Ж.Брассар

СОВРЕМЕННАЯ КРИПТОЛОГИЯ

Руководство

Москва, Издательско-полиграфическая фирма ПОЛИМЕД 1999

Эта сравнительно небольшая книжка отражает многочисленные, как теоретические, так и практические аспекты современной криптологии, которые уже стали или становятся частью повседневной жизни. Информационно очень ёмкая, она написана на концептуальном уровне, неформально и с большим мастерством.

Автор книги — известнейший специалист в области криптологии, член совета директоров Международной ассоциации криптологических исследований, главный редактор журнала «Journal of Cryptology», один из основоположников квантовой криптографии и соавтор открытия квантовой телепортации профессор Монреальского университета Жиль Брассар.

Оригинальная английская версия книги была опубликована в серии «Lecture Notes in Computer Science», где печатаются труды основных ежегодных конференций по криптологии — CRYPTO и EUROCRYPT.

Для широкого круга читателей, интересующихся проблемами криптографии и её применений.

Оглавление

Именной указатель

Предисловие переводчика

«Великая держава — это страна, которая владеет ядерными технологиями, ракетными технологиями и криптографией»

— *Дэвид Кан* [230]

Криптология — это область знаний, изучающая тайнопись (криптографию) и методы её раскрытия (криптоанализ), которая по меткому выражению Рональда Ривеста, профессора MIT — Массачусетского технологического института — и одного из авторов знаменитой криптосистемы RSA, является повивальной бабкой всей «computer science» вообще [306].

Построение современной криптологии как науки основывается на совокупности фундаментальных понятий и фактов математики, физики, теории информации и сложности вычислений, природно очень сложных для всестороннего и глубокого осмысления даже профессионалами. Однако, несмотря на органически присущую ей сложность, многие теоретические достижения криптологии, сейчас широко используются в нашей насыщенной информационными технологиями жизни, например: в пластиковых smart-картах, в электронной почте, в системах банковских платежей, при электронной торговле через Internet, в системах электронного документооборота, при ведении баз данных, системах электронного голосования и др.[1] Подобное соотношение об-

[1] **Без криптографии принципиально нельзя обойтись при защите данных, передаваемых по открытым электронным каналам связи, а также там, где необходимо подтверждать целостность электронной информации или доказывать ее авторство.**

щей внутренней сложности и практической применимости для теоретической науки, по-видимому, уникально.

С другой стороны, именно насущная потребность и широкий спектр возможностей практического использования стимулируют теоретические и прикладные исследования не только в этой области знаний и в соответствующих областях математики, физики, теории информации и теории вычислений, но также настойчиво побуждают к совершенствованию юридических и правовых норм и механизмов на государственном, международном и общечеловеческом уровне, что зачастую порождает открытые обсуждения в печати или жаркие дебаты в парламентах разных стран, и даже заставляют обсуждать связанные с этим вопросы на совещаниях глав великих держав.

Тем ценнее эта, небольшая по объему, но информационно очень емкая, книжка. Она написана неформально, очень понятно и с большим мастерством[2], на концептуальном уровне отражая многие теоретические и практические аспекты современной криптографии. В меньшей степени в ней представлены вопросы криптоанализа, что вполне объяснимо, так как для этого необходимы некоторые специальные знания, которых выбранный автором стиль изложения не предполагает[3]. Тем не менее, для полного понимания всего излагаемого материала от читателя требуется математическая и компьютерная культура. И наоборот, затрагивающая фундаментальные проблемы природы, такую культуру прививает читателю и сама книга.

Автор книги — известнейший специалист в области криптологии профессор Монреальского университета Жиль Брассар является директором Международной ассоциации криптографических исследований (`www.iacr.org`). Он известен еще как один из основоположников и ведущих исследователей в области квантовой криптографии, которой посвящена последняя седьмая глава книги. Брассар также — один из шести авторов сенсационно-

[2] Как не вспомнить здесь закономерность, подмеченную в свое время выдающимся советским математиком академиком Л.С. Понтрягиным о том, что толщина написанной книжки обратно пропорциональна квадрату труда, затраченного автором при ее написании.

[3] Эту часть современной криптологии на соответствующем уровне отлично освещает книга Коблица [237], вышедшая уже вторым изданием.

го открытия квантовой телепортации[4], идея которой восходит к Эйнштейну [162].

Помимо оригинальной версии «Modern Cryptology» [71], вышедшей на английском языке в издательстве «Springer-Verlag» в знаменитой серии «Lecture Notes in Computer Science», книга была издана также на французском и итальянском языках.

Настоящий перевод был выполнен с английского издания книги с учетом, по возможности, изменений и дополнений, сделанных автором к ее французскому изданию.

Оригинал-макет и стилевой файл книги подготовлены мною в пакете LaTeX [5].

> Фактически оригинал-макет был подготовлен осенью 1995 года, но из-за отсутствия средств, книжка так и не вышла в печать. За это время на русском языке было издано несколько подобных книг по криптографии, из которых следует отметить три:
>
> > Саломаа А., *Криптография с открытым ключом*, М, Мир, 1996, 318 стр.
> >
> > Анохин М. И., Варнавский Н. П., Сидельников В. М., Ященко В. В., *Криптография в банковском деле* (методические материалы), М, МИФИ, 1997, 274 стр.
> >
> > *Введение в криптографию*, под общей редакцией В. В. Ященко, М, МЦНМО-ЧеРо, 1998, 272 стр.
>
> Тем не менее все они не перекрывают содержания написанной в едином стиле тоненькой «Современной криптологии» и не могут сравниться с ней по мастерству изложения концептуальных основ науки, в которой автор этой книги сам давно и активно работает.

Нет слов, чтобы выразить благодарность автору Жилю Брассару. Он не только безвозмездно предоставил мне право на перевод книги на русский язык, но и оказывал внимание, поддержку и даже техническую помощь в процессе ее перевода. При этом хотелось бы сказать спасибо зам. зав. редакцией литературы по

[4] Информацию об этом открытии и о том, что такое квантовая телепортация, можно найти на `www.research.ibm.com/quantuminfo/teleportation`.

[5] В связи с этим хотелось бы выразить благодарность И. А. Маховой, Ольге Лапко, Сергею Львовскому и Александру Шеню за приобщение к TeX'у.

математическим наукам издательства «Мир» А. С. Попову, который уговорил меня, чтобы я сам подготовил уже имеющуюся у меня рукопись перевода книги, предназначавшуюся для друзей и коллег, к ее официальному изданию.

Мне хочется поблагодарить также К. П. Игнатенко, В. П. Панова и Анатолия Лебедева, помогавших мне при работе над переводом и оформлением оригинал-макета, но особенно я благодарен моему другу Андрею Карасеву, помощь которого была неоценимой. Отдельной благодарности за поддержку и долготерпение моей многодневной ежевечерней работы за компьютером заслуживают моя жена Лена и дети, Ира и Вера. Я же, помня об очаровании, испытанном мною в свое время при чтении оригинальной версии книги, только считал своим долгом сделать соответствующим настоящий перевод. Насколько это удалось — судить Вам.

Эта книга предназначена для тех, кто хочет получить представление или пополнить свои знания о необычайно глубокой теории и весьма разносторонних применениях криптологии — парадоксальной в этом смысле области знаний, поразительно бурно развивающейся на стыке современной науки и жизни. Книга написана очень доходчиво и высокопрофессионально. Купите ее, прочитайте, не пожалеете.

Михаил Ветчинин
Москва, ноябрь, 1999

Dédié à Manuel Blum,
Ami inestimable[1]

Предисловие автора

Источником предлагаемой вашему вниманию книги послужила рукопись, подготовленная мной к $3\frac{1}{2}$-часовому учебному семинару, который меня пригласили провести на *29-ой Компьютерной конференции (CompCon)*, очередной из тех, что ежегодно организуются Институтом инженеров по электротехнике и электронике (IEEE), состоявшейся в Сан-Франциско с 27 февраля по 2 марта 1987 года. В процессе подготовки книги мне, правда, пришлось существенно дополнить свои записи к семинару новым материалом, включив в рассмотрение еще несколько тем. Основная цель, которую я при этом перед собой ставил, состояла в том, чтобы обеспечить замкнутый краткий обзор недавних криптографических достижений в той форме, которая может быть понятна даже читателям, до этого совершенно ничего не знавшим о криптологии. Таким образом, предлагаемая вашему вниманию книга может использоваться как предварительный материал для чтения в качестве вводного курса. Тем не менее, поскольку в книге излагается достаточно современный материал, то она может представлять интерес даже для специалиста. Кроме того, в ней содержится обширная библиография.

Инициатором моего приглашения провести упомянутый в предыдущем абзаце *CompCon'-овский* семинар был Рассел Брэнд. Впоследствии Дэвид Кан и Рональд Л. Ривест поддержали меня в том, чтобы я «что-нибудь сделал» с рукописью, которая была подготовлена к этому семинару. Идея относительно ее переработки с целью последующей публикации в виде монографии в

[1] **Мануэлю Блюму, Другу бесценному посвящается (фр.).**

серии книг *Lecture Notes in Computer Science*, выходящих в издательстве Springer-Verlag, была предложена Линной Монц.

Многие, не считаясь со временем, великодушно помогали мне при работе над этой книгой. Бонни Бергер, Дэвид Чаум, Уво Десмедт, Беннетт Фокс, Сильвио Микэли и Жан-Жак Кискате обеспечили чрезвычайно важную для меня обратную связь, благодаря которой рукопись подверглась нескольким пересмотрам и исправлениям. С большим удовольствием выражаю также благодарность тем, кто вместе со мной работал над различными частями этой книги: Клоду Крепеу и Клоду Готье — за совместную работу над § 5.3, Дэвиду Чауму и Клоду Крепеу — за совместную работу над § 6.2 и § 6.3, и Чарльзу Х. Беннетту — за соавторство при написании всей главы 7. Кроме того необходимо отметить, что Дэвидом Чаумом собственноручно был написан весь § 6.4.

Дэвид Чаум и Жан-Жак Кискате обеспечили для меня возможность работы над этой книгой во время моего визита к ним: в Центр научно-технической информатики —Centrum voor Wiskunde Informatica (CWI) — в Амстердаме и, соответственно, в Научно-исследовательскую лабораторию фирмы Phillips в Брюсселе. Я также очень обязан Юргену Н. Е. Босу за его неутомимую и безотказную помощь во время моего пребывания в CWI. Окончательный вариант этой рукописи был подготовлен в CWI на фотонаборном устройстве Harris под управлением компьютера Boring, на котором использовалась издательская система `troff`, работающая под UNIX'ом. Мои научные исследования были поддержаны Канадским NSERC, грант A4107.

Хотя я не и не могу назвать здесь поименно всех, кого мне хотелось бы поблагодарить, так как пришлось бы перечислять очень многих, тем не менее я рад возможности выразить общую благодарность всем, с кем имел содержательные и стимулирующие обсуждения по вопросам криптологии за прошедшие годы. Именно они делают ее плодотворной и побуждают к активной работе в этой области. И наконец, причем не в меньшей степени, я выражаю огромную благодарность моей жене Изабель. Она не только не мешала моей работе над корректировкой и компьютерным набором рукописи к *CompCon'овскому* семинару *сразу же после того*, как закончились все мои треволнения, связанные с подготовкой английской версии моего учебника по алгоритми-

ке [77], но и фактически перепечатала бóльшую часть оригинала этой рукописи накануне и в самом начале Нового 1987 года.

Может быть, когда-нибудь я наберусь храбрости, чтобы переработать монографию, которую Вы, уважаемый читатель, держите в руках, в солидный учебник[2]. В связи с этим я был бы благодарен всем, кто поможет мне не допустить переноса ошибок из настоящей монографии в этот будущий учебник, и укажет на любого рода неточность(ти), которую(ые) он/она смогут найти в данной книге, в том числе и просто типографские опечатки. Для меня была бы ценной, также, информация относительно существования и координат окончательных журнальных версий статей, соответствующих библиографическим ссылкам, которые приведены здесь в списке литературы, и в качестве предварительных версий были опубликованы в трудах различных конференций. Всю корреспонденцию направляйте мне, пожалуйста, по адресу: *Département d'informatique u de recherche opérationnelle, Université de Montréal, C. P. 6128, Succursale "A", Montréal (Québec), CANADA H3C 3J7*. Всем вам я заранее благодарен.

Gilles Brassard
Брюссель, апрель 1988

[2] Летом 1994 года в ответ на вопрос, «набрался ли он храбрости», а если да, то когда можно ожидать выход в свет этого учебника, Жиль ответил мне, что уже оставил свою затею, так как «за него» такой учебник написал Дуглас Стинсон [341] (это и все дальнейшие подстрочные примечания в книге принадлежат переводчику).

Глава 1

Введение

«Тот, кто записывает свои секреты таким способом, который не позволяет скрывать их от посторонних — сумасшедший.» [12]

— *Роджер Бэкон*, около 1250 г.

На протяжении тысячелетий ***криптография*** была искусством засекречивания важной государственной информации при передаче ее по незащищенным каналам связи, а ***криптоанализ*** был двойственным криптографии искусством раскрытия такой информации. Поэтому ***криптология***, объединяющая в себе криптографию и криптоанализ, исторически ранее находилась почти исключительно в руках военных и дипломатических ведомств. Однако в нынешний период осуществления во всем мире компьютерной революции, когда огромное количество персональной, финансовой, коммерческой и технологической информации хранится в компьютерных банках данных и пересылается по информационным компьютерным сетям, чрезвычайно важным является то обстоятельство, что в обществе появляется острейшая потребность в гражданской криптографии. Отражением этого факта служат слова Дэвида Кана: «Криптография, в 1945 году тесно связанная с сохранением национальных секретов, стала теперь общедоступной» [229]. (См. также [239, 240, 241, 280].)

Кто же, в конце концов, может рассчитывать на победу в этом многовековом поединке между криптографией и крипто-

анализом? Великие умы (хотя и не специалисты) прошедших веков расходятся во мнениях. В 1769 году в своем *Философском словаре* (*Dictionnaire philosophique*) Вольтер писал: «Ceux qui se vantent de lire les lettres chiffrées sant de plus grands charlatans que ceux qui se vanteraieat d'entendre une langue qu'ils n'ont point apprise» [353] (что на русском соответствует примерно следующему: «... не зная законов языка ирокезского, можешь ли ты делать такое суждение по сему предмету, которое было бы неосновательно и глупо?»[1]). Противоположное мнение высказал Эдгар По в своем знаменитом рассказе *Золотой жук* (1843): «...едва ли разуму человека дано загадать такую загадку, которую разум его собрата, направленный должным образом, не смог бы раскрыть.» [287][2].

В настоящее время совершенно очевидно, что Вольтер был неправ, поскольку большинство известных из истории криптосистем были полностью раскрыты, причем иногда с очень интригующими последствиями [228]. С другой стороны, существуют криптографические системы, относительно которых было доказано, что независимо от «изобретательности» криптоаналитиков или мощности используемых при криптоанализе вычислительных средств они останутся нераскрываемыми (таким является, например, *одноразовый шифр*, который рассматривается в § 3.2). Однако по прежнему открытым остается тот же самый вопрос для широко применяемых сейчас на практике криптосистем *с открытым ключом* (они являются предметом темы, которая обсуждается в главе 4). В настоящее время существует убеждение в том, что засвидетельствованное в самые последние годы этого столетия увеличение мощностей вычислительных средств создает в высшей степени благоприятные условия для криптографов и в свою очередь наносит ущерб криптоаналитикам. Сложившееся положение можно считать иронией судьбы, потому что Колосс (Colossus), хронологически будучи самой первой электронно-вычислительной машиной, разрабатывался специально для криптоанализа шифров фашистской Германии [202, 302]. (Брайан Рэндел, как уже сообщалось, однажды

[1] Козьма Прутков, *Сочинения*, Художественная литература, М, 1974, стр. 128.

[2] Цитата здесь приводится в переводе А. Старцева по русскому изданию [287, стр. 184].

сказал по этому поводу: «По моим подсчетам, ENIAC был не первым компьютером, а одиннадцатым.» [372].) Таким образом, можно сказать, что криптоанализ, который, по меткому выражению Рональда Л. Ривеста, является «повивальной бабкой всей информатики» [306], по всей видимости, уже породил орудие для своей собственной погибели!

До недавнего времени предполагаемая надежность криптографической системы оценивалась количеством усилий, потраченных квалифицированными криптоаналитиками при неудачных попытках ее раскрытия. История доказала явную неправомерность такого подхода к оценке стойкости, поскольку сообщения, зашифрованные с помощью криптосистем, которые их пользователи считали нераскрываемыми, впоследствии, как правило, расшифровывались. Раскрытие Энигмы (Enigma) союзниками во время (или даже до) Второй мировой войны — простейший пример подобной ситуации [187, 303]. Читателю, интересующемуся историческим значением криптологии, мы рекомендуем прочитать замечательный обзор Кана [228][3], а также другие популярные книги, такие как [367, 99, 187, 230, 140, 362].

В нынешнем столетии математики занимались нахождением объективных критериев надежности криптографических систем, трансформировав таким образом это древнее искусство в точную науку. Свою теорию информации, которая была опубликована в [323], Клод Шеннон создал в результате работы над другой более ранней (и первоначально засекреченной) статьей в области криптографии [324]. Для самых различных криптографических систем он смог получить верхнюю оценку на длину шифртекста, которую необходимо учитывать для того, чтобы при криптоанализе достичь любого требуемого уровня достоверности его раскрытия. В связи с этим, например, Иб Мельхиор, если бы только он был знаком с теорией Шеннона, мог бы вообще не ездить в Эльсинор, когда решил, что якобы расшифровал секретную эпитафию на надгробном камне Шекспира, которая, как ему казалось, раскрывала тайну существования первого издания «Гамлета» [228, стр. 750].

В прошедшем десятилетии специалисты по компьютерным

[3] На русском языке имеется книга: СОБОЛЕВА Т. А., *Тайнопись в истории России. (История криптографической службы в России XVIII — начала XX в.)*, М., «Международные отношения», 1994.

наукам работали над обоснованием надежности криптографии исходя из более современной теории вычислительной сложности, а не из шенноновской теории информации [157, 265], результаты и выводы которой использовались ранее. Коренное различие между ними заключается в том, что при использовании теории Шеннона криптограф уповает на то, что криптоаналитик не будет располагать достаточной *информацией* для того, чтобы дешифровать криптограмму, в то время как, основываясь на теории вычислительной сложности, криптограф расчитывает только на то, что у криптоаналитика не хватит *времени*, чтобы это сделать.

Цель настоящей книги состоит в том, чтобы дать краткий обзор недавних криптографических достижений и методов, а также их многочисленных настоящих, а возможно и будущих, применений. От читателя не предполагается никакой специальной подготовки. Несмотря на то, что освещение некоторых тем по необходимости будет кратким, в книге приводится обширный (но, конечно, далеко не исчерпывающий) список литературы. В дополнение к историческим книгам, упомянутым ранее, доступны, кроме того, книги более технического характера, такие как [181, 242, 268, 142, 136, 244, 344, 293, 341][4] и, наконец, некоторые популярные статьи [227, 175, 185, 214, 332, 70], а также специальные обзорные статьи [159, 326, 249, 10, 246, 93, 208, 306] и сборник [333].

[4] Заслуживают особого внимания также фундаментальная недавно изданная книга Менезеса, Ооршота и Ванстоуна [263] и более практическая, вышедшая вторым изданием, книга Брюса Шнайера [312].

Глава 2

Определения и классификация

Цель *криптографической системы* заключается в том, чтобы любой ее пользователь — отправитель некоторого сообщения — с помощью заранее известной ему вполне определенной информации имел возможность *зашифровать* (осмысленный) *исходный текст* (также называемый *открытым текстом*) сообщения, отправляемого другому пользователю системы — получателю этого сообщения, получив в результате совершенно бессмысленный на взгляд *шифрованный текст*, или, коротко, *шифртекст* (называемый также *криптограммой*). Получатель, которому предназначен данный шифртекст, должен обладать, вообще говоря, другой, вполне определенной *секретной* информацией для того, чтобы быть способным с ее помощью *расшифровать* (говорят, также, *дешифровать*) полученный шифртекст, восстановив, таким образом, соответствующий ему открытый текст. При этом *нарушитель*, или *противник* (называемый также *криптоаналитиком* или *злоумышленником*), которому секретная информация для дешифрования неизвестна, должен быть неспособен эффективно определить, или, как говорят, *раскрыть*, исходный текст. Необходимо отметить, что при попытке раскрытия шифртекста соответствующий ему исходный текст ищется в принципе без знания секретной информации для его дешифрования. Именно в этом и заключается отличие дешифрования от раскрытия.

Раскрытием криптосистемы называется результат работы криптоаналитика, приводящий к возможности эффективного раскрытия любого, зашифрованного с помощью данной криптосистемы, открытого текста. Степень неспособности криптосисте-

мы к раскрытию называется ее *стойкостью*. Точное описание работы (математическая модель) криптографической системы называется ее *криптографической схемой*, или *криптосхемой*.

Существуют несколько способов, в соответствии с которыми могут быть расклассифицированы все криптографические системы. Обычно в качестве основной принимается следующая классификация:

- Криптосистемы ограниченного использования
- Криптосистемы общего использования
  - ⋆ с секретным ключом
  - ⋆ с открытым ключом.

Будем говорить, что криптографическая система является криптосистемой *ограниченного использования*, если ее стойкость основывается на сохранении в секрете самого характера алгоритмов шифрования и дешифрования. Простейшим историческим примером такой системы можно считать так называемый *шифр Юлия Цезаря*, который представляет собой простую замену каждого символа открытого текста третьим следующим за ним символом алфавита[1] (с циклическим переносом, когда это необходимо). Например, слово «`cleartext`» при таком шифровании превращается в «`fohduwhaw`». Криптосистемы ограниченного использования обычно разрабатываются любителями и почти всегда являются детской забавой для опытного криптоаналитика-профессионала. Гораздо важнее то, что такие системы вообще не используются в современной ситуации, когда должна обеспечиваться работа большого числа пользователей. *Шифры*, которые служат примерами криптосистем ограниченного использования, в настоящей книге не обсуждаются.

Криптографическую систему назовем криптосистемой *общего использования*, если ее стойкость основывается не на секретности алгоритмов шифрования и дешифрования, а на секретности некоторого сравнительно короткого значения, которое называется **ключом** этой криптосистемы. Такие ключи должны легко вырабатываться конкретными пользователями при помощи их

[1] Здесь и везде далее в книге автор, естественно, имеет в виду английский алфавит.

собственных ключей таким образом, чтобы при этом даже разработчик криптосистемы не мог ее раскрыть при условии, что у него нет доступа к тем ключам, которые в ней действительно используются.

В некоторых применениях (главным образом в военных, дипломатических и разведывательных ведомствах) у разработчика общей криптосистемы нет никаких причин делать общедоступным описание существа ее алгоритмов. Сохраняя такую информацию в тайне, можно обеспечить даже некоторую дополнительную безопасность. Однако решающим обстоятельством, позволяющим полагаться на секретность такой информации, это не является, поскольку ничего нельзя сказать о том, когда она может быть скомпрометирована. По этой причине, исследования надежности таких систем *всегда* должны проводиться в предположении, что потенциальному противнику о криптосистеме известно все, за исключением реально используемого секретного ключа. А если противник такими знаниями на самом деле не обладает, то это даже лучше. Для других типов применений, подобных, например, большим финансовым комплексам, в действительности лучше раскрывать, как работают их криптосистемы, поскольку в противном случае пользователи всегда будут подозревать, что возможно все-таки существует какой-то скрываемый от них метод раскрытия такой криптосистемы.

Одним из очевидных требований обеспечения стойкости общей криптографической системы является огромное количество возможных ключей, которое не позволяет провести *исчерпывающий поиск* (когда осуществляется попытка систематического дешифрования заданного шифртекста, используя при этом каждый из возможных ключей до тех пор, пока не получится некий осмысленный открытый текст). Например, при наивном подходе шифр Юлия Цезаря можно рассматривать как пример общей криптосистемы (с ключом $k = 3$), шифрование в которой заключается в замене каждого очередного символа открытого текста k-тым после него символом соответствующего алфавита, где k — некий секретный ключ. Но такое обобщение является бесполезным, потому что оно предоставляет возможность использования лишь 25 нетривиальных ключей, и тем самым обеспечивает простоту полного перебора для любого, кто предположительно знает метод шифрования (по крайней мере тогда, когда зашифрован-

ное сообщение имеет достаточную избыточность для того, чтобы у него имелась единственная осмысленная расшифровка).

Необходимо осознавать однако, что большое число ключей само по себе стойкости криптосистемы не обеспечивает. Так, например, еще одно обобщение шифра Юлия Цезаря может состоять в том, что в качестве ключа выбирается произвольная перестановка всех 26 букв алфавита, наподобие (EROX...WM), а шифрование каждого символа открытого текста производится в соответствии с этой перестановкой (A→E, B→R, ..., Z→M). При таком шифровании открытый текст «BAD DAY» преобразуется в шифртекст «REX XEW». Заметив, что существует 26! различных перестановок из 26 символов, которое является числом большим, чем 4×10^{26}, можно было бы предположить, что полный перебор на таком множестве ключей невозможен, потому что, если опробовать каждый возможный ключ, когда в течении каждой секунды проверяется миллиард ключей, то это заняло бы десять миллиардов лет! Тем не менее подобный *(одноалфавитный) шифр простой замены* является довольно легким для криптоанализа, хотя бы только из-за разницы в частотах, с которыми в естественном языке встречаются различные символы открытого текста [228, 181, 284]. Известны намного более надежные криптографические системы, которые были разработаны и использовались со значительно меньшим ключевым пространством.

Если вернуться к нашей классификации, то общая криптографическая система называется криптосистемой *с секретным ключом*, если в ней любые две стороны, перед тем, как связаться друг с другом, должны заранее договориться между собой об использовании в дальнейшем некоторой информации для шифрования и дешифрования, хранящаяся в тайне часть которой и называется секретным ключом. В предыдущем примере, когда первая сторона, назовем ее *Алисой*, зашифровывает сообщение, используя ключ (EROX...WM), и посылает шифртекст второй стороне, скажем, *Бобу*, то лучше всего, чтобы *Боб* заранее знал, какой ключ был использован при шифровании открытого текста.

Такая необходимость в секретном *распределении ключей* не была непреодолимой проблемой в те дни, когда потребность в криптографии испытывало небольшое количество пользовате-

лей, хотя при этом нужно было проявлять предусмотрительность, для того чтобы предотвратить препятствующие задержки прежде, чем секретная связь могла быть установлена. Теперь же, когда криптография стала общедоступной, было бы неразумно организовывать подобные коммуникационные сети, в которых каждой паре потенциальных пользователей заранее предоставлялся бы их совместный секретный ключ, потому что в такой сети число ключей возрастало бы квадратично с увеличением числа пользователей.

В 1976 году Уитфилд Диффи и Мартин Хеллман в [157] заложили основы для преодоления указанной трудности, введя понятия *открытого распределения ключей* и *криптографии с открытым ключом*. Сходные понятия было независимо открыты Ральфом Мерклем [265]. Вскоре последовала первая практическая реализация криптосистемы с открытым ключом, предложенная Рональдом Ривестом, Эди Шамиром и Леонардом Эдлеманом [307]. Секретная связь по незащищенным каналам связи между двумя *совершенно незнакомыми* друг с другом сторонами наконец-то стала возможна.

Основное наблюдение, которое, собственно, и привело к криптографии с открытым ключом, заключалось в том, что тот, кто зашифровывает сообщение, не обязательно должен быть способен его расшифровывать. В таких системах каждый пользователь выбирает свой *собственный секретный ключ*, на основании которого получает пару соответствующих алгоритмов. При этом он делает один из них доступным каждому из возможных своих респондентов, объявляя этот алгоритм собственным *открытым алгоритмом шифрования*, в то время как другой, двойственный этому открытому, объявляет своим *личным алгоритмом дешифрования*, и хранит его в строгом секрете. Это, например, позволяет *Бобу*, даже совершенно незнакомому с *Алисой*, применять ее (общедоступный) открытый алгоритм шифрования, чтобы зашифровать предназначенное ей сообщение, но только она сама сможет расшифровать его с помощью своего личного (секретного) алгоритма дешифрования. Само собой разумеется, что такие системы могут быть стойкими лишь тогда, когда из общедоступного алгоритма шифрования никаким образом нельзя получить соответствующий ему секретный алгоритм дешифрования.

Совсем недавно Шафи Гольдвассер и Сильвио Микэли ввели понятие *вероятностного шифрования*, которое является очень интересной вариацией на тему криптографии с открытым ключом [197, 198, 56]. В том случае, если произвольное сообщение шифруется при помощи вероятностного шифрования, то при криптоанализе шифртекста, по существу, становится одинаково трудно выяснить о сообщении какую бы то ни было информацию, которая позволила бы восстановить весь открытый текст. Кроме того следует отметить, что существует вероятностная схема шифрования, которая является более быстрой, чем предложенная до этого схема шифрования с открытым ключом RSA — см. § 4.4 и § 4.6. Подобные криптографические системы называются «вероятностным» в связи с тем, что при их использовании шифрование сообщений, которые имеют один и тот же исходный текст и шифруются на одном и том же ключе, в разное время может привести к совершенно различным шифртекстам.

Рассматривались и некоторые другие подходы к проблеме распределения ключей. Например, *бесключевая криптография*, предложенная Боуэном Альперном и Фредом Шнейдером, может эффективно использоваться в сетях связи, которые скрывают происхождение (но не содержание) сообщений [9, 370, 141]. В *идентификационной криптосистеме* Эди Шамира отпадает необходимость в распределении ключей, но требуется наличие некоего центра, которому должна быть доверена генерация секретных ключей [318]. Однако здесь мы не будем обсуждать эти новые понятия. В заключение, отметим только, что Чарльз Беннетт и Жиль Брассар, опираясь на работу Стефена Уиснера [357], разработали теорию *квантовой криптографии*, которая предлагает совершенно иную базу для современной криптологии и в своих утверждениях о секретности основывается скорее на квантовой физике, нежели на математике или теории вычислительной сложности [31, 27, 26, 28, 25]. Квантовой криптографии посвящена заключительная седьмая глава этой книги.

Глава 3

Системы с секретным ключом

§ 1. Определения и уровни атак

Криптосистема с открытым ключом состоит из *пространства ключей* $\mathcal{K}$ и, для каждого $k \in \mathcal{K}$, из *пространства открытых текстов сообщений* $\mathcal{M}_k$, *пространства шифртекстов сообщений* $\mathcal{C}_k$ и пары функций $E_k: \mathcal{M}_k \to \mathcal{C}_k$ и $D_k: \mathcal{C}_k \to \mathcal{M}_k$ таких, что $D_k(E_k(m)) = m$ для каждого сообщения открытого текста $m \in \mathcal{M}_k$. При этом необходимо, чтобы, задаваясь любым ключом k, можно было легко получать эффективные алгоритмы для вычисления E_k и D_k. Криптосистема называется *эндоморфной*, если $\mathcal{C}_k = \mathcal{M}_k$ для каждого k.

Для обеспечения секретной связи криптосистема используется следующим образом. Если, например, *Алиса* и *Боб* предполагают, что в конечном счете могут установить друг с другом конфиденциальную связь, то с самого начала они должны договориться между собой о некотором совместном секретном ключе $k \in \mathcal{K}$. Тогда, если в дальнейшем *Алиса* захочет послать некоторое специальное сообщение $m \in \mathcal{M}_k$ *Бобу*, то она, для того чтобы сформировать шифртекст $c = E_k(m)$, должна использовать алгоритм шифрования E_k; затем ей необходимо будет послать шифртекст c по незащищенному каналу *Бобу*; получив этот шифртекст c, *Боб* должен использовать алгоритм D_k, чтобы восстановить открытый текст сообщения $m = D_k(c)$.

Во многих практически используемых криптографических системах и $\mathcal{M}_k$, и $\mathcal{C}_k$ — конечные множества, часто не зависящие от самого ключа k. Таково, например, множество всех строк,

состоящих из восьми символов. В этом случае может случиться так, что в действительности сообщение m окажется слишком длинным, чтобы его можно было расшифровать непосредственно. Если это и в самом деле так, то m нужно разбить на части, а процедуру шифрования E_k применять несколько раз. В дальнейшем мы еще обсудим эту ситуацию в разделе § 5.

Первое требование, которому должна удовлетворять криптосистема с секретным ключом, заключается в том, что криптоаналитик должен быть неспособен воспроизвести открытый текст m (или, что еще хуже, k) из перехваченного шифртекста $c = E_k(m)$. Однако отметим при этом, что криптографическая система, которая невосприимчива даже к такой угрозе, может оказаться слабой при других обстоятельствах. В криптографии с секретным ключом различают три уровня криптографических атак.

- *Атака на основе только шифртекста*: криптоаналитику заданы $c_1 = E_k(m_1)$, $c_2 = E_k(m_2)$, ..., $c_1 = E_k(m_i)$, являющиеся шифртекстами i различных неизвестных открытых текстов сообщений, зашифрованных на одном и том же неизвестном ключе. Он должен определить ключ k или убедиться в своей неспособности это сделать, исходя из такого количества различных открытых текстов m_1, m_2, ..., m_i, какое для этого потребуется.

- *Атака на основе известного открытого текста*: криптоаналитику заданы шифртексты c_1, c_2, ..., c_i такие же, как определены выше, а также соответствующие им открытые тексты m_1, m_2, ..., m_i. Он должен либо определить ключ k, либо, убедившись в своей неспособности это сделать, вычислить m_{i+1} из некоторого нового шифртекста $c_{i+1} = E_k(m_{i+1})$, зашифрованного с использованием того же самого ключа.

- *Атака на основе выбранного открытого текста*: криптоаналитику предоставляется выбрать несколько сообщений m_1, m_2, ..., m_i открытого текста и вместе с тем ему передаются соответствующие им шифртексты $c_1 = E_k(m_1)$, $c_2 = E_k(m_2)$, ..., $c_1 = E_k(m_i)$. Криптоаналитик должен либо найти k, либо, убедившись в своей способности это сделать, вычислить m_{i+1} из некоторого нового шифртекста

$c_{i+1} = E_k(m_{i+1})$, зашифрованного с использованием того же самого ключа. (Ситуации, в которых может быть проведена подобная мощная атака на основе выбранного открытого текста, существуют в реальной жизни — например, в системах идентификации типа «друг-или-враг» [228].)

Различие в степени действенности этих трех уровней атак лучше всего объяснить при помощи нашего предыдущего примера шифра простой замены. Когда мы говорили, что он является «легким» для криптоанализа, то имели в виду — «легким при атаке на основе только шифртекста». Хотя это и в самом деле так, для подобного криптоанализа все-таки требуется выполнить некоторую работу. Если же проводить атаку на основе известного открытого текста, то раскрытие становится совершенно тривиальным, как только в доступных открытых текстах сообщений встретятся, по крайней мере по одному разу, все символы алфавита (естественно, при этом достаточно даже всех, кроме одного). А при атаке на основе выбранного открытого текста даже не нужно ничего ждать: ключ (то есть секретная перестановка алфавита) получается тотчас же после того, как становится доступным значение $E_k(\texttt{ABCD}\ldots\texttt{XYZ})$.

