конструирования различных шифровальных машин. В 1917 году Гильберт Вернам предложил очень эффективный способ шифрования сообщений. Шифруемое сообщение переводилось в двоичную кодировку, складывалось со значениями специальной телетайпной ленты-ключа и отправлялось по телеграфу адресату. Расшифровать сообщение, имея ленту-ключ, не представляло труда. В дальнейшем известным американским математиком Клодом Шенноном было доказано, что схема Вернама является абсолютно стойкой системой шифрования, при условии если длина ключа равна длине сообщения, ключ вырабатывается случайно и используется однократно.

Основной проблемой шифровальщиков стала выработка ключей и организация схемы обмена ими между отправителем и получателем сообщений. Так появились электромеханические шифраторы, состоящие из коммутационных дисков и механизма изменения их угловых положений. Наибольшую известность среди них приобрела немецкая шифровальная машина «Enigma», которая использовалась немецким командованием для нужд шифрования своих военных сообщений во время Второй мировой войны. Концепция работы «Enigma» была разработана немецким инженером Артуром Шербиусом, но принципы работы, устройство, да и сам факт существования этой шифровальной машины представляли собой высший государственный секрет Третьего Рейха.

Учитывая, что «Enigma» представляла собой одно из наивысших достижений «докомпьютерного» этапа развития шифровальной техники и что дешифровка ее кодов представляет собой одно из самых значительных достижений разведки союзников, которое оказало большое влияние на весь ход военных операций Второй мировой войны, имеет смысл сказать по этому поводу несколько слов.

1.2.2. Шифровальная машина «Enigma». Принцип действия и битва за шифры

«Enigma» (в русской транскрипции «Энигма») в переводе с древнегреческого означает «безымянный», «неназванный», «загадочный». Работы по применению шифровальной машины с подобным названием начались в Германии в глубокой тайне в 1928 году и активизировались с приходом к власти Гитлера. Работами руководил непосредственно германский Генеральный штаб. К началу Второй мировой войны работы по созданию военного варианта «Enigma» были закончены, машина прошла испытания и была принята на вооружение.

Рис. 1. Внешний вид шифровальной машины «Энигма»

«Enigma» относилась к классу электромеханических шифровальных машин. Ее конструкция была основана на системе из трех вращающихся барабанов, осуществлявших замену 26 букв латинского алфавита. Каждый барабан имел 26 входных контактов на одной стороне и столько же выходных контактов — на другой. Внутри каждого барабана проходили провода, связывавшие входные и выходные контакты между собой. Выходные контакты первого барабана соединялись с входными контактами второго. Когда оператор нажимал на какую-либо букву на клавиатуре машины, электрический ток подавался на входной контакт первого барабана, соответствующий этой букве. Ток проходил через первый барабан и поступал на выходной контакт, соответствующий какой-либо другой букве. Затем ток проходил последовательно через второй и третий барабаны и подавался на неподвижный рефлектор (от лат. reflecto — обращаю назад, отражаю). В конструкции рефлектора 26 контактов разбивались на пары, контакты внутри каждой пары были соединены между собой. Таким образом, рефлектор заменял букву на парную ей.

Ток, прошедший через рефлектор, подавался назад, на систему барабанов. Он вновь проходил через три барабана, но в обратном порядке. В конце концов на световом табло машины загоралась одна из 26 лампочек, соответствовавшая зашифрованной букве.

Самым важным свойством машины «Enigma» являлось вращение барабанов. Первый барабан после каждого преобразования буквы поворачивался на одну позицию. Второй барабан поворачивался на одну позицию после того, как первый совершал полный оборот, т. е. после преобразования 26 букв. Наконец, третий барабан поворачивался на одну позицию после того, как второй совершал полный оборот, т. е. после шифрования 676 букв.

Благодаря рефлектору «Enigma» на каждом шаге осуществляла перестановку букв внутри пар, и если, к примеру, буква N заменялась на S, то при том же положении роторов буква S менялась на N (ток шел по тем же проводам, но в другую сторону). Этим объяснялась особенность машины: для расшифровки сообщения достаточно было вновь пропустить его через машину, восстановив предварительно изначальное положение барабанов. Таким образом, начальное положение барабанов играло роль ключа шифрования. Это начальное положение устанавливалось в соответствии с текущей датой. Каждый оператор имел специальную книгу, задававшую положение барабанов для каждого дня. В целом получилась компактная, быстродействующая шифровальная машина, достаточно устойчивая к попыткам взлома применяемого шифра.

Очевидная слабость данной системы шифрования заключалась в том, что противнику достаточно было завладеть специальной книгой, задающей ключи шифрования, и самой машиной, чтобы дешифровать многие сообщения [5].

Немцам, несмотря на колоссальные усилия, не удалось сохранить в тайне работу над «Энигмой». Уже в 1932 году в специально созданном «Шифровальном бюро» в Варшаве начались работы над раскрытием тайны «Энигмы». Возглавлял группу молодой польский математик Мариан Ршевский, выпускник математического факультета университета в Познани. Группа имела в своем распоряжении устаревшую коммерческую шифровальную машину, купленную в Германии. Конечно, эта модель была очень далека от современных для той поры немецких военных шифровальных машин и принесла мало пользы. Поэтому главным моментом в работе ученых для решения задачи «Энигмы» было применение математики (теории групп и теории перестановок). Для раскрытия шифров «Энигмы» польские математики использовали перехваченные шифрованные сообщения и добились значительных успехов. Ими было теоретически воссоздано устройство машины, что позволило позже создать её реальную модель; были разработаны также методы восстановления ключей к шифрам на основе перехваченных сообщений.

Позднее, в 1939 году, перед началом войны все материалы по «Энигме» были поляками переданы во Францию и Англию. Англичане продолжили работы, раскрыв усовершенствования, которые были внесены в конструкцию последних немецких машин и систему кодов, используемую Германией. В этой работе, выполнявшейся большой группой ученых в местечке Блетчли в 70 км от Лондона, участвовал знаменитый математик Алан Тьюринг, широко известный как автор виртуальной «машины Тьюринга». Благодаря, главным образом, усилиям возглавляемой им группы были созданы механические вычислительные устройства, полным перебором отыскивавшие ключи к шифру на много порядков быстрее, чем это можно было сделать вручную. Подобное механическое устройство, но с возможностью его «программирования» с помощью бумажной перфоленты, созданное специально для дешифровки перехваченных сообщений «Энигмы» и названное «Colossus», некоторые исследователи считают *первым в мире по-настоящему программируемым компьютером*.

Математикам из Блетчли часто удавалось находить блестящие и в то же время простые решения, во много раз сокращавшие время вскрытия шифровок «Энигмы». Но их оригинальные идеи, в частности по организации «распределенных вычислений», приходилось воплощать по преимуществу с помощью карандаша и листа бумаги, что значительно затягивало время дешифровки перехваченных сообщений.

Наконец, осенью 1942 года, в результате спецоперации ВМС Англии, на германской подводной лодке U-571, которую немецкое командование считало затонувшей, была захвачена сама шифровальная машина «Enigma» и процесс дешифровки немецких секретных сообщений был поставлен англичанами на поток.

Вся эта работа по взламыванию немецких секретных шифров сохранялась в глубокой тайне, и немцы до самого конца войны даже не подозревали, что все их секретные сообщения становятся известны антигитлеровской коалиции. О том, какое значение придавало английское командование сохранению в секрете факта взлома немецких секретных шифров, говорит тот факт, что У. Черчилль никак не воспользовался знанием о предстоящем налете немецкой авиации на город Ковентри, сообщения о котором были перехвачены и заблаговременно расшифрованы англичанами. В результате город был подвергнут сильнейшей бомбардировке немецкой авиации, однако англичане сохранили в тайне свои возможности по дешифровке немецких секретных сообщений.

Многие историки, изучающие Вторую мировую войну, убеждены, что взлом англичанами секретных шифров значительно ускорил падение фашистской Германии и сохранил тысячи жизней.

История «Энигмы» держалась в глубокой тайне и после окончания войны; она была опубликована только 30 лет спустя, по истечении срока давности военных секретов.

Так закончилась одна из самых замечательных историй «докомпьютерной» криптографии. Вскоре после войны начался новый этап развития этой древнейшей науки. За дело взялись теоретики — математики и вычислители, вооруженные компьютером.

1.3. Математическая формализация

1.3.1. Односторонние функции и функции с секретом

1.3.1.1. Предварительные замечания. После Второй мировой войны с появлением компьютеров и развитием телекоммуникаций в основном стали использоваться алгоритмы шифрования, в которых сам метод не является секретным, сообщение может быть успешно дешифровано, только если известен сам ключ шифрования. В настоящее время разработаны специальные стандарты шифрования в системах, где отправитель и получатель сообщений пользуются одним и тем же ключом шифрования (*симметричные шифросистемы*). В США с 1977 по 1997 год действовал стандарт DES (Data Encryption Standard), с 2002 года действовал стандарт AES (Advanced Encryption Standard). В России применяется стандарт ГОСТ 28147-89.

Но задача, как сообщить друг другу секретный ключ, чтобы о нем не узнали посторонние, стала еще более актуальной. Эта проблема была решена с помощью методов асимметричной криптографии (шифрования с открытым ключом). На практике оба типа алгоритмов часто используются вместе. Алгоритм с открытым ключом используется для того, чтобы передать созданный секретный ключ, который используется отправителем для шифрования, а получателем для расшифрования сообщений. Шифрование с открытым ключом позволило не только решить проблему обмена секретными ключами, но и проблему проверки подлинности документов, проблему удостоверения подлинности подписи и многие другие. О них и пойдет речь в дальнейшем.

Для нематематиков это может показаться удивительным, но несмотря на все многообразие самых хитроумных способов шифрования и дешифрования, которые на протяжении веков были разработа-

ны самыми изощренными умами, в основе математического описания любого из них лежит всего одно математическое понятие — понятие «односторонней функции» или ее более усложненный вариант — «односторонняя функция с секретом». Уже отсюда ясна важность этих понятий в современной математической криптографии.

Однако прежде чем перейти к формальному описанию этих понятий, играющих фундаментальную роль в современной криптографии, необходимо сделать одно замечание. Функции, используемые в криптографии, — это обычные числовые функции, те функции, аргументами которых являются числа и результатом применения которых также являются числа. Поскольку эти функции используются для преобразования текстов, прежде, чем их применить, нужно буквенно-цифровой текст преобразовать в число. Для этого используется простейший стандартный процесс, называемый *перекодировкой*. Он состоит в посимвольной замене букв и других символов, содержащихся в тексте (например, знаков препинания) на соответствующие им цифры.

Например, для текста на английском языке обычно применяется следующая стандартная перекодировка: $a = 01$, $b = 02$, …, $z = 26$, пробел $= 00$, и т. д. Полученные цифры записываются друг за другом и результат рассматривается как одно целое число, эквивалентное исходному тексту. Например, текст «baza» будет рассматриваться как число 02012601.

Всюду в дальнейшем, когда будет говориться, что та или иная арифметическая операция производится над некоторым текстом, имеется в виду, что сначала этот текст стандартным образом перекодируется, а затем соответствующая операция производится над получившимся в результате перекодировки числом.

В результате перекодировки получаются достаточно большие целые числа, содержащие, быть может, несколько сотен десятичных знаков. С этими числами приходится производить различные арифметические операции: перемножать, делить, возводить в степень. И хотя компьютеры не умеют напрямую производить операции с подобными большими числами, разработаны специальные компьютерные программы, созданы даже специализированные языки программирования, которые с успехом справляются с подобными вычислениями.

И наконец, последнее замечание. Очень длинные тексты могут быть разбиты на блоки стандартной длины, а затем процесс перекодирования и последующего шифрования может быть применен к каждому получившемуся блоку текста по отдельности. В некоторых случаях к текстам произвольной длины предварительно может быть

применена специальная операция, «сжимающая» их до требуемой в дальнейших преобразованиях длины. Эта операция называется *хешированием*. Разработаны и широко применяются различные алгоритмы хеширования текстов, но их обсуждение выходит за пределы настоящего курса. Можно только добавить, что в США с 2002 года действует стандарт функции хеширования SHS (secure hash standart), в России применяется стандарт ГОСТ Р 34.11-94.

1.3.1.2. Односторонние функции. С математической точки зрения шифрование, несмотря на все огромное разнообразие применяемых шифров и способов шифрования, можно представить как процесс преобразования исходного, шифруемого текста с помощью некоторой шифрующей функции

$$f : x \to y.$$

Здесь

x — исходный, шифруемый текст (может быть, стоит в последний раз напомнить, что в этой формуле и всюду далее x — это целое число, полученное в результате стандартной перекодировки исходного буквенно-цифрового текста);

y — полученный в результате шифрования зашифрованный текст;

f — некоторая функция, преобразующая исходный текст в зашифрованный (значения функции f, то есть величины y, обычно также являются целыми числами).

В этих же терминах процесс дешифрования сводится к построению обратной (дешифрующей) функции

$$f^{-1} : y \to x.$$

Чтобы зашифровать текст, необходимо применить к исходному тексту функцию f, а чтобы расшифровать, необходимо применить к зашифрованному тексту обратную функцию f^{-1}.

Интуитивно понятно, что для того чтобы процесс шифрования был эффективным, необходимо, чтобы шифрующая функция f была достаточно простой; для того чтобы процесс дешифрования был затруднен, нужно, чтобы, зная шифрующую функцию f, было достаточно сложно найти ей обратную.

Математическим обобщением этих рассуждений является понятие односторонней функции — одно из основных в математической криптографии.

Определение. Функция f называется *односторонней*, если выполнены два условия:

1) для любого x из некоторого множества существует эффективный алгоритм вычисления $y = f(x)$;

2) не существует эффективного алгоритма обращения функции f.

Другими словами, зная функцию f, мы легко можем зашифровать любой текст x, то есть найти $y = f(x)$; но, зная шифр y, мы не можем восстановить исходный текст x. Даже зная, что зашифрованный текст y получен в результате применения известной функции f к некоторому (неизвестному) исходному тексту, мы тем не менее не можем восстановить исходный текст.

Замечание. В математической криптографии сформулированы строгие определения для односторонней функции, которые дают точный математический смысл понятиям типа «существования или несуществования эффективного алгоритма». Для практического же применения достаточно считать, что п. 2 этого определения означает, что если даже алгоритм вычисления обратной функции найдется, его работа будет занимать столько времени, что после завершения его работы знание исходного текста уже не будет иметь никакого практического значения.

Поясним приведенное на очень простом примере. Пусть нужно зашифровать обычные целые числа 1, 2, …, N. Выберем в качестве шифрующей функции обычное возведение в шестую степень:

$$f(x) = x^6. \qquad (1)$$

Например, число 2 шифруется как $2^6 = 64$, число 3 — как $3^6 = 729$ и так далее. Эта функция удовлетворяет условию 1 из определения односторонней функции, так как возведение в шестую степень — это достаточно простая операция (то есть существует эффективный алгоритм шифрования).

Посмотрим теперь, как обстоит дело с выполнением условия 2.

Для дешифрования нужно к предъявленному шифру применить функцию, обратную (1), то есть извлечь корень шестой степени. Конечно, нельзя сказать, что для этой операции не существует эффективных алгоритмов. Но данный пример иллюстрирует примечательный факт: алгоритм применения обратной функции существенно сложнее, чем алгоритм вычисления самой функции. И поэтому, хотя в строгом смысле слова функция $y = x^6$ не является односторонней, при ее применении для дешифрования приходится применить значительно больше усилий, чем для шифрования.

Например, если по каким-либо причинам время и ресурсы на дешифрирование сильно ограничены, в каких-то случаях вполне можно ограничиться такой «слабо односторонней» функцией, как $f = x^6$,

и противник вполне может и не уложиться в отведенное время, вычисляя напрямую, например, корни шестой степени из 46656, или 117649, или 8026. (Первое число в данном случае шифрует 6, второе 7, а третье вообще ничего не шифрует.)