§ 2. Теория информации

Что же мы подразумеваем под словами: «криптоаналитик должен быть неспособен воспроизвести открытый текст m»? Заслуживает некоторого пояснения, какой смысл вкладывается здесь в слова «неспособен» и «воспроизвести». При этом с самого начала отметим, что в данном параграфе под криптографической атакой обычно подразумевается атака на основе только шифртекста.

С точки зрения классической теории информации Клода Шеннона, слово «неспособность» означает «математическую невозможность вне зависимости от доступных ресурсов». Например, предположим, что вы бросаете жребий, но перед тем как самому взглянуть на то, что выпало, «орел» или «решка», просите своего друга случайным образом решить, оставить то, что получилось, или перебросить. Очевидно, что из знания конечного результата такого эксперимента, невозможно определить, каков был первичный исход подбрасывания монеты. (При этом

необходимо отметить, что в главе 4, когда мы будем обсуждать криптосистемы с открытым ключом, слово «неспособен» будет иметь совершенно другой смысл.)

Точное смысловое значение слова «воспроизвести» гораздо труднее объяснить без определения важнейших положений теории информации. Относительно формального математического изложения последней мы рекомендуем обратиться к [323, 324, 184]. Окончательная цель криптоаналитика заключается, конечно же, в том, чтобы точно и определенно вычислить ключ k криптосистемы или, хотя бы, какое-то конкретное сообщение m открытого текста. Однако он может быть удовлетворен, узнав даже некоторую *вероятностную* информацию об m. Уже само предположение о том, что открытый текст некоторого сообщения написан по-английски, предоставляет криптоаналитику определенную *априорную* информацию об этом сообщении даже *до* того, как он увидит его шифртекст. Так, например, он заранее знает, что слово «`hello`» является *более вероятным* началом сообщения, чем скажем набор букв «`xykph`». Поэтому текущая цель криптоанализа заключается в том, чтобы посредством изменения вероятностей увеличить количество этой *априорной* информации, относящейся к каждому возможному сообщению открытого текста, таким образом, чтобы правильный открытый текст сделать более вероятным, хотя, быть может, и не обязательно точным.

Рассмотрим ситуацию, при которой криптоаналитик перехватил шифртекст «`xtjja`» и знает (или предполагает), что он был зашифрован с использованием шифра простой замены. Этот шифртекст говорит ему о том, что открытый текст сообщения состоит из пяти букв, третья и четвертая из которых являются одной и той же, а все остальные отличными от нее и разными. Он не может считать, что открытый текст — это слово «`hello`», потому что это также могло бы быть, например, и слово «`teddy`». Тем не менее *апостериорные* вероятности таких открытых текстов возрастают относительно их *априорных* вероятностей. Криптоаналитик, кроме этого, совершенно уверен (в предположении, что он прав относительно характера криптосистемы) в том, что этот открытый текст не может быть ни словом «`peace`», ни словом «`rambo`», и, таким образом, *апостериорная* вероятность обоих этих открытых текстов сокраща-

ется до нуля даже вне зависимости от их *априорной* вероятности.

Шеннон называет криптографическую систему *совершенно секретной*, если, каким бы ни был зашифровываемый с ее помощью открытый текст, знания, которые могут быть получены из соответствующего ему шифртекста, не раскрывают никакой информации об исходном тексте, за исключением, возможно, его длины. Другими словами, в подобных системах *апостериорные* вероятности открытых текстов даже после просмотра шифрованных текстов остаются *точно* такими же, какими были их *априорные* вероятности. Совершенно секретные системы действительно существуют [352], и может показаться странным, почему они не являются панацеей для всех криптографических потребностей. Имеются три основных причины, объясняющих такое положение дел.

- Как и любые криптосистемы с секретным ключом, совершенно секретные системы предполагают, что проблема распределения ключей уже решена.

- Трудность проблемы распределения ключей еще больше усугубляется в связи с тем, что согласно теореме Шеннона соблюдение совершенной секретности возможно *только тогда, когда* ключевое пространство является, по крайней мере, таким же большим, как и пространство открытых текстов сообщений (что в свою очередь позволяет утверждать, что секретный ключ должен иметь по крайней мере такую же длину, как и само сообщение), *и только если* один и тот же ключ не используется при шифровании более одного раза.

- Третий недостаток совершенно секретных криптосистем состоит в том, что они (как описано в § 5.1) могут играть лишь незначительную роль для целей аутентификации.

Несмотря на это, совершенно секретные криптографические системы используются на практике в самых ответственных приложениях, наподобие «красного телефона» между Москвой и Вашингтоном.

Совершенная секретность может быть достигнута следующим образом. Пусть m — это открытый текст некоторого сообщения,

например, «hello». *Одноразовым шифром* k сообщения m называется полученная совершенно случайным образом строка символов, в точности такой же длины, как и m, например «iwpbu». Шифрование сообщения m с использованием ключа k очень похоже на то, которое применялось в нашем первом обобщении криптосистемы с использованием шифра Юлия Цезаря, и было описано в главе 2, за исключением того, что число позиций букв в алфавите, на которое в этом случае нужно сдвинуться, чтобы вместо очередного символа открытого текста получить соответствующий символ шифртекста, не является постоянной величиной. В нашем примере буква «h» открытого текста m должна быть заменена на букву «q», являющуюся девятым после «h» символом английского алфавита, потому что соответствующий символ ключа k — буква «i» — девятый символ этого алфавита. Аналогично, буква «e», циклически сдвигаясь на 23 позиции (в соответствии с положением символа «w» ключа в алфавите), становится буквой «b». Если продолжать далее таким же образом, то в результате мы получим шифртекст «qbbnj». Обратите внимание, что третий и четвертый символы открытого текста являются идентичными, что не так, если посмотреть на соответствующие символы шифртекста, в то время как, напротив, второй и третий символы в шифртексте одинаковы.

Открытый текст «hello» может быть легко восстановлен из шифртекста «qbbnj» при условии, что ключ «iwpbu», который используется для шифрования, известен. Однако без этой информации при помощи соответствующего ключа произвольный шифртекст может быть дешифрован в любые пять символов открытого текста. Например, «qbbnj» можно дешифровать в «peace», используя ключ «awake». Здесь решающим для понимания является то, что если ключ действительно выбирается наугад, как это и должно быть, то оба ключа, «iwpbu» и «awake», будут абсолютно равновероятными, даже несмотря на то, что один из них случайно оказался осмысленным словом английского языка. Это иллюстрирует важность требования, согласно которому ключ должен выбираться случайным образом. Фактически, если известно, что и ключ, и открытый текст должны быть словами английского языка, то для того чтобы эффективно восстановить и тот и другой из имеющегося шифрованного текста, обычно бывает достаточно его отрезка, длиной в несколько

десятков символов [181]. Для того чтобы избежать корреляций между шифртекстами, решающим обстоятельством является то, что для шифрования разных сообщений никакие части ключа не должны использоваться повторно.

Описанная выше схема «сложения по модулю 26» прекрасно реализуется при помощи карандаша и бумаги (и с помощью так называемой *таблицы Виженера* [142]). Однако гораздо удобнее реализовывать ее в двоичном виде как *электронный* одноразовый шифр: открытый текст перекодируется в битовую строку посредством некоторого стандартного преобразования (в котором нет никакого секрета — например, используя стандарт ASCII); одноразовым шифром является произвольная двоичная строка точно такой же длины, а шифртекст получается в результате поразрядной операции «исключающего или» над этими двумя строками (определение операции «исключающее или», которая также называется сложением по модулю 2, приведено в § 5). Дешифрование осуществляется при помощи точно такого же процесса: поразрядное «исключающее или» над зашифрованным текстом и тем же самым ключом снова приводит к открытому тексту (в битовом представлении).

Если криптографическая система не является совершенно секретной, то знание шифртекста сообщения предоставляет некоторую информацию относительно соответствующего ему открытого текста. В частности, в связи с природной избыточностью, которая присуща английскому (как и вообще любому другому естественному) языку, для большинства классических криптосистем с секретным ключом по мере увеличения длины сообщений можно провести гораздо более простое сокращение числа кандидатов на секретный ключ шифрования. Рассмотрим криптографическую систему с ключевым пространством, в котором ключи имеют фиксированную длину (и длина ключа не зависит от длины открытого текста сообщения). Обозначим через $H(K)$ *энтропию ключевого пространства* (которая приблизительно равна логарифму по основанию 2 от числа ключей), а через D — избыточность исходного языка открытого текста, измеряемую в битах на символ (которая для английского языка составляет примерно 3,5 [325]). Тогда для любой однозначно зашифровывающей и однозначно расшифровывающей эндоморфной криптографической системы ожидаемое число неправильных ключей

для дешифрования сообщения длины n равно по крайней мере $2^{H(K)-nD} - 1$ [15], причем это число близко к аналогичному значению для так называемых случайных шифров [212].

Расстояние единственности любой криптографической системы определяется как длина произвольного шифртекста, при которой ожидается существование единственного соответствующего ему осмысленного открытого текста [324]. Таким образом, расстояние единственности сообщает нам *не* то, насколько большим должен быть шифртекст, для того чтобы гарантировать *простой* его криптоанализ, а скорее то, насколько он должен быть длинным, чтобы быть уверенным в существовании правильного решения. Для классических криптосистем расстояние единственности приближенно выражается формулой $H(K)/D$ [324, 212]. Например, расстояние единственности для шифра простой замены равно примерно $log_2(26!/3,5) \approx 25$ символам. Этот теоретический результат согласуется с практикой, поскольку оказывается, что для 30-буквенной криптограммы соответствующего типа почти всегда существует уникальное решение, в то время как с лишь 20-символьными криптограммами обычно легко удается найти несколько допустимых решений. В свой статье [139] Цифер А. Дивоурс приводит хороший краткий обзор вычислений расстояния единственности для некоторых классических криптографических систем.

Шеннон определил также понятие *идеальной* секретности криптографических систем. Так называются криптосистемы, расстояние единственности которых бесконечно, при этом они могут даже не быть совершенно секретными. После криптоанализа таких криптосистем на основе шифртекста обычно остается некоторая неопределенность, поскольку вне зависимости от длины шифртекста выполненный исключительно на его основе криптоанализ может, тем не менее, не дать никакой информации о соответствующем открытом тексте.

§ 3. Рассеивание и перемешивание

При криптоанализе традиционных криптосистем с секретным ключом основная угроза раскрытия шифртекста таится в высокой избыточности языка, на котором написан открытый текст сообщения. Именно она позволяет применять различные виды

статистических атак, многие из которые описаны в классическом учебнике Фридмана [181]. В связи с этим Шеннон в [324] предложил два основных криптографических метода: рассеивание и перемешивание[1], «которые делают любой статистический анализ совершенно бесполезным».

Цель *рассеивания* заключается в перераспределении избыточности исходного языка, которая имеется в различных местах открытого текста сообщения, посредством распространения ее на весь открытый текст. Это может быть достигнуто двумя различными способами. Рассмотрим шифр, использующий *транспозиции*, которые переставляют символы (буквы или биты) исходного сообщения. Так например, перестановка $(1 \to 3,\ 2 \to 5,\ 3 \to 4,\ 4 \to 1,\ 5 \to 2)$ на множестве из пяти элементов, примененная к открытому тексту «`hello`», дает слово «`lolhe`». Пусть секретным ключом будет именно эта перестановка, и предположим, что более длинный открытый текст шифруется с использованием одного из «режимов работы», которые описаны в § 5. Тогда такое шифрование, хотя и не окажет воздействия на частоту появления самих букв (и скорее, даже облегчит раскрытие самого шифра [181]), но зато скроет частоты появления биграмм, триграмм, и так далее.

Другой подход к рассеиванию заключается в том, чтобы сделать каждый символ (или бит) открытого текста зависимым от стольких его оставшихся символов (битов), от скольких это вообще возможно. Рассмотрим, например, открытый текст сообщения $m = m_1 m_2 \ldots m_n$, где каждый символ m_i выражается целым числом от 00 до 25. Пусть $k = k_1 k_2 \ldots k_s$ для некоторого целого числа s является секретным ключом и имеет точно такое же представление, что и m. Для $0 \leqslant i < s$ положим $m_{-i} = k_{s-i}$. После этого для каждого $i < n$ определим c_i как $c_i = \left(\sum_{j=0}^{s} m_{i-j}\right) \bmod 26$ (определение оператора mod приводится в § 4.1) и рассмотрим шифртекст $c = c_1 c_2 \ldots c_n$. Когда ключ известен, расшифровать такой шифртекст не составляет труда. Обратите внимание, что здесь каждый символ шифртекста c (кроме первых s) зависит от $s+1$ символов открытого текста, и именно это обеспечивает рассеивание. Для осуществления

[1] В русской версии [324], опубликованной в книге К. Шеннон, Работы по теории информации и кибернетике, ИЛ, М, 1963, соответствующие термины «diffusion» и «confusion» переведены как «распыление» и «запутывание».

рассеивания очень хорошо подходят также некоторые из режимов работы при шифровании длинных открытых текстов, которые мы будем обсуждать в § 5.

Цель *перемешивания* состоит в том, чтобы зависимость между ключом и шифртекстом сделать настолько сложной, насколько это вообще возможно. Криптоаналитик, основываясь на результатах статистического анализа шифрованного текста, полученного в процессе перемешивания, не должен получить сколь-нибудь значительного количества полезной информации о ключе шифрования криптосхемы. Обычно перемешивание осуществляется при помощи техники *подстановок*, которые используют для получения шифртекста из открытого текста перестановки символов алфавита сообщения. Использующий подстановку шифр простой замены не дает очень уж хорошего перемешивания потому, в частности, что самым повторяющимся символом любого шифртекста в этом случае почти наверняка будет пятая запись в его ключе шифрования (соответствующая букве «E» открытого текста сообщения). Лучше использовать подстановку для блоков из нескольких символов, хотя она и приводит к тому, что ключ становится намного длиннее (во всяком случае тогда, когда подстановка задается в виде таблицы). Альтернативный подход состоит в том, чтобы для каждой позиции в открытом тексте использовать свою собственную подстановку. Это может привести, с одной стороны, к совершенной секретности, как в одноразовом шифре, а с другой — к злополучной Энигме (Enigma) [187, 303].

Необходимо учитывать, что применяемые порознь ни рассеивание, ни перемешивание сами по себе не являются очень уж действенными методами, радикально затрудняющими статистический криптоанализ шифртекста (если только ключ не будет достаточно длинным — ведь, в конце концов, в одноразовых шифрах используется только перемешивание). Тем не менее, оказывается, что криптосистема, в которой оба этих метода используются совместно, становится намного более стойкой. Возможно, самый лучший пример подобного феномена — Стандарт шифрования данных (Data Encryption Standard, или, сокращенно, DES).

§ 4. Стандарт шифрования данных (DES)

Стандарт шифрования данных (DES) — это знаменитая криптографическая система с секретным ключом, которая была предложена Национальным бюро стандартов при Министерстве торговли США (National Bureau of Standarts, или сокращенно NBS) в 1977 году [277]. Она была разработана для использования на срок от десяти до пятнадцати лет «в интересах Федерального правительства [США] для криптографической защиты наиболее значимых, но не подлежащих категорированию компьютерных данных». Несмотря на то, что с недавнего времени эта криптосистема уже не имеет сертификата, она по-прежнему широко применяется и достойна изучения. Основным преимуществом DES является то, что ее использование позволяет достичь очень высокой скорости шифрования и дешифрования. История создания DES описана Хорстом Фейстелом в [175], ее «настоящее и будущее» — Майлсом Смидом и Деннисом Брэнстэдом в [337].

Мы не будем описывать здесь детально алгоритмм DES. Для этого обращайтесь к [277, 142, 337]. Отметим лишь, что в нем шифруются 64-битовые блоки данных с использованием секретного ключа длиной в 64 бита. Однако, фактически секретными в нем являются только 56 бит, так как последние разряды байтов, составляющих ключ, т.е. его 8-ой, 16-ый, ..., 64-ый биты, отведены для контроля предшествующих разрядов соответствующих байтов по четности. Алгоритм DES предварительно преобразует секретную 56-битовую часть ключа посредством вполне определенного алгоритма *формирования ключей*, использующего перестановки и сдвиги, в шестнадцать 48-битовых *частичных ключей*, используя каждый из битов первичного ключа по нескольку раз. Затем после стандартной *начальной перестановки* над 64-битовым блоком данных открытого текста производится шестнадцать *раундов* некоего конкретного преобразования, в конце которого осуществляется инверсия начальной перестановки. При этом, следуя рекомендациям Шеннона, в каждом раунде выполняется один шаг перемешивания (с использованием соответствующего частичного ключа и так называемых *S-боксов*), за которым следует один шаг рассеивания. Примечательно, что процесс на этапе рассеивания не зависит от секретного ключа, так что стойкость достигается за счет комбинации перемешива-

ния и рассеивания, даже несмотря на то, что один из двух шагов преобразования является стандартным и общеизвестным.

Алгоритм DES разработан таким образом, что дешифрование в нем выполняется с помощью точно такого же процесса (в *прямом* порядке), что и шифрование, за исключением того, что формирование частичных ключей для дешифрования тогда будет производиться в обратном порядке. Это очень удобно, поскольку одно и то же устройство может использоваться как для процедуры шифрования, так и для процедуры дешифрования.

Является ли использование DES достаточно надежным? В 1979 году Мартин Хеллман написал статью под названием «DES будет полностью раскрыт в течении десяти лет» [213]. Подобная полемика относительно стойкости DES обычно возникает из-за того, что пространство его ключей является относительно небольшим и позволяет сделать возможным его полный перебор, даже несмотря на то, что при этом он может быть довольно дорогостоящим. Миллион процессоров, работающих в параллель и проверяющих миллион ключей за одну секунду, перебрал бы все его ключевое пространство за двадцать часов. Однако несмотря на многочисленные исследования криптосистемы DES [215, 158, 213, 348, 137, 145, 179, 113, 319, 231, 93, и т.д.], никто еще пока не смог найти в ней другие сколь-нибудь существенные изъяны. Таким образом, она, по-видимому, вполне отвечает требованиям обеспечения секретности для небольших и средних приложений.

Тем не менее использование DES с целью установления подлинности (см. § 5.1) кажется более сомнительным, потому что последствия от нескольких успешно подделанных фальсификатором сообщений могут быть намного более серьезными, чем последствия от нескольких успешных расшифровок перехваченных шифртекстов.

Простой методикой, которая может быть использована, для того чтобы сделать полный перебор более трудоемким, и без которой DES вообще не должен применяться, является *многократное шифрование*. Вместо одного 64-битового ключа необходимо использовать два (или, еще лучше, три) таких ключа. Очевидным подходом было бы зашифровывать m по формуле $c = DES_{k_1}(DES_{k_2}(m))$. Однако это не повышает стойкости, поскольку угроза простого полного перебора может возникнуть

вновь. Действительно, если имеются в распоряжении по крайней мере два соответствующих друг другу известных блока открытого и шифрованного текста, то k_1 и k_2 могут быть вычислены с большой вероятностью после примерно 2^{56} DES-шифрований и приблизительно такого же числа DES-дешифрований. Для того чтобы показать, как это делается, предположим, что m_1, m_2, c_1 и c_2 таковы, что $c_i = DES_{k_1}(DES_{k_2}(m_i))$ для $1 \leqslant i \leqslant 2$. Для каждого значения ключа k вычислим $DES_k(m_1)$ и запишем полученный результат в некоторой хэш-таблице (так, чтобы по каждому результату можно было быстро определить соответствующий ему ключ k). Затем снова для каждого значения k вычислим $DES_k^{-1}(c_1)$ и попробуем найти получившийся результат в хэш-таблице. Если он будет найден, то возможно, что соответствующее ему значение k является в действительности ключом k_2, а то значение k, которое использовалось, чтобы получить этот результат (для записи его в хэш-таблицу), как раз и есть k_1. Если же, кроме того, и $c_2 = DES_{k_1}(DES_{k_2}(m_2))$, то, вполне вероятно, что пара $\langle k_1, k_2 \rangle$ является правильной (с вероятностью ошибки, равной приблизительно 2^{-16}). В противном случае перебор значений k и поиск результатов $DES_k^{-1}(c_1)$ в хэш-таблице может быть продолжен. Ожидаемое число «ложных тревог» (неправильных k, для которых $DES_k^{-1}(c_1)$ находится в хэш-таблице) равно примерно 2^{48}. Обратите внимание на то, что хэш-функция может быть очень грубой, так как ожидаемые результаты работы DES должны быть в большинстве своем почти случайными.

Хотя описанная выше атака ненамного медленнее, чем исчерпывающий поиск при однократном шифровании, она требует значительно большего пространства в оперативной памяти для хранения соответствующей хэш-таблицы. Альтернативное решение состоит в том, чтобы результаты вычислений $DES_{k_1}(m_1)$ для каждого значения ключа k_1 (позволяющие при этом сохранять ссылки на значения k_1) записывать на одних магнитных лентах, а результаты $DES_{k_2}^{-1}(c_1)$ для каждого значения k_2 (позволяющие также сохранять ссылки на значения k_2) — на других. После сортировки содержимого таких лент последовательный просмотр позволяет легко находить подходящие пары $\langle k_1, k_2 \rangle$, которые при дальнейшей проверке с использованием m_2 и c_2 могут быть отброшены.

Вальтером Тахманом в [347] была предложена несколько луч-

шая процедура двухключевого шифрования при помощи алгоритма DES. Согласно ей шифртекст нужно вычислять по формуле $c = DES_{k_1}(DES_{k_2}^{-1}(DES_{k_1}(m)))$. Использование в этой формуле обратного преобразования на втором шаге вычисления необходимо для того, чтобы обеспечить сопряжение с однократным шифрованием, когда k_1 и k_2 имеют одинаковые значения. Хотя указанный подход предотвращает простую атаку типа «навстречу друг-другу», которая была описана выше, Ральф Меркль и Мартин Хеллман [267] нашли способ, как раскрыть такую схему также приблизительно за 2^{56} шагов (хотя им для этого потребовалась атака на основе выбранного открытого текста). По этой причине они рекомендовали схему шифрования с использованием трех независимых ключей согласно формуле $c = DES_{k_1}(DES_{k_2}^{-1}(DES_{k_3}(m)))$. Даже несмотря на то, что такое шифрование делает полный перебор в настоящее время практически неосуществимым, не ясно, приводит ли это к почти неуязвимому шифру, поскольку в криптосистеме DES могут существовать еще необнаруженные (или не до конца исследованные) слабости или лазейки. Тем не менее Дон Копперсмит написал в [122] относительно DES следующее: «Я горжусь тем, что принимал посильное участие в этом проекте [проект DES]. (...) Насколько мне известно, никто из пытавшихся сократить криптоанализ DES так и не смог сделать его существенно более простым, чем полный перебор всех ключей.».

Если DES реализуется в аппаратуре, то с его помощью может быть достигнута очень высокая скорость шифрования и дешифрования. В наиболее быстром в настоящее время коммерчески используемом шифраторе скорость шифрования достигает 90 мегабит в секунду[2]. Описание других аппаратных реализаций можно найти в [256, 13, 217, 351]. Добиться высокого быстродействия схем можно, если при их проектировании правильно использовать идеи, изложенные в [137]. Указанное быстродействие вполне достаточно для того, чтобы зашифровывать или расшифровывать данные без потери скорости при чтении или записи их на диск, и оно вполне приемлемо для большинства прикладных программ передачи данных. Довольно приличные скорости — до 650 килобит в секунду на 80-ой модели IBM PS/2 —

[2] В этом абзаце указаны данные на 1993 год, которые приведены во французском издании книги.

можно получить также посредством только программной реализации [205]. Относительно других программ, реализующих DES см. [138, 359, 160].

§ 5. Режимы операций

Рассмотрим криптографическую систему наподобие DES, в которой пространство сообщений состоит из блоков длиной по 64 бита. Как с ее помощью нужно шифровать более длинные сообщения? Очевидное решение состоит в том, чтобы разбить текст сообщения на отрезки длиной по 64 бита каждый и шифровать их последовательно и независимо, используя один и тот же секретный ключ. Однако такого подхода, который известен под названием режима *электронной кодовой книги* (ECB), необходимо максимально избегать везде, где только возможно. Наиболее очевидная его слабость заключается в том, что два одинаковых отрезка шифрованного текста однозначно указывают криптоаналитику, что два соответствующих им отрезка открытого текста также идентичны. Подобная информация может быть очень ценной отправной точкой для вычисления всего открытого текста. Если же DES используется с целью подтверждения подлинности (аутентификации), то ситуация оказывается еще хуже (см. § 5.1).

Существует по крайней мере четыре альтернативы режиму ECB. Во всех этих режимах два одинаковых блока открытого текста (почти наверняка) шифруются по-разному. В режимах CBC и CFB (которые описываются ниже) надлежащим образом используется понятие рассеивания: каждый блок шифртекста зависит от всего предшествующего ему открытого текста. Это делает их пригодными для целей аутентификации, так как таким образом в них предотвращаются попытки удаления и вставки нарушителем предварительно преобразованных частей шифртекста (см. § 5.1). Кроме того, оба этих режима являются *самосинхронизирующимися* в том смысле, что если при передаче возникает ошибка или, если в процессе шифрования и дешифрования происходит случайный сбой, или даже, если теряется некоторый, причем неизвестно какой, блок шифртекста, то при этом могут быть неправильно дешифрованы лишь несколько блоков открытого текста. В режимах OFB и счетчика (которые также описаны ниже) могут одинаково хорошо исправляться случай-

ные ошибки передачи, хотя ошибки других типов исправляются не так легко.

Теперь кратко опишем каждый из этих режимов. За более подробными сведениями обращайтесь к [278, 142]. Несмотря на то, что мы будем рассматривать все режимы только с использованием DES, должно быть очевидно, что они могут применяться точно так же и в любой другой криптографической системе, основанной на пространстве сообщений, в котором все блоки имеют заданный размер. В режиме *сцепления блоков шифра* (CBC) секретным ключом является 64-разрядный DES-ключ k (точнее его секретная 56-битовая часть) и некий 64-битовый блок c_0 (при этом секретность c_0 не является обязательной). Открытый текст m разбивается на блоки, длиной по 64 бита каждый, таким образом, что $m = m_1 m_2 \ldots m_n$. Для $i = 1, 2, \ldots, n$ блок c_i шифртекста вычисляется по формуле $c_i = DES_k(m_i \oplus c_{i-1})$, где через $\oplus$ обозначена операция поразрядного *исключающего или* ($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$ и $1 \oplus 1 = 0$). В результате получается шифртекст $c = c_1 c_2 \ldots c_n$. Для заданного шифртекста при знании k и c_0 дешифрование осуществляется посредством вычисления m_i для $i = 1, 2, \ldots, n$ по формуле $m_i = c_{i-1} \oplus DES_k^{-1}(c_i)$. Из нее становится понятно, почему не распространяются ошибки при использовании режима CBC — ведь каждый блок открытого текста m_i зависит только от двух блоков c_{i-1} и c_i шифртекста. Поэтому одна ошибка при передаче может привести к неправильному дешифрованию только двух блоков открытого текста. Этот режим может (и должен) использоваться также в криптографических системах с открытым ключом (в частности в RSA, которая обсуждается в § 4.4), когда размер шифруемого сообщения больше, чем размер блока.

В режиме *обратной связи по шифртексту* (CFB) открытый текст разбивается не обязательно на 64-битовые блоки. Размер блока задается с помощью числового параметра t, $1 \leqslant t \leqslant 64$. Открытый текст m в этом случае представляется в виде $m = m_1 m_2 \ldots m_n$, где каждая его часть m_i является t-битовым блоком. С самого начала в сдвиговый 64-разрядный регистр записывается некоторое значение s_0, которое может быть либо частью секретного ключа, либо пересылаться в открытом виде перед передачей шифртекста сообщения (таким образом, чтобы его можно было каждый раз менять). Затем для $i = 1, 2, \ldots, n$

блок c_i шифртекста вычисляется как $c_i = m_i \oplus p_i$, где через p_i обозначены t старших битов $DES_k(c_{i-1})$. Содержимое сдвигового регистра изменяется отбрасыванием t значений его старших разрядов и конкатенацией к ним справа c_i, согласно формуле $s_i = (2^t s_{i-1} + c_i) \bmod 2^{64}$. Аналогичным образом осуществляется и дешифрование: для $i = 1, 2, \ldots, n$ точно таким же способом вычисляются p_i и s_i, после чего m_i восстанавливаются как $m_i = c_i \oplus p_i$. Отметим, что при этом самые младшие $64 - t$ битов каждого результата DES-шифрования отбрасываются.

В режиме *обратной связи по выходу* (OFB) также имеется блок переменного размера t и сдвиговый регистр, который инициализируется некоторым значением s_0. Однако на этот раз при каждом новом использовании системы должны применяться разные s_0, и поэтому перед началом передачи шифртекста их необходимо пересылать в открытом виде (таким образом, секретной здесь является только 56-битовая часть ключа k самого алгоритма DES). Пусть m, как обычно, представляется в виде $m_1 m_2 \ldots m_n$. Тогда для $i = 1, 2, \ldots, n$ блок c_i вычисляется точно так же, как и в режиме CFB: $c_i = m_i \oplus p_i$, где через p_i обозначены t значений старших разрядов $DES_k(s_{i-1})$. Отличие от CFB заключается в том, что меняется содержимое сдвигового регистра. Официально утвержденная в NBS версия режима OFB устанавливает, что $s_i = (2^t s_{i-1} + p_i) \bmod 2^{64}$, однако в [135, 226] описываются слабости, существующие в этой схеме. Таким образом, при получении p_i лучше всего использовать $t \leqslant 8$, хотя для обеспечения обратной связи предпочтительнее, тем не менее, использовать $t = 64$. Поэтому, чтобы совместить эти два противоречивых требования, надежнее всего будет использовать вариант, основанный на «официальной» версии OFB режима: p_i и c_i должны вычисляться так, как это уже было указано выше (с небольшим значением t), но при этом необходимо просто заменять s_i на $DES_k(s_{i-1})$ — обратите внимание, что тогда нам уже не нужен сдвиговый регистр. В рассматриваемом режиме DES применяется для генерации псевдослучайной последовательности $p_1 p_2 \ldots p_n$, которая тем самым используется как одноразовый шифр для открытого текста (см. § 4.5). Поскольку эта последовательность не зависит от открытого текста, то всякий раз, когда k и s_0 фиксированы, будет вырабатываться один и тот же шифртекст. Вот почему s_0 каждый раз необходимо менять. Проце-

дура дешифрования должна быть очевидна. Хотя этот режим и не является самосинхронизирующимся, зашифрованный в нем с помощью DES открытый текст прекрасно восстанавливается, даже если какой-то из блоков соответствующего шифртекста был получен с ошибками.

Уитфилд Диффи предложил некоторую модификацию режима OFB: так называемый *режим счетчика*. Этот режим отличается от OFB только тем, что в нем вместо сдвигового регистра используется счетчик, а изменения задаются простой формулой $s_i = 1 + s_{i-1}$, где «+» обозначает обычное арифметическое сложение. Если DES действительно является хорошим алгоритмом шифрования, то режим счетчика должен быть ничуть не слабее, чем OFB. Более того, если при работе в таком режиме в DES возникает какая-нибудь случайная ошибка, то при этом неправильно расшифровывается только один блок.

Несмотря на свою слабость, использование режима ECB скорее всего неизбежно в прикладных программах баз данных, когда требуется произвольный доступ для чтения/записи к различным полям. Тем не менее, если произвольным должен быть только доступ для чтения, то последовательно зашифрованный файл может создаваться либо в CBC, либо в CFB режиме. Однако *никогда* не следует использовать ни режим OFB, ни режим счетчика, если шифруемый файл восприимчив к изменениям посредством добавления или стирания записей — смотрите упражнение 3.1 в [142].

Отметим, что в режимах CFB и OFB, а также в режиме счетчика, криптографическая система, которая лежит в основе их обсуждения (в нашем случае — это криптосистема DES), используется только для шифрования. По этой причине, криптографические системы с открытым ключом (см. § 4.3) типа RSA (см. § 4.4) должны использоваться только в режиме CBC, при условии, что необходимость в произвольном доступе для чтения/записи не вынудит использовать их в режиме ECB.

Глава 4

Системы с открытым ключом

§ 1. Однонаправленные функции

Два понятия — однонаправленной функции и однонаправленной функции с «потайным ходом», или «лазейкой» — являются центральными для всей криптографии с открытым ключом. Рассмотрим произвольные множества X и Y, а также некоторую функцию $f: X \to Y$. Обозначим через $f[X]$ область значений f. Функция f называется *однонаправленной*, если ее значение $f(x)$ может быть легко вычислено для каждого аргумента $x \in X$, тогда как почти для всех $y \in f[X]$ нахождение такого $x \in X$, что $f(x) = y$, является трудновычислимым. Однонаправленные функции не следует путать с функциями, которые являются математически необратимыми из-за того, что они не взаимнооднозначны или не «на» (то есть из-за того, что либо существует несколько различных значений x, таких что $f(x) = y$, либо же их нет вовсе).

Нынешнее состояние наших знаний пока еще не позволяет нам доказать, что однонаправленные функции (обоих типов) вообще существуют, так как их существование разрешило бы знаменитую ($\mathcal{P} \stackrel{?}{=} \mathcal{NP}$)-проблему [186]. Более того, теория NP-полноты не кажется вполне убедительной, чтобы обеспечить даже простую аргументацию существования таких функций [64, 169, 207]. И все же, несмотря ни на что, у нас в этом смысле имеются *кандидаты* среди функций, эффективно вычислять которые мы умеем, хотя при этом никаких эффективных алгоритмов вычисления обратных им (во всяком случае среди общедоступных!) до сих пор не известно.

Первым простым примером кандидата на однонаправленную функцию является *целочисленное умножение*. В самом деле, известно, что перемножить два любых, пусть и очень больших, числа относительно нетрудно, тогда как даже самый мощный из существующих сейчас компьютеров не в состоянии разложить на множители с помощью наилучшего имеющегося в его распоряжении алгоритма четырехсотзначное десятичное число, являющееся произведением двух примерно одинакового размера простых чисел. Конечно, необходимо понимать, что «не в состоянии» здесь означает «не в состоянии за приемлемое время (например, в течении человеческой жизни или за время, ограниченное возрастом вселенной)»[1].

Другим очень важным примером кандидата на однонаправленную функцию является *модульное возведение в степень*, или *модульное экспоненцирование* (с фиксированными основанием и модулем). Пусть n и a — целые числа, такие, что $1<a<n$, и пусть также $\mathbf{Z}_n=\{0,1,2,\ldots,n-1\}$. Тогда модульным возведением в степень, или экспоненцированием (относительно основания a и модуля n), называется функция $f_{a,n}\colon \mathbf{Z}_n \to \mathbf{Z}_n$, определяемая как $f_{a,n}(m) = a^m \bmod n$, где $i \bmod j$ — положительный остаток при делении i на j. Сразу не очевидно, что такую функцию можно вычислить эффективно, когда длина каждого из трех параметров a, n и m составляет несколько сотен знаков. То, что, возможно, это и в самом деле так, лучше всего продемонстрировать на примере:

$$a^{25} = (((a^2 \cdot a)^2)^2)^2 \cdot a \ ,$$

который показывает, что a^{25} можно вычислить с помощью всего лишь четырех возведений в квадрат и двух умножений. При вычислении $a^m \bmod n$ приведение по модулю n желательно делать после каждого возведения в квадрат и каждого умножения, чтобы избежать получения очень больших целых чисел. Если

[1] Карл Померанц [290], как-бы подтверждая, что по-американски время — деньги, разработал проект специализированного компьютера, который при использовании самого эффективного алгоритма факторизации в принципе позволяет при все более увеличивающемся объеме оборудования разлагать на множители за заданное время числа любой длины, и оценил стоимость разработки такого компьютера в зависимости от размера этих чисел. Для факторизации четырехсотзначного десятичного числа в течение одного года она равна примерно 100 миллиардам миллиардов долларов!