1.3.1.3. Функции с секретом. Односторонние функции, оставаясь важнейшим средством исследований в математическом шифровании, тем не менее не могут быть непосредственно применимы для шифрования. Дело в том, что трудности дешифрования, связанные с отсутствием эффективного алгоритма вычисления обратной функции, будут испытывать не только противники, пытающиеся взломать шифр, но и законный получатель зашифрованного сообщения.

Между тем сама идея шифрования наряду с невозможностью взлома шифра противником должна содержать и способ достаточно простого расшифрования полученного зашифрованного сообщения законным его получателем.

В криптографии подобные требования формализуются в виде определения «функции с секретом» или «односторонней функции с секретом».

Определение. Функция $f(x)$ называется *функцией с секретом*, если она удовлетворяет следующим условиям:

1) для любого x из некоторого множества существует эффективный алгоритм вычисления $y = f(x)$;

2) функция $f(x)$ обладает некоторым «секретным свойством» (обозначим его k), удовлетворяющим следующим условиям:

а) при использовании свойства k можно построить эффективный алгоритм построения обратной функции

$$f_k^{-1}(y) = x;$$

б) если свойство k неизвестно, то не существует эффективного алгоритма построения обратной функции f^{-1}.

Другими словами, если для некоторого пользователя «секретное свойство k» неизвестно (то есть или ему не сообщили об этом свойстве заранее, или он не смог догадаться о нем в процессе изучения функции f), то для такого пользователя $f(x)$ — это односторонняя функция, и, используя ее для шифрования, он не может осуществить процесс дешифрования.

Если же некоторому пользователю становится известно «свойство k», то, используя его, он может построить эффективную обратную функцию, то есть осуществлять процесс расшифрования (или дешифрования, если пользователь незаконный).

Замечание. Так же, как и в случае с односторонней функцией, в математической криптографии формулируются строгие определения таких понятий, как существование эффективного алгоритма.

Из самого определения функции с секретом ясна ее роль в процессе шифрования. Шифровать, то есть осуществлять вычисления по схеме $y = f(x)$, могут все участники процесса обмена информацией. Что же касается дешифрования, то его могут эффективно осуществлять только участники процесса, владеющие секретным «свойством k».

Таким образом, процесс шифрования/расшифрования с использованием функции с секретом осуществляется следующим образом.

- Выбирается некоторая функция с секретом и алгоритм вычисления $y = f(x)$ сообщается всем заинтересованным участникам процесса обмена информацией (этот алгоритм может быть даже опубликован). Это позволяет всем участникам схемы осуществлять шифрование по схеме $y = f(x)$.
- Некоторым участникам («законным пользователям») сообщается о секретном «свойстве k».
- Получив зашифрованное сообщение $y = f(x)$, «законный» участник может осуществить расшифрование, вычислив исходный текст по формуле $x = f_k^{-1}(y)$, поскольку такой алгоритм вычисления строится в силу свойства а).
- «Незаконный» участник схемы, обладая зашифрованным текстом y, не может, тем не менее, вычислить исходный текст x, т. е. провести дешифрование, поскольку ему не известно секретное «свойство k» шифрующей функции f, а без его знания вычисление напрямую $x = f^{-1}(y)$ невозможно практически.

Для иллюстрации сказанного относительно свойств функций с секретом рассмотрим тот же пример шифрующей функции $y = x^6$.

Мы договорились считать, что в данном примере дешифрование, т. е. вычисление обратной функции $x = y^{1/6}$, при данном значении y является сложной задачей; по крайней мере, эта задача значительно сложней, чем задача шифрования, т. е. вычисления $y = x^6$ при заданном значении x.

Чтобы проиллюстрировать понятие «секрета» функции, заметим, что обратную функцию можно переписать в виде

$$x = (y^{1/2})^{1/3}, \tag{2}$$

а это позволяет построить алгоритм вычисления обратной функции, который сводится к последовательному извлечению сначала квадратного корня, а затем кубического, что можно осуществить значительно

более эффективно, чем однократное извлечение корня шестой степени.

Приведенный пример иллюстрирует (конечно, весьма условно) идею применения функций с секретом. Конечно, этот пример плох в том смысле, что распознать приведенный секрет данной схемы шифрования может не только законный ее пользователь. В данном случае секрет, заключающийся в том, что для любого числа a

$$a^{1/6} = (a^{1/2})^{1/3},$$

вовсе не является таким уж секретом.

Однако этот пример иллюстрирует еще одну весьма важную математическую задачу. Фактически в (2) использовано разложение исходного показателя степени на простые множители, а именно, $6 = 2 \cdot 3$, и вместо извлечения одного корня шестой степени можно извлечь сначала корень второй степени, а затем корень третьей степени.

Опять-таки, разложить на простые множители число 6 — задача, не содержащая секрета. Однако эта же задача, поставленная в общем виде, дает неожиданный результат. Несмотря на колоссальные усилия математиков, особенно за последние 20—25 лет, до сих пор не найден эффективный алгоритм разложения произвольного числа на простые множители. Именно осознание этого факта позволило построить один из самых эффективных и наиболее распространенных алгоритмов шифрования, так называемый алгоритм RSA, описание которого приведено в следующем разделе.

1.3.2. Алгоритм шифрования RSA

Алгоритм шифрования RSA был предложен в опубликованной в 1977 году работе американскими математиками (Ривест, Шамир и Адлеман) и назван по первым буквам английской транскрипции их фамилий (Rivest R. L., Shamir A., Adleman L.). В настоящее время он является одним из наиболее популярных алгоритмов. Некоторые авторы считают, что RSA — наиболее часто употребляемый алгоритм шифрования вообще.

К сожалению, подробное изложение математического обоснования RSA опирается на некоторые результаты теории чисел, изложение которых выходит за пределы данных лекций. Ниже мы приведем необходимые факты без доказательств. Более подробную информацию об RSA можно найти в работе [6].

Приведем основные моменты построения алгоритма.

В качестве основной шифрующей функции в схеме RSA принята

$$f(x) = x^e \bmod m, \tag{3}$$

где x — шифруемый исходный текст, e, m — заданные натуральные числа. (Здесь и далее выражение $a \bmod m$ означает остаток от деления числа a на число m. Другими словами, для вычисления значения $f(x)$ по формуле (3) нужно x возвести в степень e и результат поделить на m. Остаток от деления и даст искомое значение функции). Пара чисел e и m считается известной, она составляет открытый ключ схемы шифрования.

Для расшифрования сообщения $y = f(x)$ необходимо, зная y, найти x из уравнения $x^e = y \bmod m$.

Авторы схемы RSA показали, что для вычисления обратной функции достаточно вычислить

$$x = y^d \bmod m,$$

где d — некоторое число, удовлетворяющее соотношению

$$de \bmod \varphi(m) = 1. \tag{4}$$

Число d является секретным ключом схемы, необходимым для расшифрования.

В уравнении (4) функция $\varphi(m)$ — так называемая функция Эйлера — хорошо изученная в теории чисел. Для любого целого m она равна числу целых чисел из интервала $1, 2, \ldots, m-1$, взаимно простых с m.

Достаточно легко проверить следующие свойства функции Эйлера:

0) $\varphi(1) = 1$;

1) $\varphi(p) = p - 1$ для любого простого p;

2) $\varphi(p^r) = p^{r-1}(p-1)$ для любого простого p и целого r;

3) $\varphi(pq) = \varphi(p)\varphi(q)$ для любых взаимно простых p и q.

Свойства 1, 2, 3 позволяют легко вычислять $\varphi(m)$, если известно разложение числа m на простые множители.

Достаточно давно была доказана теорема Эйлера, частный случай которой утверждает, что если произвольное число x взаимно просто с m, то $x^{\varphi(m)} \bmod m = 1$.

Теперь готов весь алгоритм для организации схемы шифрования RSA.

• *Организатор схемы*

— выбирает два достаточно больших простых числа p, q;

— вычисляет $m = pq$;
— выбирает $e < m$, которое должно быть взаимно простым с числом $(p-1)\cdot(q-1)$;
— с помощью алгоритма Евклида вычисляет число d из условия (4);
— числа m и e публикуются, числа d, p, q остаются секретными.

• *Любой участник схемы*, желающий зашифровать исходный текст x, вычисляет (зная e и m)

$$y = x^e \bmod m$$

и результат y отправляет организатору схемы в качестве зашифрованного текста.

• *Организатор схемы*, получив шифрованное сообщение y и зная секретное число d, вычисляет исходное сообщение

$$x = y^d \bmod m.$$

Проверим это равенство:

$$y^d \bmod m = x^{de} \bmod m = \{\text{по условию (4)}\} =$$
$$= x^{\varphi(m)} x \bmod m = \{\text{по теореме Эйлера}\} = x \bmod m.$$

• *Участник схемы*, знающий всю организацию схемы RSA, но *не знающий секретного числа* d, должен, чтобы расшифровать сообщение y, вычислить d по формуле (4). В эту формулу входит величина $\varphi(m)$, которую можно легко вычислить, зная разложение m на простые множители. Заметим, что организатор схемы, знающий разложение числа m на простые множители, легко вычислял $\varphi(m)$ по формуле $\varphi(m) = (p-1)(q-1)$. Для пользователя, не знающего секретного числа d, единственная возможность вычислить $\varphi(m)$ — разложить m на простые множители, а эта задача, как уже указывалось, не имеет эффективного алгоритма.

Пример 1. Вычислим d по заданным m, p, q, e.

В этом примере в качестве шифрующей «односторонней функции с секретом» используется степенная функция $y = x^e \bmod m$ при конкретном значении параметра $m = 91$ и ключа шифрования $e = 29$, а в качестве «секретного свойства» этой функции использовалось знание разложения параметра m на простые множители ($m = p \cdot q$), или в конкретном примере $91 = 7 \cdot 13$. Значение функции Эйлера

$$\varphi(m) = (p-1)(q-1) = 6 \cdot 12 = 72.$$

Число $e = 29$ не имеет общих делителей ни с 91, ни с 72, значит, годится в качестве ключа шифрования.

Найдем d по алгоритму Евклида. Этот алгоритм используется для отыскания наибольшего общего делителя двух целых чисел:

$$72 = 2 \cdot 29 + 14;$$
$$29 = 2 \cdot 14 + 1;$$
$$1 = 29 - 2 \cdot 14 = 29 - 2 \cdot (72 - 2 \cdot 29) = 5 \cdot 29 - 2 \cdot 72;$$
$$5 \cdot 29 = 1 \bmod 72.$$

Ответ: $d = 5$.

Пример 2 (исторический)**.** Приведем, в прямом смысле слова, исторический пример, который сыграл огромную роль в распространении схемы RSA. Для иллюстрации своего метода Ривест, Шамир и Адлеман зашифровали с помощью предложенного ими метода английскую фразу: «The magic words are squeamish ossifrage» (при этом, по-видимому, авторы не слишком старались придать шифруемому тексту какой-либо содержательный смысл).

Сначала этот текст стандартным образом (см. п. 1.3.1.1) был перекодирован в целое число x. При этом, естественно, получилось целое число, содержащее 78 десятичных цифр:

$$x = 200805001301070903002315180419000118050019172105011309$$
$$190800151919090618010705.$$

Затем к этому числу была применена процедура шифрования RSA

$$f(x) = x^e \bmod m$$

при значении $e = 9007$ и 129-значном (!) значении

$$m = 114381625757888867669325779976146612010218296721242362$$
$$562561842935706935245733897830597123563958705058989075147$$
$$599290026879543541.$$

Полученный в результате зашифрованный текст

$$f(x) = 968696137546220614771409222543558829057599911245743 1$$
$$987469512093081629822514570835693147662288398962801339199$$
$$0551829945157815154$$

был опубликован вместе с указанными значениями использованных параметров e и m. Дополнительно сообщалось, что $m = pq$, где p и q — некоторые простые числа, записываемые соответственно 64 и 65 десятичными знаками. Первому, кто дешифрует соответствующее сообщение, была обещана награда в 100 долларов.

Эта история завершилась спустя 16 лет в 1994 г., когда D. Atkins, M. Graff, A. K. Lenstra и P. C. Leyland сообщили о дешифровке фразы, предложенной выше. Соответствующие числа p и q оказались равными

$$p = 3490529510847650949147849619903898133417764638493387843990820577,$$

$$q = 3276913299326670954996198819083446141317764296799294253979828853 3.$$

Выполнение вычислений потребовало колоссальных ресурсов. В работе, возглавлявшейся четырьмя авторами проекта и продолжавшейся после предварительной теоретической подготовки примерно 220 дней, на добровольных началах участвовало около 600 человек и примерно 1600 компьютеров, объединенных сетью Internet. Понятно, что основной трудностью всего проекта, которую удалось преодолеть его участникам, было именно разложение 129-значного числа на простые множители. Детали вычислений опубликованы в статье, в заголовок которой авторы вынесли зашифрованную за шестнадцать лет до этой публикации и расшифрованную ими английскую фразу. Честным трудом заработанные 100 долларов участники этого грандиозного проекта передали в фонд развития программного обеспечения.

Описанная выше схема RSA используется во всем мире чрезвычайно широко и во многих приложениях. Вместе с тем ее использование ставит ряд вопросов. Главный из них: существуют ли другие способы решения сравнения (4)? Ведь если можно найти решение (4), не разлагая число m на простые сомножители, да еще сделать это достаточно быстро, вся система RSA разваливается. Между тем, вопрос этот чрезвычайно важен. Ведь схема RSA положена в основу защиты многих информационных систем, хранящих и государственные и коммерческие тайны. Косвенным, но все же достаточно убедительным, подтверждением стойкости системы RSA является тот факт, что за 16 лет никто так и не объявил, что смог дешифровать предложенную авторами RSA фразу. Не довольствуясь этим, специалисты по информационной безопасности принимают дополнительные меры для защиты своих систем, например, периодически с интервалом заведомо меньшим, чем возможное время, требуемое для взлома шифра, проводится смена ключей шифрования и новое «перешифрование» защищаемых текстов.

И все-таки, если когда-либо кому-нибудь из математиков удастся найти алгоритм, дешифрирующий схему RSA за приемлемое время

без знания секретного ключа d, почти наверняка этот заведомо выдающийся математический результат будет не опубликован, а засекречен, ибо битва за шифры продолжается.

1.3.3. Открытый и закрытый ключи

Как уже отмечалось, понятие ключа появилось еще в древности вместе с возникновением первых систем шифрования. Вместе с тем, описанный в предыдущих разделах математический аппарат позволяет формализовать это важнейшее понятие теории шифрования.

Схема асимметричного шифрования, основанная на использовании понятия функции с секретом, предполагает с одной стороны известный всем участникам схемы (другими словами, *открытый*) алгоритм шифрования, а с другой стороны, известный только «законным участникам» (другими словами, *закрытый*) алгоритм расшифрования, использующий «секретное свойство» шифрующей функции.

При этом каждый математический алгоритм, в том числе и любой алгоритм шифрования, сформулированный в общем виде, содержит какое-то количество параметров, которым для реальной работы этого алгоритма должны быть присвоены конкретные *значения*.

Таким образом, мы получаем два определения.

Открытым ключом схемы шифрования, использующей функцию с секретом, является совокупность конкретных значений параметров, обеспечивающих работу открытого алгоритма шифрования с помощью этой функции.

Закрытым ключом схемы шифрования, использующей функцию с секретом, является совокупность конкретных значений параметров, обеспечивающих работу алгоритма расшифрования с использованием «секретного свойства» шифрующей функции.

По установившейся традиции для открытых и закрытых ключей в криптографии используются обозначения $K1$ и $K2$ соответственно.

Поясним сказанное на конкретном примере.

Пример 3. Схема шифрования RSA.