экспонента m является числом длиной l битов то, для того чтобы вычислить $a^m \bmod n$, приведенному ниже рекурсивному алгоритму потребуется выполнить от l до $2l$ модульных умножений (считая при этом умножением и возведение в квадрат):

function *expmod*(a, m, n)
  if $m = 0$ **then return** 1
  if m четно **then return** $\left(expmod\left(a, \frac{m}{2}, n\right)\right)^2 \bmod n$
  else return $(a \cdot expmod(a, m-1, n)) \bmod n$.

По аналогии с вещественным анализом задача вычисления функции, обратной модульному возведению в степень, известна как *задача дискретного логарифмирования*: даны целые числа a, n и x, требуется найти такое целое m (если конечно оно существует), что $a^m \bmod n = x$. Например, $5^4 \bmod 21 = 16$. Так что 4 — это дискретный логарифм 16 с основанием 5 по модулю 21. При желании можно проверить, например, что число 3 вообще не имеет логарифма с основанием 5 по модулю 21. В настоящее время вычисление больших модульных экспонент можно выполнить очень быстро даже на «персоналке», но, тем не менее, на сегодняшний день неизвестно ни одного алгоритма вычисления дискретных логарифмов больших чисел за приемлемое время даже на самых мощных, самых быстродействующих суперкомпьютерах. При этом, хотя мы и не можем доказать, что эффективных таких алгоритмов вообще не существует, имеются веские основания предполагать, что модульное возведение в степень (с фиксированными основанием и модулем) действительно является однонаправленной функцией.

Очевидно, что однонаправленные функции не могут непосредственно использоваться в качестве криптосистем (когда m шифруется как $f(m)$), поскольку тогда даже *законный* получатель не смог бы определить открытый текст! Позднее мы увидим, что несмотря на это они полезны для защиты паролей (см. § 5.3). Гораздо более употребимым в криптографии является понятие *однонаправленной функции с потайным ходом (лазейкой)*. Функция $f: X \to Y$ называется однонаправленной функцией с потайным ходом (или, что то же самое, с лазейкой), если, во-первых, не только сама f, но и функция f^{-1}, обратная ей, могут быть вычислены эффективно, а во-вторых, даже если такой эффек-

тивный алгоритм вычисления f известен, то никакое, пусть самое полное, описание его работы не должно давать возможности построить эффективный алгоритм вычисления f^{-1}. Секрет, с помощью которого, тем не менее, можно эффективно вычислить функцию f^{-1}, как раз и называется *потайным ходом*, или *лазейкой* для функции f.

Наш первый кандидат на однонаправленную функцию с потайным ходом во многом похож на нашего второго кандидата на просто однонаправленную функцию. Это — модульное возведение в степень, но *с фиксированной экспонентой и модулем*. Пусть m и n — целые числа, а $\mathbb{Z}_n$ определено так же, как ранее. Тогда модульное возведение в степень (относительно экспоненты m и модуля n) есть функция $g_{m,n}: \mathbb{Z}_n \to \mathbb{Z}_n$, определяемая следующим образом: $g_{m,n}(a) = a^m \bmod n$. Необходимо быть уверенным в понимании различия между функциями $f_{a,n}$ и $g_{m,n}$. Опять по аналогии с вещественным анализом, операция, обратная $g_{m,n}$, известна как *извлечение корня m-ой степени из x по модулю n*: даны целые числа m, n и x, найти такое целое a (если оно существует), что $a^m \bmod n = x$. Например, 5 — это корень 4-ой степени из 16 по модулю 21, потому что, как мы уже видели, $5^4 \bmod 21 = 16$. Очевидно, что 2 также является корнем 4-ой степени из 16 по модулю 21. Можете ли Вы найти другие корни 4-ой степени из 16 по модулю 21? Найдите целое число x, которое не имеет корней 4-ой степени по модулю 21.

В том случае, когда экспонента m и модуль n фиксированы, мы уже приводили эффективный алгоритм вычисления $g_{m,n}(a)$ для любого основания a. В противоположность задаче дискретного логарифмирования, тем не менее известно, что существует также и эффективный алгоритм извлечения корня m-ой степени из x по модулю n (или выяснения, что такого корня нет) для любого заданного x. Любопытный феномен заключается в том, что мы не знаем, как эффективно построить этот эффективный алгоритм при заданных лишь m и n. Иными словами, известно, что функция $g_{m,n}$ на самом деле является *не* однонаправленной, поскольку мы знаем, что она может быть эффективно обращена, но, несмотря на этот факт, не знаем, как это сделать! Тем не менее, легко построить эффективный алгоритм вычисления m-ого корня по модулю n при условии, что известно разложение n на простые множители. Именно по этой причине $g_{m,n}$ и является

кандидатом на однонаправленную функцию с потайным ходом, для которой m и n используются как открытая информация, тогда как разложение служит в качестве секретного потайного хода. Мы еще увидим, каким образом это может быть использовано, когда будем изучать знаменитую криптосистему RSA (см. § 4).

Важным частным случаем модульного возведения в степень является тот, при котором экспонента равна 2, а модуль — число некоторого специального вида. Для понимания того, что это и есть еще один кандидат на однонаправленную функцию с потайным ходом, необходимы дополнительные сведения из теории чисел. Если p и q — два различных больших простых числа примерно одинакового размера, и кроме того и p, и q сравнимы с 3 по модулю 4, то мы будем говорить, что их произведение $n = p \cdot q$ является *целым числом Блюма*. Определим $\mathbb{Z}_n^*$ как множество целых чисел от 1 до $n-1$, которые не делятся ни на p, ни на q. Наконец, определим QR_n как подмножество множества $\mathbb{Z}_n^*$, состоящее из чисел, которые являются совершенными квадратами по модулю n. Элементы QR_n известны как *квадратичные вычеты* по модулю n. В качестве небольшого примера положим $p = 19$, а $q = 23$, откуда имеем $n = 437 = 19 \cdot 23$. Тогда 135 является элементом $\mathbb{Z}_n^*$, в то время как 133 — нет (поскольку $133 = 19 \cdot 7$). Более того, 135 не является квадратичным вычетом по модулю 437, так как не существует целого числа a, такого, что $a^2 \equiv 135 \pmod{437}$, тогда как 139 является таковым, поскольку $24^2 = 576 \equiv 139 \pmod{437}$.

Теперь сформулируем без доказательства несколько утверждений, необходимых для понимания всего дальнейшего изложения. Число элементов в $\mathbb{Z}_n^*$ равно $(p-1)(q-1)$, причем в точности четвертую их часть составляют квадратичные вычеты. Каждый квадратичный вычет допускает ровно четыре различных «квадратных корня» в $\mathbb{Z}_n^*$, из которых лишь один единственный является квадратичным вычетом. Этот особый корень мы назовем *примитивным (первообразным)* квадратным корнем. В нашем примере 24 — это примитивный квадратный корень из 139 по модулю 437, другими тремя корнями являются числа 185, 252 и 413. Имеющий криптографическое значение факт заключается в том, что способность определять примитивные квадратные корни по модулю такого числа n оказывается *вычислительно эквивалентной* умению раскладывать это n на множители.

Иначе говоря, тот, кто знает разложение n на множители, может эффективно вычислять и примитивные квадратные корни по модулю n, тогда как такие вычисления столь же трудны, сколь и факторизация n, для того, кто сомножителей n не знает.

Теперь наш второй кандидат на однонаправленную функцию с потайным ходом совершенно очевиден. Вы случайно выбираете p и q и вычисляете $n = p \cdot q$, которое открыто объявляете. После этого любой человек может эффективно возводить в квадрат по модулю n, но только ваш друг сможет эффективно произвести обратные вычисления (в предположении, что факторизация трудна). Открытой информацией здесь является число n, а секретным потайным ходом, опять таки, служит его разложение на множители.

Для более основополагающих сведений по вычислительной теории чисел для криптографии мы отсылаем читателя к замечательной книге Евангелоса Кранакиса [244].

§ 2. Открытое распределение ключей

Одна из основных трудностей, которую всегда необходимо учитывать в больших многопользовательских криптографических системах заключается в том, что каждая пара ее пользователей, предполагающих обмениваться друг с другом конфиденциальной информацией, должна заранее выработать свой обоюдный секретный ключ. Как же, например, должны поступить два конкретных пользователя, которые неожиданно захотели установить между собой секретную связь, но еще не сообщали друг другу никакой секретной информации? Традиционное решение для них — где-нибудь встретиться и выработать совместный секретный ключ, или использовать для его передачи доверительного курьера. Оба этих решения являются очень неоперативными и дорогостоящими. К тому же они не могут быть абсолютно надежными.

Предназначение *криптосистемы с открытым распределением ключей* как раз и заключается в том, чтобы позволить двум пользователям *Алисе* и *Бобу* выработать в результате переговоров друг с другом по несекретному каналу связи такой совместный секретный ключ, который «нарушитель» не мог бы разгадать даже после прослушивания всех этих переговоров.

Точнее говоря, хотелось бы иметь такой протокол, с помощью которого *Алиса* и *Боб* могли обмениваться сообщениями m_1 (от *Алисы* к *Бобу*), m_2 (от *Боба* к *Алисе*), ..., до тех пор пока они оба окончательно не договорятся между собой о некотором совместном ключе k, причем они должны сделать это так, чтобы определить k из знания только m_1, m_2, ...было практически невозможно. Подчеркнем еще раз, что этого необходимо добиться даже в том случае, если *Алиса* и *Боб* не обменивались заранее никакой информацией, которая не была бы известна нарушителю.

Первый кажущийся абсолютно невозможным протокол, который тем не менее достигает этой цели, был предложен Диффи и Хеллманом [157] в 1976 году. Он основывается на задаче дискретного логарифмирования, рассмотренной в § 1. Пусть n — некоторое большое целое число, и пусть g — другое целое, лежащее строго между 1 и $n-1$. В качестве первого шага протокола Диффи–Хеллмана *Алиса* и *Боб* уславливаются об n и g посредством несекретного канала связи (в качестве альтернативы n и g могли бы быть *стандартными* параметрами, применяемыми всеми пользователями системы). Затем *Алиса* выбирает некоторое большое целое число x и вычисляет $X = g^x \bmod n$. Соответственно, *Боб* выбирает число y и вычисляет $Y = g^y \bmod n$. После этого *Алиса* и *Боб* обмениваются числами X и Y по тому же несекретному каналу связи, *сохраняя в секрете* x *и* y (*Алиса* при этом знает только x, а *Боб* — только y). Наконец, *Алиса* вычисляет $Y^x \bmod n$, а *Боб*, соответственно, вычисляет $X^y \bmod n$. Оба эти значения, очевидно, равны между собой, так как каждое из них равно $g^{(x \cdot y)} \bmod n$. Это как раз и есть тот самый ключ k, который *Алиса* и *Боб* хотели совместно выработать.

Нарушитель при таком протоколе сталкивается с задачей вычисления k из пересылаемых по несекретному каналу чисел g, n, X и Y. Очевидным подходом для нарушителя было бы вычислить x из g, n и X (или по крайней мере некоторое $\hat{x}$, такое, что $g^{\hat{x}} \bmod n = X$, так как для любого подобного $\hat{x}$ всегда $Y^{\hat{x}} \bmod n = k$). Однако это в точности задача нахождения дискретного логарифма, которая считается практически невыполнимой. Кроме того, никто пока не придумал способа вычислять k эффективно из g, n, X и Y, как никто и не смог доказать, что это невозможно, или хотя бы продемонстрировать, что не

существует лучшего способа сделать это, по сути дела не находя в процессе вычисления дискретного логарифма. Вот почему, вообще говоря, правомерно предполагать, что вычисление k может быть осуществлено эффективно, даже если окажется, что решение задачи дискретного логарифмирования действительно практически невыполнимо.

Выбор g и n может оказывать существенное влияние на эффективность и надежность представленного только что проекта криптосхемы. Для того чтобы сократить размер возможных окончательных значений k, важно чтобы функция модульного возведения в степень $f_{g,n}: \mathbb{Z}_n \to \mathbb{Z}_n$ (как она определена в § 1) была настолько почти взаимнооднозначной, насколько это возможно. В том случае, когда n является простым числом, всегда существует такое g, что $g^x \bmod n$ принимает каждое из значений в промежутке от 1 до $n-1$, когда x пробегает значения из того же интервала. Подобные g, которые называются образующими элементами, или генераторами, *циклической группы* $\mathbb{Z}_n$, и были бы искомыми. При этом надежнее выбрать n таким образом, чтобы $\frac{n-1}{2}$ также было простым [288]. Кроме того, можно было бы все указанные операции проводить в полях Галуа $GF(2^k)$, описание методов вычисления в которых, к сожалению, выходит далеко за рамки этой книги [41]. Они предоставляют возможность намного более эффективно осуществлять умножение (а следовательно, и возведение в степень), так как позволяют избежать при вычислениях необходимости переносов и приведений по модулю [5, 4]. Однако в этом случае размер ключа будет намного длиннее. С другой стороны, длины всех чисел, участвующих в арифметических операциях можно существенно сократить, если использовать вычисления над эллиптическими кривыми [6].

Несмотря на то, что очень большие дискретные логарифмы возможно и являются трудновычислимыми, хотелось бы предупредить читателя, что для их вычисления существуют алгоритмы намного лучшие, чем исчерпывающий поиск. Самые лучшие из известных в настоящее время алгоритмов, когда n является простым числом, описаны в [124], а при вычислениях в $GF(2^k)$ — в [281]. При выборе размера ключа такие поля должны тщательно анализироваться. Фирмой Hewlett-Packard однажды бы-

ла разработана злополучная аппаратная реализация, использующая $GF(2^{127})$ [368]. Однако ключи в ней были быстро раскрыты с применением методов, которые изложены в [45].

В предположении, что n и g являются стандартными параметрами, интересной альтернативой описанному ранее интерактивному протоколу является организация и использование некоторого единого для всех пользователей *каталога*. Каждый пользователь записывает в этот каталог свой собственный открытый ключ X, вычисленный как $g^x \bmod n$ в соответствии с личным случайно выбранным секретным ключом x. Это позволяет любым двум пользователям сформировать свой совместный секретный ключ даже до их предварительной договоренности о нем друг с другом. Основной недостаток такого общедоступного каталога состоит в том, что он не поддерживает достаточно частые изменения пользователями личных секретных ключей. Мы намереваемся вернуться к этой теме в § 7.

Другой подход к проблеме распределения ключей обеспечивает предложенная Чарльзом Беннеттом и Жилем Брассаром *квантовая криптография* [25]. Ее надежность не зависит от недоказанных предположений о вычислительной сложности, основанных на трудности вычислений каких бы то ни было функций. Более того, она остается таковой даже тогда, когда нарушитель имеет неограниченные вычислительные ресурсы, или даже если, все-таки, $\mathcal{P}=\mathcal{NP}$ [27, 26, 28]. Квантовое открытое распределение ключей описывается в последней, седьмой главе этой книги.

§ 3. Теория криптосистем с открытым ключом

Система открытого распределения ключей, так как она описана в § 2, позволяет двум сторонам сформировать совместную часть некоторой распределенной секретной информации. Однако при этом ни одна из сторон не имеет никакого непосредственного влияния на то, какой окажется эта информация. Если бы *Алиса* захотела передать *Бобу* единственное *специальное* сообщение, то за использованием системы открытого распределения ключей должно было бы последовать использование криптосистемы с секретным ключом, в которой первоначальная совместно сформированная ими информация сохранялась бы в качестве их общего секретного ключа.

Криптосистемы же с открытым ключом могут *непосредственно* использоваться для шифрования наиболее значимых сообщений. Криптографические системы *с открытым ключом* во многом подобны криптосистемам с секретным ключом. Они состоят из *пространства ключей* $\mathcal{K}$, и для каждого $k \in \mathcal{K}$, из *пространства открытых сообщений* $\mathcal{M}_k$, *пространства шифртекстов* $\mathcal{C}_k$ и пары функций $E_k: \mathcal{M}_k \to \mathcal{C}_k$ и $D_k: \mathcal{C}_k \to \mathcal{M}_k$, таких, что $D_k(E_k(m)) = m$ для любого открытого текста $m \in \mathcal{M}_k$. Так же как и в криптосистемах с секретным ключом, эффективные алгоритмы для вычисления и E_k, и D_k, должны легко получаться для каждого ключа k. Мы будем называть алгоритмы, полученные таким образом, *естественными алгоритмами*. Важной новой отличительной особенностью является то, что E_k должна быть однонаправленной функцией с потайным ходом — необходимо, чтобы было практически невозможно построить никакого эффективного алгоритма для вычисления D_k (но только естественного такого алгоритма), зная описание естественного алгоритма для вычисления E_k. В частности, это означает, что значение k *не должно* присутствовать в явном виде в естественном алгоритме шифрования.

Криптографические системы с открытым ключом используются следующим образом. Каждый пользователь раз и навсегда выбирает для себя некоторый случайный ключ $k \in \mathcal{K}$. Этот ключ он использует для получения обоих естественных алгоритмов E_k и D_k. Затем он делает публично доступным свой алгоритм шифрования E_k, возможно посредством использования некоторого справочника, но при этом хранит в строгой тайне свой алгоритм дешифрования D_k. Тогда, если один из пользователей криптосистемы захочет послать другому некоторое сообщение, то он найдет в справочнике его открытый алгоритм шифрования и использует этот алгоритм для того, чтобы зашифровать свое сообщение. В этом случае после его пересылки, используя собственный секретный ключ, только законный получатель сможет расшифровать полученный шифртекст. Заметим, что в противоположность криптосистемам с секретным ключом, если *Алиса*, зашифровав некоторое сообщение m для *Боба*, сохранит шифртекст c, но потеряет соответствующий ему открытый текст (или забудет все, что в нем содержалось), то она не будет иметь никаких преимуществ перед нарушителем в раскрытии m из c.

Также в противоположность криптосхемам с секретным ключом, если нарушитель, перехватив шифртекст, знает, для кого он был зашифрован, то он может использовать открытый алгоритм шифрования для проверки любого конкретного предположения о том, каким может быть соответствующий открытый текст. Возможность формировать шифртексты из открытых сообщений по своему выбору приводит к сокращению числа классических уровней атак, которые обсуждались в связи с криптосистемами с секретным ключом (см. § 3.1). Однако иногда может применяться следующая важная атака:

- *Атака на основе выбранного шифртекста*: против определенного пользователя, функциями шифрования и дешифрования которого являются E_k и, соответственно, D_k, криптоаналитик задается выбранными шифртекстами c_1, c_2, ..., c_i и подбирает отвечающие им открытые тексты $m_1 = D_k(c_1)$, $m_2 = D_k(c_2)$, ..., $m_i = D_k(c_i)$ при условии, что такие осмысленные открытые тексты существуют. Его цель заключается в том, чтобы либо определить ключ k, либо построить какой-нибудь эффективный алгоритм вычисления D_k, либо (за неимением и той, и другой возможности) пытаться найти m_{i+1} из некоторого нового шифртекста $c_{i+1} = E_k(m_{i+1})$.

Криптосистемы с открытым ключом, в принципе, могут существовать лишь в том случае, если существуют не только просто однонаправленные функции, но *и* однонаправленные функции с потайным ходом. В самом деле, функции шифрования должны быть однонаправленными функциями с потайным ходом, а процесс, который с помощью ключа обеспечивает естественный алгоритм шифрования, должен реализовывать просто однонаправленную функцию. В результате всего этого получается, что мы не знаем, как доказать существование криптосистем с открытым ключом. Таким образом, все наши рассуждения о них могут быть не более, чем осуществляемый весьма своеобразным способом разговор о пустом множестве, даже несмотря на то, что возможная реализуемость самого понятия криптосистемы с открытым ключом уже была формально продемонстрирована в относительных утверждениях [68, 65, 66].

Поэтому точно так же, как и в случае однонаправленных

функций (с потайным ходом), мы должны быть удовлетворены своими кандидатами на криптосистемы с открытым ключом. Две из наиболее известных криптосистем, претендующих на роль таких кандидатов, были предложены сразу после того, как Диффи и Хеллман ввели понятие криптографии с открытым ключом [157]. Одна из них — это так называемая *ранцевая* криптосистема Меркля и Хеллмана [266, 214], которая вскоре была раскрыта [150, 161, 317, 3, 88, 245, 89]. И хотя существуют еще нераскрытые варианты оригинальной криптосхемы, этим вариантам, по видимому, не стоит доверять. Брикелл написал очень подробную историю криптоанализа ранцевой криптографической системы [90]. Другая наиболее известная криптосистема с открытым ключом все еще остается несокрушимой, и мы в следующем параграфе опишем ее достаточно детально. Были предложены и другие системы, такие как системы Макэлиса [262] и Эль-Гамаля [166], но мы не будем обсуждать их здесь. По поводу исчерпывающего обзора криптографических систем с открытым ключом обращайтесь к [246, 279]. Захватывающая история создания криптографии с открытым ключом отражена с статье самого Уитфилда Диффи [156].

§ 4. Криптосистема RSA

Самой первой криптографической системой с открытым ключом, из тех что были предложены в открытой литературе вообще, является криптосистема Ривеста, Шамира и Эдлемана [307]. Она стала известна под названием RSA или MIT криптосистемы. Эта криптосистема основывается на вере в то, что модульное возведение в степень при фиксированных экспоненте и модуле является однонаправленной функцией с потайным ходом (см. § 1). Для первоначального ознакомления с этой криптосистемой обращайтесь к статье Мартина Гарднера [185].

Пусть p и q — два больших различных простых числа, $n = p \cdot q$, а e — некоторое целое, взаимно простое с $(p-1)(q-1)$. Тогда для криптосистемы RSA в качестве *личного (секретного)* ключа может быть выбрана любая такая тройка $k = \langle p, q, e \rangle$. Пусть также каждое из соответствующих пространств открытых текстов $\mathcal{M}_k$ и шифрованных сообщений $\mathcal{C}_k$ суть $\mathbb{Z}_n$, т.е. множество неотрицательных целых чисел, меньших n. (Если при этом ре-

альные сообщения окажутся слишком длинными, чтобы принадлежать $\mathbb{Z}_n$, то их необходимо разбивать на части и зашифровать, используя режим шифрования со сцеплением блоков (CBC), который описан в § 3.5). Тогда соответствующая ключу k функция шифрования $E_k: \mathcal{M}_k \to \mathcal{C}_k$ определяется как $E_k(m) = m^e \bmod n$. Для того чтобы полностью определить естественный алгоритм ее вычисления, достаточно записать n и e в открытый справочник. Такая пара $\langle n, e\rangle$ чисел называется открытым ключом. Он легко вычисляется простым перемножением p и q из личного (секретного) ключа $\langle p, q, e\rangle$.

Вспомним из § 1, что E_k является кандидатом на однонаправленную функцию с потайным ходом, и хотя существует эффективный алгоритм вычисления обратной ей функции D_k, но мы не знаем, как получить его эффективно, задаваясь только естественным алгоритмом вычисления E_k (т.е. только при заданных n и e). Тем не менее такой эффективный алгоритм вычисления D_k легко построить, задав дополнительную секретную информацию, а именно — числа p и q, являющиеся множителями числа n. С этой целью, используя обобщеный алгоритм Евклида для нахождения наибольшего общего делителя, необходимо вычислить целое число d, такое, что $e \cdot d \equiv 1 \pmod{\varphi(n)}$, где $\varphi(n) = (p-1)(q-1)$. Тогда по известной теореме Эйлера $m^{e\cdot d} \equiv m \pmod{n}$ для каждого целого числа m и, следовательно, $m^{e\cdot d} \bmod n = m$ при условии, что $0 \leqslant m < n$, которое обеспечивается, когда $m \in \mathcal{M}_k$. Функция дешифрования $D_k: \mathcal{C}_k \to \mathcal{M}_k$ в связи с этим определяется как $D_k(c) = m^d \bmod n$, и для ее вычисления может быть использован также эффективный алгоритм модульного возведения в степень. (Для заданных чисел p и q существует даже более эффективный алгоритм вычисления D_k, который основан на китайской теореме об остатках [292].)

Таким образом, суммируя все сказанное, каждый пользователь криптосистемы RSA должен совершенно случайно и раз и навсегда выбрать для себя подходящие целые числа p, q и e и вычислить с их помощью d. После чего он должен сделать свой открытый ключ $\langle e, n\rangle$ доступным в пользовательском справочнике, но при этом сохранять d в секрете. Это дает возможность остальным пользователям зашифровывать посылаемые ему сообщения, которые только он один потом сможет расшифровать. Для того чтобы эта идея могла быть реализована на практи-

ке, решающим является удовлетворение требования, чтобы генерация больших случайных простых чисел и вычисление d были легкоосуществимы. К счастью, проверка чисел на простоту оказывается действительно легче их разложения на множители, благодаря вероятностным алгоритмам Роберта Соловея и Волкера Штрассена [339] и Майкла Рабина [298]. Проверке чисел на простоту посвящено несколько интересных работ, в частности, [289, 233, 271]. По поводу тесно связанного с криптографией вопроса о *генерации* подходящих простых чисел (но не их *проверки* на простоту) обращайтесь к [125, 16, 260]. Относительно описания расширенного алгоритма Евклида для вычисления d смотрите [7, 142] (но если же вы предпочитаете переоткрыть расширенный алгоритм Евклида самостоятельно, то можете найти полезными те указания, которые даны к задаче 8.5.12b из [77]).

В качестве небольшого примера, пусть $p = 19$ и $q = 23$, так что $n = 437$, а $\varphi(n) = 396$. Пусть также $e = 13$, а значит $d = 61$, поскольку $13 \cdot 61 = 793 = 2\varphi(n) + 1$. Тогда сообщение открытого текста $m = 123$ будет зашифровано как $c = 123^{13} \bmod 437 = 386$. Действительно, $386^{61} \bmod 437 = 123$.

Особо отметим ситуацию, когда криптоаналитик, перехвативший шифртекст $c = E_k(m)$, посланный известному пользователю, знает естественный алгоритм шифрования E_k, который используется отправителем для вычисления c. Такое предположение имеет два важных следствия. Если бы нарушитель мог в точности угадать открытый текст сообщения m, то он был бы в состоянии точно так же, как и отправитель, вычислить $E_k(m)$, а затем проверить свою догадку, сравнив полученный результат с шифртекстом c. Учитывая возможность исчерпывающего поиска, такая угроза является довольно опасной, если число возможных открытых текстов сравнительно невелико. Если же добавлять в короткие сообщения случайные биты, то эта трудность может быть до некоторой степени разрешена, однако, более приемлемым решением несомненно является использование *вероятностного шифрования* (см. § 6). При желании можете обратиться также к [308].

Более существенным для криптосистемы RSA является другое следствие из того факта, что нарушителю доступен открытый алгоритм шифрования. Действительно, он знает, что $c = m^e \bmod n$ для известных значений c, e и n (хотя и не знает m).

Поэтому, если бы он мог разложить n на множители (раскрыв таким образом личный секретный ключ $\langle p, q, e\rangle$ законного получателя шифртекста c), то он мог бы получить и $\varphi(n)=(p-1)(q-1)$, после чего, применив расширенный алгоритм Евклида, вычислить d, чтобы затем найти $m = c^d \mod n$. К счастью, неизвестно никакого алгоритма, с помощью которого можно было бы разложить на множители четырехсотзначное десятичное число за приемлемое время, и поэтому считается абсолютно надежным выбирать числа p и q длиной примерно в двести (десятичных) знаков. Выбирать p и q необходимо с особой тщательностью, чтобы при использовании известных алгоритмов факторизации не предоставить криптоаналитику никаких «зацепок». В частности, наибольший общий делитель чисел $p-1$ и $q-1$ должен быть небольшим, а оба они, как $p-1$, так и $q-1$, должны иметь большие простые делители. Блэкли и Борош предложили выбирать p и q так, чтобы и $\frac{p-1}{2}$, и $\frac{q-1}{2}$ также были простыми числами [47]. Для более широкого обсуждения подобных вопросов рекомендуем обратиться к [142]. Джон Гордон в [203] и Ули Маурер в [260] предлагают эффективные способы выбора таких подходящих (сильных) простых чисел. См. также [218].

Даже если считать, что факторизация действительно трудна, остается неизвестным, является ли столь же трудным само раскрытие RSA. Ведь вполне вероятно, что d может быть вычислено из открытой информации e и n вообще без разложения n на множители. Но возможно также и то, что значение d (а, стало быть, и значения множителей числа n) действительно трудно вычислить практически из e и n, даже если существует какой-то иной эффективный алгоритм раскрытия m из e, n и $m^e \mod n$. Майклом Рабином в [297] и Хуго Уилльямсом в [360] были предложены другие криптосистемы с открытым ключом, для которых они доказали, что раскрытие в них открытого текста из всей доступной нарушителю информации оказывается таким же трудным, как и разложение на множители больших чисел. Однако эти криптосистемы сразу же раскрываются при атаке на основе выбранного шифртекста. Тем не менее теоретически существуют криптосистемы, которые являются вероятностно секретными при атаке на основе выбранного шифртекста [276]. Наконец, даже если m и в самом деле практически трудно вычисляется

из той информации, которая доступна нарушителю, быть может еще остается какой-нибудь способ получить эффективно некоторую *частичную* информацию об открытом тексте, например, такую, как половина битов сообщения m. Все эти возможные слабости снимает применение вероятностного шифрования, которое мы рассмотрим в § 6, при единственном предположении, что разложение на множители является действительно трудной задачей.

Пользователи криптосистемы RSA должны знать о том, что эта криптосхема является слабой при некоторых видах атак на основе выбранного шифртекста [134, 143]. Предположим, например, что нарушитель перехватил некоторое $c = m^e \mod n$, где e и n — открытая информация. Ему хотелось бы определить m. При атаке на основе выбранного шифртекста ему предоставляется возможность передать некоторое $\hat{c}$ законному получателю, с тем чтобы получить в ответ соответствующее $\hat{m}$, такое, что $\hat{c} = \hat{m}^e \mod n$. Разумно ожидать, что получатель может уклониться от ответа, если нарушитель попытается использовать непосредственно $\hat{c} = c$ (в противном случае никакая криптосистема, скорее всего, не должна быть надежной). Однако нарушитель в состоянии *замаскировать* свой вопрос, выбрав случайным образом некоторое x из $\mathbb{Z}_n^*$ и вычислив $\hat{c} = (x^e) \cdot c \mod n$. Исходный открытый текст m может быть тогда получен эффективно (при использовании обобщенного алгоритма Евклида), поскольку $m = \hat{m} \cdot x^{-1} \mod n$. Неизвестно, существует ли более хитроумная атака на основе выбранного шифртекста, *действительно* способная раскрыть RSA в том смысле, что позволит нарушителю вычислить множители n или, хотя бы, секретную экспоненту дешифрования d. Если бы это было так, то все последующие шифртексты можно было бы расшифровывать без какого бы то ни было обращения к законному получателю. Для первого знакомства в связи с этим смотрите [149].

Несмотря на все свои преимущества перед криптографическими системами с секретным ключом, RSA все же значительно медленнее, чем DES. Напомним, что аппаратная реализация DES в настоящее время способна достигать скорости 90 мегабит в секунду. Это заставляет рассматривать перспективу коммерческой применимости даже самых быстрых RSA устройств для шифрования скорее как не совсем удовлетворительную.

Так, например, на вентильной матрице Кочанского [238, 338] можно достичь примерно 5 килобит в секунду с 512-ти битовым ключом (что является безусловно самым нижним пределом на размер ключа, который вообще имеет смысл рассматривать). Было сообщение, опубликованное в [280] о более быстрой реализации, которая принадлежит Джиму Омуре, но и она (на время ее разработки была) более чем в тысячу раз медленнее, чем DES. Существовала также знаменитая «микросхема Ривеста», которая так никогда и не была отлажена [304, 305] и проект Брикелла, который в случае его реализации теоретически смог бы достичь 25 килобит в секунду. Об еще более быстрой разработке для практических применений сообщалось в [314]. Читайте также [282]. Обзор этих и некоторых других аппаратных реализаций RSA дан Эрнестом Брикеллом в [91]. Консультируйтесь также в [156, 279].

Под руководством Джина Виллемина была разработана очень быстрая микросхема, в которой скорость шифрования достигала 225 килобит в секунду при длине ключа в 508 бит [322]. В ней существенно использовалось знание сомножителей, составляющих модуль.

Необходимо отметить только, что чисто программные версии алгоритма RSA гораздо медленнее аппаратных. Наиболее быстрая (соответствующая) программная реализация RSA-шифрования, имевшая коммерческое применение, шифрует со скоростью 20 килобит в секунду (при длине модуля в 1024 бита) и 40 килобит в секунду (при длине модуля в 512 бит). Кроме того, интересна описанная в [160] программно-аппаратная реализация RSA, в которой используются возможности арифметики сигнального процессора Motorola DSP56000.

Следует упомянуть также о шифраторе CryptoCom компании Algorithmic Research, который был спроектирован для работы на IBM PC. Он шифрует очень быстро, так как в экспоненте у него в качестве открытого ключа всегда использует $e = 3$, но при этом RSA (поскольку CryptoCom — это гибридная система) применяется только при смене DES-ключей шифрования. Обычно использование малых экспонент в RSA очень опасно [210], хотя в приведенной ситуации и не слишком. По поводу криптоанализа RSA при использовании коротких секретных экспонент см. [356]. Один 512-битовый блок CryptoCom дешифрует примерно за 9 секунд, или, при разбиении записи открытого текста на отдельные

блоки, со скоростью около 57 бит в секунду! Фактически же, т.к. это гибридная система, то за достаточно большой промежуток времени она работает намного быстрее (см. § 7).

Относительно реализации RSA в университете Ватерлоо (Канада), в которой используется арифметика в полях характеристики 2 и скорость шифрования и дешифрования достигает 300 килобит в секунду, см. [5].

§ 5. Генерация псевдослучайных чисел

Двоичная последовательность называется *псевдослучайной*, если она выглядит как бессистемная и вероятностная, хотя на самом деле создавалась посредством вполне детерминированного процесса, который называется *псевдослучайным генератором*. Такие генераторы начинают свою работу с действительно случайной исходной последовательностью, называемой *начальной*, и детерминировано вырабатывают с ее помощью гораздо более длинную псевдослучайную последовательность. В этом смысле псевдослучайные генераторы можно рассматривать как своего рода *распространителей* случайности. В качестве энциклопедии по классической генерации псевдослучайных последовательностей и тестам, цель которых заключается в том, чтобы с их помощью иметь возможность отличать псевдослучайные последовательности от истинно случайных, мы рекомендуем [236].