Сказанного выше достаточно, чтобы понять, что в схеме RSA, описанной в п. 1.3.2, открытый и закрытый ключи определяются соотношениями $K1 = (m, e)$, $K2 = (p, q, d)$, поскольку тот, кто знает параметры m и e, может осуществлять шифрование по формуле

$$y = x^e \bmod m,$$

а тот, кто знает параметры p, q и d, может еще и осуществлять расшифрование по формуле

$$x = y^d \bmod m.$$

Важное замечание. Понятие ключа шифрования является важнейшим в современной криптографии. Важно понять, что закрытый ключ является единственным действительно секретным элементом криптографической схемы. В отличие от закрытых ключей *криптографические алгоритмы* вовсе не являются секретными. Их можно обсуждать и публиковать. Более того, они являются коммерческим товаром, а их производство весьма прибыльно.

1.3.4. Криптография и «трудные» математические задачи

В предыдущих разделах в качестве примера, реализующего ключевую для криптографии концепцию односторонней функции, приводился алгоритм RSA, в котором «трудной» математической задачей являлась задача разложения большого числа на простые множители. Однако не следует думать, что это единственная «трудная» математическая задача, которая может быть с успехом применена в криптографии. Более того, за много веков своего развития математика накопила большое число подобных «трудных» задач. Про одни из них было доказано, что они не имеют решения, про другие было установлено, что их решение на самых современных компьютерах займет не менее нескольких миллионов лет, другие просто не поддавались решению.

Казалось бы, какая может быть практическая польза от этих математических доказательств невозможности решения? Ну, доказали, что какую-то задачу нельзя решить. Ну и что? Вот если бы нашли решение, тогда другое дело. Тогда еще, возможно, кто-нибудь, когда-нибудь да и нашел бы этому решению практическое применение.

Но все изменилось в 70-х годах прошлого века с появлением математической криптографии и концепции односторонней функции. Эти самые абстрактные «трудные» математические задачи оказались востребованы для решения весьма актуальных практических задач. В самом деле, ведь если доказано, что некоторую задачу нельзя решить, то ее не сможет решить и противник, и такая задача может с успехом использоваться в криптографических схемах. Подобные метаморфозы часто случаются в науке: вчера еще абстрактная теоретическая наука, результаты которой интересуют от силы пару десятков специалистов-энтузиастов во всем мире, — сегодня неожиданно становится основой целой отрасли прикладной практической деятельности для тысяч исследователей, инженеров, производственников и обывателей.

Возвращаясь к криптографии, отметим, что, используя различные «трудные» задачи, математики построили большое количество криптографических алгоритмов. В основе каждого из них лежит та или иная «трудная» математическая проблема. Эти алгоритмы зачастую

решают одну и ту же криптографическую задачу, результаты их работы можно сравнивать по тем или иным критериям (например, по быстродействию), и выбор того или иного алгоритма представляет собой достаточно сложную организационную, экономическую, а иногда и государственную задачу.

Приведем для иллюстрации, без математических подробностей и доказательств, еще один пример «трудной» математической задачи, которая нашла широкое применение в криптографии.

Задача дискретного логарифмирования

Рассмотрим в качестве шифрующей функции выражение

$$F(x) = a^x \bmod p, \tag{5}$$

где p — большое простое число, a — некоторое специально подобранное целое число, зависящее от выбора p.

Тогда для вычисления $F(x)$ имеются эффективные алгоритмы, реализующие возведение целого числа в большую степень. В то же время, вычисление x по известному $F(x)$ — это «трудная» математическая задача, известная как задача дискретного логарифмирования, для которой нет эффективных алгоритмов. Поэтому (5) — односторонняя функция и на ее основе построено большое количество криптографических схем.

Учитывая указанную «множественность» криптографических алгоритмов, часто в тех случаях, когда не требуется специально указывать, какой именно алгоритм использовался для шифрования и расшифрования, многие авторы применяют символические обозначения для этих процессов.

Пусть x, как обычно, обозначает исходный текст. Тогда процесс шифрования (и зашифрованный текст) обозначают $E(x)$, а соответствующий процесс расшифрования, применяемый к зашифрованному тексту, обозначают $D(E(x))$. То, что алгоритмы E и D соответствуют друг другу, выражается тождеством $D(E(x)) = x$ для любого исходного текста x.

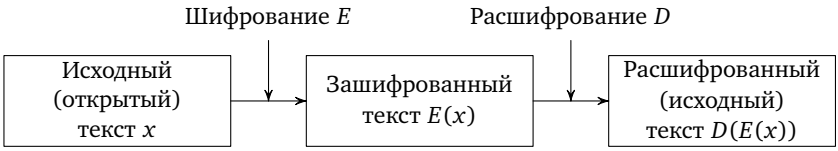

Рис. 2. Шифрование и расшифрование исходного текста

Если применяемые алгоритмы шифрования и расшифрования используют открытый ($K1$) и закрытый ($K2$) ключи, соответствующие процессы записываются следующим образом. Шифрование с использованием открытого ключа $K1$ обозначается через $E_{K1}(x)$, расшифрование с использованием закрытого ключа $K2$ — через $D_{K2}(E_{K1}(x))$, а тот факт, что алгоритмы E и D, а также конкретные значения ключей $K1$ и $K2$ соответствуют друг другу, выражается тождеством $D_{K2}(E_{K1}(x)) = x$.

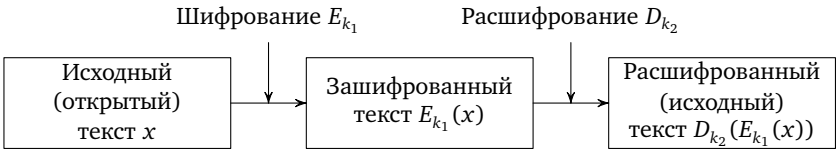

Рис. 3. Шифрование и расшифрование исходного текста с использованием пары ключей: открытого — k_1 и закрытого — k_2

2. Криптографические протоколы

2.1. Что такое криптографические протоколы

Концепция односторонних функций и функций с секретом появилась как математическая формализация процессов шифрования и расшифрования. Однако очень быстро было замечено, что эта концепция может быть с успехом применена к решению и многих других не менее важных задач криптографии, т. е. таких задач, когда, взаимодействуя с удаленным партнером по компьютерным сетям, необходимо обеспечить надежность передачи информации своему партнеру, а с другой стороны, гарантировать, что эта информация не будет перехвачена противником. Кроме того, широкое развитие передачи информации по сетям поставило и ряд совсем новых вопросов. Нужно, например, убедиться, что удаленный партнер именно тот, за кого он себя выдает, а если даже и тот, то нужны гарантии того, что он вас не обманет. Например, не откажется от своей подписи, или не откажется подписать контракт после того, как получит вашу подпись.

Большое количество подобных задач нашло свое решение в виде так называемых *криптографических протоколов*. В простейшем случае криптографический протокол — это последовательность взаимодействий двух удаленных пользователей (которые не видят друг друга и не слишком друг другу доверяют), предназначенная для решения какой-либо простейшей задачи. Тем не менее, точное выполнение протокола заведомо гарантирует строгое выполнение поставленной задачи, даже если один из пользователей попытается обмануть другого. Такие криптографические протоколы называются примитивными, и в различных публикациях по криптографии их приводится несколько десятков.

Объединяя в нужной последовательности несколько примитивных протоколов, можно получить прикладные криптографические схемы, в которых участвует практически неограниченное число пользователей и которые предназначены для решения вполне серьезных и вполне актуальных задач, например, организации электронных торгов, или электронного голосования, или организации электронной платежной системы. Важно подчеркнуть, что, как и в случае примитивных протоколов, прикладные криптографические схемы

разрабатываются таким образом, чтобы обеспечить надежность и защиту информации и самой схемы от возможных атак противников.

Ниже приведено несколько примеров как примитивных протоколов с двумя участниками, так и упомянутых протоколов организации электронных торгов и электронного голосования.

И наконец, последнее замечание. Большинство описанных в последующих разделах протоколов приведены в форме, использующей алгоритм RSA. Однако именно это не является обязательным. Учитывая то, что было сказано в п. 1.3.4, нетрудно понять, что для конкретной реализации того или иного криптографического протокола могут быть применены разные односторонние функции, основанные на использовании различных «трудных» математических задач. В результате получаются различные варианты одного и того же протокола, решающие одну и ту же задачу, но обладающие различными характеристиками, например, по быстродействию или по каким-либо другим параметрам [3].

2.2. Протокол подбрасывания монеты по телефону

Подбрасывание монеты, пожалуй, наиболее распространенная форма организации жребия между двумя участниками. Она предельно проста, если оба участника присутствуют при жеребьевке лично. Однако ситуация сильно изменяется, если оба участника жеребьевки пытаются бросить жребий, находясь на разных концах телефонного провода или будучи связаны компьютерной сетью. Ситуация еще более ухудшается, если оба участника не слишком доверяют друг другу.

Действительно, пусть два сослуживца обсуждают по телефону вопрос «кто первый сообщит неприятную новость начальнику» и пытаются решить его с помощью жеребьевки. Один из них подбрасывает монету, а его партнер на другом конце провода пытается угадать результат этого подбрасывания. При этом проверка этого угадывания целиком будет опираться только на честность подбрасывающего монетку участника и, по всей видимости, вероятность того, что сообщать новость начальнику придется его телефонному собеседнику, в этой ситуации значительно превысит одну вторую.

Казалось бы, подобная задача не имеет решения. Тем не менее она решается, и даже довольно несложно, с помощью аппарата односторонних функций.

Рассмотрим одностороннюю функцию $f(x)$, удовлетворяющую следующим условиям:

1) $f(x)$ определена на некотором множестве целых чисел, которое содержит одинаковое количество четных и нечетных чисел;
2) функция $f(x)$ такова, что если $f(x_1) = f(x_2)$, то x_1 и x_2 имеют одинаковую четность;
3) функция $f(x)$ такова, что по заданному значению $f(x)$ «трудно» вычислить четность неизвестного аргумента x.

Допустим, что функция, удовлетворяющая указанным свойствам, выбрана и известна участникам жеребьевки. Пусть А подбрасывает монету, а В пытается угадать результат. Тогда протокол обмена между абонентами, решающий задачу, состоит из следующих шагов.

а) *А выбирает случайное значение x (подбрасывает монету), зашифровывает x и полученное значение $f(x)$ посылает В.*
б) *В, получив $f(x)$, пытается угадать четность x и направляет свою догадку А.*
в) *А, получив догадку от В, сообщает В, угадал тот или нет, и направляет ему в подтверждение выбранный x.*
г) *В проверяет, не обманул ли его А, для чего он вычисляет $f(x)$ и сравнивает его с полученным от А на первом шаге.*

Если $f(x)$, вычисленный абонентом В на шаге г), совпадает с полученным им от А на шаге а), то результат жеребьевки, который А объявил на шаге в), приходится принять. Действительно, даже если А попытается схитрить и отправит В для проверки другое значение x', для которого $f(x') = f(x)$, то x и x' будут иметь одинаковую четность в силу свойства 2 функции $f(x)$, и эта хитрость не удастся.

Приведем один из возможных примеров функции, удовлетворяющей условиям 1—3. Выберем для начала стандартную схему RSA и пусть (m, e) — открытый ключ этой схемы. Выберем, кроме того, некоторое четное число p меньше m. Тогда функция

$$f(x) = (x \bmod p)^e \bmod m \tag{6}$$

удовлетворяет требуемым условиям.

Действительно, условие 1 просто накладывает требования на область возможного выбора аргумента x.

Далее, пусть $x_1 = a_1 \cdot p + r_1$, $x_2 = a_2 \cdot p + r_2$. Тогда из равенства $f(x_1) = f(x_2)$ следует

$$(x_1 \bmod p)^e \bmod m = (x_2 \bmod p)^e \bmod m.$$

Возводя в степень d (где $de = 1 \bmod \varphi(m)$), получим

$$x_1 \bmod p = x_2 \bmod p,$$

откуда $r_1 = r_2$ и, следовательно, $x_1 - x_2 = (a_1 - a_2) \cdot p$, т. е. x_1 и x_2 в силу четности числа p имеют одинаковую четность, и тем самым выполнено свойство 2.

Что касается свойства 3, то поскольку число m в схеме RSA всегда нечетно, вычисление по $\bmod m$ не обязательно сохраняет четность. Другими словами, число $a \bmod m$ может быть как четным, так и нечетным, независимо от четности числа a. Например, $13 \bmod 3 = 1$ и $16 \bmod 3 = 1$, а $14 \bmod 3 = 2$ и $17 \bmod 3 = 2$. Поэтому, зная значение $f(x)$, вычисленное по формуле (6), четность аргумента x можно определить, только вычислив само значение x, а это «трудная» задача по самому построению схемы RSA. Тем самым выполнено и условие 3.

Тому, кто прочитал данный раздел, может показаться, что задача «подбрасывания монеты по телефону» носит чисто развлекательный характер и не может иметь никакого практического значения. Однако это совсем не так. Во-первых, протокол а)—г) достаточно очевидным образом обобщается на задачу с испытаниями, имеющими не два, а более исходов (аналог задачи «бросания игрального кубика по телефону»). Дальше можно будет поиграть по телефону в покер или в нарды, а там недалеко и до виртуального казино, а это уже совсем не умозрительные задачи.

2.3. Протокол аутентификации

Базы данных компьютеров хранят огромное количество информации. Часть ее общедоступна и с нею может ознакомиться любой желающий. Однако кроме общедоступной информации в базах данных может также храниться информация, представляющая служебную, коммерческую или государственную тайну. Ясно, что эта информация не может быть общедоступной, она является закрытой. Но также ясно, что если закрытая информация имеется, значит, кто-то должен иметь к ней доступ, даже к самой секретной. Этот процесс разделения всех пользователей на категории в зависимости от вида доступной информации называется обычно *разграничением доступа*. К этому понятию относят также и способ обработки информации, доступный тому или иному пользователю — одному пользователю разрешается один фрагмент информации только читать, а другой фрагмент еще и изменять, либо совсем уничтожать, либо создавать новый фрагмент информации. А другой пользователь может всю информацию использовать только в режиме чтения. Короче говоря, концепция разграничения доступа предусматривает следующее условие: прежде

чем допустить какого-либо пользователя к информации, необходимо определить его полномочия, а для этого нужно сначала этого пользователя идентифицировать. Криптографический процесс идентификации пользователя, защищающий информационную систему от несанкционированного доступа, и называется *аутентификацией*.

В протоколе аутентификации участвуют два пользователя: A — пользователь, пытающийся пройти аутентификацию, и B — проверяющий. Рассмотрим протокол аутентификации, основанный на алгоритме RSA. Он состоит из следующих шагов:

а) *A строит стандартную схему RSA. Пусть ее открытый ключ* (m, e) *и закрытый ключ* (m, d). *Открытый ключ публикуется, закрытый остается известен только A.*

б) *B посылает A случайную строку* s.

в) *A вычисляет с помощью известного только ему закрытого ключа число* g *по формуле*

$$g = s^d \bmod m$$

и отправляет полученное число g *проверяющему B.*

г) *Проверяющий B, получив число* g, *выбирает из публикации открытый ключ абонента A и вычисляет с его помощью число* s' *по формуле*

$$s' = g^e \bmod m.$$

Если при этом $s' = s$, *то аутентификация принята, если же* s' *не равно* s *— аутентификация отвергнута.*

Смысл указанного протокола в том, что проверяющий B убеждается в том, что абонент, проходящий процедуру аутентификации, действительно знает закрытый ключ схемы RSA, парный открытому ключу абонента A, а значит, он этот самый A и есть. Любой другой абонент, пытающийся выдать себя за абонента A с целью получить его полномочия по доступу к информационной системе, должен для успешного прохождения процедуры аутентификации либо взломать схему RSA, построенную абонентом A, либо попросту украсть у A его закрытый ключ. При этом первое невозможно при условии грамотного построения абонентом A схемы RSA. Второе невозможно при условии грамотного соблюдения абонентом A правил хранения секретной информации. Выполнение обоих этих условий возлагается на абонента A, поскольку это целиком в его интересах.