Случайность и криптография вообще очень сильно взаимосвязаны. Ведь основная цель криптографических систем состоит в том, чтобы преобразовать неслучайные осмысленные открытые тексты в кажущуюся случайной беспорядочную мешанину символов. Эта способность криптосистем может быть использована для генерации псевдослучайных последовательностей следующим образом. Пусть E_k — некоторый алгоритм шифрования, и пусть x_0 — произвольный открытый текст. Рассмотрим последовательность, которая определяется как $x_{i+1} = E_k(x_i)$ для $i > 0$. Если E_k вполне подходит для криптографических целей, то по всей видимости последовательность $x_1, x_2, \ldots$ — является последовательностью без явных признаков системности (хотя совершенно очевидно, что она обязательно должна зацикливаться). Может быть, для того чтобы уменьшить корреляцию в этой

последовательности, было бы предпочтительнее из каждого x_i оставлять только несколько битов информации.

Другой аспект взаимосвязи между свойствами случайных последовательностей и криптографией еще более интересен. Даже самые лучшие криптосистемы были бы бесполезны, если бы криптоаналитик мог угадывать используемый в них ключ (причем это замечание в одинаковой степени относится как к криптосистемам с секретным, так и к криптосистемам с открытым ключом). И по всей видимости, не существует лучшего способа предотвратить эту угрозу, чем выбирать ключ действительно случайным образом. Ведь именно отсутствие случайности в «телеграфных ключах» как раз и было главной причиной раскрытия Энигмы (*Enigma*) [187, 303]. В качестве простейшего примера рассмотрим одноразовый шифр, который уже был описан в § 3.2. Как мы там отмечали, эта криптосистема обладает совершенной секретностью, если только ее ключ действительно выбирается случайно, но в то же время может быть без особого труда раскрыта, если ключом в ней будет открытый текст на английском языке. Напомним, что основное неудобство одноразовых шифров заключается в том, что их ключи должны быть не только случайными, но также иметь в точности такую же длину, как и само шифруемое сообщение, и при этом использоваться только один раз.

В режиме обратной связи по выходу (OFB), который описан в § 3.5, криптография и случайность прекрасно объединяются: в нем для генерации псевдослучайной последовательности на основе двух коротких исходных векторов (k и s_0; из которых только k секретно) используется криптосистема, и полученная таким образом последовательность применяется как одноразовый ключ для шифрования реальных сообщений открытого текста. Преимущество подобного подхода состоит в том, что секретный ключ k намного короче, чем открытый текст, и может быть использован повторно столько раз, сколько вы этого захотите, поскольку s_0 при этом все равно каждый раз меняется (напомним, что s_0 задается в открытом виде как часть шифртекста, так что для обеспечения соответствующей стойкости необходимо лишь однажды передать начальный ключ k и все). Неизбежной платой за использование такого короткого ключа является, конечно, поте-

ря совершенной секретности. Хотя, в принципе, может быть, что этот режим обратной связи по выходу является ни чем иным, как принятием желаемой за действительную эвристикой.

Скажем, что псевдослучайный генератор является *криптографически сильным*, если последовательность, которую он вырабатывает из (секретного) короткого начального двоичного вектора, является по существу точно такой же, как и по-настоящему случайная последовательность, применяемая для той цели, для которой она используется в одноразовом шифре. Под словами «по существу точно такой же» мы понимаем, что никакое практически *легкоосуществимое* вычисление не может позволить криптоаналитику получить какую бы то ни было информацию об открытом тексте после перехвата его шифртекста (за исключением разве что с пренебрежимо малой вероятностью). Другими словами, он ведет себя точно так же, как если бы был совершенно секретным по Шеннону, до тех пор, пока для его криптоанализа не будет затрачено невообразимо большое количество времени. Подобные генераторы могут использоваться для реализации криптосистем с секретным ключом, если обе стороны договорились об исходном секретном векторе, который они собираются использовать, при условии что никогда не будут использовать один и тот же такой исходный вектор дважды. Намного менее очевидно, что криптографически сильные псевдослучайные генераторы могут быть использованы и при реализации криптосистем с *открытым* ключом, но в следующем параграфе мы покажем, что такое тоже возможно.

Первый шаг к обоснованию криптографически сильных генераторов был сделан Эди Шамиром [316]. Впоследствии Мануэлем Блюмом и Сильвио Микэли в [57] было введено ключевое понятие псевдослучайных генераторов — так называемая *непредикативность влево* — когда криптоаналитик, который знает, как работает генератор, но не знает, какой исходный вектор используется при его работе, для того чтобы угадать первый бит, выработанный генератором, не может найти ничего лучшего, чем после анализа всей сгенерированной в результате последовательности, подбросить жребий (если, конечно, ему при этом анализе не слишком повезет, или если он не окажется в состоянии проделать практически невыполнимые вычисления). Как и обычно,

мы не знаем, существуют ли подобные генераторы, но первый кандидат на такой генератор был предложен Блюмом и Микэли [57], которые доказали, что их генератор является непредикативным влево в предположении, что вычисление дискретных логарифмов является практически трудноосуществимым. То, что введение понятия непредикативности генераторов влево было вполне уместно, установил Эндрю Яо, доказав, что любой такой генератор является криптографически сильным [364]. Наконец, Леонид Левин дал необходимые и достаточные условия для существования подобных генераторов [250]. См. также [219, 211].

Сейчас мы приведем описание более простого и в вычислительном отношении более эффективного кандидата на криптографически сильный псевдослучайный генератор, который известен как BBS-генератор, названный так в честь Леоноры Блюм, Мануэля Блюма и Майка Шуба [49]. Он основывается на нашем втором кандидате на однонаправленную функцию с потайным ходом, который был преставлен в § 1. Напомним, что n называется *целым числом Блюма*, если оно является произведением двух различных простых чисел p и q, оба из которых сравнимы с числом 3 по модулю 4. Напомним также, что в данном случае функция возведения в квадрат по модулю n является перестановкой квадратичных вычетов по модулю числа n, и считается, что она является однонаправленной функцией с потайным ходом, так как было доказано, что трудность ее обращения (то есть вычисления *примитивных квадратных корней* по модулю n) вычислительно эквивалентна разложению n на множители. В предположении, что факторизация числа n является трудной задачей, может быть доказана даже более сильная теорема: для почти всех квадратичных вычетов y по модулю n, то есть $y = x^2 \bmod n$, к наилучшей (из всех возможных) выполнимой оценке того, каким должен быть первый значащий разряд x, приводит лишь подбрасывание жребия [349, 350, 8, 117]. Другими словами, оказывается, что практически трудно не только вычислить сам примитивный квадратный корень, но даже получить вероятностную информацию о его первом значащем бите (наподобие следующей: «Я не убежден, но мне все-таки кажется, что этот бит больше похож на нуль, чем на единицу»).

Теперь мы готовы описать BBS-генератор и доказать, что при

соответствующем предположении он является криптографически сильным. Пусть n — целое число Блюма с неизвестным разложением на множители. Возьмем в качестве исходного числа случайным образом выбранный квадратичный вычет x_0 (для того, чтобы это сделать, выберем наугад целое x взаимно простое с n, а x_0 вычислим как $x_0 = x^2 \bmod n$). Для $i \geqslant 0$ определим x_i рекурсивно по формуле $x_{i+1} = x_i^2 \bmod n$ и обозначим через b_i первый значащий бит x_i. Тогда для любого целого t первые t битов, которые таким образом будут сгенерированы из x_0, мы и объявим искомой псевдослучайной последовательностью $BBS_{n,t}(x_0) = b_0 b_1 b_2 \ldots b_{t-1}$.

Когда мы говорим, что BBS-генератор непредикативен влево, то это означает, что никто не может отгадать b_0, основываясь лишь на знании n и $b_1 b_2 \ldots b_{t-1}$. Если бы это было не так, то первый значащий бит неизвестного примитивного квадратного корня x любого заданного квадратичного вычета y можно было бы определить следующим образом. Сначала вычислим псевдослучайную последовательность $BBS_{n,t-1}(y)$ и *объявим*, что она на самом деле является последовательностью $BBS_{n,t}(x)$ с удаленным первым битом. Затем отгадаем этот пропущенный бит и отметим, что по определению он и является тем самым первым значащим битом x, который мы искали. Из этого следует, что BBS-генератор должен быть непредикативен влево в предположении, что факторизация целого числа n является трудной задачей. Следовательно, по теореме Яо, такой генератор будет криптографически сильным.

Дополнительное привлекательное свойство этого генератора заключается в том, что для всех, кто знает разложение n на множители, он допускает *прямое определение* тех отдельных битов, которые в нем вырабатываются. Для этого заметим, что $x_i = x_0^{2^i} \bmod n$. Но по теореме Эйлера $x^{\varphi(n)} \equiv 1 \pmod n$, где $\varphi(n) = (p-1)(q-1)$. Следовательно, посредством двух применений быстрого алгоритма модульного возведения в степень, все числа $x_i = x_0^{2^i \bmod \varphi(n)} \bmod n$ могут быть вычислены эффективно, исходя из начального вектора x_0 и определяющего каждое из этих чисел индекса i.

§ 6. Вероятностное шифрование

С одной стороны, криптография с открытым ключом в значительной степени решает задачу распространения ключей, которая является довольно серьезной проблемой для криптографии с секретным ключом. А с другой, как мы указывали выше, злоумышленнику при перехвате шифртекста $c = E_k(m)$ в любом случае становится известной некоторая информация об открытом тексте m, поскольку он всегда может без посторонней помощи вычислить значение открытой функции шифрования E_k для любого нужного ему открытого текста. Стало быть, задаваясь по своему выбору сообщением $\hat{m}$, он может также легко выяснить, верно ли, что $m = \hat{m}$, так как это справедливо только, если $E_k(\hat{m}) = c$. Даже если нахождение m из c и в самом деле было бы трудно осуществить при знании только естественного алгоритма шифрования, что пока еще не доказано, ничего нельзя сказать о том, сколь велика и в чем состоит возможная частичная утечка информации об m. Если использовать метафору Шафи Гольдвассер, то применение криптографии с открытым ключом подобно попытке прятать верблюда под одеялом: можно утаить, какой из верблюдов спрятан, но не число его горбов.

Целью *вероятностного шифрования* — понятия, введенного Шафи Гольдвассер и Сильвио Микэли в [197] — является такое криптографическое преобразование над открытым текстом сообщения, при котором никакое легко выполнимое вычисление на основе шифртекста не может дать какой бы то ни было информации о соответствующем открытом тексте (кроме как, быть может, с пренебрежимо малой вероятностью). Это напоминает системы, являющиеся совершенно секретными в смысле Шеннона, но с дополнительными преимуществами, которые предоставляют короткие ключи, и возможностью для каждого пользователя обнародовать свои открытые алгоритмы шифрования. Конечно, от таких систем нельзя ожидать настоящей совершенной секретности — они в принципе несекретны для криптоаналитика, обладающего неограниченной вычислительной мощностью.

Основное техническое различие между вероятностным шифрованием и системами с открытым ключом состоит в том, что

естественные алгоритмы шифрования являются при этом вероятностными, а не детерминированными: одно и то же сообщение открытого текста может привести к возникновению большого числа различных шифртекстов. В результате криптоаналитик, имеющий, по его предположению, истинный открытый текст, не сможет долго проверять свою догадку посредством шифрования этого открытого текста (с помощью естественного алгоритма его законного получателя) и последующего сравнения получившегося результата с перехваченным шифртекстом.

Формально система вероятностного шифрования состоит из *пространства ключей* $\mathcal{K}$ и, для каждого $k \in \mathcal{K}$, из *пространства открытых текстов* $\mathcal{M}_k$ сообщений, *пространства шифртекстов* $\mathcal{C}_k$, *вероятностного пространства* $\mathcal{R}_k$ и пары функций $E_k: \mathcal{M}_k \times \mathcal{R}_k \to \mathcal{C}_k$ и $D_k: \mathcal{C}_k \to \mathcal{M}_k$, таких что $D_k(E_k(m, r)) = m$ для любого сообщения открытого текста $m \in \mathcal{M}_k$ и случайного числа $r \in \mathcal{R}_k$. С помощью любого $k \in \mathcal{K}$ должны легко получаться эффективные естественные алгоритмы для вычисления как E_k, так и D_k, но построение какого-либо эффективного алгоритма вычисления D_k, на основе только естественных алгоритмов вычисления E_k, должно быть практически невозможно.

Использование системы вероятностного шифрования очень похоже на использование систем с открытым ключом. Каждый пользователь раз и навсегда выбирает для себя ключ $k \in \mathcal{K}$, который используется для получения обоих естественных алгоритмов вычисления E_k и D_k. Он делает алгоритм шифрования E_k общедоступным, но сохраняет в секрете алгоритм дешифрования D_k. В том случае, когда другой пользователь захочет послать ему свое сообщение m, он находит E_k в справочнике, случайно выбирает некоторое $r \in \mathcal{R}_k$ и вычисляет шифртекст $c = E_k(m, r)$. При этом только законный получатель, используя свою секретную лазейку, может легко определить m из c.

Вследствие того, что при использовании вероятностного шифрования каждому открытому тексту соответствует довольно большое количество шифртекстов, неизбежно происходит некоторое *раскрытие данных*: шифртекст всегда длиннее, чем соответствующий ему открытый текст. Заметим, что для криптосистемы RSA, это было не так. Несмотря на все свои замечательные теоретические свойства, оригинальная система вероят-

ностного шифрования Гольдвассер-Микэли допускала столь существенное раскрытие данных, что не имела большого практического значения. Поэтому мы не будем описывать ее здесь.

Тем не менее, вероятностное шифрование достигло такого состояния, что уже в настоящее время существует криптосистема даже более эффективная, чем RSA. Для обеспечения конфиденциальности (но не аутентификации — см. § 5.1) механизмы работы этой системы Блюма-Гольдвассер, которая описывается ниже, являются наилучшими из того, что наука в настоящее время может предложить. Система основана на вере в то, что возведение в квадрат по модулю целого числа Блюма — однонаправленная функция с потайным ходом (см. последний пример из § 1), и на том, что псевдослучайный битовый генератор, который был описан в § 5, является криптографически сильным. Естественно, в качестве такого генератора для получения псевдослучайного одноразового шифра необходимой длины из его собственного случайным образом выработанного исходного вектора отправителем сообщения используется BBS-генератор. Способность законного получателя вычислять квадратные корни (основанная на его личной информации о соответствующем потайном ходе) позволяет ему найти этот шифр, а затем с его помощью определить открытый текст.

Более формально, пусть p и q — два случайно выбранных различных простых числа, сравнимых с 3 по модулю 4, которые вместе образуют секретный ключ. Их произведение $n = p \cdot q$ является открытым ключом. Пространство сообщений открытого текста — это множество *всех* конечных двоичных строк произвольной длины. Любое сообщение может в этом случае шифроваться непосредственно без разбиения на части, используя при этом один из режимов, которые были описаны в § 3.4, точно так же, как это было в случае с RSA. (Правда, это только чисто теоретически, поскольку на самом деле длина битовой строки, представляющей открытый текст не должна быть больше периода псевдослучайной последовательности, вырабатываемой BBS-генератором, хотя на практике этот период в столь большое число раз превышает размер модуля n, что позволяет фактически не ограничивать длину открытого текста.) Вероятностным пространством служит множество QR_n, то есть множе-

ство квадратичных вычетов по модулю n. Пространством сообщений шифрованного текста является множество пар, образованных квадратичными вычетами по модулю n и конечными двоичными строками.

Пусть m — некоторое t-разрядное сообщение. И пусть также x_0 — это случайный квадратичный вычет по модулю n. Предположим далее, что $BBS_{n,t}(x_0)$ и x_t определяются точно так же, как в § 5. Тогда шифрование m, использующее исходный вектор x_0 и открытый ключ n, задается в виде пары чисел $\langle x_t, m \oplus BBS_{n,t}(x_0)\rangle$, где «$\oplus$» означает поразрядное сложение по модулю 2. Здесь $BBS_{n,t}(x_0)$ используется в качестве одноразового шифра к открытому тексту m. Значение x_t включается в шифртекст только для того, чтобы обеспечить законному получателю эффективное дешифрование, но тем самым никак не помогая в этом нарушителю. Напомним, что необходимым и достаточным условием для эффективного вычисления примитивных квадратных корней является знание сомножителей числа n [297]. Простейший алгоритм дешифрования заключается в вычислении всей псевдослучайной последовательности, начиная с x_t, в *обратном порядке*, и используя для этого рекуррентное уравнение $x_i = \sqrt{x_{i+1}} \bmod n$. В результате такого вычисления сначала однозначно восстанавливается значение $BBS_{n,t}(x_0)$, а затем из известного шифртекста легко получается открытый текст.

Обозначим через l число разрядов модуля n. Эффективность только что описанного алгоритма шифрования полностью сопоставима с эффективностью алгоритма криптосхемы RSA, потому что в нем для каждого бита открытого текста используется одна операция модульного *возведения в квадрат*, в то время как в RSA для каждого $(l-1)$-блока открытого текста требуется одно модульное *возведение в степень*, и при этом каждое возведение в степень требует l возведений в квадрат и, кроме того, l умножений (см. § 1). Простейший алгоритм дешифрования, который был предложен выше, не очень хорош, поскольку для каждого бита открытого текста он требует вычисления одного *примитивного квадратного корня*, а на каждое такое вычисление расходуется примерно столько же времени, сколько на одно модульное возведение в степень. К счастью, знание множителей n позволяет определять все отдельные биты случайной последовательности

непосредственно не только в прямом порядке так, как это описано в конце § 5, но и в обратном. Такое вычисление обходится примерно во столько же, во сколько обходится одно возведение в степень (или RSA-дешифрование одного блока), и позволяет законному получателю сообщения вычислить x_0 непосредственно из x_t, а затем для того, чтобы получить $BBS_{n,t}(x_0)$, проделать то же самое в прямом порядке так же эффективно, как это сделал его отправитель.

Сейчас мы представим этот эффективный алгоритм вычисления x_0 из x_t при известном разложении $n = p \cdot q$. В качестве предварительного шага в нем сначала с помощью обобщенного алгоритма Евклида раз и навсегда вычисляются целые числа a и b, такие что $ap + bq = 1$. Затем выполняются следующие операции:

$$\begin{aligned}
&\alpha \leftarrow \mathit{expmod}\left(\frac{p+1}{4}, t, p-1\right)\\
&\beta \leftarrow \mathit{expmod}\left(\frac{q+1}{4}, t, q-1\right)\\
&u \leftarrow \mathit{expmod}((x_t \bmod p), \alpha, p)\\
&v \leftarrow \mathit{expmod}((x_t \bmod q), \beta, q)\\
&\textbf{return}\ (bqu + apv) \bmod n \quad .
\end{aligned}$$

Схема вероятностного шифрования Блюма-Гольдвассер может быть сделана даже еще быстрее. Более тщательный анализ псевдослучайного генератора BBS [349, 350, 8, 117] показывает, что в нем после каждой операции модульного возведения в квадрат можно использовать более одного значащего бита очередного квадратичного вычета x_i. А именно, окончательная псевдослучайная последовательность не ослабится, если в нее для каждого индекса i будут выбираться (приблизительно) $log_2(l)$ первых значащих битов x_i. Благодаря такому улучшению вероятностное шифрование будет осуществляться быстрее, чем шифрование в RSA, примерно в $log_2(l)$ раз, причем то же самое справедливо и для дешифрования длинных сообщений (так как в этом случае достаточно проделать вычисление в обратном порядке только один раз). В заключение отметим, что криптосхема вероятностного шифрования не только быстрее, чем RSA, но и также *доказано*, что трудность ее раскрытия вычислительно

эквивалентна разложению на множители (тогда как раскрытие RSA может оказаться проще, чем факторизация), и кроме того она не предоставляет никакой частной информации об открытом тексте, если факторизация действительно трудна (в то время как RSA заведомо предоставляет некоторую частичную информацию, и может обладать таким свойством, даже если в открытый текст при ее использовании будут случайным образом добавляться лишние символы). Однако следует отметить, что эта криптосхема вообще несекретна по отношению к атаке на основе выбранного шифртекста. Мы предлагаем читателю самому разобраться, почему это так.

§ 7. Гибридные системы

Несмотря на все преимущества криптосистем с открытым ключом и схем вероятностного шифрования, ни одна из предложенных до настоящего времени их реализаций не может конкурировать по *быстродействию* с такими криптосистемами с секретным ключом, как, например, DES. Когда необходимо передавать большое количество информации, то может статься, что применение RSA будет серьезно замедлять работу, тогда как использование DES по каким-то причинам либо окажется невозможным (например, из-за отсутствия совместного секретного ключа), либо перестанет отвечать необходимым требованиям секретности.

В такой ситуации очень полезным может быть использование *гибридной криптосистемы*, в которой один раз перед началом каждого сеанса передачи шифрованных сообщений с помощью криптосистемы с открытым ключом формируется небольшая (совместная и секретная для других) часть информации, а затем в последующей передаче используется в качестве ключа к шифратору (и дешифратору) для тех текущих сообщений открытого текста, которые будут зашифровываться (и расшифровываться) при помощи криптосистемы с секретным ключом.

Если передаваемое сообщение является достаточно длинным, то лучше всего в течение каждой передачи использовать криптосистему с открытым ключом по несколько раз, с тем чтобы секретные ключи можно было почаще менять. Не вызывая

особого замедления протокола, это значительно повысит стойкость гибридной системы по двум причинам: во-первых, при атаке на основе только шифртекста, являющейся единственным типом атаки, которая имеет смысл в данной ситуации, криптосистему с секретным ключом легче раскрыть, если доступен большой шифртекст, и во-вторых, даже если криптоаналитику и удастся определить какой-нибудь один из секретных ключей, то он сможет расшифровать лишь соответствующую ему часть всего сообщения.

Глава 5

Аутентификация и подпись

Несмотря на то, что с самого начала исторической движущей силой криптографии было стремление передавать секретные сообщения по несекретным каналам связи, осуществление только такой передачи информации не является ее единственной целью.

§ 1. Аутентификация

До сих пор мы имели дело лишь с понятием *пассивного* криптоаналитика, то есть с тем, чья цель заключалась только в прослушивании канала связи. *Активный* криптоаналитик (называемый также *фальсификатором*) идет дальше: не удовлетворяясь прослушиванием канала связи, он может также вводить свои собственные сообщения в надежде на то, что получатель при их расшифровке может поверить, что они были посланы кем-то другим. Излишне говорить, что, например, финансовые сделки должны быть защищены в первую очередь именно от подобной фальсификации, чем быть непременно засекреченными.

Целью системы *аутентификации*, или иначе системы *удостоверения авторства*, точно так же, как и системы *подтверждения целостности*, является выявление указанного выше фальсификатора-самозванца. Всякий раз, когда *Боб* получает сообщение, в котором утверждается, что оно было послано от *Алисы*, система должна позволить ему убедиться не только в том, что это сообщение действительно исходит от *Алисы*, но и в том, что оно не было изменено при передаче. Мы допускаем, что фальсификатор в состоянии прослушивать столько аутенти-

фицированных (то есть подтверждающих свою собственную подлинность) сообщений, сколько он хочет, и его цель состоит в том, чтобы добиться именно такой подделки сообщения, которая позволит ему избежать ее обнаружения. Это подделанное сообщение может либо полностью отличаться от уже перехваченных, либо только минимально отличаться от одного из них, либо быть чем-то средним между этими двумя крайностями. Поэтому для того, чтобы обнаружить подмену, важно, чтобы каждое сообщение включало временну́ю отметку или некий порядковый номер.

Бытует убежденность, что любая криптосистема с секретным ключом может использоваться непосредственно как с целью защиты информации, так и для того, чтобы обеспечить ее аутентификацию. Пусть k — секретный ключ, общий как для *Алисы*, так и для *Боба*, и пусть E_k и D_k — соответствующие алгоритмы шифрования и дешифрования. Если бы *Боб* получил от *Алисы* некоторый шифртекст c, он мог бы расшифровать его как $m = D_k(c)$. В том случае, когда получившийся открытый текст m имеет смысл, *Боб* может чувствовать себя уверенным в том, что шифртекст c и в самом деле был получен как $c = E_k(m)$ единственным кроме него самого человеком, знающим секретный ключ k, а именно *Алисой*. Аргументацией этого является то, что фальсификатор, не зная k, был бы не в состоянии составить фальшивый шифртекст так, чтобы он при расшифровке не превратился бы в бессмысленную мешанину символов. Однако *эта вера ошибочна*, потому что фальсификатор может, зная пары соответствующих открытых и шифрованных текстов, оказаться способным объединить отдельные части шифртекста в нечто вполне осмысленное. Помимо того, даже самые лучшие криптосистемы могут быть абсолютно бесполезными для целей аутентификации, особенно если они используются в режиме электронной кодовой книги (см. § 3.5).

Для того чтобы понять, почему это так, давайте рассмотрим криптосистему, которая является совершенно секретной по Шеннону. Предположим, что для передачи информации основному банковскому компьютеру кассовый автомат использует одноразовый шифр, и допустим, что открытый текст пересылаемой информации представлен в формате, который известен фальсификатору. Тогда последний может подойти к автомату, сделать вклад, скажем, в десять долларов, перехватить зашифрованное

сообщение об этом вкладе на его пути к компьютеру, с помощью зашифрованного сообщения и своего знания о том, каким должен быть открытый текст, установить тот шифр, который при этом действительно использовался, и модифицировать необходимые позиции в шифртексте, заменив указанную в нем сумму на ту, что ему больше нравится. Даже если шифртекст не известен в точности, достаточно предположительно знать его формат для того, чтобы проследить, как может быть воспринята подделка, без опасения слишком большого риска быть обнаруженным.

Любая *аутентификационная система* состоит из *пространства ключей* $\mathcal{K}$ и для каждого $k \in \mathcal{K}$ из *пространства сообщений* $\mathcal{M}_k$, *пространства меток* $\mathcal{T}_k$ и *аутентификационной функции* $A_k: \mathcal{M}_k \to \mathcal{T}_k$. При этом для любого заданного k должен легко получаться эффективный алгоритм вычисления A_k. Аутентификационная система используется следующим образом. Если *Алиса* и *Боб* ожидают, что со временем они могут обмениваться друг с другом подтверждающими свою подлинность сообщениями, то они сначала должны договориться о некотором секретном ключе $k \in \mathcal{K}$. Всякий раз, когда *Алиса* захочет аутентифицировать некоторое сообщение $m \in \mathcal{M}_k$ для *Боба* (то есть подтвердить ему свое авторство данного сообщения), он вычисляет его метку $t = A_k(m)$ и посылает ее вместе с m. Для того чтобы удостовериться в подлинности этого сообщения, *Боб* также вычисляет $A_k(m)$ и сравнивает его с той меткой t, которую он получил. Конечно, это не исключает возможности при этом зашифровывать сообщение m, если требуется не только подтверждение его подлинности, но и соблюдение секретности.

Аутентификационная функция не обязана быть взаимнооднозначной и пространство меток может быть существенно меньше, чем пространство сообщений. Однако оно не должно быть слишком маленьким, с тем чтобы случайно выбранная метка все-таки имела пренебрежимо малую вероятность быть правильной для конкретного удачно измененного фальсификатором сообщения.

Как и в случае криптографии с секретным ключом, здесь также можно выделить различные уровни атаки:

- *Атака на основе известного сообщения*: фальсификатор подслушал несколько аутентифицированных пар $\langle m_1, t_1 \rangle$, $\langle m_2, t_2 \rangle$, ..., $\langle m_i, t_i \rangle$, таких что $t_j = A_k(m_j)$ для каждого j, $1 \leqslant j \leqslant i$, при неизвестном k; тогда он сможет:

- или вычислить k или оказаться неспособным это сделать;
- или вычислить $t_{i+1} = A_k(m_{i+1})$ для *выбранного им самим* сообщения m_{i+1} или оказаться неспособным на это;
- или вычислить $t_{i+1} = A_k(m_{i+1})$ для любого m_{i+1}, отличного от m_j, $1 \leqslant j \leqslant i$ (в надежде сбить с толку), или, если даже и это окажется невозможным, то
- вычислить некоторую пару $\langle m_{i+1}, t_{i+1} \rangle$ для нового m_{i+1} такого, что t_{i+1} должна быть правильной меткой $A_k(m_{i+1})$ с вероятностью большей, нежели некоторая пренебрежимо малая величина.

- *Атака на основе выбранных сообщений*: фальсификатор вначале формирует некоторые сообщения m_1, m_2, ..., m_i, затем ему предоставляются соответствующие этим сообщениям метки $t_1 = A_k(m_1)$, $t_2 = A_k(m_2)$, ..., $t_i = A_k(m_i)$, после чего он решает одну из упомянутых выше задач (такой тип атаки может быть подготовлен с помощью временно устроившегося на работу в банк служащего, который имеет доступ на ограниченное время к аутентификационному черному ящику автомата).

Несмотря на все, что было сказано ранее, криптосистема с секретным ключом зачастую все же может использоваться в системах аутентификации в качестве составного блока. Это возможно, если криптосистема имеет блок фиксированного размера и если сообщение, которое должно аутентифицироваться, является достаточно длинным, и поэтому занимает несколько блоков. Идея заключается в том, чтобы зашифровывать сообщения либо в режиме шифрования со сцеплением блоков, либо в режиме шифрования с обратной связью и разрешить использовать для аутентификационной метки только таким образом получающийся в результате блок. В этом случае, согласно определению каждого из режимов, метка будет зависеть от всего сообщения. Если же добиваться и секретности, и аутентификации одновременно, и если для обеспечения обеих этих функций используется одна и та же криптосистема с секретным ключом, то тогда существенно надежнее будет использовать два *различных* секретных ключа [226].

Когда криптосистема с секретным ключом используется для обеспечения секретности, то ее надежность значительно увеличивается при частой смене ключей. Довольно странно, что для целей аутентификации это не так [146]. К тому же последствия редких подделываний меток, которые остаются необнаруженными, кажутся мне намного более драматичными, чем последствия редких нарушений секретности, которые происходят благодаря успешному дешифрованию сообщений криптоаналитиками противника. Вследствие этого более критичным является тщательный выбор криптосистемы, которая потом будет использоваться для целей аутентификации, а не для обеспечения секретности.

Комбинируя понятия одноразового шифра и универсального хеширования [98], Марк Вегман и Ларри Картер разработали аутентификационные метки совершенно другой природы: они предложили систему аутентификации, которая является нераскрываемой в теоретико-информационном смысле [355]. Как и в случае классического применения одноразовых шифров, в ней количество необходимой секретной информации пропорционально числу сообщений, которые должны аутентифицироваться (хотя оно и не зависит от длины этих сообщений). Другими словами, когда какое-то сообщение аутентифицируется, то некоторые биты секретного ключа используются неоднократно. Если пользователи захотят использовать для этого t и более бит на одно сообщение, Вегман и Картер показали, что даже противник с неограниченными вычислительными ресурсами, применяющий атаку на основе выбранных сообщений, не сможет подделать никакую их метку с вероятностью успеха большей, чем 2^{-t}. Они также доказали, что эта вероятность оптимальна.

Жиль Брассар показал, как следует скомбинировать формирование меток Вегмана-Картера с псевдослучайным генератором BBS (см. § 4.5) для того, чтобы добиться того же самого с фиксированным коротким ключом, не зависящим от числа сообщений. Однако получившаяся система оказывается секретной только относительно практически выполнимых вычислений (без неограниченных вычислительных ресурсов), и только если факторизация целых чисел действительно трудна [67]. Одедом Голдрейчем, Шафи Гольдвассер и Сильвио Микэли была предложена также вычислительно секретная аутентификационная система [192], в

которой в качестве применения использовано введенное ими понятие полислучайного семейства функций [193] (см. также § 6.5).

§ 2. Цифровая подпись

Несмотря на то, что схема аутентификации позволяет *Бобу* достичь большой уверенности в том, что сообщение, которое он получил, исходило именно от *Алисы*, эта схема не позволяет ему убедить кого бы то ни было еще в том, что *Алиса* и в самом деле посылала полученное им сообщение. Таким образом, аутентификационные схемы могут использоваться в отношениях между двумя доверяющими друг другу сторонами, но они не способны обеспечивать улаживание возникающих между ними разногласий. Такая возможность предоставляется только благодаря существованию однонаправленных функций с потайным ходом и основанного на них более сильного понятия *цифровой*, или *электронной*, подписи.

Если *Алиса* посылает *Бобу* сообщение, подписанное своей электронной подписью, то *Боб* при его получении не только сам сможет убедиться в том, что это сообщение подписано ни кем иным, как его составителем (т.е. *Алисой*), но и будет также способен *доказать в суде*, что именно *Алиса*, и никто иной, подписала это сообщение. Понятие цифровой подписи было введено Уитфилдом Диффи и Мартином Хеллманом [157]; читайте также [307, 296, 297, 258, 201]. Цифровая подпись и электронная почта сулят значительные преимущества, которые даже в принципе неосуществимы в традиционном бумажном документообороте. Так, в § 6.5 мы покажем, что они, например, делают возможной не только электронную почту *с удостоверением* о получении сообщений, но и электронное заключение *контрактов*.

Хотелось бы пояснить, почему схема аутентификации не обеспечивает цифровой подписи. Для этого заметим, что если *Алиса* аутентифицирует сообщение для *Боба* так, как это описано в § 1, то *Бобу* было бы столь же легко получить соответствующую метку и самому. Следовательно, единственной причиной для *Боба* быть уверенным в том, что такое вычисление было выполнено *Алисой*, является то, что *Боб* знает, что он сам этого не делал. Разумеется, что это не должно быть столь же очевидно и судье.

Криптосхема с открытым ключом позволяет обеспечить *воз-*

можность цифровой подписи, если для каждого ключа $k \in K$, $\mathcal{M}_k = \mathcal{C}_k$ и если $E_k(D_k(m)) = m$ для каждого сообщения $m \in \mathcal{M}_k$ (второе условие является следствием первого, если $\mathcal{M}_k$ конечное множество). Для осуществления цифровой подписи такая схема используется следующим образом. Пусть a — некоторый секретный ключ *Алисы* и пусть E_a и D_a — ее функции шифрования и, соответственно, дешифрования. Тогда, если криптосистема является секретной, то только *Алиса* сможет вычислить D_a эффективно, хотя при этом каждый знает, как вычислять E_a.

Рассмотрим далее некоторый открытый текст m и положим $s = D_a(m)$. Очевидно, что любой пользователь криптосистемы может эффективно вычислить $E_a(s)$ и выяснить, что ее значением является m. Однако только *Алиса* обладает теми знаниями, которые необходимы, чтобы получить такое s, что $E_a(s) = m$. В этом смысле s может рассматриваться как подпись самой *Алисы* под сообщением m. Если *Боб* покажет s судье, и если $E_a(s) =$ **Я должна Бобу тысячу долларов**, судья вынужден будет согласиться, что никто другой, кроме *Алисы*, не мог подписать этого утверждения. Другими словами, секретный алгоритм дешифрования D_k может рассматриваться в этом случае как алгоритм, осуществляющий цифровую подпись, а открытый алгоритм шифрования E_k — как соответствующий алгоритм подтверждения этой подписи.

В криптосистемах с открытым ключом цифровая подпись может использоваться совместно с шифрованием, если требуется также соблюдать и секретность. Предположим, что b — секретный ключ *Боба*. Тогда, если *Алиса* захочет послать *Бобу* подписанное секретное сообщение m, то она использует свой секретный алгоритм подписи D_a и его открытый алгоритм шифрования E_b, чтобы получить $c = E_b(D_a(m))$. Если *Алиса* пошлет сообщение с *Бобу* по несекретному каналу, то тот сможет вычислить подпись *Алисы* под этим сообщением как $s = D_b(c)$, а затем восстановить его в открытом виде как $m = E_a(s)$. При этом предполагается, конечно, что в заголовке сообщения явно говорится, что оно исходит от *Алисы*, так что *Боб* знает, чей алгоритм подтверждения подписи применять для того, чтобы получить открытый текст. Более того, если *Боб* сохранит s, то, как мы уже объясняли ранее, он сможет доказать судье, что именно *Алиса*, и никто другой, послала ему сообщение m.

Естественным альтернативным решением для *Алисы* было бы послать $\hat{c} = D_a(E_b(m))$, которое позволяет дешифровать $\hat{c}$ как $m = D_b(E_a(\hat{c}))$. Однако такое решение желательно не применять, так как оно позволило бы фальсификатору (имеющему секретный ключ t), назовем его *Томасом*, перехватить $\hat{c}$ до того, как его получит *Боб*. Если бы затем *Томас* вычислил $\bar{c} = D_t(E_a(\hat{c}))$, то он смог бы переслать шифртекст $\bar{c}$ *Бобу*, сделав вид, будто это сообщение с самого начала именно от него и исходит. Тогда ничего не подозревающий *Боб*, вычислив $D_b(E_t(\bar{c})) = m$, был бы абсолютно убежден в том, что m подписано именно *Томасом* (конечно, если m не начиналось словами «**Меня зовут Алиса**»). Суть заключается в том, что указанное решение позволяет *Томасу* фактически подписывать сообщение, содержание которого он сам не в состоянии прочитать. Хотя и трудно представить себе, почему *Томас* мог бы захотеть это сделать, все же будет лучше не предоставлять ему такой возможности вовсе.

Если RSA используется как для обеспечения секретности, так и для цифровой подписи, может быть предпочтительней, чтобы каждый пользователь хранил для двух разных целей две различные пары функций с потайным ходом. Тогда у каждого пользователя была бы одна запись в открытом справочнике шифрования, а другая — в открытом справочнике проверки подписей. Такое разделение целесообразно по двум причинам. Во-первых, оно позволяет избежать проблемы переразбиения на блоки, которая в противном случае возникает, если модуль отправителя оказывается больше модуля получателя [307]. Во-вторых, как мы уже видели (в § 4.4), криптосистема RSA является слабой относительно атаки на основе выбранного шифртекста. И такую атаку может быть труднее проводить, если процедура цифровой подписи отличается от процедуры дешифрования.

Необходимо проявлять некоторую осторожность в утверждении, что сам факт того, что *Боб* знает s такое, что $E_a(s) = m$, должен непременно рассматриваться как то, что m должно было быть подписано именно *Алисой*. В конце концов, *Боб* мог бы выбрать некоторое случайное s, вычислить $m = E_a(s)$, которое почти наверняка не имеет вообще никакого смысла, и затем, демонстрируя его, уверять, что *Алиса*, должно быть, сошла с ума, подписав такую бессмыслицу, как m. Подобное утверждение особенно неправомерно тогда, когда плотность осмысленных

сообщений достаточно высока (что справедливо скорее для некоторых типов числовых данных, а не для обычных текстов), и *Боб* сможет выбирать случайно различные s до тех пор, пока ему не повезет настолько, что он наткнется на некоторое такое s, для которого $E_a(s)$ будет осмысленным. По этим причинам благоразумно договориться о некотором виде нормальной формы сообщений и признавать s в качестве подписи под m, только если $m = E_a(s)$, а m записано в этой *нормальной форме*. К сожалению, до сих пор неясно, какой должна быть надежная нормальная форма для RSA. Например, вставка фиксированного числа нулей является ненадежной [223, 224].

Возможность цифровой подписи предоставляет, например, криптосистема RSA. Однако ее использование обнаруживает недостатки, сходные с теми, которые она имеет как схема шифрования с открытым ключом. В частности, неизвестно, является ли ее раскрытие таким же трудным, как и разложение на множители больших целых чисел. Даже если и в самом деле трудно для выбранного открытого текста m и какого-то открытого ключа $\langle e, n\rangle$ вычислить подпись s такую, что $m = s^e \bmod n$, то может оказаться намного проще сделать то же самое при известной, кроме этого, паре чисел $\langle \hat{s}, \hat{m}\rangle$, где $\hat{s}$ — подпись законного пользователя под некоторым сообщением $\hat{m}$, которое лишь немного отличается от m. Другими словами, может оказаться легче подделать подпись под законным сообщением после того, как станут известны истинные подписи нескольких похожих сообщений.

Предыдущий абзац подсказывает, что для того, чтобы разрабатывать схемы для цифровой подписи, можно было бы попробовать применить вероятностное шифрование, а не криптографию с открытым ключом. Однако такая надежда оказывается необоснованной: схема вероятностного шифрования Блюма-Гольдвассер столь же бесполезна для обеспечения цифровой подписи, сколь бесполезен для аутентификации одноразовый шифр. К счастью, были разработаны более сильные криптосхемы цифровой подписи с доказательством (в предположении, что факторизация целых чисел является трудновычислимой проблемой) того, что они выдержат атаку даже на основе выбранных сообщений (соответствующее определение, приведенное в § 1, легко переносится на случай цифровых подписей). Правда эти схемы довольно сложны и поэтому мы не будем описывать их

здесь [190]. Для широкого обсуждения цифровой подписи и связанных с ней проблем, а также некоторых других решений мы отсылаем к [200]. Кроме того, читайте [190, 318, 18, 72, 275].

§ 3. Идентификация пользователей

(написано вместе с Клодом Крепеу и Клодом Готье)

Основной вопрос идентификации заключается в следующем: если потенциальный пользователь (крипто)системы обращается к услугам компьютера (либо в том случае, когда он, например, захочет воспользоваться автоматическим кассиром, либо тогда, когда такому кассиру понадобится получить доступ к центральному компьютеру банка), то каким образом при каждом таком обращении этот центральный компьютер сможет удостовериться, что тот, кто к нему обращается, является именно тем абонентом, за которого он себя выдает? Классическое решение этой проблемы заключается в использовании паролей. Предполагая, что пользователь и компьютер распределили между собой некоторую конфиденциальную часть информации, последний может потребовать эту информацию от обратившегося к нему в том случае, когда его подлинность сомнительна. Ясно, что ценность такой схемы решающим образом зависит от ее способности сохранять пароли в секрете, что указывает на два потенциально слабых звена: существование где-то в памяти компьютера таблицы паролей и угрозу перехвата сообщений до их попадания в него.

Сам факт, что пароли находятся в какой-то области памяти компьютера, является очень опасным. Вне зависимости от того, насколько они хорошо спрятаны и защищены, искусный противник всегда найдет их, где бы они ни были (предпочтительно, но совсем не обязательно, если он имеет физический доступ к этому компьютеру, его вспомогательной памяти и листингу состояния операционной системы). Более того, в составе персонала, обслуживающего работу большинства компьютеров, существует по крайней мере один человек, так называемый администратор системы, то есть тот ее (самый) привилегированный пользователь, который обладает всеми правами легального доступа к директории, где хранятся пароли. Сколь большим доверием другие пользователи могут облечь такого человека? Очевидно, что в

самом крайнем случае в разных системах предусмотрительные пользователи должны использовать различные пароли.

Рассмотренная выше угроза может быть легко предотвращена на практике [167, 291, 358, 272, и т.д.]. Ключевая мысль заключается в том, что компьютеру в такой ситуации совершенно не обязательно *знать* действительные пароли пользователей — для него необходима лишь способность *подтверждения* данных паролей. Это может быть реализовано посредством использования однонаправленных функций. Компьютеру достаточно хранить лишь *образы* пользовательских паролей, являющиеся результатами вычисления некоторой фиксированной однонаправленной функции от каждого пароля как аргумента. Поскольку функция однонаправленна, то хранить в секрете ее саму совершенно не обязательно. Тогда всякий раз, как только какой-нибудь пользователь захочет доказать свою подлинность, он сообщает свой истинный пароль, а компьютер сразу же вычисляет от него значение этой однонаправленной функции. Затем результат вычислений сверяется с хранящейся в компьютере таблицей. Очевидно, что в данном случае противник в силу своей неспособности вычислять прообразы значений соответствующей однонаправленной функции будет вынужден признать бесполезным для определения истинных паролей пользователей распечатку такой преобразованной таблицы.

Эта схема может оказаться довольно слабой, если пользователи склонны употреблять пароли, которые можно легко угадать. В такой ситуации с помощью кого-то, кто имеет доступ к таблице (преобразованных) паролей и к алгоритму вычисления однонаправленной функции, может, зачастую довольно успешно, применяться следующая атака. Задавшись, скажем, тысячью предположительно наиболее употребимых паролей и вычислив значения однонаправленной функции от них, этот «кто-то» может затем эффективно сравнить полученные результаты с преобразованной таблицей паролей (используя при этом технику хеширования или сортировки). Тогда каждое такое успешное сравнение предоставит ему в распоряжение некоторый истинный пользовательский пароль. Ясно, что подобная атака не позволяет нарушителю раскрыть все пароли, которые он хотел бы узнать, но тем не менее она очень продуктивна в нахождении некоторых из них [176].

Хотя и не делая ничего дополнительного для защиты индивидуальных паролей, так называемая *случайная добавка* является той техникой, которая почти исключает описанные выше коллективные угрозы. При формировании таблицы паролей с каждым именем пользователя связывается некоторая случайным образом сгенерированная битовая строка (в открытом виде), а хранящееся значение однонаправленной функции вычисляется от *конкатенации* истинного пароля каждого пользователя и этой его случайной строки. В результате оказывается, что два различных пользователя, которые могли бы случайно выбрать одинаковые пароли, будут иметь разные записи в преобразованной системной таблице.

Несмотря на такое улучшение, остается одна основная слабость. Всякий раз, когда пользователь захочет, чтобы была подтверждена его подлинность, он должен вводить свой настоящий пароль в компьютер. Пароль в этом случае становится уязвимым к перехвату либо тогда, когда он проходит по линии связи к компьютеру, либо внутри самого компьютера вплоть до того момента, пока он не будет преобразован посредством вычисления соответствующей однонаправленной функции.

Для того чтобы избавиться от подобной угрозы, необходимо, чтобы компьютер не только не имел паролей в памяти, но и вообще не выдавал бы их ни при каких обстоятельствах. Это требование кажется парадоксальным. Тем не менее оно может быть удовлетворено посредством комбинации одной идеи времен второй мировой войны — системы идентификации типа «друг-или-враг» (I.F.F.) [228] — с понятием однонаправленной функции с потайным ходом. Для этого требуется, чтобы пользователи имели в распоряжении свои собственные интеллектуальные терминалы или, что предпочтительнее, персональные smart-карточки [264, 116, 209].

Для того чтобы решить проблему идентификации пользователей какой-нибудь системы, каждый ее пользователь вырабатывает случайным образом личный (секретный) ключ в соответствии с некоторой криптосхемой с открытым ключом. Этот ключ он использует для конкретизации собственной согласованной пары алгоритмов: шифрования и дешифрования. Затем алгоритм шифрования он предоставляет в распоряжение центрального компьютера системы, а на своем терминале программирует алгоритм

дешифрования. Всякий раз, когда нужно убедиться в подлинности (того, кто, быть может, только действует от имени легального) пользователя системы, центральный компьютер вырабатывает случайное сообщение, вычисляет от него, как от аргумента, значение функции шифрования этого пользователя, и посылает результат на его терминал в качестве опроса. Пользователь в ответ на опрос запускает свою хранящуюся в секрете от всех программу дешифрования и вычисляет само исходное (случайное) сообщение, которое в качестве удостоверения пересылает назад в компьютер. Как следует непосредственно из определения однонаправленных функций с потайным ходом, только законный пользователь способен пройти подобные опросы эффективно, даже если его функция шифрования (программа которой хранится внутри центрального компьютера) известна нарушителю. Как и было объявлено ранее, при таком механизме работы системы по типу «опрос-ответ» личный пароль пользователя, который фактически является его эффективным алгоритмом дешифрования, никогда не пересылается в центральный компьютер.

Заметим, что в описанной ситуации нарушителю не поможет перехват опросов и ответов, которыми обмениваются компьютер и конкретный пользователь системы, поскольку он (вероятностно говоря) мог бы и сам, основываясь лишь на знании открытой информации пользователя, произвести точно такой же обмен. Кроме того, если бы он попытался выдать себя за легального пользователя, то почти наверняка столкнулся бы с запросами, отличными от тех, которые он уже перехватил.

С другой стороны, истинный пользователь системы, очевидно, не должен доверять компьютеру в том, что запросы и в самом деле формируются случайным образом. Из этого следует, что криптографическая схема, на которой основывается предлагаемый выше механизм работы системы, должна быть стойкой против атак на основе выбранного шифртекста (см. § 4.3). Поэтому здесь ***нельзя*** использовать криптосхему вероятностного шифрования Блюма-Гольдвассер (см. § 4.6).

В действительности для того чтобы получить работоспособную систему идентификации пользователей, нужно принять во внимание все нюансы теории однонаправленных функций с потайным ходом. Предположим, например, что в некоторой криптосхеме любое достаточно короткое сообщение, если иметь в

своем распоряжении личный секретный ключ дешифрования, может быть дешифровано менее, чем за одну секунду, но оно также может быть осуществлено и без него за несколько часов. Столь быстрый криптоанализ означал бы крах всей такой криптографической системы при ее использовании для шифрования. Однако для системы идентификации пользователей она была бы приемлемой, потому что центральный компьютер всегда может проверить, как долго пользовательский терминал был занят для ответа. Правильный, но вычисляемый слишком долго, ответ будет точно так же бесполезен для нарушителя, как и отсутствие ответа вообще.

Механизм, который мы описали выше, является на самом деле не схемой идентификации *пользователей*, а скорее схемой идентификации *терминалов*. В ней любой человек, имеющий физический доступ к конкретному терминалу, сможет воспользоваться компьютером, даже если этот доступ к терминалу является незаконным. Для идентификации пользователей одно из решений заключается в требовании, что пользователи должны подтвердить собственные полномочия на своем терминале до того, как начнут использовать его в составе системы. Это может быть сделано посредством классической системы паролей, если и пользовательские пароли, и алгоритм дешифрования хранятся в черном ящике данного терминала, который саморазрушается тогда, когда в него пытаются залезть.

Лучшая же идея сводится к тому, чтобы алгоритм дешифрования реализовать с помощью персональной интеллектуальной карточки пользователя [264, 116, 209]. При таком решении от терминалов требуется лишь минимальная интеллектуальность, причем все они могут быть одинаковыми и позволять пользователю входить в систему физически с любого места, которое оборудовано собственно таким терминалом. Важные результаты, касающиеся опасностей использования интеллектуальных (smart) карточек, упомянуты в [188]. Вообще говоря, было бы намного надежнее, если бы такие карточки имели собственную клавиатуру и дисплей.

Описанный механизм идентификации может использоваться в режиме *непрерывного опроса*, когда центральный компьютер постоянно опрашивает терминал через регулярно повторяющиеся промежутки времени. Такой режим снимает дальнейшие

угрозы со стороны нарушителя, который будет вынужден поджидать законного пользователя, с тем чтобы установить соединение до подключения канала связи к его собственному терминалу (или, что еще лучше во избежание возникающих подозрений, ожидать до те пор, пока законный пользователь не пошлет команду «`logout`», с тем чтобы сразу же после этого перехватывать его канал). Если дополнительные вычисления и соответствующие задержки в такой ситуации являются допустимыми, то было бы надежнее еще и аутентифицировать каждое отдельное обращение к системе, исходящее от пользователя.

Для практических применений лучше всего, если подобный опрос может быть *построен* очень быстро, даже если *подтверждение* самого опроса требует более значительных вычислений. Это связано с тем, что основной компьютер может обслуживать большое количество терминалов, тогда как каждый индивидуальный терминал способен обрабатывать такие опросы в фоновом режиме (тем самым с точки зрения пользователя обеспечивая функционирование с высоким быстродействием). Например, можно было бы использовать принципы работы криптосистемы RSA, но при этом открытую экспоненту каждый раз полагать равной 3 (поскольку известной слабостью применения небольших экспонент при шифровании [210] в данном случае вообще никак нельзя воспользоваться).

Конечно, проблема проверки полномочий кого бы то ни было не ограничивается лишь использованием компьютеров. В общем виде рассматриваемая здесь задача идентификации по типу «вопрос-ответ» может быть решена, используя *технику минимального раскрытия* — понятия, которое рассматривается ниже в § 6.3. Более подробному обсуждению идентификационных протоколов с минимальным раскрытием (точнее, с *нулевым знанием*) посвящена статья Урила Фейге, Эмоса Фиата и Эди Шамира [171]. Описание того, как распознавать потенциальных мошенников, приводится в работах Уво Десмедта и его коллег [148, 147, 94, 24].

Глава 6

Применения

В настоящее время современная криптография (как с секретным, так и с открытым ключом) имеет самые разнообразные и многочисленные применения в нашем управляемом информацией обществе. В этой главе мы детально опишем одни из них и коротко перечислим (в § 5) некоторые другие интересные применения более теоретического свойства.

§ 1. Бросание жребия

В 1982 году на 24-ой компьютерной конференции IEEE (*24th IEEE Computer Conference — CompCon'82*) Мануэлем Блюмом была прочитана чрезвычайно плодотворная лекция под названием «Бросание жребия по телефону: протокол для решения неразрешимых проблем» [51]. Эта лекция оказалась прологом к последующему завораживающему развертыванию событий, в ходе которого решались все более и более «неразрешимые проблемы». Для обзора некоторых из таких наиболее выдающихся недавних криптологических достижений, которые стали возможными благодаря академическим исследованиям, обращайтесь к [69]. На лекции Блюм ввел понятие бросания жребия с помощью примерно следующего очаровательного повествования.

> *Алиса и Боб хотят бросить жребий по телефону. (Они только что развелись, живут в разных городах и хотят решить, кому достанется машина.) Жребий бросает Алиса. Боб считает, что выпадет решка и сообщает об этом по телефону Алисе, после чего слышит,*

как Алиса (на другом конце провода) говорит: «Ну, хорошо... Я бросаю... Орел! Ты проиграл!»

Затруднение в этом наивном протоколе, конечно же, состоит в том, что он позволяет *Алисе* подменить результат жеребьевки после того, как *Боб* сообщил ей о своем предположении. Для того чтобы предотвратить подобный вид мошенничества, *Боб* должен потребовать, чтобы *Алиса* сначала бросила монету и зафиксировала результат. Только после этого *Боб* мог бы высказать свою догадку. При реализации этой идеи возникают два требования, выполнение которых приводит к осуществимому на практике протоколу: *Алиса* не должна иметь возможности бросать жребий после того, как она услышит предположение *Боба*, а *Боб* в свою очередь не должен иметь возможности узнать, как упадет монета до того, как он объявит о своей догадке.

Пусть $f: X \to Y$ — однонаправленная функция и предположим, что X — это конечное множество натуральных чисел, среди которых содержится одинаковое количество четных и нечетных. В предположении, что *Алиса* и *Боб* заранее договорились о функции f, Блюмом и Микэли впервые был предложен следующий протокол:

Алиса выбирает случайно число $x \in X$, затем вычисляет значение $y = f(x)$ и предъявляет его *Бобу*;

Боб пытается отгадать, является ли число x четным или нечетным, и объявляет свою догадку *Алисе*;

Алиса сообщает *Бобу*, оказался он прав или нет, и доказывает ему это, раскрывая x;

Боб посредством проверки $f(x) = y$ либо подтверждает, либо опровергает честность *Алисы*.

Суть представленного протокола заключается в том, что *Алиса* связывает себя (в качестве обязательства) числом x, ввиду того что она сообщает *Бобу* значение $y = f(x)$, однако *Боб*, зная y, не может вычислить x самостоятельно, потому что функция f является однонаправленной. (В § 2 мы предпримем более детальное обсуждение общего понятия подобного *битового обяза-*

тельства.) Интуитивно ясно, что этот протокол является конструктивным, хотя если функция f была выбрана недостаточно тщательно (даже если она и в самом деле однонаправленная), то он все равно оставляет дверь открытой для мошенничества как со стороны *Алисы*, так и со стороны *Боба*.

Если функция f не является взаимнооднозначной, то возможно, что *Алиса* знает такие числа x и $\hat{x}$, для которых $f(x)=f(\hat{x})$ и при этом сумма $x+\hat{x}$ которых нечетна (что заранее не включалось в определение однонаправленности). В таком случае она может объявить $y=f(x)$, фактически не связывая себя обязательством ни посредством x, ни посредством $\hat{x}$. Следовательно, если бы это было так, то *Алиса* могла бы смошенничать. С другой стороны, может быть так, что функция f взаимнооднозначна и что вычисление x из $f(x)$ практически трудноосуществимо, но его то как раз делать заведомо и не обязательно, поскольку оказывается, что *четность* числа x может быть легко определена из вида самого значения $f(x)$. Но даже, если бы четность x будет трудно определить в точности для данного значения $f(x)$, может статься, что ее можно угадать с вероятностью успеха более, чем 50%. В этой ситуации мог бы мошенничать *Боб*.

В предположении, что факторизация больших целых чисел является практически невыполнимой задачей, эти потенциальные трудности можно обойти с помощью следующего протокола:

Алиса выбирает случайным образом какое-нибудь число Блюма n и некоторое $x\in\mathbb{Z}_n$, затем вычисляет $y=x^2 \bmod n$ и $z=y^2 \bmod n$, после чего сообщает числа n и z *Бобу*;

Боб объявляет *Алисе* свое предположение о том, является ли число y четным или нечетным;

Алиса раскрывает *Бобу* числа x и y, а также позволяет ему убедиться в том, что n является числом Блюма;

Боб проверяет, что и в самом деле $y=x^2 \bmod n$, а $z=y^2 \bmod n$.

Иными словами, *Алиса* передает *Бобу* квадратичный вычет z по модулю целого числа Блюма, а *Боб* проверяет, является ли его примитивный квадратный корень четным или нечетным. В

этом случае *Боб*, как мы уже видели в § 4.5, чтобы высказать свое предположение, не может найти ничего лучшего, чем бросить жребий. С другой стороны, критично, что n является целым числом Блюма, так как в противном случае могло бы быть так, что z имеет два квадратных корня различной четности, оба из которых являются квадратичными вычетами. Вот почему *Алиса* должна позволить *Бобу* «убедиться в том, что n является целым числом Блюма». Для этого *Алиса* может сообщить *Бобу* разложение n на множители (теперь для *Боба* будет уже слишком поздно использовать эту информацию, с тем чтобы повлиять на результат своего выбора). Если же *Алиса* хочет применять свое целое число Блюма n еще и для других целей или использовать его в нескольких жеребьевках, то она, для того чтобы убедить *Боба* в своей добропорядочности, может воспользоваться интерактивным протоколом минимального раскрытия [206]. Смотрите также § 3.

Дополнительное преимущество этого протокола состоит в том, что он позволяет бросать жребий как бы *в колодец*. Это означает, что наступает такой момент в протоколе, когда *Алиса* знает исход события и не может его изменить, хотя сама в состоянии будет отсрочить то, что в результате должно быть раскрыто *Бобу*. По отношению к *Бобу* это аналогично тому, как если бы и в самом деле жребий бросался на дно глубокого колодца, расположенного рядом с *Алисой*: она может видеть, как упала монета, но не может ничего изменить, а *Боб* находится слишком далеко, чтобы ее увидеть, но *Алиса* может в конце концов позволить ему подойти поближе и заглянуть в колодец.

В [27] обсуждается протокол для бросания жребия в колодец, который основан на принципах квантовой физики. Он остается секретным даже тогда, когда либо одна, либо обе стороны имеют неограниченные вычислительные ресурсы. Несмотря на то, что в нем одна из сторон в принципе могла бы мошенничать, это потребовало бы такой технологии, которую невозможно будет реализовать в сколь-нибудь обозримом будущем.

§ 2. Схемы битовых обязательств

(написано вместе с Клодом Крепеу и Дэвидом Чаумом)

Понятие *битового обязательства* является сильным и общим примитивом для разработки секретных криптографических протоколов. Цель битового обязательства заключается в том, чтобы позволить *Алисе* взять на себя в качестве обязательства значение некоторого бита информации таким способом, который не позволяет *Бобу* узнать это значение без ее помощи, но который также не позволяет и самой *Алисе* его изменить. Если возможна нераскрываемая схема битовых обязательств, то бросание жребия (так, как оно описано в § 1) становится тривиальным в реализации: *Алиса* берет на себя в качестве обязательства значение некоторого случайно выбранного бита, *Боб* угадывает это значение и аннонсирует свою догадку, а *Алиса* показывает ему, догадался ли он на самом деле или нет. Более тщательно проработанное применение схемы битовых обязательств описывается в следующем параграфе.

Битовое обязательство реализуется посредством некоторого основного понятия, которое мы назовем «*блобом*». Каждый блоб используется как обязательство либо 0, либо 1. Из соображений общности мы не накладываем никаких ограничений на природу блобов. Они могут быть сделаны даже из «улыбки чеширского кота» (или из поляризованного фотона!), если это окажется полезным. Под словами: «*Алиса* берет (принимает на себя) в качестве обязательства некоторый блоб», или, короче: «*Алиса* задается некоторым блобом», мы понимаем, что *Алиса* хранит конкретное значение, соответствующее данному блобу «в уме» и во всех своих последующих действиях придерживается именно этого значения. Если блоб сам по себе может быть представлен в виде битовой строки, как оказывается в большинстве практических случаев, то демонстрацией принятия блоба в качестве обязательства может быть просто показ его в явном виде.

Абстрактными свойствами, которые определяют блобы, являются следующие:

(a) *Алиса* может принять блобы в качестве обязательств: задаваясь некоторым блобом, она в сущности принимает в качестве обязательства некоторый бит.

(b) *Алиса* может *раскрыть* любой блоб, который она приняла в качестве обязательства. При этом она может убедить *Боба* в том, что действительно задавалась именно указанным ею значением соответствующего бита информации, когда принимала в качестве обязательства данный блоб. Таким образом, никакой из блобов не может быть «раскрыт» ею и как 0, и как 1.

(c) *Боб* не в состоянии узнать ничего о том, каким образом *Алиса* может раскрыть какой бы то ни было нераскрытый блоб, который она приняла как обязательство. Это остается справедливым даже после того, как другие блобы уже были раскрыты *Алисой*.

(d) Блобы не содержат никакой *побочной информации*: блобы сами по себе, так же как и тот процесс, посредством которого *Алиса* принимает их в качестве обязательств и раскрывает, не коррелируется никаким другим секретом, который она может захотеть утаить от *Боба*.

Рассмотрим следующую иллюстрацию реализации блоба. Когда *Алиса* хочет принять в качестве обязательства некоторый бит (свойство (a)), она пишет его значение на полу и до того, как *Боб* сможет его увидеть, заклеивает его непрозрачной липкой лентой. Тогда, с одной стороны, *Боб* не может сказать, какой бит скрыт под лентой (свойство (c)), а с другой — *Алиса* после этого в присутствии *Боба* уже не может изменить значение заклеенного бита. Для того чтобы «раскрыть блоб» (свойство (b)), *Алиса* позволяет *Бобу* отклеить ленту и взглянуть на этот бит. Свойство (d) удовлетворяется, если ни способ, с помощью которого значение бита было записано на полу, ни сама лента, ни ее местоположение не коррелируются никаким секретом, который *Алиса* могла бы захотеть скрыть от *Боба*.

Если блобы используются в рамках общепринятого криптографического протокола, в процессе которого происходит обмен электронными сообщениями, то сами они также должны быть представлены в цифровом виде. В этом случае, казалось бы, налицо явное противоречие между свойствами, которым должны удовлетворять блобы. Из свойства (b) следует, что блобы, которые *Алиса* может открыть как 0, должны отличаться от тех,

которые она может открыть как 1. Но тогда что же воспрепятствует *Бобу* обнаружить это различие и нарушить, таким образом, свойство (c)?

Суть одного из возможных ответов на такой вопрос состоит в том, что значение принятого бита должно определяться не самим по себе блобом, а скорее знанием *Алисы* о нем. Если эту идею реализовать аккуратно, то для *Боба* становится невозможным узнать что бы то ни было о том, каким образом *Алиса* может раскрыть любой еще нераскрытый ею блоб, который она приняла в качестве обязательства. Причем можно добиться, что это будет справедливо в самом сильном теоретико-информационном смысле. Тем не менее дополнительные знания могут в принципе позволить *Алисе* нарушить свойство (b).

С другой стороны, если мы настаиваем на том, чтобы свойство (b) выполнялось безусловно, требуется принять двойственное решение, заключающееся в том, что блобы, используемые *Алисой* как обязательство 0, должны быть отличимы от тех, которые она использует в качестве обязательства 1. В этом случае она неизбежно задается неким особым битом всякий раз, когда принимает в качестве обязательства некоторый блоб, и, как следствие этого, *Боб* может в принципе нарушить свойство (c). Несмотря на то, что оба подхода предоставляют возможность одной из сторон смошенничать, это может быть сделано вычислительно неосуществимым при соответствующих криптографических предположениях.

В качестве элементарного (хотя, быть может, и не очень надежного) примера рассмотрим два изоморфных графа G и H, о которых *Алиса* и *Боб* условились заранее. Предположим, что *Алиса* убеждена в том, что они изоморфны, хотя в действительности и не знает никакого изоморфизма между ними. Более того, предположим, что она неспособна в вычислительном смысле найти такой изоморфизм за приемлемое время. (Давайте отложим до § 3 вопрос о том, каким образом *Алиса* может убедиться в том, что данные графы являются изоморфными, не узнав при этом самого изоморфизма.) В этих предположениях *Алиса* договаривается с *Бобом*, что любой граф, для которого она сможет указать изоморфизм с G (соответственно с H) является принятием в качестве обязательства бита 0 (соответственно 1).

Что касается свойств, определяющих блобы, то в приведен-

ном примере свойство (a) выполнено потому, что *Алиса* может задаться битом 0 (соответственно 1), случайным образом переставив вершины G (соответственно H) и предъявив получившийся граф-блоб *Бобу*. Для того чтобы раскрыть блоб, *Алисе* достаточно указать *Бобу* этот известный ей изоморфизм для любого из двух графов, G или H. Свойство (b) выполняется, если и только если *Алиса* не может найти изоморфизма между G и H. Точнее, для того чтобы *Алиса* могла нарушить свойство (b), она должна получить такую информацию, которая позволит ей легко обнаружить подобный изоморфизм. Свойство (c) безусловно справедливо, потому что блобы, используемые *Алисой* в качестве обязательства 0, в теоретико-информационном смысле неотличимы от тех, что используются как обязательство 1 — ведь и те, и другие являются случайными изоморфными копиями G (а, следовательно, и H). Наконец, свойство (d) удовлетворяется, потому что *Алиса* переставляет вершины G случайным образом, то есть таким способом, при котором перестановка не коррелируется ничем вообще.

Таким образом, приведенный выше пример иллюстрирует именно тот случай, в котором не блоб сам по себе (некий граф, который изоморфен и G, и H) определяет некоторый бит информации, а, скорее, знание *Алисы* об этом блобе (фактический изоморфизм, известный *Алисе*, между этим графом и либо графом G, либо графом H).

Более практичный, хотя в техническом отношении и более сложный пример такого же типа схемы битовых обязательств основан на предположении о трудности решения задачи дискретного логарифмирования (см. § 4.1) [112, 62]. Пусть p — большое простое число и пусть α является примитивным (первообразным) корнем по модулю p (порождающим $\mathbb{Z}_p^*$ — мультипликативную группу обратимых элементов кольца классов вычетов по модулю p). Напомним, что для любого заданного целого числа y можно легко вычислить $\alpha^y \bmod p$, но до сих пор не известно никакого эффективного алгоритма обращения этой процедуры. Более того, предположим, что вычисление y из α, p и $\alpha^y \bmod p$ остается практически невыполнимым, даже если известно разложение числа $p-1$.

Сначала *Алиса* и *Боб* договариваются о подходящем простом числе p, для которого им обоим известно разложение числа $p-1$.

Они также договариваются об α, генераторе $\mathbb{Z}_p^*$. Благодаря своему знанию множителей $p-1$, они оба могут с определенностью убедиться, что p и в самом деле простое число, а α является генератором $\mathbb{Z}_p^*$ [236]. Указанные параметры p и α могут быть сделаны общедоступными в том смысле, что к ним можно предоставить доступ всем, кто захочет принять участие в протоколах битовых обязательств. Вначале *Боб* выбирает, помимо всего прочего, случайное число $s \in \mathbb{Z}_p^*$ такое, что $s \neq 1$, и передает его *Алисе*. Согласно нашему предположению, *Алиса* не может вычислить e, $0 \leqslant e \leqslant p-2$, для которого $s \equiv \alpha^e \pmod{p}$.

Затем, для того чтобы принять в качестве обязательства значение некоторого бита b, *Алиса* выбирает случайным образом число y, лежащее в интервале от 0 до $p-2$, и вычисляет $x = s^b \alpha^y \bmod p$. После этого она передает x, которое и является блобом, *Бобу*, но сохраняет в секрете y в качестве своего *удостоверения* этого блоба, позволяющего ей открывать x как бит b. Ясно, что как в качестве обязательства 0, так и в качестве обязательства 1, *Алисой* может быть использован любой элемент $\mathbb{Z}_p^*$, который зависит только от ее знаний о нем. Более того, элементы $\mathbb{Z}_p^*$, получающиеся с помощью этого процесса, имеют равномерное распределение вероятностей, независящее от того бита, который *Алиса* пожелает взять в качестве обязательства. Следовательно, свойство (c) здесь также безусловно справедливо — блобы, принятые в качестве обязательств *Алисой*, не содержат никакой информации о том способе, которым она могла бы их раскрыть. Свойство (b) удовлетворяется в вычислительном смысле, потому что *Алиса* могла бы легко получить e (что, как мы предположили, является для нее практически неосуществимым) из знания y_0 и y_1 таких, что $\alpha_0^y \equiv s \cdot \alpha_1^y \pmod{p}$, поскольку тогда $e = y_1 - y_2 \bmod (p-1)$. Наконец, свойство (d) выполняется, потому что *Алиса* выбирает y случайно.

Предположение о трудности решения задачи дискретного логарифмирования может также использоваться для реализации схемы битовых обязательств дуального типа [78]. Снова, пусть p — большое простое число, и пусть α является генератором $\mathbb{Z}_p^*$. Пусть u — наименьшее целое, такое, что 2^u не делит $p-1$. Для любого заданного $s \in \mathbb{Z}_p^*$ обозначим через $\mathit{hard}(s)$ u-тый младший значащий бит единственного целого числа e, такого, что $0 \leqslant e \leqslant p-2$ и $s = \alpha^e \bmod p$. В предположении о сложности

задачи дискретного логарифмирования практически ничего невозможно узнать о $\mathit{hard}(s)$ (при заданных лишь p, α и случайно выбранном s), потому что доказано, что решение этой задачи является столь же трудным, сколь и само дискретное логарифмирование (в сильном вероятностном смысле) [285] (хотя $u-1$ младших значащих битов e могут быть легко вычислены для данного s).