Отметим при этом одно важное обстоятельство: при использовании указанного протокола аутентификации секретный ключ d пользователя A не раскрывался.

2.4. Доказательство с нулевым разглашением

Приведенный протокол иллюстрирует одно из замечательных новых понятий современной математической криптографии — так называемое *доказательство с нулевым разглашением*. Традиционное математическое доказательство на протяжении веков строилось и строится таким образом, что любой читатель, проверив доказательство какого-либо математического факта и убедившись в его правильности, способен в дальнейшем самостоятельно без помощи автора представить это доказательство любому другому третьему лицу. Можно даже сказать, что подобным образом строится не только математическое, но и вообще любое логическое доказательство.

Однако приведенный выше протокол аутентификации представляет принципиально иную природу математического доказательства. Действительно, абонент A, прошедший аутентификацию, доказал проверяющему B, что он знает секретный ключ схемы RSA, но выполнил это доказательство таким образом, что проверяющий не только не узнал этот секретный ключ, но и не получил в процессе доказательства никаких сведений, которые позволили бы ему этот ключ узнать впоследствии. Это и есть пример доказательства с нулевым разглашением.

Подобное понятие, появившееся совсем недавно, уже нашло широкое применение и в криптографии, и за ее пределами. В самом деле, аргумент: «Я докажу вам, что я это знаю, но проведу доказательство таким образом, что вы не сможете самостоятельно повторить мое доказательство никому другому» может с успехом быть использован в самых разных областях деятельности.

2.5. Электронная подпись

С развитием электронной почты с компьютера на компьютер стало поступать огромное количество посланий самого разного свойства. Среди личных посланий, поздравлений, рекламных проспектов и прочего в электронной почте стали попадаться и *документы*. А обычные строки текста превращаются в документ, когда под ними появляется *подпись*. К этому нас приучил многовековой опыт «бумажного» обращения с текстами.

Более того, даже один и тот же текст может иметь совершенно разный смысл в зависимости от того, кто его подписал. Одно дело, если текст подписал министр, и совсем другое дело, если тот же самый текст подписал третий секретарь четвертого отдела министер-

ства. Однако если имеется подпись, то имеется и проблема борьбы с ее подделкой. И переход к электронным документам эту проблему только обострил. Поскольку, например, передача по сети сканированного документа, снабженного ручной подписью, проблемы подделки ни в коем случае не решает, а для любой мало-мальски серьезной экспертизы подлинности подписи требуется оригинал «бумажного» документа. Так возникла задача создания электронной цифровой подписи.

Электронной цифровой подписью (ЭЦП) называется реквизит электронного документа, появление которого в документе получателем подтверждает два факта: то, что документ дошел до получателя без искажений, и то, что он подписан именно отправителем.

В качестве примера рассмотрим протокол формирования ЭЦП, основанный на алгоритме RSA. В обмене участвуют два абонента: отправитель A и получатель B. Пусть A отправляет получателю B текст b и желает, чтобы это сообщение было снабжено ЭЦП.

Протокол, решающий указанную задачу, состоит из следующих шагов:

а) *A строит стандартную схему RSA. Пусть ее открытый ключ* (m, e) *и закрытый ключ* (m, d). *Открытый ключ публикуется, закрытый остается известен только A.*

б) *A вычисляет с помощью известного только ему закрытого ключа число* s *по формуле*

$$s = b^d \bmod m,$$

где b *— отправляемый текст, и отправляет получателю B исходный текст* b *и полученное число* s *в качестве его ЭЦП.*

в) *Проверяющий B, получив число* s, *выбирает из публикации открытый ключ абонента A и вычисляет с его помощью число* b' *по формуле*

$$b' = s^e \bmod m.$$

Если при этом $b' = b$, *то подпись правильна и это означает, что исходный текст принят без искажений и этот текст подписан действительно A. Если же* $b' \neq b$, *то это означает, что либо в исходный текст внесены изменения, либо он подписан не отправителем A, либо и то и другое вместе.*

Смысл указанного протокола в том, что абонент B, как и в случае протокола аутентификации, убеждается, что подпись s, полученная им вместе с исходным текстом b, вычислялась с использованием закрытого ключа (m, d), парного к открытому ключу отправителя. А такую операцию по определению мог проделать только сам отправитель

А, поскольку только он знает закрытый ключ. И при этом опять-таки получатель не приобретает никакой информации о самом закрытом ключе отправителя, т. е. и в этом протоколе вновь работает схема нулевого разглашения.

Схема ЭЦП имеет еще одно важнейшее свойство. Оно заключается в том, что данная конкретная подпись может относиться только к одному конкретному тексту. Действительно, если отправитель А попытается подобрать другой текст b', так, чтобы он имел ту же подпись, что и исходный текст b, ему придется найти новый текст b' из уравнения

$$(b')^d \bmod m = b^d \bmod m.$$

А такое уравнение, по крайней мере в классе осмысленных текстов, может иметь только одно решение $b' = b$.

Это обстоятельство определяет важнейшее свойство ЭЦП — *невозможность отправителя подменить подписанный им текст либо отказаться от подписи*. Это свойство, иногда для краткости называемое *неотказуемостью* либо *неотвергаемостью подписи*, ставит в жесткие рамки отправителя сообщения, в то же время страхуя получателя от возможных последствий недобросовестности отправителя. С другой стороны, отправитель может быть уверен, что его ЭЦП никто не может подделать, поскольку закрытый ключ знает только он сам. Эти два обстоятельства делают тексты, подписанные с помощью ЭЦП, юридически значимыми. Они могут служить основой сделок, контрактов, могут приниматься в судах в качестве вещественных доказательств и вообще могут признаваться там, тогда и в той степени, где, когда и в какой степени могут признаваться обычные собственноручные личные подписи.

2.6. Электронные торги

Такие свойства ЭЦП, как ее «неподделываемость» и «неотказуемость», находят самое широкое применение. Рассмотрим, для примера, протокол организации электронных торгов, которые в силу их мобильности, возможности значительного расширения состава участников и сокращения большого количества бюрократических процедур получили в последнее время широкое распространение.

Как известно, любые торги организуются, если отвлечься от деталей, по следующей схеме:

1) организатор торгов объявляет условия торгов, критерии определения победителей и срок приема заявок от участников;

2) участник торгов в установленный срок направляет организатору свою заявку с предложениями в соответствии с объявленными условиями;
3) организатор торгов, закончив прием заявок от всех участников, сравнивает их предложения между собой и объявляет победителя, который предложил лучшие условия в соответствии с заранее объявленными в условиях торгов критериями.

Такова вкратце схема организации торгов. Особенностью любой такой схемы является обеспечение конфиденциальности содержания заявок участников с момента подачи ими своей заявки и до подведения итогов. В обычной практике это требование обеспечивается достаточно просто. Все участники торгов направляют свои предложения организатору в запечатанных конвертах, которые вскрываются все одновременно в момент подведения итогов.

Рассмотрим, как могут быть выполнены все условия организации торгов, включая соблюдение конфиденциальности заявок участников, в случае когда организатор торгов связан с участниками компьютерной сетью.

Приведенный ниже протокол основан опять-таки на схеме RSA.

а) *Организатор торгов создает стандартную схему RSA с открытым ключом* (m, e) *и закрытым ключом* (m, d). *Открытый ключ публикуется, с тем чтобы он стал известен всем участникам торгов, закрытый ключ остается известен только организатору.*

б) *Каждый участник торгов* V_i $(i = 1, 2, \ldots, N)$ *формирует свое исходное предложение* s_i *и направляет организатору зашифрованное предложение* f_i, *определяемое по формуле*

$$f_i = (s_i)^e \bmod m.$$

в) *Организатор торгов по истечении срока поступления заявок публикует реестр участников, принявших участие в торгах, и их зашифрованные предложения* f_i. *Это нужно для того, чтобы каждый участник имел возможность проверить правильность получения организатором своего зашифрованного предложения и для возможности осуществления последующего контроля.*

г) *После публикации реестра зашифрованных предложений организатор осуществляет вычисление исходных заявок участников по формуле*

$$s_i = (f_i)^d \bmod m,$$

публикует исходные заявки, сравнивает предложения участников и объявляет победителя.

Замечание 1. Сравнивая приведенный протокол электронных торгов с протоколом ЭЦП, можно увидеть, что зашифрованные заявки — это фактически ЭЦП исходных заявок. Это позволяет перенести на протокол электронных торгов все результаты, относящиеся к ЭЦП, в частности, «неподделываемость» и «неотказуемость».

Замечание 2. Замена исходной заявки ее цифровой подписью является криптографическим аналогом упомянутого конверта, в который запечатывается исходная заявка с целью сохранения ее конфиденциальности.

Убедимся, что приведенный протокол решает все задачи организации торгов. В самом деле:

— до момента подведения результатов сохраняется конфиденциальность исходных заявок участников, поскольку провести расшифровку опубликованных зашифрованных заявок можно только с использованием закрытого ключа, а он известен только организатору;
— после того как опубликован реестр зашифрованных заявок, в силу действия «принципа неотказуемости» невозможно даже по сговору с организатором подменить ни одну из исходных заявок таким образом, чтобы сохранить без изменения зашифрованную заявку;
— и наконец, после объявления победителя и публикации его исходной заявки любой участник торгов может осуществить проверку факта получения организатором именно этой заявки до начала процедуры подведения итогов торгов. Для этого он должен взять опубликованную исходную заявку победителя (обозначим ее $s_{r'}$), повторить процедуру шифрования, т. е. вычислить $f_{r'} = (s_{r'})^e \bmod m$, и сравнить с зашифрованной заявкой f_r из реестра, опубликованного организатором. Равенство $f_{r'} = f_r$ возможно только в случае $s_{r'} = s_r$, а это и будет означать, что организатор при подведении итогов рассматривал ту самую заявку, которую получил на первом этапе конкурса.

Особенно эффективно проведение электронных торгов в случае регулярных торгов, для которых критерий определения результатов также может быть формализован. Условия для таких торгов разрабатываются один раз, а результат каждых торгов определяется автоматически без сбора комиссий, вскрытия конвертов и сравнения присланных заявок по зачастую нечетким критериям. Примером такого эффективного применения электронных торгов могут быть ежедневные торги на Нью-Йоркской нефтяной бирже.

2.7. Протокол электронного голосования

Допустим, что в голосовании участвуют N избирателей $V_1, V_2, \ldots, V_N$, которые могут передавать данные по сети в электронной форме. Допустим также для простоты, что голосование каждого избирателя состоит в простейшем выборе: «за», «против» или «воздержался».

Сформулируем естественные требования к протоколу электронного голосования (е-голосования):

1) голосование должно быть тайным;

2) один избиратель имеет один голос;

3) должна быть обеспечена процедура проверки правильности подсчета голосов.

В рассматриваемой ниже схеме, наряду с избирателями, фигурирует также некий Центр, с которым избиратели могут обмениваться данными. Он создается для организации процесса голосования и проведения подсчета голосов и является, таким образом, электронным аналогом избирательной комиссии.

Для удобства весь протокол разделен на четыре этапа:

— подготовка к голосованию;
— голосование;
— подсчет результатов голосования;
— проверка подсчета результатов.

Этап 1. Подготовка к голосованию

Центр начинает свою работу по организации выборов с создания схемы шифрования по алгоритму RSA, которая будет использоваться при е-голосовании. Для этого Центр выбирает в соответствии со схемой RSA значения (m, e) в качестве открытого ключа и значения (m, d) в качестве закрытого ключа. Открытый ключ сообщается всем избирателям, закрытый ключ остается секретным, он известен только Центру.

Далее устанавливаются следующие правила голосования для избирателей. Каждый избиратель V_i назначает секретное значение параметра b_i в зависимости от своего выбора по следующему правилу:

$$b_i = \begin{cases} 2, & \text{если он голосует «за»;} \\ 3, & \text{если он голосует «против»;} \\ 1, & \text{если он воздерживается.} \end{cases} \qquad (7)$$

Значение b_i играет роль электронного аналога избирательного бюллетеня для голосования, заполненного избирателем V_i.

И наконец, последней подготовительной процедурой является электронный аналог регистрации избирателей, в качестве которого для каждого избирателя требуется пройти в Центре процедуру аутентификации. Если предположить, что эта процедура будет проходить в соответствии с протоколом аутентификации, изложенном в п. 2.3, то каждый избиратель V_i, имеющий право и желающий принять участие в голосовании, должен создать свой открытый ключ (m_i, e_i) и закрытый ключ (m_i, d_i) и разместить открытый ключ так, чтобы он был доступен Центру.

После завершения этих подготовительных оргмероприятий первого этапа можно приступать к самому процессу е-голосования.

Этап 2. Голосование

Прежде чем описать сам процесс е-голосования, следует сделать несколько предварительных замечаний относительно процесса шифрования е-бюллетеней. Прежде всего, следует заметить, что «лобовая» схема типа «каждый избиратель шифрует свой е-бюллетень b_i с помощью открытого ключа по формуле

$$p_i = b_i^e \bmod m$$

и полученный зашифрованный е-бюллетень p_i пересылает Центру» в данном случае не подходит. Действительно, поскольку любой е-бюллетень b_i может принимать в соответствии с (7) только одно из трех значений (1, 2 или 3), их зашифрованные образы будут принимать тоже всего лишь три различных значения (ведь открытый ключ у всех избирателей один и тот же), и ни о какой тайне голосования не может быть речи.

Подобные ситуации довольно часто встречаются в криптопротоколах, и для их разрешения применяют стандартный прием: шифрование по формуле $y = f(x)$ заменяют шифрованием по формуле $y_1 = f(x, s)$, где s — случайный параметр. Этот прием называют «затенением». Случайный параметр s должен удовлетворять двум условиям. Во-первых, как правило, должно выполняться $f(x, s_1) \neq f(x, s_2)$, что не позволяет угадать значение x по $f(x)$. Во-вторых, добавление параметра s не должно мешать при расшифровке.

Возвращаясь к схеме е-голосования, проведем «затенение» е-бюллетеня b_i избирателя V_i по формуле

$$t_i = b_i \cdot q_i, \tag{8}$$

где q_i — случайно выбранное избирателем V_i простое число, для которого в рассматриваемом простейшем случае голосования из двух альтернатив достаточно потребовать выполнение условия $q_i \geqslant 5$.

«Затененные» подобным способом е-бюллетени уже можно шифровать по принятой схеме RSA:

$$f_i = t_i^e \bmod m. \tag{9}$$

При этом если два избирателя V_i, V_j проголосовали одинаково (т. е. $b_i = b_j$), то их зашифрованные е-бюллетени будут, вообще говоря, различны в силу случайного выбора соответствующих «затеняющих» множителей q_i, q_j.

Таким образом, каждый избиратель на этапе голосования V_i выполняет следующую последовательность действий:

0) *проходит процедуру аутентификации в Центре*;
1) *голосует, т. е. выбирает b_i в соответствии с* (7);
2) *затеняет, т. е. вычисляет t_i по формуле* (8);
3) *шифрует, т. е. вычисляет f_i по формуле* (9);
4) *пересылает полученное значение f_i в Центр.*

Разумеется, сам избиратель лично выполняет только п. 1 и только в том случае, если он успешно прошел процедуру аутентификации. Избиратели, не прошедшие эту процедуру, к голосованию не допускаются. Остальные пункты выполняются программно и только в том случае, если b_i в п. 1 действительно удовлетворяет условиям (7).

По ходу голосования, получив очередной е-бюллетень от избирателя V_i, Центр вносит его ФИО и, быть может, еще какие-то данные в реестр избирателей, принявших участие в голосовании. При успешном прохождении процедуры аутентификации очередным избирателем он проверяется на наличие его ФИО в этом реестре, с тем чтобы исключить повторное участие одного и того же избирателя в голосовании. (Один избиратель — один голос).

После завершения этапа 2 в Центре будут собраны N зашифрованных е-бюллетеней, принадлежащих всем избирателям, принявшим участие в голосовании. Получив все бюллетени, Центр публикует таблицу зашифрованных е-бюллетеней вида:

ФИО избирателя	1	.....	f_1
ФИО избирателя	2	.....	f_2
.....			
ФИО избирателя	N	.....	f_N.

Публикация подобной таблицы, не нарушая тайного характера голосования, позволяет каждому избирателю проверить правильность операций шифрования его бюллетеня. На этом завершается второй этап протокола — этап голосования.

Этап 3. Подсчет результатов голосования

Опять-таки, необходимо начать с нескольких замечаний относительно неприемлемости «лобовой» схемы подсчета результатов голосования. Действительно, такая схема действий Центра могла бы выглядеть следующим образом:

— расшифровка с помощью секретного ключа (m, d)

$$t_i = f_i^d \bmod m;$$

— определение b_i путем проверки делимости t_i на 2 или на 3;
— подсчет бюллетеней «за» и «против» и объявление результатов голосования.

Подобная процедура имеет по крайней мере два недостатка, каждый из которых делает ее неприемлемой. Во-первых, эта схема предполагает непосредственное вычисление исходных избирательных бюллетеней b_i и поэтому для обеспечения тайны голосования должны быть приняты дополнительные меры. Во-вторых, без знания секретного ключа d подобную схему подсчета голосов невозможно проверить. При этом секретный ключ известен, по определению, только Центру, а недобросовестные люди, как известно, могут встретиться не только среди избирателей, но и среди работников избирательных комиссий.

Вариант процедуры подсчета голосов, свободный от указанных недостатков, мог бы выглядеть следующим образом.

1. *Вычисляется число F, равное произведению всех зашифрованных е-бюллетеней*

$$F = \prod_{i=1}^{N} f_i.$$

2. *К полученному числу F применяется процедура дешифровки с использованием закрытого ключа (m, d):*

$$Q = F^d \bmod m = \left(\prod f_i\right)^d \bmod m = \prod \left(f_i^d \bmod m\right) = \prod t_i = \prod b_i \cdot q_i,$$

где произведения $\prod$ берутся по всем i от 1 до N.

3. *Принимая во внимание, что все b_i могут принимать значения только 2 или 3, представляем число Q в виде:*

$$Q = \prod b_i \cdot q_i = (2^r) \cdot (3^p) \cdot R,$$

где $R = \prod q_i$ — произведение всех «затеняющих» множителей.

4. *Полученные показатели степени r и p есть не что иное, как общее число голосов, поданных «за» и «против» соответственно. Число*

воздержавшихся и определяется вычитанием из общего числа избирателей $u = N - (r + p)$.

5. *Центр публикует числа* r, p, *и в качестве результатов голосования и число* R *в качестве контрольного числа, которое может быть использовано для проверки правильности подсчета голосов.*

Приведенная процедура не предполагает вычисления исходных бюллетеней b_i каждого избирателя, поскольку она вычисляет сразу общее число бюллетеней, поданных «за» и «против» и поэтому сохраняет тайну голосования. Что касается контроля правильности подсчета голосов, то этот вопрос рассматривается в следующем разделе.

Этап 4. Контроль правильности подсчета голосов

Обычная общественная практика показывает, что опасения в том, что опубликованные результаты голосования кое-где и кое-когда не соответствуют действительности, имеют место. В связи с этим возникает вопрос о контроле. На практике этот вопрос зачастую решается путем пересчета голосов. Посмотрим, как этот вопрос может быть решен в предложенной схеме электронного голосования.

Итак, желающие проверить результаты электронного голосования имеют в своем распоряжении:

а) открытый ключ (m, e), опубликованный Центром на этапе организации голосования;

б) зашифрованные с помощью этого ключа бюллетени всех участников голосования f_i $(i = 1, 2, \ldots, N)$, опубликованные Центром после завершения этапа голосования;

в) числа r, p и R, опубликованные Центром в качестве результатов голосования.

Предлагаемая процедура проверки состоит из следующих шагов:

1) *восстанавливается число* $Q = (2^r) \cdot (3^p) \cdot R$ *и проводится процедура шифрования полученного числа с помощью открытого ключа* (m, e), *который известен проверяющим*;

2) *вычисляется число* F, *равное произведению всех зашифрованных* e*-бюллетеней, взятых из реестра, опубликованного Центром*;

3) *проверяется выполнение равенства*: $Q^e \bmod m = F$*; если это равенство имеет место, то число* Q *вычислено Центром правильно, если нет — Центр некорректно провел подсчет результатов*;

4) *остается убедиться, что при правильно подсчитанном* Q *числа* r *и* p *подсчитаны также верно. Для этого, учитывая, что контрольное число* R *представляет собой произведение простых чисел* $q_i \geqslant 5$, *достаточно проверить, что число* R *не делится ни на* 2, *ни на* 3.

Таким образом, обладая только открытой информацией и не нарушая тайны голосования, заинтересованные лица могут убедиться в том, что Центр корректно провел подсчет голосов, либо уличить его в фальсификации.

Заключительные замечания

Замечание. Приведенный выше протокол электронного голосования легко обобщить на случай выбора из более чем одной альтернативы, а также на случай так называемого мягкого, рейтингового голосования, когда избиратель может голосовать сразу за несколько кандидатур. В этих случаях естественным образом изменяются правила заполнения исходного е-бюллетеня. Например, в случае выбора из четырех кандидатур Центром устанавливается правило:

$$b_i = \begin{cases} 2 & \text{в случае голосования за кандидата № 1,} \\ 3 & \text{в случае голосования за кандидата № 2,} \\ 5 & \text{в случае голосования за кандидата № 3,} \\ 7 & \text{в случае голосования за кандидата № 4,} \\ 11 & \text{в случае голосования против всех;} \end{cases}$$

«затеняющие» множители в этом случае должны случайным образом выбираться из множества простых чисел, начиная с 13.

В случае же рейтингового голосования в е-бюллетень должно заноситься число, равное произведению простых чисел, соответствующих голосованию за выбранных кандидатов.

Приведенный выше протокол представляет собой лишь общую схему, демонстрирующую принципиальную возможность организации е-голосования, основанного на криптографических протоколах. Для превращения ее в реально применяемый программный продукт необходимо обеспечить криптографическую защиту всех ее компонентов.

Например, должны быть приняты специальные меры для защиты реестра избирателей (точно так же, как соответствующие меры принимаются для защиты списков избирателей в случае традиционного голосования). Должна быть защищена процедура проверки заполнения е-бюллетеня в соответствии с принятыми правилами (это аналог процедуры признания бюллетеня действительным в случае традиционного голосования) и т. д.

Тем не менее, е-голосование, основанное на применении криптографических протоколов, широко применяется во многих странах.

Правда, общенациональные электронные выборы пока не проводятся ни в одной стране. Некоторые эксперты говорят о возможности реализации подобных проектов в наиболее «компьютеризованных» странах в 2010—2015 годах.

Однако системы электронного голосования, например, в парламентах, работают во многих странах. При этом, если, во-первых, такая система построена в строгом соответствии с требованиями криптографической защиты информации, и, во-вторых, парламентарии представляют себе, как подобные системы работают, все подозрения в возможности фальсификации результатов электронного голосования не должны иметь места. Однако периодическое возобновление этой темы говорит о том, что либо первое условие, либо второе, либо оба вместе выполнены не полностью.

2.8. Задачи, решаемые только с использованием криптографических протоколов. Закрытый информационный обмен между двумя партнерами

Криптографические протоколы, часть которых приведена в предыдущем разделе, позволили найти решение многих новых задач, в том числе и тех которые раньше вообще считались неразрешимыми (например, задача о подбрасывании монеты по телефону). Более того, криптографические протоколы стали основой развития многих новых направлений информатики, из которых следует в первую очередь отметить *информационную безопасность*.

Однако сами по себе криптографические протоколы только предоставляют математическую основу решения проблемы. Они вполне достаточны для организации защищенного компьютерного обмена между двумя абонентами.

Пусть, например, два абонента хотят наладить между собой компьютерный обмен так, чтобы гарантировать его конфиденциальность, даже в случае вторжения активных противников. Криптографические протоколы дают им для решения подобной задачи разнообразные инструменты. Во-первых, они могут договориться об использовании в своем обмене процедуры взаимной *аутентификации*. Во-вторых, они могут *шифровать* свои послания. Они могут также договориться об использовании в своем обмене *электронной цифровой подписи*. При этом они могут в зависимости от взаимных договоренностей подписывать послания, содержащие открытые тексты либо уже зашифрованные. Они могут даже шифровать саму электронную подпись. И для реализации всего этого многообразия

возможных вариантов защиты партнерам достаточно завести каждому по три пары открытых и закрытых (секретных) ключей (по одной паре для каждой из операций: аутентификации, шифрования и подписи) и обменяться открытыми ключами.

Пусть, например, два партнера A и B желают зашифровать свой компьютерный обмен. Тогда они могут действовать следующим образом:

а) *A и B выбирают общий алгоритм шифрования E и соответствующий ему алгоритм расшифрования D.*

б) *Партнер A выбирает пару ключей k_1, k_2, открытый ключ k_1 он сообщает своему партнеру, секретный ключ k_2 остается известен только ему.*

в) *Аналогично, партнер B выбирает пару ключей q_1, q_2 и открытый ключ q_1 сообщает своему партнеру.*

г) *Сообщение x от A к B перед отправкой шифруется отправителем с использованием известного ему открытого ключа получателя q_1 и полученный зашифрованный текст $E_{q_1}(x)$ пересылается получателю.*

д) *Получатель B расшифровывает полученный текст $E_{q_1}(x)$ с использованием своего секретного ключа q_2, который известен только ему*: $D_{q_2}(E_{q_1}(x)) = x$.

е) *Аналогично сообщение от B к A сначала шифруется отправителем B с использованием открытого ключа получателя k_1, и зашифрованный текст $E_{k_1}(x)$ пересылается получателю A, который расшифровывает его с использованием своего закрытого ключа*: $D_{k_2}(E_{k_1}(x)) = x$.

Примерно по такой же схеме партнеры могут организовать процедуру электронной цифровой подписи для своих сообщений. При этом отправитель должен организовать вычисление ЭЦП своего сообщения с использованием своего закрытого ключа, а получатель организует проверку полученной подписи с использованием заранее ему известного открытого ключа отправителя.

И все будет прекрасно работать, и оба партнера будут довольны. По крайней мере до тех пор, пока один из партнеров не утратит свой секретный ключ или он не станет известен каким-либо нежелательным третьим лицам.

2.9. Криптографические протоколы и «честное слово»

Задачи, подобные тем, что решаются с помощью криптографических протоколов, существовали издавна с тех пор, как люди начали

взаимодействовать друг с другом и, к сожалению, заметили, что в этих взаимодействиях случаются обманы. Любопытно рассмотреть, как решались подобные задачи в «докриптографическую» эпоху. Рассмотрим несколько примеров взаимодействия не слишком доверяющих друг другу партнеров, не владеющих техникой криптографии.

Пример 4. Взаимодействуют два партнера (обозначим их A и B), каждый из которых владеет некоторой секретной информацией, представляющей интерес для другого партнера.

A: — *Я владею некоторой секретной информацией.*

B: — *И я тоже владею некоторой секретной информацией.*

A: — *Давай обменяемся.*

B: — *Давай, но только ты первый.*

A: — *Нет, ты первый, а то ты обманешь*: *узнаешь мой секрет, а свой не расскажешь.*

B: — *Так ведь и ты можешь сделать то же самое.*

A: — *Ну хорошо, я даю тебе* ***честное слово****, что расскажу свой секрет, как только узнаю твой.*

Пример 5. Взаимодействуют два партнера, один из которых утверждает, что владеет некоторой информацией и хочет убедить в этом партнера, но при этом не хочет, чтобы сама информация стала известна другому партнеру.

A: — *Я знаю, кого назначат новым начальником, только это секрет.*

B: — *Да не можешь ты этого знать.*

A: — *Нет, я в самом деле знаю.*

B: — *Ну тогда скажи мне.*

A: — *Не хочу тебе говорить, ты всем разболтаешь.*

B: — *Я даю тебе* ***честное слово****, что никому не скажу.*

Пример 6. Взаимодействуют два партнера, которые должны подписать уже согласованный контракт, но оба боятся, что партнер его обманет: получит его подпись, а от своей откажется.

A: — *Контракт готов, можно подписывать.*

B: — *Подписывайте, я после Вас.*

A: — *Почему же я первый. Давайте сначала Вы.*

B: — *Боитесь, что я обману. Хорошо, я даю* ***честное слово****, что подпишу после Вас.*

Подобные примеры можно продолжать.

Все приведенные «протоколы» обрываются на обещании одного из партнеров дать **честное слово**. Что произойдет потом, не

ясно, поскольку само понятие честного слова невозможно формализовать. Иногда ему можно верить, и тогда протокол завершается. Иногда честного слова недостаточно, и тогда «протокол» приходится существенно усложнять. Приведем пример подобного существенно усложненного протокола обмена денег на расписку в их получении, в котором участвуют два не доверяющих друг другу партнера.

«*— Хорошо, дайте же сюда деньги.*

— На что-ж деньги? У меня вот они в руке! Как только напишете расписку, в ту же минуту их возьмете.

— Да позвольте, как же мне писать расписку? Прежде нужно видеть деньги.

Чичиков выпустил из рук бумажки Собакевичу, который, приблизившись к столу и накрывши их пальцами левой руки, другою написал на лоскутке бумаги, что задаток двадцать пять рублей государственными ассигнациями за проданные души получен сполна...»

Этот пример заимствован из пятой главы «Мертвых душ».

С тех пор, с появлением криптографии ситуация существенно изменилась. Криптография позволяет делать подобные протоколы обмена не доверяющих друг другу партнеров по-настоящему честными без привлечения такого субъективного понятия, как «честное слово».

Ситуацию из примера 4 решает криптографический *протокол обмена секретами*. Ситуация примера 5 — это сфера применения криптографического понятия *доказательство с нулевым разглашением*. А героям Н. В. Гоголя пригодился бы *протокол подписания контракта*.

Ясно, что приведенные примеры можно наполнять самым разнообразным актуальным содержанием. А это означает, что концепция криптографических протоколов может с успехом применяться во всех тех областях формализованного человеческого общения, где необходимо обеспечить защиту информации и гарантию от возможной нечистоплотности партнера или атак противника.

3. Криптография и массовые информационные коммуникации

3.1. Какие задачи хотелось бы уметь решать. Массовые информационные коммуникации

В предыдущем разделе показано, как криптографические протоколы могут быть эффективно использованы для организации информационного обмена между двумя партнерами. Для организации подобного защищенного обмена партнерам достаточно предварительно обменяться своими открытыми ключами.

Однако вся сложность жизни в современном информационном сообществе не укладывается в схему обмена между двумя абонентами. Каждый участник подобного сообщества ведет обмены одновременно с сотнями или даже тысячами абонентов. Более того, невозможно заранее составить полный список таких абонентов.

Так, банк не может заранее знать всех своих клиентов, а мэр города, организуя свой электронный почтовый ящик для переписки с горожанами, не может заранее назвать всех жителей города, которые пожелают к нему обратиться. Между тем очевидно, что банк, прежде чем проводить какую-либо операцию со счетом клиента, должен по крайней мере провести процедуру его аутентификации, равно как мэр города должен убедиться, что пришедшее к нему по электронной почте письмо написано именно этим жителем города, поскольку он не желает иметь дело с анонимками.

Набор примеров подобного рода без труда может быть продолжен, поскольку они иллюстрируют ситуацию новой реальности, в которую вступило общество к концу XX столетия, — эпоху *массовых коммуникационных обменов*. При этом вполне естественным представляется требование обеспечить соответствующий уровень криптографической защищенности подобных обменов.