Сначала *Алиса* и *Боб* договариваются о p и α точно так же, как и в предыдущем примере. Для того чтобы задаться некоторым битом b, *Алиса* случайным образом выбирает целое число y, лежащее на отрезке от 0 до $p-2$ и, кроме того, полагает b равным u-тому младшему значащему биту y. Затем она вычисляет блоб $x = \alpha^y \bmod p$ и передает его *Бобу*. Ясно, что тогда свойства (a) и (d), определяющие этот блоб, выполняются точно так же, как и в двух предыдущих реализациях битовых обязательств. Однако свойство (b) теперь справедливо безусловно, потому что α является генератором $\boldsymbol{Z}_p^*$. Следовательно, дискретный логарифм x определяется однозначно, а это значит, что также однозначно задается и u-тый младший значащий бит y. Наконец, свойство (c) выполняется с вычислительной точки зрения при сделанном ранее криптографическом предположении, так как бит, закодированный блобом s, суть ни что иное, как $\mathit{hard}(s)$.

В [78] были предложены некоторые другие схемы битовых обязательств. Существует также реализация, которая остается нераскрываемой, даже если и *Алиса*, и *Боб* имеют неограниченные вычислительные ресурсы, но она основывается на постулатах квантовой физики. Кроме того, еще одно применение является безусловно нераскрываемым для всех сторон, которые взаимодействуют в *многопользовательской* среде (смотрите § 5) в предположении, что не более одной трети всех участников этого взаимодействия являются нечестными [111, 37].

§ 3. Доказательства с наименьшим раскрытием

(написано вместе с Клодом Крепеу и Дэвидом Чаумом)

Предположим, что *Алиса* хочет *доказать*, что ей известна некая информация. Например, это может быть какая-то теорема или разложение большого целого числа на простые множители. Предположим далее, что информация *Алисы* является *проверяе-*

мой в том смысле, что существует эффективная процедура, способная подтвердить справедливость этой информации. Тогда, если *Боб* захочет *проверить*, что *Алисе* и в самом деле известна та информация, о знании которой она заявляет, то *Алиса*, для того чтобы убедить в этом *Боба*, может просто предоставить ему эту информацию, чтобы он сам смог выполнить проверяющую процедуру. Подобное доказательство было бы доказательством *с наибольшим раскрытием*, так как в результате *Бобу* стала бы известна вся эта информация. Поэтому в дальнейшем он сам мог бы не только сообщать ее кому угодно, но даже утверждать, что она изначально является его собственной информацией.

В этом параграфе мы опишем общий и эффективный протокол для проведения доказательств *с наименьшим раскрытием*. Этот протокол позволяет *Алисе* убедить *Боба* после любого его разумного сомнения в том, что она действительно обладает некой информацией, которая с успехом пройдет проверяющую процедуру, но способ, которым она это делает, никоим образом не помогает ему определить саму эту информацию. Например, если такой информацией *Алисы* является доказательство какой-нибудь теоремы, то *Боб* будет в полной уверенности, что *Алиса* действительно знает, как ее доказывать, и, следовательно, что теорема верна. Однако *Бобу* при этом не предоставится никакого намека на то, как такое доказательство (за исключением разве что верхней границы на его длину) можно было бы осуществить ему самому. Следовательно, несмотря на то, что оригинальная информация *Алисы* является проверяемой, та убежденность в ее справедливости, которую получает *Боб*, фактически ничего ему не дает. В частности, проведение протокола с *Алисой* не обязано (а во многих случаях и *не будет*) предоставлять возможности *Бобу* впоследствии убедить в этом таким же образом кого-нибудь еще.

В общем случае это достигается с помощью интерактивного протокола, состоящего из нескольких раундов. В каждом раунде *Бобу* позволяется задавать *Алисе* один из двух возможных очень трудных вопросов. Эти вопросы задаются таким образом, что *Алиса* сможет ответить правильно на оба вопроса *Боба* лишь в том случае, если она действительно владеет тем секретом, о знании которого заявляет. Однако способ, которым *Алиса* это делает, не предоставляет *Бобу* никакой информации о сохраняемом

ею в секрете утверждении. Поскольку *Алиса* не может раньше времени предугадать, какой из вопросов задаст *Боб*, то каждый раунд протокола увеличивает его уверенность в справедливости ее секрета. В действительности она может надеяться обдурить его за k успешных раундов с экспоненциально уменьшающейся вероятностью 2^{-k}. Мы будем называть такую технику приемом *разрезания-и-выбора*, поскольку каждый ее раунд подобен классическому «протоколу», с помощью которого двое детей делят кусок торта — один из них режет, а другой выбирает ту часть, которая ему больше нравится. Необычайная полезность приема «разрезания-и-выбора» состоит в том, что она обеспечивает экспоненциально возрастающую уверенность ценой лишь линейного увеличения числа раундов.

Самое раннее использование техники «разрезания-и-выбора», которое нам известно в связи с криптографическими интерактивными протоколами, было предложено Майклом Рабином в 1977 году [296]. Понятие *интерактивного протокола* было впоследствии формализовано, а Шафи Гольдвассер, Сильвио Микэли и Чарльзом Раковом в [199] было введено понятие *нулевого знания*. (Понятие нулевого знания сходно с понятием минимального раскрытия, за исключением того, что оно требует, чтобы *Боб* не мог получить из протокола вообще *ничего*, кроме знания о том, что *Алиса* обладает той информацией, о которой она заявляет.) Ласло Бабаи формализовал также понятие, сходное с понятием интерактивных протоколов [11]. Модели, представленные в [199, 11], предполагают, что доказывающий обладает неограниченными вычислительными ресурсами. Дэвидом Чаумом была предложена несколько иная модель, в которой большое внимание уделено безусловной нераскрываемости (в смысле Шеннона) секретной информации доказывающего [101], даже если проверяющий имеет неограниченные вычислительные мощности. Здесь же мы сосредоточимся на модели, в которой предполагается, что все затрагиваемые стороны должны иметь «разумные» вычислительные ресурсы.

Рассмотрим довольно простой, но элегантный пример предложенного Одедом Голдрейчем, Сильвио Микэли и Эви Вигдерсоном [196] протокола доказательства с минимальным раскрытием, который позволяет *Алисе* убедить *Боба* в том, что она знает некоторый изоморфизм между двумя заданными графами G и H,

таким образом, что это никак не помогает *Бобу* разгадать этот изоморфизм. Данный пример иллюстрирует применение принципа «разрезания-и-выбора» (и раскрывает ссылку, которая была сделана в конце § 2 в том абзаце, где приводится описание примера схемы битового обязательства, основанной на изоморфизме графов). В произвольном раунде этого протокола *Алиса* случайно переставляет вершины G, чтобы получить некоторый граф K, изоморфный как графу G, так и графу H, и передает K *Бобу*. Очевидно, что если *Алиса* правдива, то она должна знать также и изоморфизм между H и K (посредством композиции сделанной перестановки с перестановкой, определяющей изоморфизм между G и H, о знании которого она заявила).

После этого *Боб* выбирает G или H и предлагает *Алисе* предоставить ему изоморфизм между выбранным им графом и K. Ясно, что *Алиса* в состоянии решить эту трудную задачу, только если она не врет. Однако *Алиса*, вне зависимости от того, каким способом она получает K (даже полагая, что она попытается мошенничать), не сможет быть готова ответить на оба возможных вопроса, если она в действительности не знает изоморфизма между G и H. Следовательно, поскольку *Алиса* не может предсказывать, какой из вопросов будет поставлен *Бобом*, любая ее попытка смошенничать будет выявлена им с вероятностью равной по крайней мере одной второй. Вероятность не выявленного обмана может быть, таким образом, сделана сколь угодно малой при помощи повторения этого процесса достаточно большое число раз. Несколько труднее, однако, увидеть, что этот протокол не в состоянии помочь *Бобу* определить изоморфизм между графами G и H. Для выяснения дальнейших деталей обращайтесь к [196].

Описанный протокол хорошо работает для такой специфической проблемы, как изоморфизм графов. Однако для того чтобы показать, что справедливость *любой* эффективно проверяемой информации может быть подтверждена с помощью доказательства с минимальным раскрытием, нам необходим более сильный результат. С помощью теоремы Стевена Кука [120] техника доказательства с минимальным раскрытием для задачи о выполнимости (или любой другой NP-полной проблемы [186]) будет автоматически давать такое доказательство для любого позитивного утверждения, относящегося к произвольному языку из NP. При

подходящих криптографических предположениях такие протоколы были получены Одедом Голдрейчем, Сильвио Микэли и Эви Вигдерсоном для такой *NP*-полной проблемы, как 3-раскраска графов [196], а Дэвидом Чаумом — для задачи о выполнимости [103]. Сходные результаты для булевых схем были получены независимо, но несколько позднее, Жилем Брассаром и Клодом Крепеу [79, 78].

Понятие доказательства с минимальным раскрытием естественным образом распространяется на *вероятностно* проверяемую информацию [78]. В принципе можно даже *публиковать* доказательства с минимальным раскрытием (так что интерактивность, вообще говоря, не является обязательной) [55]. По этому поводу читайте также [183, 54, 114, 21, 206, 345, 346, 78]. О дискуссии на эту тему относительно Министерства обороны Соединенных Штатов сообщается в [189, 103]. Сейчас же мы приведем конструкцию Чаума в деталях.

Предположим, что *Алисе* известно какое-то выполнимое присваивание истинностных значений для некоторой булевой формулы. Протокол, который будет представлен ниже, позволяет *Алисе* убедить *Боба* в том, что она знает это присваивание, не выдавая при этом никакой информации о нем. (Заметим, что булева формула сама по себе не является секретной — таково лишь то выполнимое присваивание, которое *Алиса* хочет утаить от *Боба*.) В качестве примера рассмотрим булеву формулу

$$\Psi = \big[(p \wedge q) \oplus (\bar{q} \vee r)\big] \wedge \overline{\big[(\bar{r} \oplus q) \vee (p \wedge \bar{r})\big]}$$

и предположим, что $\langle p = \mathtt{true}, q = \mathtt{false}, r = \mathtt{true}\rangle$ — выполнимое секретное присваивание *Алисы*. (Само собой разумеется, что рассматриваемый пример нельзя считать реальным, так как для *Боба* было бы слишком легко выяснить, каким образом можно сделать выполнимым столь простое булево выражение.)

На первом шаге *Алиса* и *Боб* договариваются о реализации булевой схемы, вычисляющей Ψ. Для простоты мы используем в качестве базиса только двухвходовые вентили и инверторы. Схема для Ψ изображена на рис. 6.1 на стр. 101. Кроме того, на этом рисунке приведено выполнимое присваивание *Алисы* и представлены таблицы истинности для каждого из вентилей (за исключением вентилей отрицания).

Отметим, что в каждой таблице истинности выделена одна из

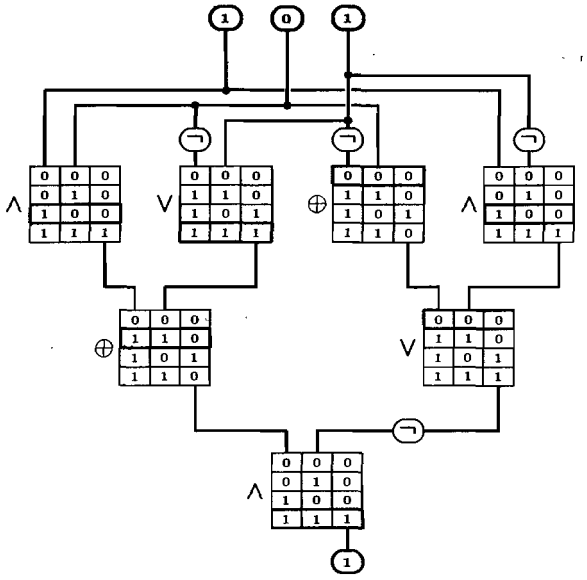

Рис. 6.1. Булева схема с полными таблицами истинности

строк, которая соответствует результатам работы рассматриваемой схемы в соответствии с выполнимым присваиванием *Алисы*. Анализируя эти строки, довольно легко проверить, что Ψ является выполнимой. Это можно выяснить с помощью простых независимых проверок на совместность каждой цепи. Например, выходное значение для верхнего левого вентиля $\wedge$ равно 0, которое, в свою очередь, является первым входным значением в средней строке таблицы истинности для вентиля $\oplus$. К тому же первые входы верхнего левого и верхнего правого вентиля $\wedge$ являются одинаковыми, как это и должно быть, так как они соответствуют одной и той же входной переменной. Наконец, выход последнего вентиля равен 1. Заметим также, что просмотр этих выделенных строк выявляет и соответствующее выполнимое присваивание (даже если оно не было выписано явно). Поэтому нам потребуется такой протокол, который позволил бы *Алисе* убедить *Боба*, что она знает, каким образом должно быть сделано вы-

деление строк в каждой таблице истинности, не раскрывая при этом какой бы то ни было информации о том, что это за строки.

Как и ранее, это достигается посредством интерактивного протокола, состоящего из нескольких раундов. В каждом раунде *Алиса* сначала «перетасовывает» все таблицы истинности рассматриваемой схемы и принимает в качестве обязательства соответствующие им семейства блобов (см. § 2). После этого *Боб* задает *Алисе* один из двух трудных проверочных вопросов: в качестве ответа на один из них от *Алисы* требуется, чтобы она показала, что принятые ею блобы и в самом деле кодируют правильное перетасовывание таблиц истинности схемы; для ответа на другой вопрос *Алисе* необходимо открыть строку, которая должна быть выделенной, в предположении, что перетасовывание сделано правильно. Как и всегда, вопросы формулируются так, чтобы *Алиса* была в состоянии правильно ответить на *оба* из них, только если она знает, как обеспечить выполнимость схемы, но отвечая на какой-то *один* из них, она не должна предоставлять никакой информации о том, как это можно сделать. Как было показано ранее, поскольку у *Алисы* нет возможности заранее предугадать, какой из вопросов будет задан *Бобом*, то с каждым успешно проведенным раундом доверие *Боба* к утверждению *Алисы* будет возрастать.

Перетасовывание каждой таблицы истинности, которое осуществляет *Алиса*, состоит из случайной перестановки ее строк в сочетании со случайным инвертированием содержимого ее столбцов. Проиллюстрируем этот механизм на примере. На рис. 6.2 на стр. 103 в части (a) изображена таблица истинности для булевой конъюнкции ($\wedge$). Строки этой таблицы случайным образом переставляются и получается таблица, которая приведена на рис. 6.2 (b). (При этом каждая из 24 возможных перестановок, включая тождественную перестановку, может быть выбрана с одинаковой вероятностью.) Затем для каждого из трех столбцов таблицы истинности случайно задается значение некоторого бита. И наконец, для каждого из столбцов, тогда и только тогда, когда соответствующий ему случайный бит равен 1, вычисляется его дополнение (при котором инвертируются все биты, составляющие данный столбец) так, как это показано на трех промежуточных таблицах. Окончательный результат представлен на рис. 6.2(c). Заметим, что полностью перетасованная таким образом таблица

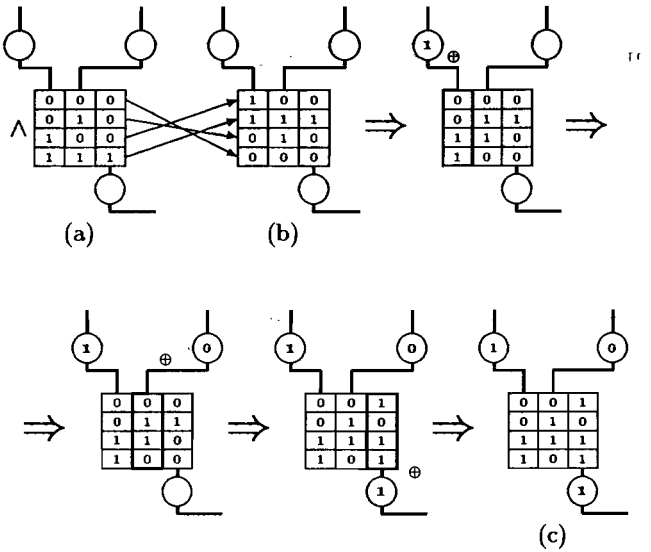

Рис. 6.2. Пример перетасовывания таблиц истинности

истинности может быть, тем не менее, безошибочно опознана как представляющая булеву конъюнкцию (при условии, что биты соответствующих операций дополнения столбцов, которые указываются внутри кружочков для отвечающих им цепей, заранее определены).

Дополнения при этом должны браться согласованно: биты всех столбцов таблиц истинности, соответствующие одной и той же цепи схемы, должны либо инвертироваться, либо оставаться теми же, что и были. Это достигается выбором битов дополнений случайно и независимо для каждой соответствующей цепи. (Для простоты мы никогда не инвертируем выход результирующего вентиля.) На рис. 6.3 на стр. 104 приведен результат некоторых случайных перестановок и дополнений столбцов в таблицах истинности для нашей исходной схемы, изображенной на рис. 6.1.

После того как с проверяемой схемой сделано то же самое, что в результате в качестве примера показано на рис. 6.3, *Алиса* принимает ее в качестве обязательства: для каждого бита табли-

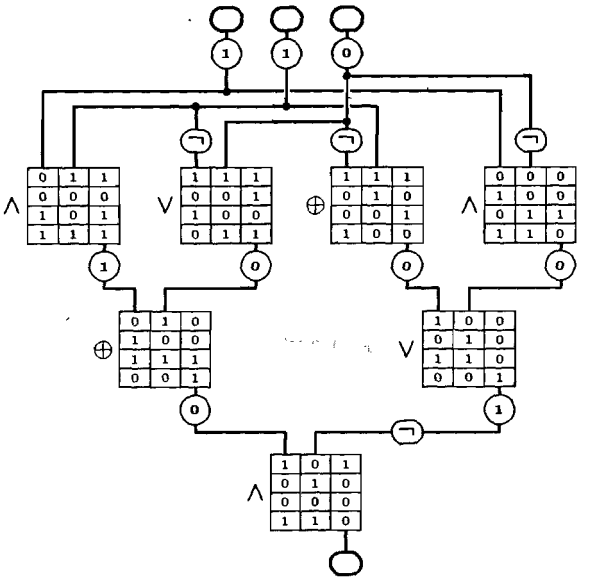

Рис. 6.3. Схема с перетасованными таблицами истинности

цы истинности *Алиса* задается некоторым блобом, о котором она знает, как он, соответствующим образом, должен быть открыт. (В действительности для нее не обязательно задаваться битами дополнений, но они в тот момент должны оставаться в секрете.) Можно представлять себе, что *Алиса* изобразила рис. 6.3 на полу, но до того, как позволить *Бобу* взглянуть на него, заклеила все его биты непрозрачной липкой лентой. В этот момент процесс фазы «разрезания» для *Алисы* заканчивается и начинается фаза «выбора» для *Боба*. Он просит *Алису* убедить его в ее честности, предлагая дать ответ на выбранную им случайным образом одну из двух задач, задачу **A** или задачу **B**, которые заключаются в следующем:

Задача **A**. *Алиса* должна открыть все без исключения блобы, которые она только что приняла в качестве обязательств. Более того, она должна также показать все биты допол-

нений, которые использовались ею в процессе перетасовывания. Продолжая наше предыдущее образное представление, *Алиса*, для того чтобы продемонстрировать *Бобу* свой эквивалент схемы, вроде той, что изображена на рис. 6.3, должна отклеить все заклеенные ею места. Это позволяет *Бобу* проверить, что информация, которая была скрыта посредством блоба, соответствует правильным перестановкам и дополнениям таблицы истинности проверяемой булевой схемы.

Задача **B.** *Алиса* открывает только блобы, которые соответствуют одной строке в каждой таблице истинности. Строки, которые должны открываться, будут в точности такими же, как те, что обведены на рис. 6.1, но, возможно, с новым расположением, которое определяется перестановкой строк. По-прежнему, как бы продолжая наше образное представление, *Алиса* должна выборочно отклеить кусочки липкой ленты, чтобы показать *Бобу* часть схемы, которая соответствует той, что изображена на рис. 6.4 на стр. 106. Это предоставляет ему возможность проверить согласованность каждой цепи и то, что значением выхода всей проверяемой схемы действительно является 1.

Для того чтобы можно было доказать корректность приведенного протокола, должны удовлетворяться три требования. В предположении, что выполнены только те четыре свойства, которыми определяются блобы (см. § 2), в [78] доказано, что за исключением, возможным, разве что, с экспоненциально малой вероятностью, справедливо следующее:

1) *Алиса* может довести до конца свою часть протокола при условии, что она знает выполнимое присваивание для Ψ. (Конечно, никакой протокол не сможет заставить *Боба* удостовериться в честности *Алисы*, даже если та сообщит ему выполнимое присваивание в явном виде, поскольку он всегда может отказаться ее слушать.)

2) Если *Алиса* не знает выполнимого присваивания для Ψ, то к каким бы уловкам во время протокола она не прибегала, чтобы убедить *Боба* в своей правоте, *Боб* всегда сможет уличить ее в мошенничестве.

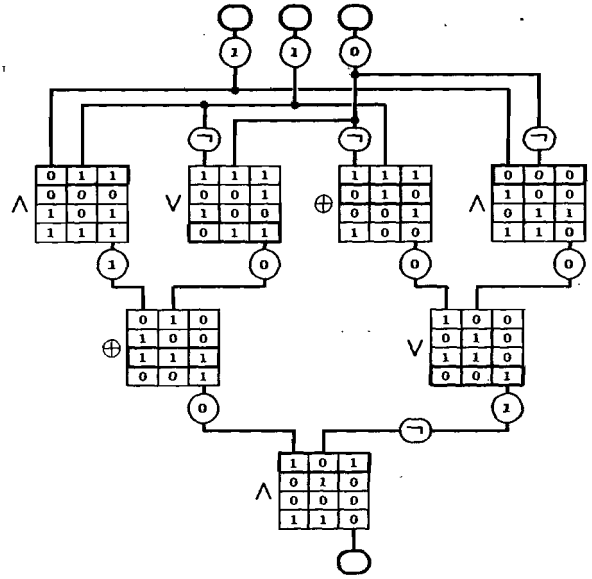

Рис. 6.4. Демонстрация существования выполнимого присваивания

3) Если *Алиса* знает некоторое выполнимое присваивание для Ψ, и если она в соответствии с данным присваиванием неукоснительно следует своей части протокола, то в ходе его она не предоставит *Бобу* никакой информации, которая могла бы помочь ему определить ни это конкретное, ни какое-то другое выполнимое присваивание (или хотя бы какую-то частичную информацию о нем). Все это остается справедливым даже тогда, когда *Боб* произвольно отклоняется от поведения, предписанного ему протоколом.

Несмотря на то, что в нашем протоколе третье требование удовлетворяется, из этого, вообще говоря, *не* следует, что *Боб* не может получить какую-нибудь дополнительную информацию, кроме того факта, что *Алиса* действительно знает выполнимое присваивание для Ψ. Например, возможно, что только *Алиса* владеет технологией и знаниями, которые необходимы для того,

чтобы принять блобы в качестве обязательства. В таком случае *Бобу* может быть доступна некоторая информация, которую он не в состоянии получить самостоятельно, хотя она и не будет иметь отношения к выполнимому присваиванию. Иными словами, наш протокол — это протокол с минимальным раскрытием, но он может и не быть протоколом с «нулевым знанием» в терминологии [199].

Интуитивно ясно, что некий протокол будет протоколом с *нулевым знанием*, если третье требование, которому он должен удовлетворять, усилить до такого, согласно которому *Боб* в ходе протокола не может выяснить вообще ничего, кроме того, что *Алисе* известно выполнимое присваивание. Точнее, *Боб* должен быть в состоянии воспроизвести полностью свой диалог с *Алисой* фактически без какого бы то ни было разговора с ней. За формальным определением такого протокола обращайтесь к [199].

Однако и наш протокол будет протоколом с нулевым знанием, если в нем свойство 4, определяющее блоб (см. § 2), усилить так, чтобы быть уверенным, что *Боб* не извлечет *ничего* из того процесса, согласно которому *Алиса* принимает блобы в качестве обязательств, а также что *Боб* получит *только* биты, определяющиеся ходом того процесса, в соответствии с которым *Алиса* открывает ему некоторые из них. Следуя технике доказательства Шафи Гольдвассер, Сильвио Микэли и Чарльза Ракова из [199], будем говорить, что блобы являются *моделируемыми*, если, в дополнение к свойствам 1, 2 и 3, они удовлетворяют свойству

4′) *Боб* может смоделировать то, что должно быть обеспечено им в результате процесса принятия *Алисой* в качестве обязательств блобов, которые она может открыть как 0, и блобов, которые она может открыть как 1. *Боб* в состоянии также смоделировать процесс, посредством которого *Алиса* будет открывать блобы, принятые ею в качестве обязательств.

Заметим, что все реализации блобов, описанные в § 2, являются моделируемыми.

Если в *теоретико-информационном* смысле невозможно различить блобы, которые используются *Алисой* для того, чтобы принять их в качестве обязательства 0, и блобы, которые используются для того, чтобы принять их в качестве обязательства 1,

и если эти блобы являются моделируемыми, то описанный выше интерактивный протокол для выполнимости булевых выражений становится совершенным протоколом с нулевыми знаниями в терминологии [196]. Это достигается, например, если он использует в качестве блобов те, что основаны на сложности задачи дискретного логарифмирования, и являются первыми из тех, что описаны в § 2. Заметим, что такие блобы не соответствуют модели [199], так как бесконечные вычислительные ресурсы доказывающего могут помочь ему смошенничать, что объясняет, почему результат Фортноу [177] в этом случае становится неприменимым. (Здесь хотелось бы обратить ваше внимание на то, что совершенный интерактивный протокол с нулевыми знаниями для задачи о выполнимости в [79] определен некорректно.)

Интересная ситуация возникает, если рассматривать некоторый вариант протокола, в котором все раунды проводятся в параллель: *Алиса* принимает в качестве обязательств все до единого блобы, соответствующие k схемам таким, как на рис. 6.3, *Боб* передает ей строку с проверочными вопросами, и *Алиса* открывает блобы так, как того требует ответ на соответствующий вопрос. (В некоторых ситуациях этот вариант протокола может быть более эффективным.)

Предположим для простоты, что блобы — это битовые строки и что *Алиса* задается некоторым блобом, показывая его явно. Рассмотрим следующую стратегию для *Боба*: после получения от *Алисы* блобов, которые соответствуют всем k схемам со случайно переставленными строками и инвертированными столбцами таблиц истинности, он объединяет эти блобы вместе (конкатенируя их в одну строку) и использует полученный результат в качестве аргумента некоторой однонаправленной функции. Затем он использует первые k битов значения этой функции для определения k проверочных вопросов, которые будет ей задавать. Даже если проведение этого модифицированного протокола с *Алисой* и не помогает *Бобу* узнать что-либо о ее секрете, сама запись его может помочь ему убедить кого-нибудь еще в существовании этого секрета, поскольку *Боб* почти наверняка не сможет воспроизвести такую запись каким-то другим способом (даже если блобы являются моделируемыми). Это приводит к любопытному феномену: запись параллельной версии протокола может не содержать никакой информации о действительном

секрете *Алисы* (в смысле шенноновской теории информации), даже несмотря на то, что такая запись может быть использована для убеждения кого-то еще в существовании этого секрета!

Если важно, чтобы протокол проводился в параллель (возможно, из соображений большей эффективности), то полезно знать, что он остается протоколом с нулевым знанием, если определяющее его свойство 4' еще более усилить. Мы скажем, что блобы являются *хамелеонами*, если в дополнение к свойствам 1, 2 и 3 они удовлетворяют свойству

4'') *Боб* может смоделировать все, что должно быть обеспечено им в том процессе, с помощью которого *Алиса* принимает блобы в качестве обязательств. Более того, для каждого из этих блобов *Боб* может смоделировать как процесс, с помощью которого *Алиса* открывала бы их как 0, так и процесс, с помощью которого она открывала бы их как 1.

Иными словами, блобы-хамелеоны позволяют *Бобу* самому делать именно то, что свойство 2 предписывает делать *Алисе*, тем самым избавляя ее от этого. Преимущество блобов-хамелеонов заключается в том, что они предоставляют возможность *Бобу* моделировать напрямую весь свой диалог с *Алисой*. Это остается справедливым даже тогда, когда *Боб* произвольно отступает от предписанного ему поведения. Даже если *Алиса* и *Боб* имеют примерно одинаковые вычислительные ресурсы, это свойство может иногда достигаться в том случае, когда *Боб* обладает некоторой дополнительной информацией. Так, например, блобы (описанные в § 2), основанные на изоморфизме графов, являются хамелеонами, если и только если *Боб* знает изоморфизм графов G и H, о которых идет речь. Соответственно первый из блобов, основанных на дискретном логарифме, является хамелеоном тогда и только тогда, когда *Боб* знает дискретный логарифм s. Это возможно, если вместо того чтобы выбирать случайным образом s, как это предлагалось ранее, он в интервале от 1 до $p-2$ выбирает случайным образом число e, а s вычисляет как $\alpha^e \bmod p$. Однако такие двойственные блобы, основанные на дискретном логарифме, конечно, не являются хамелеонами, так как каждый из них определяет один бит однозначно.

Возникает естественный вопрос: кому лучше доверять — до-

казывающему или проверяющему? «Мошенничество» приобретает разный смысл в зависимости от того, кто имеется при этом в виду, *Алиса* или *Боб*. Для *Боба* мошенничество означает, что он узнаёт что-то еще, кроме того факта, что *Алиса* имеет доступ к той информации, о владении которой она заявляет. Быть может, например, он и не получит полностью гамильтонову цепь, которую безуспешно пытается найти, но узнает достаточно для того, чтобы существенно сократить время ее поиска. С другой стороны, мошенничество для *Алисы* означает, что ей удается убедить *Боба* в том, что она обладает той информацией, которая будет подвергнута процедуре проверки, в то время как в действительности это не так.

Интересно также делать различие между *удачным* и *дерзким* типами успешного мошенничества [27]. Первый тип имеет отношение к отгадыванию *Алисой*, или *Бобом*, вопреки всем расчетам, той части информации, которая поможет ей, или ему, спокойно осуществить свой обман с уверенностью, что он будет успешным и не обнаружится. Ко второму типу обмана относится совершение *Алисой*, или *Бобом*, некоторых незаконных действий, которые в результате почти наверняка приведут к тому, что ее, или его, мошенничество будет обнаружено в определенный момент в будущем, но при этом, тем не менее, может с экспоненциально малой вероятностью привести ее, или его, к успеху. Наконец, мошенничество следует назвать *ретроактивным* (или *автономным*), если оно может иметь успех спустя какое-то время после завершения протокола при просмотре его записи. Такое мошенничество называется мошенничеством *в реальном времени*, если оно должно завершаться во время осуществления протокола.

Если используется блоб, который является безусловно надежным для *Боба* (в соответствии с протоколами, что приведены в [196, 78]), то *Алиса* может так никогда и не воспользоваться никакой частью своих знаний — *Бобу* может позволить ретроактивно смошенничать открытие какого-нибудь нового алгоритма, даже если этот новый алгоритм и не является достаточно быстрым для ответа за то реальное время, в течение которого проводится указанный выше протокол. Далее, если конкретное криптографическое предположение окажется не вполне обоснованным, для *Боба* все еще имеется (очень небольшая) вероятность удачного

(и, следовательно, без обнаружения) мошенничества. С другой стороны, невзирая ни на какие предположения, единственный тип мошенничества, который может позволить *Алисе* добиться своего в данной ситуации, является дерзким.

В противоположность предыдущему, если используются блобы, которые являются безусловно надежными для *Алисы* (в соответствии с протоколами, приведенными в [103, 79]), то убежденность *Боба* в том, что *Алиса* не может смошенничать, зависит от его веры в соответствующее криптографическое предположение. В соответствии с первой (из описанных в § 2) реализаций блобов, что основаны на дискретном логарифме, *Алиса*, например, сможет «открыть» любой блоб либо как 0, либо как 1, причем так, как ей это будет угодно, только если она будет в состоянии получить дискретный логарифм s до окончания первого раунда протокола, в котором *Боб* задает ей соответствующую сложную задачу. Получение этого логарифма в любое более позднее время было бы для нее бесполезно, так как в противном случае она не сможет подготовиться к ответу. Более того, если указанное криптографическое предположение является вполне обоснованным, то *Алиса* все еще имеет (очень небольшой) шанс попытаться отгадать его наудачу, но она должна набраться дерзости, чтобы предложить провести весь протокол в надежде, что ей и здесь повезет. С блобами, основанными на дискретном логарифме, у *Боба* нет вообще никакой возможности мошенничать, даже несмотря на везение и вычислительные ресурсы. Наконец, в рассматриваемой ситуации ретроактивное мошенничество является бессмысленным, как для *Алисы*, так и для *Боба*.

Если используются блобы, которые являются безусловно надежными для *Алисы*, то дополнительная защита для *Боба* достигается посредством обращения к *Алисе* повторять весь протокол всякий раз с другими типами блобов. Тогда, если бы *Алиса* захотела смошенничать, то это вынудило бы ее быть способной опровергнуть сразу несколько различных криптографических предположений. Например, используя в данной ситуации реализацию блобов из [79], она должна была бы знать эффективно работающие в реальном времени алгоритмы не только факторизации, но и вычисления дискретных логарифмов. Любопытно, что при использовании блобов, которые являются безусловно надежными для *Боба*, результат получается противоположным — повто-

рение протокола с разными типами блобов приводит к тому, что *Бобу* становится *легче* мошенничать, так как он может сделать это, опровергнув (возможно автономно, то есть не в реальном времени) любое (одно) из упомянутых выше криптографических предположений. Тем не менее, *можно* обеспечить большую надежность, если скомбинировать различными способами несколько типов безусловно надежных для *Боба* блобов: каждый раз, когда *Алиса* хочет принять в качестве обязательства некоторый бит b, она для каждого типа блобов случайным образом задается одним из них, а b при этом получается как сумма по модулю 2 соответствующих битов указанных блобов. Естественно, что использование подобной стратегии с блобами, которые являются безусловно надежными для Алисы, приводит только к тому, что в этом случае ей становится легче мошенничать.

Итак, кому же все-таки предпочтительнее доверять, *Алисе* или *Бобу*? Ответа на этот вопрос мы не знаем, но, разумеется, как замечательно иметь выбор! Наконец, рассмотрим следующую провокационную ситуацию. Предположим, что *Алиса* утверждает, будто ей известно доказательство теоремы **T**, и что она, для того чтобы убедить в этом скептически настроенного *Боба*, использует блобы, основанные на дискретном логарифме. Если *Алиса* правдива, то в конце протокола, безотносительно каких бы то ни было криптографических предположений, *Боб* убедится в том, что *Алиса* либо на самом деле знает доказательство теоремы **T**, либо обладает новейшими результатами по вычислению дискретных логарифмов! При этом, в частности, если *Алиса* утверждает: «Я знаю эффективный алгоритм вычисления дискретных логарифмов», то не потребуются никакие криптографические предположения вообще.