Как могут в этих условиях помочь криптографические протоколы? Мы видели в предыдущем разделе, как криптографические протоколы с успехом решают задачу информационного обмена между двумя партнерами. Однако попытка напрямую повторить такую же схему по принципу «каждый с каждым» в условиях массовых информационных коммуникаций вряд ли может быть признана удачной.

В самом деле, для организации такого обмена каждый его участник должен знать открытый ключ любого другого участника. Если же число участников обмена заведомо составляет многие тысячи и при этом к тому же общее их число заранее не фиксируется, а именно такую ситуацию мы имеем в условиях массовых информационных коммуникаций, — проблема взаимного обмена открытыми ключами и поддержания их в актуальном состоянии превращается в самостоятельную проблему, требующую отдельного решения.

Попытки решить эту проблему привели в последней четверти XX века к выделению на стыке криптографии и информационной безопасности новой специализированной дисциплины, которая получила название *инфраструктура открытых ключей*, и созданию новых видов организаций — *удостоверяющих центров*.

3.2. Инфраструктура открытых ключей и удостоверяющие центры

3.2.1. Предпосылки создания

В 1977 году американские математики У. Диффи и М. Э. Хеллман предложили перейти к системе, основанной на создании единого общедоступного хранилища всех открытых ключей. В несколько упрощенном и полемически заостренном варианте их предложение звучало примерно так: «Давайте издадим общедоступную книгу — справочник с открытыми ключами всех участников информационного обмена. Это полностью решит проблему хранения каждым участником большого количества ключей».

Поначалу это предложение произвело неоднозначный и даже шокирующий эффект. Однако авторы показали большое число преимуществ, которые обеспечивались его реализацией. Действительно, такая система полностью избавляла огромное число пользователей от трудоемкой работы по запоминанию, хранению и уничтожению своих открытых ключей, перекладывая эти заботы на администрацию книги-справочника. Кроме того, для пользователей значительно упрощалась процедура поиска открытых ключей других участников обмена. Для этого пользователю достаточно найти в книге-справочнике своего адресата и извлечь хранящийся в ней его открытый ключ.

Предложение открывало и другие возможности. Например, пользователи освобождаются от необходимости самим генерировать свои открытый и закрытый ключи, это можно было делать в централизо-

ванном порядке в едином центре. За пользователем остается только священная обязанность следить, чтобы ничего неприятного не случилось с его закрытым ключом.

Осознание этих преимуществ привело к тому, что уже в начале 80-х годов прошлого века появились первые практические реализации предложенной идеи. Ядро подобной реализации составили общедоступные централизованные хранилища открытых ключей, которые получили названия *удостоверяющих центров*. Кроме собственно функции *хранения* открытых ключей, удостоверяющие центры выполняют и еще много функций. В качестве важнейших из них укажем функции *регистрации* пользователей и *формирования ключей* пользователей. Что касается самого названия, то оно связано с тем, что с самого начала одной из важнейших функций удостоверяющих центров было обеспечение инфраструктуры электронной цифровой подписи в процессе информационного обмена. Поэтому еще одной функцией удостоверяющих центров стало *удостоверение полномочий* участников обмена.

Рассмотрим, в общих чертах, как удостоверяющий центр (УЦ) осуществляет выполнение функций, связанных с обеспечением ЭЦП всех участников информационного обмена.

3.2.2. Обеспечивающие алгоритмы

Напомним вкратце, какими криптографическими алгоритмами обеспечивается технология использования ЭЦП в электронном документообороте. Таких алгоритмов три:

— формирование ключей клиента (алгоритм I);

— формирование ЭЦП сообщения (алгоритм G);

— проверка ЭЦП сообщения (алгоритм V).

В основу их функционирования могут быть положены различные математические методы. Если абстрагироваться от конкретных математических методов, то эти алгоритмы можно определить следующим образом.

Формирование ключей клиента A

Алгоритм

$$I \to \left(\overline{\overline{K}}_A, \overline{K}_A\right) \tag{10}$$

запускается во время регистрации клиента A в удостоверяющем центре Y. В результате клиент получает:

— $\overline{\overline{K}}_A$ — закрытый ключ, известный только ему;

— $\overline{K}_A$ — соответствующий закрытому ключу $\overline{\overline{K}}_A$ открытый ключ, который становится известен всем участникам информационной системы;
— C_A — сертификат ЭЦП клиента A — совокупность записей, содержащих различные сведения о клиенте, в частности его ФИО, полномочия, срок действия сертификата и т.п. Эта информация может быть использована в алгоритмах обмена. В разных системах структура сертификата может иметь незначительные различия. Для дальнейшего изложения удобно считать, что сертификат ЭЦП клиента A в обязательном порядке содержит открытый ключ клиента $\overline{K}_A$ и открытый ключ удостоверяющего центра $\overline{K}_Y$.

Формирование ЭЦП сообщения

Алгоритм запускается клиентом A:

$$\left(m, \overline{\overline{K}}_A\right) \xrightarrow{G} (m, s), \tag{11}$$

где m — сообщение, которое клиент A подписывает перед отправкой; s — строка, представляющая собой ЭЦП сообщения.

Говорят, что s является ЭЦП сообщения m, которая сформирована с помощью закрытого ключа $\overline{\overline{K}}_A$. Если важно указать на факт подписи сообщения с помощью данного закрытого ключа, может быть использована другая запись действия алгоритма G:

$$\left(m, \overline{\overline{K}}_A\right) \xrightarrow{G} (m)\, s/\overline{\overline{K}}_A. \tag{11'}$$

Проверка ЭЦП сообщения

Алгоритм запускается клиентом, принявшим сообщение m, подписанное с помощью ЭЦП клиентом A:

$$(m, s, \overline{K}_A) \xrightarrow{V} \{0; 1\}. \tag{12}$$

Или, используя обозначения (11):

$$\left((m)\, s/\overline{\overline{K}}_A, \overline{K}_A\right) \xrightarrow{V} \{0; 1\}, \tag{12'}$$

где m — принятое сообщение; s — подпись сообщения m, сформированная клиентом A с помощью своего закрытого ключа $\overline{\overline{K}}_A$ с использованием алгоритма G; $\overline{K}_A$ — открытый ключ клиента A.

Результатом проверки является 0, если проверка выполнена, либо 1, если проверка не выполнена. Выполнение проверки означает одновременное выполнение двух условий:

— закрытый ключ $\overline{\overline{K}}_A$, использовавшийся при формировании подписи, соответствует открытому ключу $\overline{K}_A$, использовавшемуся при

проверке. Другими словами, пара ключей $\overline{\overline{K}}_A$ и $\overline{K}_A$ сформированы алгоритмом (10);

— сообщение m, поданное на вход алгоритма проверки (12), совпадает с сообщением m, которое было подписано с использованием алгоритма (11).

Соответственно, невыполнение проверки означает, что либо в алгоритме проверки (12) используется открытый ключ, не соответствующий закрытому ключу из алгоритма (11), либо сообщение m на входе алгоритма проверки не совпадает с сообщением, которое подписывалось при помощи алгоритма (11).

3.2.3. Обмен между клиентами одного удостоверяющего центра

Рассмотрим, как с использованием криптографических алгоритмов (10)—(12) решается задача пересылки сообщения, подписанного клиентом A, клиенту B, при условии, что оба эти клиента зарегистрированы в одном удостоверяющем центре УЦ.

Всю совокупность операций, обеспечивающих выполнение указанной задачи, можно разбить на пять групп:

— операции, выполняемые при создании удостоверяющего центра УЦ;
— операции, выполняемые при регистрации клиента A в удостоверяющем центре УЦ;
— операции, выполняемые при регистрации клиента B в том же удостоверяющем центре УЦ;
— операции, выполняемые клиентом A при отправке сообщения;
— операции, выполняемые клиентом B при получении сообщения.

Рассмотрим указанные операции по шагам.

3.2.3.1. Операции, выполняемые при создании удостоверяющего центра. (i) При создании удостоверяющего центра выбирается некоторая тройка криптографических алгоритмов (I, G, V), обеспечивающих технологию использования ЭЦП. (В дальнейшем все клиенты этого УЦ будут при регистрации обеспечиваться этими же алгоритмами.)

(ii) Запускается алгоритм формирования ключей I, который на выходе дает пару ключей УЦ

$$I \to \left(\overline{\overline{K}}_Y, \overline{K}_Y\right). \tag{13}$$

(iii) **Замечание.** Вообще говоря, для создания реально работающего УЦ должно быть выполнено большое количество условий, обеспе-

чивающих легитимность, в том числе и юридическую, его деятельности. Однако здесь и всюду далее мы будем останавливаться только на криптографическом обеспечении деятельности УЦ. А для этого достаточно при создании УЦ выполнить только одну операцию (13).

3.2.3.2. Операции, выполняемые при регистрации клиента. Для легитимного участия в процессе обмена информации любой участник должен пройти процедуру *регистрации* в УЦ, становясь тем самым его *клиентом*. При регистрации клиента A УЦ выполняет следующие операции.

(i) С помощью алгоритма I формируются открытый $\overline{K}_A$ и закрытый $\overline{\overline{K}}_A$ ключи клиента A.

(ii) Формируется *сертификат ЭЦП клиента* A — совокупность записей, содержащих различные сведения о клиенте, в частности, его ФИО, полномочия, срок действия сертификата и т. д. Эта информация может использоваться в алгоритмах обмена. В разных информационных системах структура и состав сертификата могут иметь незначительные различия. Для дальнейшего удобно считать, что сертификат ЭЦП клиента A в обязательном порядке содержит открытый ключ клиента $\overline{K}_A$ и открытый ключ удостоверяющего центра $\overline{K}_Y$. В дальнейшем сертификат клиента A обозначается C_A, или, если требуется подчеркнуть, какие ключи содержит сертификат клиента A, — $C_A(\overline{K}_A, \overline{K}_Y)$.

(iii) Сертификат клиента подписывается (разумеется, имеется в виду ЭЦП с выполнением алгоритма G) с использованием закрытого ключа УЦ, т. е. выполняется операция:

$$\left(C_A, \overline{\overline{K}}_Y\right) \xrightarrow{G} (C_A)\, s/\overline{\overline{K}}_Y.$$

(iv) УЦ пересылает клиенту A:

- его сертификат C_A, подписанный закрытым ключом удостоверяющего центра $(C_A)\, s/\overline{\overline{K}}_Y$ и содержащий открытый ключ удостоверяющего центра $\overline{K}_Y$, который проводил регистрацию клиента;
- его открытый и закрытый ключи $\overline{K}_A$ и $\overline{\overline{K}}_A$. При этом передаче закрытого ключа уделяется особое внимание. Физически он может быть передан по защищенному каналу связи, либо по обычному каналу в зашифрованном виде, может быть передан на «секретной» дискете или каким-либо другим способом.

(v) Клиент A, получив от УЦ перечисленные в предыдущем пункте параметры, проводит проверку правильности регистрации, для чего, используя полученный открытый ключ удостоверяющего центра $\overline{K}_Y$,

запускает алгоритм проверки V по схеме:

$$\left((C_A)\, s/\overline{\overline{K}}_Y, \overline{K}_Y\right) \xrightarrow{V} 0.$$

Если при этом результат проверки равен 0, регистрация проведена правильно.

3.2.3.3. Аналогично проводится регистрация клиента B в том же УЦ.

3.2.3.4. Операции, выполняемые клиентом A при отправке сообщения. Для отправки сообщения m, подписанного ЭЦП, клиент A выполняет следующие операции:

(i) Подписывает с использованием своего закрытого ключа сообщение m, то есть выполняет операцию

$$\left(m, \overline{\overline{K}}_A\right) \xrightarrow{G} (m)\, s/\overline{\overline{K}}_A.$$

(ii) Пересылает клиенту-получателю B сообщение m, подписанное своим закрытым ключом

$$(m)\, s/\overline{\overline{K}}_A \to B.$$

(iii) Пересылает B свой сертификат C_A, подписанный закрытым ключом УЦ, в котором клиент A зарегистрирован:

$$(C_A)\, s/\overline{\overline{K}}_Y \to B.$$

3.2.3.5. Операции, выполняемые клиентом B при получении сообщения. (i) При получении сообщения m от клиента A клиент B должен убедиться в выполнении трех условий:

— сообщение m подписано именно A;
— сообщение m дошло без искажений;
— клиент A зарегистрирован в том же УЦ, что и сам клиент B.

Мы уже отмечали, что выполнение первых двух условий обеспечивается самим механизмом использования ЭЦП. Что касается третьего условия, то его выполнение, в силу юридических функций удостоверяющих центров, обеспечивает подтверждение полномочий участников информационного обмена.

(ii) Для проверки выполнения этого условия клиент B проверяет подпись сертификата, полученного от клиента A, с использованием открытого ключа УЦ, который клиент B получил при своей регистрации. Обозначим временно этот ключ $\overline{K}_{Y_1}$ и рассмотрим результат выполнения проверки

$$\left((C_A)\, s/\overline{\overline{K}}_Y, \overline{K}_{Y_1}\right) \xrightarrow{V} \{0; 1\}.$$

Выполнение указанной проверки означает, что клиент-отправитель A и клиент-получатель B зарегистрированы в одном и том же УЦ.

Действительно, клиент B извлекает открытый ключ удостоверяющего центра $\overline{K}_{Y_1}$ из *своего собственного сертификата*, о котором заведомо известно, что он выдан удостоверяющим центром УЦ во время регистрации клиента B. Затем этот ключ используется для проверки правильности подписи полученного сертификата отправителя. Однако сертификат отправителя подписан закрытым ключом удостоверяющего центра, в котором зарегистрирован отправитель. И если проверка выполнена, то это означает, что ключи $\overline{K}_{Y_1}$ и $\overline{\overline{K}}_Y$ соответствуют друг другу, а значит, ключи $\overline{K}_{Y_1}$ и $\overline{K}_Y$ совпадают и, следовательно, клиент A и клиент B зарегистрированы в одном и том же удостоверяющем центре.

(iii) Убедившись, что клиент, отправивший сообщение, зарегистрирован в том же УЦ, клиент-получатель B может проверить правильность подписи под полученным сообщением. Для этого он извлекает открытый ключ отправителя $\overline{K}_A$ из полученного вместе с сообщением сертификата отправителя C_A и использует его для проверки подписи полученного сообщения:

$$\left((m)\, s/\overline{\overline{K}}_A, \overline{K}_A\right) \xrightarrow{V} 0.$$

Если результат проверки равен 0, сообщение подписано правильно и, следовательно, выполнены и первые два условия.

(iv) Описанное в предыдущем пункте извлечение открытого ключа отправителя из его сертификата, полученного из удостоверяющего центра, и есть конкретное использование идеи У. Диффи и М. Э. Хеллмана о создании единого общедоступного хранилища всех открытых ключей. Все открытые ключи централизованно хранятся в удостоверяющем центре и извлекаются оттуда пользователями по мере надобности.

3.2.3.6. Если клиенты зарегистрированы в разных УЦ. Рассмотрим в качестве полезного упражнения, что происходит, если приведенную выше схему обмена пробуют использовать клиенты, зарегистрированные в разных УЦ.

Предположим, что клиент A зарегистрирован в удостоверяющем центре Y, а клиент B — в другом удостоверяющем центре Y'. При регистрации они получают сертификаты, содержащие открытые ключи самого клиента и регистрирующего удостоверяющего центра и подписанные закрытыми ключами соответствующих удостоверяющих центров. Во время обмена сообщениями клиент A направляет клиенту B свой сертификат, подписанный закрытым ключом удостоверяющего центра Y, и сообщение m, подписанное с помощью своего закрытого ключа.

Получив сертификат клиента-отправителя, клиент B, как уже отмечалось, первым делом из собственного сертификата извлекает открытый ключ удостоверяющего центра, в котором он сам зарегистрирован. В данном случае это будет открытый ключ удостоверяющего центра Y' (обозначим его $\overline{K}_Y^1$).

Поскольку Y и Y' — два разных удостоверяющих центра, которые имеют две разные пары ключей, открытый ключ $\overline{K}_Y$ удостоверяющего центра Y не будет соответствовать закрытому ключу $\overline{\overline{K}}_Y^1$ другого удостоверяющего центра Y', и проверка, которую проводит клиент-получатель в соответствии с п. 3.2.3.5 (ii), не будет выполнена.

Вообще говоря, клиент B может проигнорировать это досадное обстоятельство и выполнить следующую операцию, извлекая из полученного сертификата C_A открытый ключ отправителя $\overline{K}_A$ (а он, собственно, только и нужен для проверки правильности подписи под полученным сообщением). Однако подобные действия являются недопустимыми, поскольку нарушают всю концепцию информационной безопасности, основанную на функциях удостоверяющих центров по подтверждению полномочий участников информационной системы.

3.2.4. Система взаимодействующих удостоверяющих центров

Таким образом, приведенная система успешно работает только для клиентов, зарегистрированных в одном удостоверяющем центре. Другими словами, она предполагает регистрацию всех клиентов (а в глобальной информационной системе таковых будут уже многие миллионы) в одном-единственном удостоверяющем центре. При этом, учитывая, что информационные обмены, вообще говоря, носят межгосударственный характер, рассматриваемый гипотетический удостоверяющий центр также должен носить транснациональный характер.

Разумеется, создание такого единого «всемирного удостоверяющего центра» невозможно по многим соображениям, в том числе технологического и организационного характера. Кроме того, правительства многих стран рассматривают создание и функционирование системы удостоверяющих центров в качестве важной компоненты национальной безопасности и предпочитают сохранять жесткий государственный контроль над ее деятельностью. Напротив, в других странах деятельность удостоверяющих центров рассматривается как чисто коммерческая, и удостоверяющие центры представляют собой достаточно прибыльные коммерческие предприятия, деятельность которых регулируется достаточно либеральным национальным за-

конодательством. И наконец, третья группа стран до сих пор не определилась в этом вопросе. К сожалению, к ним относится (по состоянию на конец 2009 года) и Россия.

Между тем создание единой информационной системы требует обеспечения режима взаимного признания сертификатов ключей ЭЦП всех участников системы, независимо от того, в каком удостоверяющем центре зарегистрированы конкретные ее участники. Отказ от этого требования делает невозможным создание единой среды информационного обмена, разрывая ее на большое число несвязанных между собою региональных и/или отраслевых сегментов. Такая ситуация создает массу неудобств для пользователей. В каком-то смысле она похожа на ситуацию, которая сложилась бы, если бы паспорта, удостоверяющие личность, действовали бы только на территории какого-то данного региона или признавались бы только данным ведомством. Можно представить себе проблемы, с которыми столкнулась бы в подобной ситуации личность, которая переезжает из одного региона в другой, да при этом еще и взаимодействует с различными ведомствами.

Конечно же, подобная ситуация крайне нежелательна, и для ее разрешения предложено несколько способов решения. Ниже приводится одна из возможных схем решения задачи взаимного признания сертификатов ключей ЭЦП, основанная на построении единой иерархической системы удостоверяющих центров.

3.2.5. Иерархическая система удостоверяющих центров

Рассмотрим снова ситуацию двух клиентов, зарегистрированных в разных удостоверяющих центрах, и введем следующие обозначения:

Y_1, Y_2 — два разных удостоверяющих центра,

A_1, A_2 — клиенты, зарегистрированные в удостоверяющих центрах Y_1 и Y_2 соответственно,

$\bar{A}_1, \bar{A}_2$ — открытые ключи соответствующих клиентов,

$\bar{Y}_1, \bar{Y}_2$ — открытые ключи соответствующих удостоверяющих центров,

$\bar{\bar{Y}}_1, \bar{\bar{Y}}_2$ — их закрытые ключи.

Для сертификатов клиентов A_1 и A_2, содержащих открытые ключи клиентов и их удостоверяющих центров и подписанных закрытым ключом соответствующего удостоверяющего центра, введем обозначения:

$$C(A_1) = (\bar{A}_1, \bar{Y}_1)s/\bar{\bar{Y}}_1 \quad \text{и} \quad C(A_2) = (\bar{A}_2, \bar{Y}_2)s/\bar{\bar{Y}}_2$$

соответственно.

Будем считать, что удостоверяющие центры Y_1 и Y_2 являются удостоверяющими центрами нижнего уровня. С целью взаимного признания ЭЦП всех клиентов удостоверяющих центров Y_1 и Y_2 создадим в информационной системе еще один удостоверяющий центр Y, который будет по отношению к Y_1 и Y_2 удостоверяющим центром более высокого уровня.

Потребуем далее, чтобы при своем создании удостоверяющие центры Y_1 и Y_2 прошли стандартную процедуру регистрации в удостоверяющем центре Y. При этом в соответствии с протоколом регистрации удостоверяющие центры Y_1 и Y_2 получат свои сертификаты $C(Y_1)$ и $C(Y_2)$, содержащие открытый ключ регистрирующего удостоверяющего центра Y и подписанные его закрытым ключом. В соответствии с принятой системой обозначений указанные сертификаты в дальнейшем обозначим:

$$C(Y_1) = (\overline{Y}_1, \overline{Y})s/\overline{\overline{Y}} \quad \text{и} \quad C(Y_2) = (\overline{Y}_2, \overline{Y})s/\overline{\overline{Y}}$$

соответственно.

Потребуем далее, чтобы при регистрации клиентов A_1 и A_2 в соответствующих удостоверяющих центрах Y_1 и Y_2 клиенты вместе с их собственными сертификатами $C(A_1)$ и $C(A_2)$ получали также соответствующие сертификаты регистрирующих их удостоверяющих центров $C(Y_1)$ и $C(Y_2)$, которые удостоверяющие центры получили во время своей регистрации в удостоверяющем центре Y.

Общая схема взаимодействия удостоверяющих центров Y_1, Y_2 и Y представлена на рис. 4.

Рассмотрим последовательность операций при обмене сообщениями между клиентом A_1 и клиентом A_2, действующими в информационной системе, образованной удостоверяющими центрами Y_1, Y_2 и Y. изображенной на рис. 4. Прежде всего, будем считать, что в процессе

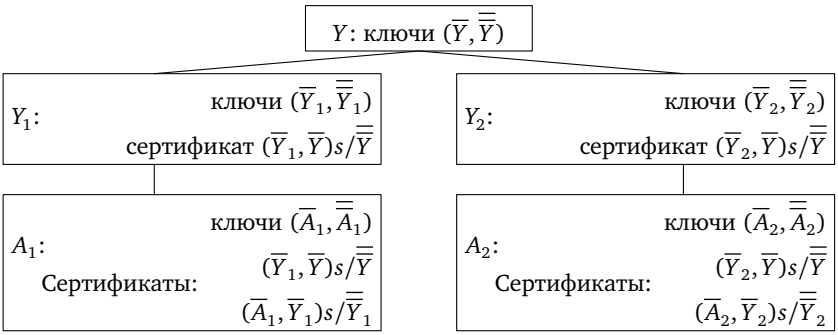

Рис. 4. Общая схема взаимодействия удостоверяющих центров Y_1, Y_2 и Y

выполнения стандартных процедур регистрации все субъекты информационной системы (удостоверяющие центры и клиенты) получили ключи и сертификаты, как это показано на рис. 4.

Тогда клиент-отправитель A_1 при отправке сообщения клиенту-получателю A_2 должен кроме трех операций, приведенных в разделе 3.2.3.4, выполнить еще одну — а именно, переслать клиенту A_2 сертификат своего удостоверяющего центра. Приведем всю последовательность операций клиента-отправителя A_1 в новых обозначениях:

$$(m, A_1) \xrightarrow{G} (m)\, s/A_1 \text{ — формирование ЭЦП сообщения } m; \quad \text{(A1.1)}$$

$$(m)\, s/A_1 \to A_2 \text{ — отправка подписанного сообщения;} \quad \text{(A1.2)}$$

$$\left(\bar{A}_1, \bar{Y}_1\right) s/\bar{\bar{Y}}_1 \to A_2 \text{ — отправка своего сертификата;} \quad \text{(A1.3)}$$

$$\left(\bar{Y}_1, \bar{Y}\right) s/\bar{\bar{Y}} \to A_2 \text{ — отправка сертификата своего удостоверяющего центра.} \quad \text{(A1.4)}$$

Клиенту-получателю необходимо выполнить дополнительные операции по сравнению с приведенными в разделе 3.2.3.5, поскольку проверка (A2.2) (см. ниже) в данной ситуации заведомо дает отрицательный результат, так как Y_1 и Y_2 — разные удостоверяющие центры. Вся последовательность операций клиента-получателя приведена ниже.

$$\left(\bar{A}_2, \bar{Y}_2\right) s/\bar{\bar{Y}}_2 \xrightarrow{E} \bar{Y}_2 \text{ — извлечение открытого ключа } Y_2 \text{ из собственного сертификата } A_2; \quad \text{(A2.1)}$$

$$\left(\left(\bar{A}_1, \bar{Y}_1\right) s/\bar{\bar{Y}}_1, \bar{Y}_2\right) \xrightarrow{V} 1 \text{ — проверка подписи присланного сертификата;} \quad \text{(A2.2)}$$

$$\left(\bar{Y}_2, \bar{Y}\right) s/\bar{\bar{Y}} \xrightarrow{E} \bar{Y} \text{ — извлечение открытого ключа } Y \text{ из имеющегося сертификата } Y_2; \quad \text{(A2.3)}$$

$$\left(\left(\bar{Y}_1, \bar{Y}\right) s/\bar{\bar{Y}}, \bar{Y}\right) \xrightarrow{V} 0 \text{ — проверка подписи присланного сертификата } Y_1; \quad \text{(A2.4)}$$

$$\left(\bar{Y}_1, \bar{Y}\right) s/\bar{\bar{Y}} \xrightarrow{E} \bar{Y}_1 \text{ — извлечение открытого ключа } Y_1 \text{ из присланного сертификата } Y_1; \quad \text{(A2.5)}$$

$$\left(\left(\bar{A}_1, \bar{Y}_1\right) s/\bar{\bar{Y}}_1, \bar{Y}_1\right) \xrightarrow{V} 0 \text{ — проверка подписи присланного сертификата } A_1; \quad \text{(A2.6)}$$

$$\left(\bar{A}_1, \bar{Y}_1\right) s/\bar{\bar{Y}}_1 \xrightarrow{E} \bar{A}_1 \text{ — извлечение открытого ключа } A_1 \text{ из присланного сертификата } A_1; \quad \text{(A2.7)}$$

$$\left((m)\, s/\bar{\bar{A}}_1, \bar{A}_1\right) \xrightarrow{V} 0 \text{ — проверка подписи сообщения } m. \quad \text{(A2.8)}$$

Покажем, что последовательность операций (А2.1)—(А2.8) удовлетворяет всем требованиям информационной безопасности, связанным с функциями удостоверяющих центров по подтверждению полномочий всех участников информационной схемы.

Действительно, невыполнение проверки (А2.2) означает, что клиент-отправитель не зарегистрирован в том же удостоверяющем центре, в котором зарегистрирован клиент-получатель.

Убедившись в этом, клиент-получатель с помощью пары операций (А2.3)—(А2.4) приходит к выводу, что удостоверяющие центры Y_1 (где зарегистрирован клиент-отправитель) и Y_2 (где зарегистрирован клиент-получатель) оба зарегистрированы в одном удостоверяющем центре Y. Тем самым «чужой» с точки зрения клиента-получателя удостоверяющий центр Y_1 становится легитимным участником информационной схемы. И клиент-получатель может признать присланный ему отправителем по операции (А1.4) сертификат этого удостоверяющего центра.

Следующей парой операций (А2.5)—(А2.6) клиент-получатель убеждается, что присланный ему по операции (А1.3) сертификат действительно является сертификатом клиента A_1, зарегистрированного в удостоверяющем центре Y_1.

Поэтому из присланного сертификата клиент-получатель может извлечь открытый ключ клиента $\bar{A}_1$ и использовать его для проверки ЭЦП сообщения m. Что он и делает операциями (А2.7)—(А2.8).

Рассмотренная простейшая схема, содержащая лишь два удостоверяющих центра нижнего уровня, естественным образом обобщается на случай произвольного числа таких удостоверяющих центров.

Действительно, пусть в схеме уже задействовано $n-1$ удостоверяющих центров нижнего уровня. Тогда при добавлении к схеме еще одного удостоверяющего центра нижнего уровня Y_n необходимо выполнить следующие операции:

1. Зарегистрировать Y_n в удостоверяющем центре Y с выдачей ему стандартного сертификата $C(Y_n) = (\bar{Y}_n, \bar{Y})s/\bar{\bar{Y}}$.
2. При регистрации в Y_n клиентов выдавать каждому из них пару сертификатов: $C(A_n) = (\bar{A}_n, \bar{Y}_n)s/\bar{\bar{Y}}_n$ — собственный сертификат клиента A_n и $C(Y_n) = (\bar{Y}_n, \bar{Y})s/\bar{\bar{Y}}$ — сертификат удостоверяющего центра Y_n.

Выполнение действий 1 и 2, первое из которых производится только один раз при регистрации добавляемого в схему удостоверяющего центра, позволяет включить в единую информационную схему всех клиентов всех удостоверяющих центров $Y_1, Y_2, \ldots, Y_n$.

Важным преимуществом указанной схемы является то обстоятельство, что при добавлении в схему нового удостоверяющего центра вообще не нужно знать, сколько удостоверяющих центров уже задействовано в схеме. Это выгодно отличает ее от схем типа попарных обменов сертификатами между всеми удостоверяющими центрами, действующими в схеме.

Еще одним преимуществом указанной схемы является отсутствие необходимости каких-либо обменов информацией между клиентом-получателем и удостоверяющими центрами, задействованными в информационной системе.

Приведенная схема обобщается также и на произвольную систему удостоверяющих центров, устроенную иерархическим образом. В этом случае произвольный набор удостоверяющих центров, объединенных в иерархическую информационную систему, естественно изображать в виде *графа*, то есть множества узлов (вершин), причем некоторые из них соединены связями — ребрами графа. Узлами графа будут являться удостоверяющие центры. Два удостоверяющих центра Y_i и Y_k соединены ребром графа, если Y_i зарегистрирован в удостоверяющем центре Y_k. Поскольку каждый удостоверяющий центр может быть зарегистрирован только в одном удостоверяющем центре более высокого уровня, граф, их изображающий, является *деревом* (быть может, несвязным). Один удостоверяющий центр самого высокого уровня, не зарегистрированный ни в каком другом, называется *корневым* удостоверяющим центром.

В любом удостоверяющем центре системы могут быть также зарегистрированы пользователи-клиенты.

При получении сообщения от клиента A_i, зарегистрированного в удостоверяющем центре Y_i, клиент-получатель A_j, зарегистрированный в удостоверяющем центре Y_j, должен убедиться, что отправитель A_i зарегистрирован в одном из удостоверяющих центров, входящих в общую систему с удостоверяющим центром Y_j, в котором зарегистрирован получатель A_j.

Для решения этой задачи может быть использован механизм сертификатов удостоверяющих центров.

Будем по-прежнему называть сертификатом удостоверяющего центра Y_i набор записей, содержащий открытый ключ самого удостоверяющего центра $\overline{Y}_i$, открытый ключ удостоверяющего центра $\overline{Y}_k$, в котором зарегистрирован Y_i, и подписанный закрытым ключом удостоверяющего центра $\overline{\overline{Y}}_k$. Будем обозначать сертификат удостоверяющего центра Y_i через

$$C(Y_i) = \left(\overline{Y}_i, \overline{Y}_k\right)s/\overline{\overline{Y}}_k.$$

Пусть $C(Y_i) = (\overline{Y}_i, \overline{Y}_k)s/\overline{\overline{Y}}_k$, $C(Y_j) = (\overline{Y}_j, \overline{Y}_l)s/\overline{\overline{Y}}_l$ — сертификаты двух удостоверяющих центров: Y_i — зарегистрированного в Y_k и Y_j — зарегистрированного в Y_l.