§ 4. Защита конфиденциальности

(Написан Дэвидом Чаумом; печатается с его любезного разрешения)

В самое ближайшее время, с помощью почти повсеместного доступа к глобальным компьютерным сетям, вы будете в состоянии оплатить покупку, заказать фильм, который хотели бы посмотреть у себя дома, сделать изменения в своем страховом полисе,

или, быть может, послать электронное «письмо» другу.[1] Хотя, до создания такой единой интегрированной системы, разумеется, все еще довольно далеко, постепенное ее построение уже ведется. Автоматизированная выдача наличных денег с применением банкоматов и даже непосредственные электронные платежи, например, в магазинах, уже используются в многих странах и в еще большем числе планируются. Эти системы обладают огромным потенциалом, с одной стороны, с точки зрения сокращения организационных расходов, увеличения защищенности и эффективности, а с другой — как средство повышения удобства потребителя. Однако, широко распространенный сейчас подход к построению подобных систем таит в себе и некоторые очень серьезные опасности.

Такой распространенный подход требует, чтобы каждый человек идентифицировал себя в такой системе всякий раз, когда он ее использует. При этом многочисленные методы идентификации, аналогичные тем, что применяются при использовании современных пластиковых карточек, в которых запоминаются секретные номера, или подобные сопоставлению отпечатков пальцев в дактилоскопии, являются по существу эквивалентами универсальных идентификационных номеров, используемых, например, при социальном страховании или в паспортах. Такие идентификаторы предоставляют возможность, сопоставляя объединенные воедино с помощью компьютерной сети финансовые, служебные, медицинские и различные другие персональные данные каждого человека, отслеживать его частную жизнь в совершенно беспрецедентной степени.

Более того, в связи с новой легитимностью успехов в таком «автоматическом распознавании образов» снова возникают давнишние опасения относительно электронного Большого Брата (Big Brother)[2]. При помощи подобных методов любое подключение к тем широкомасштабным системам, которые в настоящее время планируются, позволит автоматически распределить по категориям всех людей в соответствии со «следами» их обмена

[1] Несмотря на то, что этот параграф, как, впрочем, и вся книга, был написан в 1988 году, и многие из указанных в нем «будущих» возможностей использования компьютерных сетей стали реальностью, актуальность темы, которой он посвящен, только возрастает.

[2] Зловещий символ знаменитого романа Дж. Оруэла «1984».

электронной информацией: во всех местах, где они платят, в тех организациях, с которыми они общаются, там, куда эта информация передается и т. д. — в форме некоего невидимого наблюдения за всеми гражданами. Допустимые механизмы, сталкивающиеся с легкостью просмотра и изменения автоматизированных данных, кажутся бессильными предотвратить подобные опасности или защитить каждого отдельного человека от возможных в результате этого ошибок и дискриминации.

Тем не менее всех перечисленных проблем можно избежать при помощи нового подхода с использованием персональных «компьютерных карточек». Они должны быть похожи на «суперинтеллектуальную» кредитную карточку Visa-Toshiba, с помощью которой уже осваивается этот современный подход. В такую карточку встроен символьный дисплей и клавиатура, наподобие той, что имеется в карманном калькуляторе. Как и во всех современных «карточных» технологиях, надежность применения таких карточек основывается на сложности их подделки со стороны тех лиц, в руки которых эти карточки попадают. Разработанная с учетом нового подхода такая компьютерная карточка отличается тем, что она является вашим собственным персональным карманным компьютером. Поэтому вы свободны в приобретении как аппаратуры, так и программных средств для него любым способом, какой вы сами выберете, и во время осуществления взаимных операций с его помощью вам никогда не потребуется выпускать этот компьютер из рук. Взамен дорогостоящей, но все еще несовершенной стойкости карточек от подделки, которая позволяет обеспечить защиту только для организаций, в новом подходе для предоставления оптимальной защиты, причем как организаций, так и отдельных граждан, предполагается использование передовой техники шифрования.

Компьютер вашей электронной карточки позволяет выполнить любую операцию обращения к системе только после того, как вы наберете на его клавиатуре свой личный секретный авторизационный номер. Такое условие делает вашу карточку совершенно бесполезной для кражи. Кроме того вы можете хранить резервные копии этих номеров у себя дома, а также у друзей или у родственников, так что даже если ваша карточка потеряется или повредится, вы легко сможете завести новую.

Оплата покупок с помощью электронного бумажника

Оплата чего бы то ни было при помощи вашей компьютерной карточки внешне будет осуществляться даже проще, чем это происходит сейчас. Продавцу магазина нужно только ввести цены покупок на электронном кассовом аппарате и через имеющийся в нем интерфейс передать эти числа на соответствующий порт вашей карточки, которые сразу же высветятся на дисплее ее компьютера. Если вы согласны с правильностью чисел, отражающих цены и общую стоимость сделанных вами покупок, которые указаны на дисплее карточки, то единственное, что от вас потребуется, это набрать свой секретный авторизационный ключ, и на табло кассового аппарата сразу же появятся слова: «Оплата принята».

Система работает подобно электронным наличным деньгам — вместо бумажных банкнот различного достоинства, в вашей карточке хранятся специальные номера, которыми обозначены различные денежные номиналы. Такие номера (своего рода электронные банкноты), как раз и предназначены для использования в качестве законных платежных средств. После того, как вы ввели свой авторизационный ключ, компьютер передает набор соответствующих номеров таких банкнот в кассовый аппарат. Аппарат проверяет, правильно ли номиналы указанных электронных банкнот представляют общую сумму платежа, и по глобальной вычислительной сети пересылает их в банк. Центральный компьютер банка, получив эту информацию, в свою очередь, сначала убеждается в законности всех номеров этих электронных банкнот, а затем делает запрос в клиринговую палату с тем, чтобы удостовериться, что они не были использованы ранее. После этого банковское подтверждение «перечисления денег» в магазин отображается высвечиванием слов «Оплата принята» на табло его соответствующего кассового аппарата.

В том случае, когда вы переводите такие деньги из банка в свой электронный бумажник, то получаете в нем записи с новыми номерами банкнот. Бумажные наличные деньги, которые вы изымаете из банка, в принципе легко восстановить по вашему счету, так как их серийные номера могут, очевидно, быть известны в банке. Однако в соответствии с новым подходом, все происходит так, как если бы вы забирали деньги из банка, пере-

давая в него чек, помещенный в конверт из копировальной бумаги, на котором банк ставит штамп с надписью, скажем, «сумма в один доллар». Получив такой конверт со штампом, вы разрываете и выбрасываете его, а проштампованный банком с помощью копировальной бумаги чек используете для того, чтобы сделать покупку. Тот же самый принцип применяется и в предлагаемой электронной системе, только в ней аналогами конверта, чека и надписи являются специальным образом зашифрованные целые числа. Вы не можете подделать такую зашифрованную цифровую подпись, а банк не сможет выяснять, какой числовой «чек» в том или ином «конверте» был им подписан. Фактически, эти ваши платежи являются «безусловно неотслеживаемыми»: простое математическое доказательство показывает, что их нельзя отследить по вашему счету, причем не имеет значения, сколь изобретательной будет криптографическая атака на такую систему, даже если при этом допускается применение самых мощных компьютеров.

Предъявление удостоверений без идентификации

Иногда в государственных и коммерческих организациях официально вводят обязательное предъявление удостоверяющих документов, которые выдаются другими организациями. Раньше такие документы имели вид подтверждающих личность удостоверений, выполненных на бумаге, подобных паспортам, водительским правам или членским билетам. Первоначальная цель большинства подобных удостоверений заключалась в надежной квалификации граждан, подтверждающей их личность, и группирующей в соответствии с экономическим положением или возрастной категорией, водительскими правами или ученой степенью и т.д. Идентификация при этом была лишь средством, позволяющим провести такую классификацию. Однако сегодня при переводе подобных методов, использующих бумажные носители, на автоматизированную безбумажную технологию идентификация позволяет не только сопоставлять, но и непосредственно обращаться к компьютерным файлам других организаций, фактически создавая единую огромную базу данных обо всех частных лицах.

Возможное будущее развитие рассмотренной выше системы

платежей при помощи компьютерных карточек предотвращают возникновение связанных с такой базой данных проблем защиты личности посредством предоставления возможности отказа от идентификации. Но что еще важнее — они фактически защищают законные интересы организаций даже лучше, чем персонификация или ведение личных дел (досье).

Удостоверяющие документы выдаются организациями в виде закодированных подписей, которые невозможно подделать и которые ваша компьютерная карточка обрабатывает точно так же, как и электронные наличные деньги. Компьютеры других организаций определяют свои требования для подтверждения полномочий в виде различных возможных комбинаций таких удостоверений. Если вы хотите показать, что обладаете такими полномочиями, то ваша компьютерная карточка может предоставить убедительное доказательство того, что она содержит одну из этих комбинаций, совершенно не раскрывая при этом, о ком собственно идет речь. Использование шифрования гарантирует организациям, что вы не можете изменить, передать или отменить принадлежность вашего удостоверения личности. А благодаря тому, что ваша карточка зашифровывает «бланки» удостоверений в «конверты» до того, как они подписываются организациями, их никак нельзя связать с вашими удостоверениями, точно так же, как этого не может быть и при ваших электронных платежах. Таким образом, вы сохраняете полный контроль над своей персональной информацией, точно так же, как если бы те компьютерные записи, которые организации содержат о вас сегодня (то есть ваше электронное личное дело), хранились бы только на вашей компьютерной карточке.

Защита электронной почты

Электронная почта с какого-то момента, видимо, будет широко использоваться потребителями. Спектр связанных с ней потенциальных проблем может быть очень широким, начиная от вызывающей раздражение продажи с рассылкой электронных каталогов, содержащих всякую компьютерную ерунду, забивающую каналы связи, до более зловещей возможности, наподобие автоматического «анализа трафика информации», с помощью которого отслеживаются и раскрываются факты человеческих взаимоот-

ношений, заключающиеся в отправлении и получении электронных сообщений, даже тогда, когда содержание самих сообщений надежно зашифровано.

Новый подход принимает во внимание эти проблемы отчасти посредством передачи сообщений на все компьютеры, объединенные системой электронной почты, и не только предотвращает возможность отследить любого конкретного получателя, но и гарантирует, что потенциальный получатель не сможет отрицать факт получения сообщения. Кроме того, современные средства шифрования позволяют персональному компьютеру каждого пользователя системы электронной почты неотслеживаемо скрыть от кого бы то ни было, какое сообщение он посылает. Другие средства сохраняют конфиденциальность содержания сообщения, и дают возможность его получателям убедиться самим — и, в случае необходимости, убеждать других — в авторстве пославшего им данное сообщение. Такая система, естественно, позволяет осуществлять платежи и предъявлять удостоверения личности непосредственно из дома вместе со всеми теми удобствами и защитой, которые предоставляются компьютерными карточками.

Момент решения

У организаций уже есть причины, побуждающие их принять этот новой подход, среди которых: прямые стимулы повышения эффективности, улучшенная защита и уменьшение общей стоимости услуг. По мере того как все большему количеству людей становится известно и об опасностях, присутствующих в нынешнем подходе к автоматизации, и о преимуществах, предлагаемых в альтернативном подходе, эти стимулы могут возрасти в связи с переориентацией потребителей, изменением общественного мнения и даже с помощью принятия соответствующих законодательных актов. Кроме того, наверняка однажды всем станет очевидно, что нет никакой разумной причины, чтобы не принять этот новый подход, и организация, которая отклонит его, может быть заподозрена в предвзятости при подборе информации, а также в стремлении осуществлять тайную слежку и манипулирования людьми.

В настоящее время, однако, такой подход на основе центра-

лизованных данных является очень выгодным для определенных ведомств движением по инерции. Использование автоматов для выдачи наличных денег и непосредственные электронные платежи с отслеживанием, которое они позволяют сделать, являются только верхушкой огромного айсберга капиталовложений, требующихся для создания этих крупномасштабных систем.

Мы быстро приближаемся к моменту решающего и возможно необратимого выбора не просто между двумя видами систем передачи данных, но и между двумя видами общества. Если продолжится нынешняя тенденция, то ужасающие возможности тотального надзора в указанных автоматизированных системах не только сделают призрачными большинство узаконенных гарантий конфиденциальности и других защит, но могут также серьезно охладить желание участвовать в общественной и политической жизни. Если же, с другой стороны, возобладает новый подход, то разрушение информационных прав личности может быть предотвращено, причем при этом добавятся новые права, а именно, право, позволяющее с помощью персональных компьютерных карточек раскрывать при групповых операциях только необходимую информацию. В эру всеохватывающей компьютеризации возможность управления информацией является ключом к социальному, экономическому и политическому могуществу. Компьютерные карточки восстановили бы нарушенное сейчас равновесие, как бы возвратив этот ключ, буквально и фигурально выражаясь, в руки простых граждан.

Для того, чтобы получить больше информации относительно тем, которые обсуждались в настоящем разделе, читайте [100, 102, 101, 104].

§ 5. Дополнительные применения

Криптографические методы имеют так много разнообразных применений, что мы даже не будем и пытаться описать здесь их все. Некоторые из этих применений остаются пока еще в основном академическими, другие же уже давно реализованы и с успехом используются на практике. Позволю себе лишь кратко упомянуть некоторые из наиболее существенных достижений и указать литературу для получения большей информации.

Рассеянная передача (*oblivious transfer*) позволяет *Алисе* послать сообщение *Бобу* таким образом, что оно будет правильно получено ровно в половине случаев. Более того, на эту вероятность не может повлиять ни *Алиса*, ни *Боб*, а *Алиса* при этом не будет знать, правильно ли получено ее сообщение. Понятие рассеянной передачи принадлежит Майклу Рабину [299]. См. также [357] и последующие работы [232, 129, 130, 233]. Относительно применений этого понятия обращайтесь к [50, 299]. Рекомендуем прочитать также [40, 128].

Любопытной разновидностью подбрасывания жребия являются развлекательные игры, в которых основную роль играет случайность. Особенный интерес представляет *покер без карт* (*mental poker*). Каким образом два или более человек могли бы сыграть в покер по телефону, если ни один из них не доверяет другому? Первая схема такой игры (вместе с доказательством ее невозможности при доступе к неограниченным компьютерным ресурсам в случае с двумя игроками) была предложена Эди Шамиром, Рональдом Ривестом и Леонардом Эдлеманом [321], однако Ричард Липтон и Дон Копперсмит продемонстрировали ее ненадежность, поскольку в ней могла произойти утечка частичной информации о картах, которая по предположению всегда должна была оставаться скрытой [251, 121]. Эта специфическая трудность была снята Шафи Гольдвассер и Сильвио Микэлив [197] применением вероятностного шифрования. Они представили также полное решение задачи, но из него не следовало никакого очевидного способа, как расширить их протокол до протокола для игры с более чем двумя игроками. Безусловно гарантированное решение в том случае, если имеются по крайней мере три игрока и никакие коалиции между ними не допустимы, было предложено Имре Барани и Золтаном Фюреди [14]. Кроме того, для этой задачи различными авторами были даны все более и более тонкие решения [369, 178, 126]. Последнее (теоретическое) решение, в конце концов, было представлено Клодом Крепеу в [127], которое даже позволяет игрокам сохранить суть электронного покера! По поводу обобщений покера на любую игру без карт обращайтесь к работе Одеда Голдрейча, Сильвио Микэли и Эви Вигдерсона [195].

Предположим, что вы являетесь владельцем некоторой ценной секретной информации, такой, например, как разложение

на множители вашего главного ключа. Если вы храните его взаперти только в своем офисе, то подвергаетесь риску утратить его при пожаре; если же вы храните несколько копий этого разложения в различных местах, то увеличивается риск того, что одна из них попадет еще в чьи-то руки. (N, k)-*пороговая схема* позволяет вам *распределить вашу тайну* по n ее *дубликатам* таким образом, что для вас будет достаточно иметь доступ к любым k из них, чтобы восстановить весь секрет целиком, тогда как никакие $k-1$ из этих n дубликатов не предоставят вам абсолютно никакой информации относительно этого секрета (в смысле теории информации Шеннона). Конструкция этой схемы была разработана независимо Эди Шамиром и Бобом Блэкли [315, 46]. О дальнейших исследованиях на эту тему см. в [243, 22, 92, и т.д.], а также [335].

Пусть *Алиса* и *Боб* знают различные секреты, допустим, разложение на множители их соответствующих RSA модулей. Предположим также, что они хотят ими обменяться. Как можно осуществить этот *обмен секретами* так, чтобы ни одна из сторон не подвергалась никакому риску выяснить в конце протокола, что ею только что было получено совсем не то, о чем она договаривалась с другой стороной, в обмен на свой подлинный секрет? Первоначально этот вопрос был изучен Мануэлем Блюмом и Майклом Рабином [50, 299, 52], а затем он привлек внимание многих других исследователей [58, 253, 342, 343, 211]. См. также [366].

Почта с удостоверением (о получении) является усложненной версией цифровой подписи. В этой задаче *Алиса* хочет послать сообщение *Бобу* таким способом, чтобы *Боб* получил это сообщение, если и только если *Алисе* будет предъявлено подписанное им подтверждение о его получении. Над этой проблемой, поставленной Уитфилдом Диффи, работали Мануэль Блюм и Майкл Рабин [50, 52, 58, 349]. Решающий ее протокол реализует *экстремальную* (*all-or-nothing*) удостоверяющую почту — понятие, также введенное Блюмом, которое означает, что получатель не может узнать *абсолютно* ничего о содержании сообщения до тех пор, пока отправитель не получит подтверждения о его получении, и наоборот [53].

Подписание контрактов еще сложнее (хотя и является достаточно легким для реализации, если использовать в качестве

примитива протокол конкретной почты с удостоверением). Оно позволяет *Алисе* и *Бобу* подписывать контракты по компьютерной сети таким образом, что ни один из них не сможет прервать протокол подписывания (за исключением, которое возможно разве что с очень малой вероятностью) и поставить подписи другой стороны под контрактом до тех пор, пока она не будет с ним согласна [52, 300, 58, 168].

Предположим, что вы располагаете некоторым количеством секретов и хотите продать один из них. *Раскрывающий протокол* позволяет вам сделать это таким способом, что его покупателю будет предоставлена возможность выбора того секрета, который он получит, но не предоставится никакой информации о том, каков именно этот секрет. Такой протокол является *экстремальным* (*all-or-nothing*), если у покупателя нет никакой возможности заполучить отдельные биты или целые части нескольких секретов. Эта проблема была сформулирована и решена Жилем Брассаром, Клодом Крепеу и Жан-Марком Робертом [83, 84], а также, независимо, Дэвидом Чаумом. Ее решение полезно, в частности, при реализации протокола для многостороннего покера без карт, такого как протокол Крепеу [127], и может служить примитивом для более сложных криптографических протоколов. См. также [131].

Предположим, что *Алиса* и *Боб* хотят разделить друг с другом истинно случайную двоичную последовательность, но опасаются, что некоторые из ее разрядов могут попасть в руки нарушителя. Тогда, задавая только верхнюю границу на количество информации, которую допустимо скомпрометировать, есть возможность *повысить секретность посредством открытых обсуждений*, с тем чтобы остановиться на более короткой и намного более секретной двоичной последовательности. Это остается возможным, даже если из-за ошибок при передаче и после злонамеренной подделки исходная последовательность вашего друга оказалась бы слегка отличной от вашей. Данный вопрос был исследован независимо Чарльзом Беннеттом, Жилем Брассаром и Жан-Марком Робертом [36], а также Бенни Чором, Одедом Голдрейчем, Йоханом Хастадом, Джоэлем Фридманом, Стевеном Рудичем и Романом Смоленским [118]. Решение этого вопроса, как описано в следующей главе, является основной составной частью для установления квантового канала. См. также [32, 311].

Полислучайным семейством функций называется набор *эффективно* вычисляемых функций, которые являются полиномиально неотличимыми от истинно случайных функций. Это понятие было введено Одедом Голдрейчем, Шафи Гольдвассер и Сильвио Микэли [193], которые предложили также его многочисленные применения [192]. Одним из таких применений являются вычислительно надежные аутентификационные метки, для которых требуются короткие секретные разделенные ключи. Майком Люби и Чарльзом Раковом в [254] были построены полислучайные семейства *перестановок*. Довольно странно, что эти перестановки были получены с помощью некоего процесса, похожего на DES.

Сублиминальный канал — это канал, который допускает два уровня шифрования. *Секретное сообщение* может быть восстановлено от криптограммы, используя «регулярный» секретный ключ, но чтобы получить *сублиминальное сообщение*, необходим некий дополнительный ключ. Это понятие, которое было введено Густавом Симмонсом, позволяет двум сторонам осуществлять конфиденциальную связь, в то время как ничего не подозревающий нарушитель остается в полном убеждении, что он может читать сообщения обеих сторон [328, 331]. См. также [148].

Несмотря на то, что до сих пор никакой практически значимой криптографической схемы для *защиты программного обеспечения* пока еще не предложено, интересные теоретические идеи в этом отношении были представлены Амиром Херцбергом и Шломитом Пинтером [216] и Одедом Голдрейчем [191]. В частности, Голдрейч делает различие между защитой от незаконного копирования и защитой от незаконного воспроизведения дистрибутива. См. также [283].

Проблема о миллионере, которая впервые была поставлена Эндрю Яо, формулируется следующим образом: два миллионера хотят выяснить, кто из них самый богатый, но так, чтобы ни один из них не смог узнать, каково богатство другого [365]. Это — специальный случай *секретного распределенного вычисления*, с помощью которого *Алиса* и *Боб* хотят вычислить значение некоторой согласованной функции $f(a, b)$ с секретным, известным *Алисе*, входом a и входом b, также секретным, но известным *Бобу*, таким способом, что ни одна из сторон не может получить никакой информации, чтобы узнать значение секретного входа

другой (кроме той, что может быть извлечена из значения $f(a,b)$ и собственного секрета одной из сторон). Яо в качестве применения такого вычисления показал, что оно позволяет двум сторонам сотрудничать при выборе некоторого целого числа, которое, как им известно, является произведением в точности двух простых чисел так, что ни одна из сторон не сможет вычислить эти сомножители в том случае, когда они перестанут сотрудничать друг с другом [366]. См. также [232, 129, 130, 233].

Секретное распределенное вычисление естественным образом обобщается на *произвольное число сторон*. Криптографические решения обобщенной проблемы секретных распределенных вычислений были даны Одедом Голдрейчем, Сильвио Микэли и Эви Вигдерсоном [195], и независимо Дэвидом Чаумом, Йваном Дамгардом и Йереоном ван де Граафом [112]. Зви Галил, Стюарт Хабер и Моти Юнг изучали секретные устойчивые к ошибкам протоколы в модели системы с открытым ключом [182]. Позднее и независимо Дэвидом Чаумом, Клодом Крепеу и Йваном Дамгардом в [111], а также Майклом Бен-Ором, Шафи Гольдвассер и Эви Вигдерсоном в [38] было показано, что безусловно секретные распределенные вычисления могут быть выполнены при одном-единственном предположении, что нечестными являются не более, чем одна треть от общего числа участников. См. также [301, 106].

Схема выборов является специальным случаем предыдущей проблемы. Как могут несколько сторон участвовать в голосовании по компьютерной сети таким образом, что будут в состоянии вычислить результат голосования, но при этом все индивидуальные бюллетени сохранятся в тайне? Этот вопрос изучался многими исследователями, включая Джоша Беналоха [Коэна] и Майкла Фишера [119], Майкла Бен-Ора и Натана Линиэла [39], а также и Беналоха и Моти Юнга [23].

Еще одна интересная задача — это *вычисления с зашифрованными данными*, которая основана на идее, сначала исследованной Джоэном Фейгенбаумом [174], и впоследствии усовершенствованной им совместно с соавторами [1]. В этой задаче *Алиса* хочет узнать значение $f(x)$ некоторой открытой функции f от секретного аргумента x, но испытывает при этом недостаток ресурсов (или знаний), чтобы ее вычислить. У *Боба* же такие способности есть, и он хочет помочь *Алисе*. Функция f называется

шифруемой, если *Алиса* способна использовать *Боба* не только так, чтобы тот помог ей вычислить $f(x)$, но и так, чтобы при этом *Боб* не смог узнать никакой информации об x, за исключением значения $f(x)$. Жиль Брассар и Пауль Брэтли указали на прямую связь между этим понятием и техникой проектирования алгоритмов, которая известна под названием *стохастическая предпосылка* [77]. См. также [17, 255, 320, 73] и [74].

Множество других замечательных сведений о криптологии вы можете узнать также, если прочитаете [2, 48, 327, 252], или обратитесь к трудам конференций CRYPTO, EUROCRYPT, FOCS и STOC или попробуете заглянуть в журналы *Journal of Cryptology* и *Cryptologia*. Кроме того, другие научные журналы регулярно публикуют специальные выпуски, посвященные криптологии. Смотрите, например, майский выпуск ТИИЭР 1988 года.

Глава 7

Квантовая криптография

(Написана совместно с Чарльзом Х. Беннеттом)

В предыдущих главах мы обсуждали глубокую взаимосвязь криптологии с такими базисными понятиями современной науки, как информация, случайность, сложность, алгоритм, подчеркивая при этом, что ее эффективность коренным образом зависит от результатов и выводов соответствующих теорий. В настоящей главе мы покажем, что криптография, как это ни покажется на первый взгляд странным, тесно связана еще с одной фундаментальной теорией, позволяющей познавать природу в малых масштабах, на ее дискретном уровне, а именно — с квантовой механикой [59], и ее основным постулатом — принципом неопределенности Гейзенберга. Такая многогранность и многосложность криптологии, по-видимому, объясняется основополагающим существом предмета, который она изучает.

§ 1. Введение

Когда для пересылки цифровой информации применяются такие элементарные квантовые системы, как поляризованные фотоны, то использование принципа неопределенности Гейзенберга позволяет достичь совершенно нового криптографического феномена, который абсолютно недостижим для обычных средств передачи информации. Например, можно получить такой канал связи, в котором в принципе невозможно какое бы то ни было прослушивание без наличия таких нарушений при передаче, которые не были бы обнаружены с очень большой вероятностью. Посред-

ством такого канала достигается одно из основных преимуществ криптографии с открытым ключом: он позволяет осуществить безопасное распределение ключей между *Алисой* и *Бобом*, которые первоначально не обладали никакой совместно используемой секретной информацией, при условии, что они имеют доступ, кроме квантового, еще и к обычному каналу, который допускает не только пассивное прослушивание, но и активную фальсификацию. (Более того, распределение секретных ключей можно выполнить даже при наличии активного фальсификатора, если *Алиса* и *Боб* при этом с самого начала используют некоторую совместную секретную информацию, но при условии, что фальсификация не настолько активна, чтобы полностью подавить всю связь.) Возникающий в результате соответствующих действий сторон *квантовый канал* является вероятностно секретным даже для противника с более передовой компьютерной технологией и имеющего неограниченные вычислительные ресурсы (и даже если, все-таки, $\mathcal{P} = \mathcal{NP}$!), при единственном условии, что в этом процессе не нарушаются коренным образом общепризнанные физические законы.

В традиционной теории информации и криптографии считается не требующим доказательств то, что цифровая связь может всегда пассивно просматриваться или копироваться даже лицом, которое не осведомлено об информации, которая передается. Это может быть полезно, например, тому, кто надеется быть способным вычислить (или разузнать) секретный ключ в дальнейшем по прошествии какого-то времени, возможно, после того, как будет накоплен достаточный объем шифртекста. В противоположность этому, когда информация закодирована в неортогональных квантовых состояниях, например, в одиночных фотонах с направлениями поляризации 0°, 45°, 90° и 135°, то получается такой канал связи, что почта в нем даже в принципе не может с достоверностью ни читаться, ни копироваться нарушителем, не знающим некую информации о ключе, которая используется при формировании пересылаемого сообщения. Нарушитель не может извлечь никакой частичной информации об этой пересылке, позволяющей отличить ее от случайной, и таким способом, который не поддавался бы контролю и который не смогли бы обнаружить законные пользователи канала.

Недостаток подобного подхода на практике заключается в

том, что квантовая передача информации является, безусловно, очень слабой и не может усиливаться при прохождении. Более того, квантовая криптография вообще не в состоянии обеспечить цифровую подпись (см. § 5.2) и связанные с ней характерные возможности, наподобие почты с удостоверением (см. § 6.5) или способности улаживать споры до суда. (Однако можно доказать, что эти ограничения присущи любой схеме, сохраняющей свою секретность при атаке противника, обладающего неограниченными вычислительными ресурсами.) Тем не менее, Эрнест Брикелл и Эндрю Одлыжко заканчивают свой очень обстоятельный обзор новейших достижений в криптоанализе такими словами: «Если подобные системы [квантовой криптографии] станут практически реализуемыми, то все рассмотренные нами [в этой статье] криптографические методы окажутся попросту бесполезными.» [93].

Впервые квантовое шифрование было предложено Стефеном Уиснером [357] наряду с двумя его применениями: созданием денег, которые в принципе невозможно подделать, и мультиплексной передачей двух или трех сообщений таким образом, что при чтении одного из них остальные разрушаются, что очень похоже на протокол экстремального раскрытия по принципу «все-или-ничего» (all-or-nothing), который был предложен Майклом Рабином [299] и упомянут в § 6.5 (см. также [83, 84]).

Более чем десятилетие спустя Чарльз Беннетт, Жиль Брассар и Сет Брейдбард совместно со Стефеном Уиснером [31] показали, как использовать это квантовое шифрование вместе с методами из криптографии с открытым ключом для построения нескольких схем неподделываемых жетонов в метро. Позже Беннетт и Брассар [27] изобрели описанный выше квантовый канал, а также *квантовое подбрасывание жребия* (для обсуждения обычного подбрасывания жребия см. § 6.1). Естественно, что квантовый протокол подбрасывания жребия порождает и понятие *квантовых блобов*, которые принимаются в качестве квантовых битовых обязательств [25, 34] (сравните с обычными битовыми обязательствами, рассмотренными в § 6.2). См. также [81, 33, 232, 129, 130, 233].

В настоящей главе описывается использование основного квантового канала только для открытого распределения ключей. Подробное обсуждение вопросов квантовой криптографии пред-

ставлено в научных работах [25, 34, 371], а также в популярных статьях [204, 354, 154, 286, 163, 340]. См. также [30, 26, 28, 29].

§ 2. Основные свойства поляризованных фотонов

Поляризованный свет можно получить, пропуская обычный световой луч через какое-нибудь поляризующее устройство, вроде поляроидного фильтра или кристалла кальцита. Ось поляризации луча определяется ориентацией поляризующего устройства, через которое проходит луч. Вообще говоря, можно порождать и одиночные поляризованные фотоны, выделяя их из поляризованного светового луча, хотя чисто технологически это может быть неосуществимо. В следующем параграфе мы принимаем для простоты, что такие одиночные фотоны с определенными направлениями поляризации уже имеются, но затем в § 4 показываем, как можно избавиться от этого предположения.

Несмотря на то, что направление поляризации является величиной непрерывной, принцип неопределенности Гейзенберга не допускает такого измерения состояния любого *одиночного* фотона, которое раскрывало бы более одного бита информации (в вероятностном смысле) об угле его поляризации. Например, если луч света с осью поляризации, направленной под углом α, попадает в фильтр, ориентированный под углом β, то все отдельно взятые фотоны ведут себя дихотомическим и совершенно непредсказуемым образом, проходя через такой фильтр с вероятностью $\cos^2(\alpha-\beta)$ и поглощаясь, соответственно, с вероятностью $\sin^2(\alpha-\beta)$. Детерминировано все фотоны ведут себя только тогда, когда обе направляющих либо параллельны друг другу (тогда все фотоны проходят через фильтр), либо перпендикулярны (в этом случае все фотоны поглощаются). (Любая другая элементарная квантовая система с двумя состояниями, наподобие атома со спином 1/2, ведет себя точно таким же дихотомическим вероятностным образом.)

Если оси не перпендикулярны друг другу, то некоторые фотоны должны проходить через фильтр, и это позволяет надеяться на то, что можно было бы выяснить дополнительную информацию об α, проведя для них после прохождения повторные измерения при помощи какого-нибудь поляризатора, который будет

ориентирован под неким третьим углом. Однако такое измерение оказывается совершенно бесполезным, потому что все прошедшие через β-поляризатор фотоны, оказываются поляризованными в точности под углом β, потеряв при этом какую бы то ни было информацию о своей предыдущей поляризации под углом α. Конечно, если известно, что луч состоит из нескольких одинаково поляризованных фотонов, то для того, чтобы получить более одного бита информации относительно их общего угла поляризации, можно для разных фотонов сделать различные измерения.

Другими словами, можно было бы надеяться узнать более одного бита информации об одиночном фотоне, не измеряя напрямую угол его поляризации, а скорее так или иначе расширить один фотон до ансамбля из одинаково поляризованных фотонов, чтобы впоследствии выполнить над ними различные измерения. Однако эта надежда также оказывается тщетной, потому что существование такого ансамбля, как это можно показать, не согласуется с основными положениями квантовой механики [361].

Формально в квантовой механике внутреннее состояние квантовой системы (такое, как поляризация фотона) представляется в виде вектора единичной длины в линейном пространстве над полем комплексных чисел, то есть в так называемом гильбертовом пространстве. Размерность этого гильбертова пространства зависит от самой системы и может быть довольно большой (или даже бесконечной) для более сложных систем. Каждое физическое измерение, которое может выполняться в системе, соответствует некоторому разложению гильбертова пространства на ортогональные подпространства, каждое из которых отвечает одному из возможных результатов этого измерения. Таким образом, число возможных результатов измерения ограничено размерностью d рассматриваемого гильбертова пространства. Наиболее полными измерениями являются такие, которые соответствуют разложению гильбертова пространства на d одномерных подпространств.

Гильбертово пространство для одиночного поляризованного фотона является 2-мерным. Таким образом, состояние фотона может быть полностью описано в виде линейной комбинации, к примеру, двух единичных векторов $r_1 = (1, 0)$ и $r_2 = (0, 1)$, представляющих соответственно горизонтальную и вертикальную по-

ляризации. В частности фотон, поляризованный под углом α к горизонтали, описывается вектором состояния $(\cos\alpha, \sin\alpha)$. В том случае, когда такой фотон подвергается измерению на предмет горизонтальности или вертикальности своей поляризации, то в действительности он как бы выбирает, стать ли ему горизонтально поляризованным с вероятностью $\cos^2\alpha$ и вертикально поляризованным с вероятностью $\sin^2\alpha$. Два ортогональных вектора r_1 и r_2, таким образом, служат примером разложения 2-мерного гильбертова пространства на 2 ортогональных одномерных подпространства. С этого момента мы будем говорить, что r_1 и r_2 составляют *прямоугольный* базис рассматриваемого гильбертова пространства.

Другой возможный базис того же гильбертова пространства задается двумя диагональными векторами $d_1 = \frac{\sqrt{2}}{2}(1,1)$ и $d_2 = \frac{\sqrt{2}}{2}(1,-1)$. В этом *диагональном* базисе d_1 представляет фотон, поляризованный под углом 45°, а d_2 — фотон с поляризацией под углом 135°. Два базиса (например, прямоугольный и диагональный) называются *сопряженными*, если каждый вектор одного базиса имеет проекции одинаковой длины на все векторы другого базиса. Попросту говоря, это означает, что система, подготовленная в данном состоянии в одном из таких базисов, будет вести себя совершенно непредсказуемо и потеряет всю информацию о себе, отражающуюся в этом базисе, после того, как подвергнется измерению, которое соответствует другому базису.