Введем две формальных операции с сертификатами.

а) *Соответствие сертификатов*

Будем говорить, что сертификаты $C(Y_i)$ и $C(Y_j)$ соответствуют друг другу, если выполнено условие

$$\left((\overline{Y}_i, \overline{Y}_k)s/\overline{\overline{Y}}_k, \overline{Y}_l\right) \xrightarrow{V} 0. \tag{14}$$

Это условие может быть выполнено только в том случае, если $\overline{Y}_l = \overline{Y}_k$, а это в свою очередь означает, что оба удостоверяющих центра Y_i и Y_j зарегистрированы в одном удостоверяющем центре, т. е. удостоверяющий центр Y_k совпадает с удостоверяющим центром Y_l.

При этом, если $C(Y_i)$ соответствует $C(Y_j)$, то и $C(Y_k)$ соответствует $C(Y_l)$, т. е. если выполнено (14), то выполняется и симметричное условие

$$\left((\overline{Y}_j, \overline{Y}_l)s/\overline{\overline{Y}}_l, \overline{Y}_k\right) \xrightarrow{V} 0. \tag{14'}$$

б) *Проверка регистрации*

Пусть по-прежнему $C(Y_i)$ и $C(Y_j)$ — сертификаты двух удостоверяющих центров. Для того чтобы проверить, действительно ли удостоверяющий центр Y_i зарегистрирован в Y_j, достаточно проверить выполнение условия

$$\left((\overline{Y}_i, \overline{Y}_k)s/\overline{\overline{Y}}_k, Y_j\right) \xrightarrow{V} 0. \tag{15}$$

Действительно, проверка (15) может быть выполнена только в случае, если $\overline{\overline{Y}}_k$ соответствует $\overline{Y}_j$, а это означает, что $Y_k = Y_j$ и сертификат $C(Y_i)$ может быть переписан в виде $(\overline{Y}_i, \overline{Y}_j)s/\overline{\overline{Y}}_j$, а это, в свою очередь, означает, что удостоверяющий центр Y_i зарегистрирован в удостоверяющем центре Y_j.

Теперь может быть сформулирован *алгоритм определения открытого ключа отправителя*.

Пусть A — клиент-отправитель, зарегистрированный в удостоверяющем центре Y_0, а A' — клиент-получатель, зарегистрированный в удостоверяющем центре Y_0' (см. рис. 5).

Здесь

$$Y_0 \to Y_1 \to \ldots \to Y_{n-1} \to Y_n \text{ — цепочка регистрации клиента } A.$$

$$Y_0' \to Y_1' \to \ldots \to Y_{m-1}' \to Y_m' = Y_{n-1} \to Y_n \text{ — цепочка регистрации клиента } A'.$$

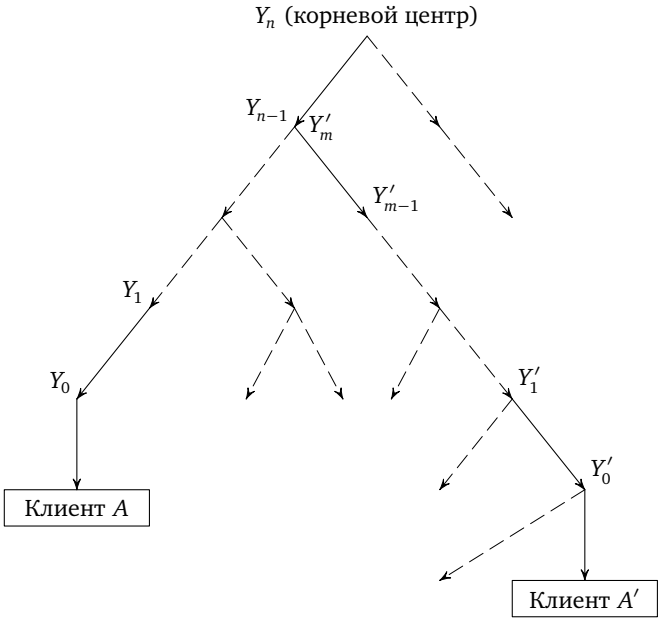

Рис. 5. Цепочки регистрации клиентов

В соответствии с соглашением при своей регистрации в Y_0 клиент A получает свой собственный сертификат $C(A)$ и всю цепочку сертификатов удостоверяющих центров вышестоящих уровней, начиная с Y_0:

$$\begin{aligned} C(A) &= (\bar{A}, \bar{Y}_0)s/\bar{\bar{Y}}_0, \\ C(Y_0) &= (\bar{Y}_0, \bar{Y}_1)s/\bar{\bar{Y}}_1, \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ C(Y_{n-1}) &= (\bar{Y}_{n-1}, \bar{Y}_n)s/\bar{\bar{Y}}_n. \end{aligned} \tag{16}$$

Аналогично, клиент-получатель A' при своей регистрации в удостоверяющем центре Y'_0 получает цепочку сертификатов

$$\begin{aligned} C(A') &= (\bar{A}', \bar{Y}'_0)s/\bar{\bar{Y}}'_0 \\ C(Y'_0) &= (\bar{Y}'_0, \bar{Y}'_1)s/\bar{\bar{Y}}'_1 \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ C(Y'_{m-1}) &= (\bar{Y}'_{m-1}, \bar{Y}'_m)s/\bar{\bar{Y}}'_m \quad \text{и, учитывая, что } Y'_m = Y_{n-1}, \\ C(Y_{n-1}) &= (\bar{Y}_{n-1}, \bar{Y}_n)s/\bar{\bar{Y}}_n. \end{aligned} \tag{16'}$$

Цепочки (16) и (16′) могут быть, вообще говоря, разной длины.

Во время обмена информацией клиент A отправляет клиенту A' сообщение m, подписанное своим закрытым ключом $(m)\,s/\overline{\overline{A}}$, и всю цепочку сертификатов $C(A), C(Y_0), C(Y_1), \ldots, C(Y_{n-1})$.

Чтобы проверить правильность принятого сообщения m, клиент-получатель должен иметь открытый ключ клиента-отправителя $\overline{A}$, который ему прислан в составе сертификата $C(A)$. Для того чтобы убедиться в легитимности его использования, клиенту A' необходимо проверить, что удостоверяющие центры Y_0 и Y_0' находятся в одной информационной системе. Другими словами, что в цепочках удостоверяющих центров, имеющих сертификаты (16) и (16′), имеется общий узел. Выполнение этого условия проверяется путем попарного сравнения сертификатов из цепочек (16) и (16′) с помощью операции (14).

Если ни для одной из возможных пар сертификатов $\left(C(Y_p), C(Y_r')\right)$, где $C(Y_p)$ — сертификат из цепочки (16), $C(Y_r')$ — сертификат из цепочки (16′), операция соответствия не выполняется, то это означает, что удостоверяющий центр Y_0 не находится в одной информационной системе с удостоверяющим центром Y_0'.

Если же для какой-то пары сертификатов $\left(C(Y_p), C(Y_r')\right)$ операция соответствия (14) выполнена, то это означает, что удостоверяющий центр Y_p из цепочки (16) и Y_r' из цепочки (16′) зарегистрированы в одном удостоверяющем центре.

Для завершения процедуры проверки легитимности использования открытого ключа клиента-отправителя A, присланного в составе сертификата $C(A)$, достаточно убедиться, что все удостоверяющие центры в цепочке (16), начиная с Y_p и включая Y_0, легитимно зарегистрированы.

Для этого необходимо последовательно проверить, что для всех сертификатов, начиная с $C(Y_p)$ до $C(Y_0)$ включительно, выполняется условие

$$\left((\overline{\overline{Y}}_k, \overline{\overline{Y}}_{k+1})s/\overline{\overline{\overline{Y}}}_{k+1}, \overline{\overline{Y}}_{k+1}\right) \xrightarrow{V} 0, \qquad (17)$$

где k пробегает все значения от p до 0. Если хотя бы одна из подобных проверок (17) не выполнена, цепочку регистрации Y_0 нельзя считать легитимной.

Если же выполнены все проверки (17), клиент-получатель может из присланного ему сертификата $C(A)$ извлечь открытый ключ отправителя $\overline{A}$ и проверить с его помощью правильность подписи сообщения m с использованием стандартного алгоритма проверки ЭЦП:

$$\left((m)\,s/\overline{\overline{A}}, \overline{A}\right) \xrightarrow{V} \{0; 1\}.$$

3.2.6. Общий случай

После прочтения предыдущего раздела может сложиться впечатление, что иерархическая система удостоверяющих центров решает все задачи по созданию единого пространства ЭЦП. Действительно, к такой системе может быть легко подсоединен новый пользователь. Для этого ему достаточно зарегистрироваться в любом уже существующем в системе удостоверяющем центре. В процессе регистрации он получит всю цепочку сертификатов вплоть до сертификата корневого удостоверяющего центра и с помощью описанных алгоритмов сможет осуществлять обмен с любым пользователем, зарегистрированным в любом удостоверяющем центре. Более того, к подобной иерархически устроенной системе достаточно легко присоединить новый удостоверяющий центр, который достаточно зарегистрировать в любом удостоверяющем центре, уже существующем в системе. При этом, поскольку общее количество удостоверяющих центров в подобной системе принципиально не ограничивается, она может обслуживать практически любое число пользователей.

Однако даже такая мощная система не способна решить все задачи. Прежде всего, особенность подобной системы (как и всякой иерархической системы вообще) в том, что она должна строиться сверху, по единому замыслу. В дальнейшем система может наращиваться, но изменение возникшей при ее создании структуры потребует огромных усилий. Действительно, как уже отмечалось, к существующей структуре легко может быть добавлен новый удостоверяющий центр, поскольку это не требует изменения цепочек сертификатов уже зарегистрированных в системе пользователей. Однако попытка «врезать» в существующую систему новый узел немедленно приведет к изменению цепочек сертификатов всех пользователей, расположенных в системе на нижележащих уровнях, т. е. потребует проведения серьезной и дорогостоящей реструктуризации всей системы.

Но даже не этот недостаток является самым главным. Как это ни покажется странным, сами математики создали наиболее серьезную проблему на пути создания единого информационного пространства. Дело в том, что рассмотренная выше иерархическая система работает только в том случае, если для каждого ее пользователя используются единые версии криптографических алгоритмов формирования ключей, генерации и проверки ЭЦП. Между тем, как уже неоднократно указывалось, таких алгоритмов математики придумали великое множество. Более того, даже один и тот же математический алгоритм допускает различные версии своей программной реализации. При этом

совершенно ясно, что если два удостоверяющих центра использовали для своих клиентов разные криптографические алгоритмы или даже разные версии одного и того же алгоритма, ничего хорошего из попытки обмена между этими клиентами не получится.

Здесь нужно обратить внимание на то обстоятельство, что единого информационного пространства по единому глобальному замыслу никто не создавал. Просто по мере осознания преимуществ, которые дает развитие компьютерных коммуникаций, наиболее «продвинутые» коммерческие структуры, правительственные ведомства, территориальные администрации и другие субъекты подобной деятельности в разных странах и независимо друг от друга для своих нужд стали создавать свои «сегменты» подобного пространства. Ничего удивительного нет в том, что в этих условиях каждый из них использовал свои варианты криптографических алгоритмов, создавал свои правила, инструкции и регламенты работы в своем «сегменте».

Когда же в результате накопления опыта встал вопрос об обеспечении взаимодействия между отдельными уже работающими сегментами и появились международные стандарты и национальное законодательство, в большинстве стран посчитали нецелесообразным «причесывать» уже действующие системы под единые жесткие требования иерархической централизованной структуры, предоставив каждому владельцу возможность самому выбирать политику и технологию использования ЭЦП, оставаясь при этом в рамках общих требований к этой политике. А для обеспечения взаимодействия между различными субъектами информационного обмена, реализующими в своих сегментах различную технологию, в подобной распределенной системе были созданы многочисленные технологические продукты, такие как *кросс-сертификаты, электронный нотариат, сертификационные мосты* и др., обсуждение которых выходит за рамки настоящих лекций.

3.3. Промежуточные итоги

Несмотря на огромные усилия и всеобщее понимание огромных преимуществ, которые может дать комплексное внедрение информационных технологий в государственном управлении и бизнесе, продвижения на этом направлении даются с огромным трудом. Тем не менее, число людей, использующих в своей практической деятельности электронную среду взаимодействия, непрерывно увеличивается. Быть может, стоит коротко перечислить основные условия, которые

обеспечивают современные средства электронного обмена, для любого его участника. Главных из таких условий по крайней мере четыре:

1) *тот, кто посылает информацию, и тот, кто ее принимает, должны быть идентифицированы, т. е. каждая сторона, участвующая в обмене, должна знать, кто отправил информацию и что эта информация отправлена по нужному адресу и получена нужным лицом*;
2) *стороны должны быть уверены, что переданная информация не подверглась искажению*;
3) *в случае служебного или судебного разбирательства можно юридически точно определить, что отправляющий информацию ее действительно отправил, а получающий — действительно получил*;
4) *информация должна быть защищена от несанкционированного доступа в случае, если требуется выполнение условия конфиденциальности.*

Ясно, что выполнение указанных условий было бы невозможным без достижений криптографии и того ее раздела, который развивает концепцию *криптографических протоколов*.

Другой раздел криптографии, развитие которого обеспечило выполнение указанных условий, — это концепция *открытого ключа пользователя*, разработка которой позволила организовать эффективное применение ЭЦП и обеспечить условия для участия в электронном документообороте всем желающим (населению, коммерческим структурам, неправительственным учреждениям, правительственным ведомствам и т. д.). Правда, *инфраструктура открытых ключей*, для которой обычно используется обозначение PKI (от английского Public Key Infrastructure), — это не только программное и криптографическое обеспечение. PKI — это именно инфраструктура, представляющая собой сочетание программных продуктов, услуг, средств, политики, регламентов, соглашений и людей, которые во взаимодействии поддерживают связи в публичных компьютерных сетях.

С другой стороны, организация юридически значимого электронного документооборота населения и коммерческих структур с правительственными ведомствами лежит в основе проектов создания *электронного правительства*, разработки которого включены в программы работы правительств многих стран [4].

Но это уже совсем другая тема.

Литература

1. *Музыкантский А. И., Фурин В. В. и др.* Примеры применения математических моделей к решению социально-гуманитарных проблем. М.: Макс Пресс, 2005.
2. Введение в криптографию / Под общ. ред. В. В. Ященко. СПб.: Питер, 2001.
3. *Алферов А. П., Зубов А. Ю., Кузмин А. С., Черемушкин А. В.* Основы криптографии: Учебное пособие. М.: Гелиос АРВ, 2001.
4. *Леонтьев К. Б.* Комментарий к Федеральному закону «Об электронной цифровой печати». М.: ТК Филби, 2003.
5. *Черчхаус Р.* Коды и шифры, Юлий Цезарь, «Энигма» и Интернет. М.: Весь мир, 2005.
6. *Бауэр Ф.* Расшифрованные секреты. Методы и принципы криптологии. М.: Мир, 2007.

Александр Ильич Музыкантский
Виктор Владимирович Фурин

ЛЕКЦИИ ПО КРИПТОГРАФИИ

Подписано в печать 27.02.2013 г. Формат $60 \times 90\ 1/16$. Бумага офсетная.
Печать офсетная. Печ. л. 4,25 Тираж 1000 экз. Заказ № .

Издательство Московского центра
непрерывного математического образования.
119002, Москва, Большой Власьевский пер., д. 11. Тел. (499) 241–74–83

Отпечатано с готовых диапозитивов в ООО «Принт Сервис Групп».
105187, Москва, ул. Борисовская, д. 14.

Книги издательства МЦНМО можно приобрести в магазине «Математическая книга»,
Большой Власьевский пер., д. 11. Тел. (499) 241–72–85. E-mail: biblio@mccme.ru