Благодаря сложной природе своих коэффициентов, двумерное гильбертово пространство допускает также третий базис, сопряженный и с прямоугольным и с диагональным базисами и состоящий из двух векторов так называемой *круговой* поляризации $c_1 = \frac{\sqrt{2}}{2}(1,i)$ и $c_2 = \frac{\sqrt{2}}{2}(i,1)$, где $i := \sqrt{-1}$. Тем не менее для обоснования рассматриваемого в следующем параграфе квантового распределения открытых ключей нам понадобятся только прямоугольный и диагональный базисы.

С практической же точки зрения достаточно понимать только, что имеются два простых прибора. Один из этих приборов может различать горизонтально поляризованные фотоны от вертикально поляризованных, а другой может различать фотоны с разной диагональной поляризацией. Однако если первый прибор используется для определения состояния диагонально поляризованного фотона (а второй — для прямоугольно поляризо-

ванного), то в такой ситуации фотон поведет себя совершенно случайным и непредсказуемом образом, и подобное измерение вообще не позволит определить угол его поляризации. На практике эти устройства очень несовершенны, но их можно построить так, что они почти никогда не будут приводить к неправильному ответу, хотя иногда могут и не дать ответ вообще. Так, например, прибор, который был разработан для того, чтобы различать фотоны с двумя типами прямоугольной поляризации, мог бы отвечать: «фотон был вертикально поляризован», «фотон был горизонтально поляризован», или «я не могу сообщать, как был поляризован фотон». Если входящий фотон был действительно поляризован прямоугольно, то любой из первых двух ответов будет почти наверняка правильным, тогда как третий не предоставляет абсолютно никакой информации.

§ 3. Квантовое распределение открытых ключей

Цель квантового распределения открытых ключей заключается в том, чтобы, используя квантовый канал, обеспечить передачу последовательности случайных битов между двумя пользователями, которые до этого не имели никакой совместно используемой и секретной для других информации. Достигается это таким способом, что пользователи, следящие за их диалогом по обычному, не квантовому, каналу связи (допускающему прослушивание без каких бы то ни было ограничений) могут с большой вероятностью определить, была ли нарушена такая первоначальная квантовая передача информации во время ее перехвата в канале. (Подобное достоинство, которое присуще исключительно квантовому каналу, фактически вынуждает сводить любой тип прослушивания к активной фальсификации.)

Если квантовая передача не нарушалась, то пользователи могут с уверенностью применять эту совместную секретную последовательность в качестве исходного секретного ключа в любой традиционной криптосистеме (наподобие одноразового шифра — см. § 3.2) для того, чтобы скрыть содержание всей последующей связи. В качестве альтернативы она может использоваться также для других криптографических целей, требующих совместной секретной случайной информации, например, для аутентификационных меток Вегмана-Картера, которые основаны

на универсальном хешировании [355] (см., также, § 5.1). С другой стороны, если обнаружится, что передача была нарушена, то пользователи могут не принимать во внимание только что полученную двоичную последовательность и должны попытаться произвести квантовую передачу еще раз. Поэтому для обеспечения намеченной цели они вынуждены будут задерживать некую важную связь до тех пор, пока не осуществят успешную передачу достаточного количества случайных битов через квантовый канал. Поэтому, несмотря на то, что нарушитель, перекрывая квантовый канал передачи информации, может препятствовать связи между пользователями, он окажется неспособным ввести их в заблуждение до такой степени, чтобы они были уверены, что передача прошла успешно, когда на самом деле это было не так.

Давайте теперь рассмотрим более подробно, каким образом *Алиса* и *Боб* могут осуществить открытое распределение ключей с использованием квантового канала. Мы предполагаем наличие нарушителя, которого, точнее которую назовем *Евой*. В качестве первого шага *Алиса* выбирает произвольную битовую строку и произвольную последовательность поляризационных базисов (прямоугольных и диагональных). Затем она посылает *Бобу* ряд фотонов, каждый из которых является носителем одного бита информации этой выбранной двоичной строки, причем значение каждого бита определяется поляризацией соответствующего ему фотона в базисе, номер позиции которого в выбранной последовательности (поляризационных базисов) совпадает с порядковым номером бита в строке. Так, например, горизонтально или под углом в 45° поляризованные фотоны могут использоваться в качестве носителя двоичного 0, в то время как вертикально или под углом в 135° поляризованные фотоны будут определять двоичную 1.

С одной стороны, хорошо, что, если *Ева* захочет измерить поляризацию (некоторых из) тех фотонов, которые *Алиса* посылает *Бобу*, то она не будет знать, в каких базисах это необходимо делать. С другой стороны, плохо то, что *Боб* также не знает, какие базисы нужно использовать. Несмотря на это, *Боб*, как только получает фотоны, решает абсолютно случайным для каждого фотона образом и совершенно независимо от *Алисы* измерить, какую поляризацию имеют все фотоны, прямоугольную или диа-

гональную. После этого он интерпретирует то, что получилось, либо как двоичный 0, либо как двоичную 1, в зависимости от исхода соответствующего измерения. Как уже объяснялось в § 2, получающийся в результате ответ должен быть случайным, и при попытке измерить, например, прямоугольна ли поляризация диагонально поляризованного фотона (или наоборот), вся заключенная в нем информация будет потеряна. Таким образом, в итоге *Боб* получает содержащие информацию данные только от приблизительно половины фотонов, которые он проверяет (от тех, для которых он угадал базис поляризации), хотя сам пока даже не знает, от каких. Кроме того, на практике количество этой информации, полученной *Бобом* может быть еще меньше в связи с тем, что некоторые из фотонов могут быть потеряны при передаче или невосприняты обнаруживающими фотоны детекторами *Боба* из-за их несовершенства.

Последующие шаги протокола выполняются в обычном открытом канале связи. Сначала предположим, что этот канал восприимчив только к прослушиванию, но не к введению новых, или изменению порядка поступающих, сообщений. Сначала *Алиса* и *Боб* определяют посредством открытого обмена сообщениями, какие из фотонов были получены в действительности и какие из них измерялись *Бобом* в том базисе, в котором их и нужно было измерять. Если квантовая передача не была искажена, то *Алиса* и *Боб* смогут в результате договориться о битах, закодированных этими фотонами, даже несмотря на то, что эти данные никогда не обсуждались ими по открытому каналу связи. Другими словами, каждый из этих фотонов в соответствии со сделанными предположениями переносит один бит случайной информации (типа того, вертикально или горизонтально поляризован прямоугольно поляризованный фотон), которая становится известной только *Алисе* и *Бобу* и больше никому.

Из-за того, что прямоугольно и диагонально поляризованные фотоны чередуются в квантовой передаче случайным образом, любой нарушитель рискует при перехвате изменить передачу таким способом, что это породит расхождение между *Алисой* и *Бобом* в некоторых из тех битов, о правильности которых, как они думают, между ними должно быть достигнуто согласие. Характерно, и это можно доказать [25], что при пересылке любого фотона никакое измерение его состояния нарушителем, которо-

му исходный базис этого фотона становится известен только после того, как он выполнит такое измерение, не может предоставить более 1/2 ожидаемых битов информации о ключе, значение соответствующего разряда которого кодируется данным фотоном. Кроме того, любые такие измерения, дающее b-тую часть битов ожидаемой информации ($b \leqslant 1/2$), должны приводить к расхождению с вероятностью по крайней мере $b/2$, если уже измеренный фотон (или любая предпринятая его подделка) будет впоследствии переизмеряться (*Бобом*) в исходном базисе. (Этот оптимальный компромисс для *Евы* возникает, например, тогда, когда она перехватывает, измеряет и сама передает дальше все фотоны в прямоугольном базисе. В этом случае она узнает правильные поляризации у половины фотонов и породит расхождения в 1/4 тех фотонов, которые будут позже повторно измеряться *Бобом* в исходном базисе.)

Теперь остается только выяснить, как *Алиса* и *Боб* смогут определить, являются ли их получившиеся в результате битовые строки идентичными (показывая с высокой вероятностью, что в квантовом канале никакого нарушения не произошло, или, также, что это нарушение было на очень малом числе фотонов) или они различны (показывая, таким образом, что квантовый канал был подвергнут серьезному прослушиванию). Простое решение заключается в том, чтобы *Алиса* и *Боб* открыто сравнили некоторые из битов, относительно которых, как они думают, они должны прийти к соглашению. Позиции таких «особо проверяемых» битов должны быть выбраны случайно уже после того, как квантовая передача будет завершена, чтобы лишить *Еву* информации о том, какие фотоны она может измерять без опаски. Конечно, подобный процесс приносит в жертву секретность этих битов. Если совокупность позиций битов, используемых при этом сравнении, является произвольным подмножеством (скажем, одной трети) всех правильно полученных битов, то перехват, допустим, более десятка фотонов, позволяющий избежать обнаружения, маловероятен. Если все сравнения подтверждаются, то *Алиса* и *Боб* могут заключить, что квантовая передача прошла без существенного перехвата. Следовательно, большинство оставшихся битов, которые были посланы и получены в одном и том же базисе, могут спокойно использоваться в качестве одноразового ключа для последующей связи по открытому кана-

лу. Когда этот одноразовый ключ будет полностью использован, протокол передачи новой порции случайным образом сгенерированной информации по квантовому каналу повторяется. Иллюстрация протокола, который мы только описали, приведена на рис. 7.1.

Такой протокол обнаружения перехвата фотонов довольно расточителен, поскольку для того, чтобы он был выявлен с большой вероятностью, должна быть пожертвована значительная

1	0	1	1	0	1	1	0	0	1	0	1	1	0	0	1
2	×	+	×	+	+	+	+	+	×	×	+	×	×	×	+
3															
4	+	×	×	+	+	×	×	+	×	+	×	×	×	×	+
5	1		1		1	0	0	0		1	1	1		0	1
6	+		×		+	×	×	+		+	×	×		×	+
7			✓		✓			✓				✓		✓	✓
8			1		1			0				1		0	1
9					1									0	
10					✓									✓	
11			1					0				1			1

Передача по квантовому каналу

1. Случайная битовая строка *Алисы*.
2. Базисы передачи, выбранные *Алисой* случайным образом.
3. Посланные *Алисой* фотоны.
4. Случайные базисы, выбранные *Бобом* при приеме.
5. Битовая строка, полученная *Бобом*.

Обсуждение по открытому каналу

6. *Боб* сообщает базисы измерений полученных фотонов.
7. *Алиса* отмечает, какие базисы были угаданы правильно.
8. Информация, которую можно использовать совместно (если нарушения не было).
9. *Боб* указывает некоторые выбранные наугад биты ключа.
10. *Алиса* подтверждает эти биты.

Результат

11. Оставшиеся совместно секретные биты.

Рис. 7.1. Квантовое открытое распределение ключей

часть битов, даже если этот перехват предпринят *Евой* только для нескольких фотонов. К тому же вероятность того, что с возникающими в результате строками *Алиса* и *Боб* согласятся полностью, не может быть сделана сколь угодно близкой к 1, если не пожертвовать при этом большим числом первоначально переданных битов. Обе такие трудности могут быть решены в соответствии с более тонким протоколом проверки, который принадлежит Чарльзу Беннетту и Жилю Брассару, а также Жан-Марку Роберту [36, 25]. Этот протокол основан на универсальном хешировании Картера и Вегмана [98, 76], которое упоминается в § 5.1. Однако здесь мы его описывать не будем. Отметим только, что в [36] приводится также эффективный для *Алисы* и *Боба* способ *повышения секретности* (см. § 6.5) посредством сокращения количества информации, известной *Еве*, об их совместной строке.

Как уже было сказано ранее, требование о том, что открытый (не квантовый) канал в схеме квантового распределения не должен подвергаться активной фальсификации, может быть ослаблено, если *Алисой* и *Бобом* заранее был сформирован небольшой секретный ключ, который они будут использовать, чтобы создавать для своих сообщений в классическом канале аутентификационные метки Вегмана-Картера [355]. Точнее говоря, аутентификационная схема многократных сообщений Вегмана-Картера может быть использована при выработке из короткого случайного ключа для любого сколь угодно длинного сообщения некоторой зависящей от него метки. Это делается таким способом, что *Ева*, которая неосведомлена об этом коротком ключе, способна, разве только с пренебрежимо малой вероятностью, сгенерировать любую другую допустимую пару, состоящую из сообщения и соответствующей ему метки. Причем это остается справедливым, даже если у *Евы* имеются неограниченные вычислительные ресурсы. Таким образом, подобная метка обеспечивает доказательство того, что сообщение является подлинным, и то, что оно не было сгенерировано или изменено *Евой*. Для того чтобы такую доказательно безусловную защиту системы было невозможно скомпрометировать, биты секретных ключей в схеме Вегмана-Картера должны расходоваться постепенно, и их нельзя использовать по нескольку раз. Несмотря на то, что в представленном применении эти биты могут заменяться новыми случайными битами, которые были успешно переданы по кванто-

вому каналу, *Ева* может, тем не менее, предотвратить связь между *Алисой* и *Бобом*, подавляя передачу сообщений в открытом канале, поскольку она всегда может либо перехватывать, либо чрезмерно возмущать фотоны, которые посылались по квантовому каналу. Однако в этом случае *Алиса* и *Боб* могли бы еще раз с высокой вероятностью определить, что их связь подавляется, и не будут введены в заблуждение, полагая, что это не так.

§ 4. Практическая применимость

С одной стороны, квантовая схема распределения открытых ключей, которая была описана в предыдущем параграфе, теоретически очень хороша, но с другой — совершенно непрактична с точки зрения нынешней технологии. В частности, в ней были проигнорированы среди прочих следующие две проблемы:

1) Намного проще иметь дело с пучками (или импульсами) фотонов, чем с одиночными фотонами.

2) Даже если никакого нарушения в квантовом канале не произошло, и даже тогда, когда *Боб* правильно угадывает выбранный *Алисой* базис, нужно ожидать, что некоторые из фотонов по пути будут переполяризованы, или что в результате ошибок в измеряющем приборе Боб может неправильно их интерпретировать.

Тем не менее, на практике обе эти трудности могут быть успешно решены. Детальное описание предлагающего такие решения криптографического квантового устройства и соответствующих протоколов приведено в техническом отчете, который был написан Беннеттом и Брассаром [28]. См. также [25]. Ниже мы упомянем только самые основные его идеи.

Всякий раз когда в идеальном протоколе § 3 должны посылаться одиночные фотоны, первая проблема решается с помощью посылки очень слабых световых импульсов. Если использовать лазер, то легко выработать такой импульс, в котором число фотонов удовлетворяет распределению Пуассона при известных математическом ожидании и дисперсии. Кроме того, использование поляризаторов, удвоителей импульсов и фильтров с нейтральной плотностью позволяет производить световые импульсы заранее

определенной поляризации, а в промежуток времени от одного такого импульса до другого (поскольку в конструкции этих приборов нет никаких движущихся частей) можно очень быстро переключаться на нужную поляризацию.

Проблема с световыми импульсами (которые заменяют одиночные фотоны) заключается в том, что каждый раз, когда за время одного и того же импульса выпускается более одного фотона, *Ева* получает возможность (по крайней мере в принципе) измерить один из этих фотонов в случайно выбранном базисе. Такое нарушение может быть необнаруживаемо при условии, что *Ева* будет очень осторожной и позволит другим фотонам того же самого импульса дойти до *Боба* неискаженными. Основная идея значительного уменьшения этой угрозы заключается в том, чтобы использовать действительно очень слабые импульсы. Если, например, ожидаемое число фотонов в каждом импульсе равно одной тысячной (10^{-3}), то можно считать, что мульти-фотонный импульс будет вырабатываться приблизительно только один раз на каждые два миллиона импульсов. Следовательно, 99,95% непустых импульсов должны содержать одиночный фотон. Только обладая очень большими ресурсами, *Ева* могла бы выявить те редкие случаи, когда передается более одного фотона и проверить соответствующий бит с вероятностью 1/2, измеряя один из фотонов в случайно выбранном базисе, но это привело бы лишь к тому, что *Ева* смогла бы узнать примерно 0,025% той битовой строки, которой *Алиса* обменивается с *Бобом*. (Обладая еще большими ресурсами, *Ева* могла бы хранить этот дополнительный фотон, не измеряя его, до тех пор, пока открытое обсуждение между *Алисой* и *Бобом* не покажет, в каком базисе этот фотон должен читаться, увеличивая, таким образом, процент своего успеха до 0,05%.) Для дальнейшего сокращения любой доли однозначной информации об их окончательной совместной битовой строке, которая может быть известна *Еве*, *Алиса* и *Боб* могут впоследствии использовать упомянутые ранее методы повышения секретности [36].

Вторая проблема, которая связана с наивной реализацией квантовой криптологии и описана в § 3, состоит в том, что некоторые биты могут быть получены неправильно даже при отсутствии перехвата, из-за недостатков в аппаратуре. *Алиса* и *Боб* никогда бы не смогли обменяться случайной битовой стро-

кой, если бы не начинали делать это заново каждый раз, когда обнаруживают даже одиночную ошибку в квантовой передаче. Следовательно, для того чтобы обнаружить и исправить приемлемое число различий между исходной случайной строкой *Алисы* и строкой, полученной *Бобом*, необходим протокол в открытом канале. Подходящий для *Алисы* способ исправления таких ошибок заключается в том, чтобы генерировать проверяющие последовательности, используя при этом согласованный код, исправляющий ошибки, наподобие несистематического сверточного кода [184]. Такая последовательность может затем быть послана *Бобом* по открытому каналу. Если код имеет достаточную избыточность, то *Боб* может однозначно декодировать доступную ему информацию, с высокой вероятностью восстановив, таким образом, исходную строку *Алисы*. Из этого следует, что при использовании протоколов, повышающих секретность [36], информация, которая попадает к *Еве*, может быть доведена, в конце концов, почти до нуля. Далее см. [309, 25, 311].

Когда никакого перехвата не происходит, то размер требуемой избыточности зависит от ожидаемой вероятности неправильно полученных битов. До тех пор, пока *Алиса* и *Боб* находятся в двусторонней связи, они могут достичь правильной избыточности, даже если не знают заранее этой вероятности: *Алиса* вычисляет достаточно много контрольных разрядов, но посылает *Бобу* столько, сколько тот считает необходимым для эффективного декодирования. Когда *Боб* будет удовлетворен тем, как исправлены все ошибки, и убедится в том, что с большой вероятностью строка *Алисы* ему (почти) доподлинно известна, он сообщит ей действительный процент ошибок, полученных с помощью своих измерений. Из этой доли ошибок, начальной интенсивности импульса и квантовой эффективности детекторов *Боба* можно будет оценивать степень перехвата. Если окажется, что перехват, по всей видимости, произошел, но *Ева*, похоже, получила только небольшое количество информации, то следует решить, можно ли продолжить безопасное распределение ключей, чтобы потом, используя протоколы повышения секретности, лишить *Еву* почти всей информации, которую она получила. С другой стороны, если будет обнаружен существенный перехват, то *Алиса* и *Боб* не должны больше исключать возможности того, что *Ева* может обладать почти полной информацией. В этом случае они

должны, не принимая во внимание полученные ранее результаты, начать всё сначала, используя при этом новую квантовую передачу.

Практическая реализация протокола должна принимать во внимание и некоторые дополнительные ограничения. Объяснение более технического характера, как можно достичь всего, что обсуждалось в этой главе, приведено подробно в [28].

Далее по поводу квантовой криптографии см. [36, 165, 81, 33], а относительно квантовых вычислений — [152, 153, 155, 225, 42].

Литература

[1] ABADI, M., FEIGENBAUM, J., KILIAN, J., "On hiding information from an oracle", *Journal of Computer and System Sciences*, vol. 39, 1989, pp. 21 – 50.

[2] ADLEMAN, L. M., "Implementing an electronic notary public", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 259 – 265.

[3] ADLEMAN, L. M., "On breaking generalized knapsack public key cryptosystems", *Proceedings of 15th ACM Symposium on Theory of Computing*, April 1983, pp. 402 – 412.

[4] AGNEW, G. B., BETH, TH., MULLIN, R. C., VANSTONE, S. A., "Arithmetic operations in GF(2^m)", *Journal of Cryptology*, на этапе подготовки.

[5] AGNEW, G. B., MULLIN, R. C., ONYSZCHUK, I. M., VANSTONE, S. A., "An implementation for a fast public-key cryptosystem", *Journal of Cryptology*, vol. 3, 1991, pp. 63 – 79.

[6] AGNEW, G. B., MULLIN, R. C., VANSTONE, S. A., "A fast elliptic curve cryptosystem", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 706 – 708.

[7] AHO, A.V., HOPCROFT, J. E., ULLMAN, J. D., *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading, MA, 1974; (есть русский перевод: А. Ахо, Дж. Хопкрофт, Дж. Ульман, *Построение и анализ вычислительных алгоритмов*, М., Мир, 1979).

[8] ALEXI, W., CHOR, B.–Z., GOLDREICH, O., SCHNORR, C. P., "RSA and Rabin functions: Certain parts are as hard as the whole", *SIAM Journal on Computing*, vol. 17, 1988, pp. 194 – 209.

[9] ALPERN, B., SCHNEIDER, F. B., "Key exchange using 'keyless cryptography'", *Information Processing Letters*, vol. 16, 1983, pp. 79 – 81.

[10] ANGLUIN, D., LICHTENSTEIN, D., "Provable security of cryptosystems: A survey", технический отчет YALEU/DCS/TR-288, Department of Computer Science, Yale University, 1983.

[11] BABAI, L., MORAN, S., "Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes", *Journal of Computer and System Sciences*, vol. 36, 1988, pp. 254-276.

[12] BACON, R., *Secret Works of Art and the Nullity of Magic*, circa 1250.

[13] BANERJEE, S. K., "High speed implementation of DES", *Computers and Security*, vol. 1, 1982, pp. 261-267.

[14] BÁRÁNY, I., FÜREDI, Z., "Mental poker with three or more players", *Information and Control*, vol. 59, 1983, pp. 84-93.

[15] BEAUCHEMIN, P., BRASSARD, G., "A generalization of Hellman's extension to Shannon's approach to cryptography", *Journal of Cryptology*, vol. 1, 1988, pp. 129-131.

[16] BEAUCHEMIN, P., BRASSARD, G., CRÉPEAU, C., GOUTIER, C., POMERANCE, C., "The generation of random numbers that are probably prime", *Journal of Cryptology*, vol. 1, 1988, pp. 53-64.

[17] BEAVER, D., FEIGENBAUM, J., "Hiding instances in multioracle queries", *Proceedings of 7th Symposium on Theoretical Aspects of Computer Science*, February 1990, Springer-Verlag, pp. 37-48.

[18] BELLARE, M., MICALI, S., "How to sign given any trapdoor permutation", *Journal of the ACM*, vol. 39, 1992, pp. 214-233.

[19] BELLARE, M., MICALI, S., OSTROVSKY, R., "Perfect zero-knowledge in constant rounds", *Proceedings of 22nd ACM Symposium on Theory of Computing*, May 1990, pp. 482-493.

[20] BELLARE, M., PETRANK, E., "Making zero-knowledge provers efficient", *Proceedings of 24th ACM Symposium on Theory of Computing*, May 1992, pp. 711-722.

[21] BENALOH, J. D., "Cryptographic capsules: A disjunctive primitive for interactive protocols", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer-Verlag, pp. 213-222.

[22] BENALOH, J. D., LEICHTER, J., "Generalized secret sharing and monotone functions", *Advances in Cryptology — Crypto '88 Proceedings*, August 1988, Springer-Verlag, pp. 27-35.

[23] BENALOH, J. D., YUNG, M., "Distributing the power of a government to enhance the privacy of voters", *Proceedings of 5th ACM Symposium on Principles of Distributed Computing*, August 1986, pp. 52-62.

[24] Bengio, S., Brassard, G., Desmedt, Y. G., Goutier, C., Quisquater, J.–J., "Secure implementation of identification systems", *Journal of Cryptology*, vol. 4, 1991, pp. 175 – 183.

[25] Bennett, C. H., Bessette, F., Brassard, G., Salvail, L., Smolin, J., "Experimental quantum cryptography", *Journal of Cryptology*, vol. 5, 1992, pp. 3 – 28.

[26] Bennett, C. H., Brassard, G., "An update on quantum cryptography", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 475 – 480.

[27] Bennett, C. H., Brassard, G., "Quantum cryptography: Public-key distribution and coin tossing", *Proceedings of the International Conference on Computers, Systems and Signal Processing*, Bangalore, India, December 1984, pp. 175 – 179.

[28] Bennett, C. H., Brassard, G., "Quantum public key distribution system", *IBM Technical Disclosure Bulletin*, vol. 28, 1985, pp. 3153 – 3163.

[29] Bennett, C. H. Brassard, G., "The dawn of a new era for quantum cryptography: The experimental prototype is working", *Sigact News*, vol. 20, no. 4, 1989, pp. 78 – 82.

[30] Bennett, C. H., Brassard, G., Breidbart, S., "Quantum cryptography II: How to re-use a one-time pad safely even if $\mathcal{P} = \mathcal{NP}$", рукопись можно получить у авторов, November 1982.

[31] Bennett, C. H., Brassard, G., Breidbart, S., Wiesner, S., "Quantum cryptography, or unforgeable subway tokens", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 267 – 275.

[32] Bennett, C. H., Brassard, G., Crépeau, C., Maurer, U. M., "Privacy amplification against probabilistic information", на этапе подготовки.

[33] Bennett, C. H., Brassard, G., Crépeau, C. Skubiszewska, M.–H., "Practical quantum oblivious transfer", *Advances in Cryptology — Crypto '91 Proceedings*, August 1991, Springer – Verlag, pp. 351 – 366.

[34] Bennett, C. H., Brassard, G. Ekert, A. K., "Quantum cryptography", *Scientific American*, October 1992, pp. 50 – 57.

[35] Bennett, C. H., Brassard, G. Mermin, N. D., "Quantum cryptography without Bell's theorem", *Physical Review Letters*, vol. 68, 3 February 1992, pp. 557 – 559.

[36] Bennett, C. H., Brassard, G., Robert, J.–M., "Privacy amplification by public discussion", *SIAM Journal on Computing*, vol. 17, 1988, pp. 210 – 229.

[37] Ben–Or, M., Goldwasser, S., Kilian, J., Wigderson, A., "Multi-prover interactive proofs: How to remove intractability assumptions", *Proceedings of 20th ACM Symposium on Theory of Computing*, May 1988, pp. 113 – 131.

[38] Ben–Or, M., Goldwasser, S., Wigderson, A., "Completeness theorems for non-cryptographic fault-tolerant distributed computation", будет опубликована в *Journal of Computer and System Sciences*; предварительный вариант в: *Proceedings of 20th ACM Symposium on Theory of Computing*, May 1988, pp. 1 – 10.

[39] Ben–Or, M., Linial, N., "Collective coin flipping", в: *Advances in Computing Research 5: Randomness and Computation*, S. Micali (editor), JAI Press, Greenwich, CT, 1989.

[40] Berger, R., Peralta, R., Tedrick, T., "A provably secure oblivious transfer protocol", *Advances in Cryptology: Proceedings of Eurocrypt 84*, April 1984, Springer – Verlag, pp. 379 – 386.

[41] Berlekamp, E. R., *Algebraic Coding Theory*, McGraw–Hill, New York, NY, 1968; переработанное издание: Aegean Park Press, Laguna Hills, CA, 1984; (есть русский перевод первого издания: Э. Берлекэмп, *Алгебраическая теория кодирования*, М., Мир, 1971).

[42] Berthiaume, A., Brassard, G., "The quantum challenge to structural complexity theory", *Proceedings of 7th Annual IEEE Conference on Structure in Complexity Theory*, June 1992, pp. 132 – 137.

[43] Biham, E., Shamir, A., "Differential cryptanalysis of DES-like cryptosystems", *Journal of Cryptology*, vol. 4, 1991, pp. 3 – 72.

[44] Biham, E., Shamir, A., "Differential cryptanalysis of the full 16–round DES", *Advances in Cryptology — Crypto '92 Proceedings*, August 1992, Springer – Verlag.

[45] Blake, I. F., Fuji–Hara, R., Mullins, R. C., Vanstone, S. A., "Computing logarithms in finite fields of characteristic two", *SIAM Journal on Algebraic Discrete Methods*, vol. 5, 1984, pp. 276 – 285.

[46] Blakley, G. R., "Safeguarding cryptographic keys", *Proceedings of AFIPS National Computer Conference*, vol. 48, 1979, pp. 313 – 317.

[47] Blakley, G. R., Borosh, I., "Rivest–Shamir–Adleman public key cryptosystems do not always conceal messages", *Computers & Mathematics with Applications*, vol. 5, 1979, pp. 169 – 178.

[48] BLAKLEY, G. R., MEADOWS, C., "A database encryption scheme which allows the computation of statistics using encrypted data", *Proceedings of IEEE Symposium on Computer Security and Privacy*, 1985, pp. 116 – 122.

[49] BLUM, L., BLUM, M., SHUB, M., "A simple unpredictable pseudo-random number generator", *SIAM Journal on Computing*, vol. 15, 1986, pp. 364 – 383.

[50] BLUM, M., "Three applications of the oblivious transfer: Part I: Coin flipping by telephone; Part II: How to exchange secrets; Part III: How to send certified electronic mail", Department of EECS, University of California, Berkeley, CA, 1981.

[51] BLUM, M., "Coin flipping by telephone: A protocol for solving impossible problems", *Proceedings of 24th IEEE Computer Conference (CompCon)*, 1982, pp. 133 – 137; перепечатано в: *Sigact News*, vol. 15, no. 1, 1983, pp. 23 – 27.

[52] BLUM, M., "How to exchange (secret) keys", *ACM Transactions on Computer Systems*, vol. 1, 1983, pp. 175 – 193.

[53] BLUM, M., "All-or-nothing certified mail", *Workshop on Mathematical Aspects of Cryptography*, Endicott House, MIT, 1985 (не является отчетом).

[54] BLUM, M., "How to prove a theorem so that no one else can claim it", представлено на *International Congress of Mathematicians*, Berkeley, CA, 1986.

[55] BLUM, M., FELDMAN, P., MICALI, S., "Non-interactive zero-knowledge and its applications", *Proceedings of 20th ACM Symposium on Theory of Computing*, May 1988, pp. 103 – 112.

[56] BLUM, M., GOLDWASSER, S., "An *efficient* probabilistic public-key encryption scheme which hides all partial information", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 289 – 299.

[57] BLUM, M., MICALI, S., "How to generate cryptographically strong sequences of pseudo-random bits", *SIAM Journal on Computing*, vol. 13, 1984, pp. 850 – 864.

[58] BLUM, M., VAZIRANI, U. V., VAZIRANI, V. V., "Reducibility among protocols", *Advances in Cryptology: Proceedings of Crypto 83*, August 1983, Plenum Press, pp. 137 – 146.

[59] BOHM, D., *Quantum Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1951; (есть русский перевод: Д. БОМ, *Квантовая теория*, 2-е изд., М., Наука, 1965).

[60] BOYAR, J. F., BRASSARD, G. PERALTA, R., "Subquadratic zero-knowledge", *Proceedings of 32nd IEEE Symposium on Foundations of Computer Science*, October 1991, pp. 69–78.

[61] BOYAR, J. F., CHAUM, D., DAMGÅRD, I. B., PEDERSEN, T., "Convertible undeniable signatures", *Advances in Cryptology — Crypto '90 Proceedings*, August 1990, Springer–Verlag, pp. 189–205.

[62] BOYAR, J. F., KRENTEL, M. W., KURTZ, S. A., "A discrete logarithm implementation of perfect zero-knowledge blobs", *Journal of Cryptology*, vol. 2, 1990, pp. 63–76.

[63] BOYAR, J. F., LUND, C., PERALTA, R., "On the communication complexity of zero-knowledge proofs", *Journal of Cryptology*, в процессе подготовки; предварительная версия в: *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer–Verlag, pp. 507–525.

[64] BRASSARD, G., "A note on the complexity of cryptography", *IEEE Transactions on Information Theory*, vol. IT–25, 1979, pp. 232–233.

[65] BRASSARD, G., "A time-luck tradeoff in relativized cryptography", *Journal of Computer and System Sciences*, vol. 22, 1981, pp. 280–311.

[66] BRASSARD, G., "An optimally secure relativized cryptosystem", *Advances in Cryptography, a Report on Crypto 81*, технический отчет no. 82–04, Department of ECE, University of California, Santa Barbara, CA, 1982, pp. 54–58; перепечатано в: *Sigact News*, vol. 15, no. 1, 1983, pp. 28–33.

[67] BRASSARD, G., "Computationally secure authentication tags requiring short secret shared keys", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 79–86.

[68] BRASSARD, G., "Relativized cryptography", *IEEE Transactions on Information Theory*, vol. IT–29, 1983, pp. 877–894.

[69] BRASSARD, G., "Cryptology in academia: A ten year retrospective", *Proceedings of 29th IEEE Computer Conference (CompCon)*, 1987, pp. 222–226.

[70] BRASSARD, G., "Cryptology", *Encyclopædia of Mathematics*, D. Reidel Publishing Company, Dordrecht, Pays–Bas.

[71] BRASSARD, G., *Modern Cryptology*, Springer–Verlag, Lecture Notes in Computer Science, vol. 325, 1988.[1]

[1] Рецензия на русском языке: ВЕТЧИНИН М. П., «Современная криптология: рецензия на книгу Brassard, G., "*Modern Cryptology*"», *Новые книги за рубежом*, 1994, № 9 (сентябрь), стр. 30–34.

[72] BRASSARD, G., "How to improve signature schemes", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 16 – 22.

[73] BRASSARD, G., "Cryptology column — Hot news on interactive protocols", *Sigact News*, vol. 21, no. 1, 1990, pp. 7 – 11.

[74] BRASSARD, G., "Cryptology column — Hiding information from oracles", *Sigact News*, vol. 21, no. 2, 1990, pp. 5 – 11.

[75] BRASSARD, G., "Cryptology column — How convincing is your protocol?", *Sigact News*, vol. 22, no. 1, 1991, pp. 5 – 12.

[76] BRASSARD, G. BRATLEY, P., *Algorithmique: conception et analyse*, Masson, Paris, 1987.

[77] BRASSARD, G., BRATLEY, P., *Algorithmics: Theory and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[78] BRASSARD, G., CHAUM, D., CRÉPEAU, C., "Minimum disclosure proofs of knowledge", *Journal of Computer and System Sciences*, vol. 37, 1988, pp. 156 – 189.

[79] BRASSARD, G. CRÉPEAU, C., "Non-transitive transfer of confidence: A *perfect* zero-knowledge interactive protocol for SAT and beyond", *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, October 1986, pp. 188 – 195.

[80] BRASSARD, G. CRÉPEAU, C., "Sorting out zero-knowledge", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 181 – 191.

[81] BRASSARD, G. CRÉPEAU, C., "Quantum bit commitment and coin tossing protocols", *Advances in Cryptology — Crypto '90 Proceedings*, August 1990, Springer – Verlag, pp. 49 – 61.

[82] BRASSARD, G., CRÉPEAU, C., LAPLANTE, S., LÉGER, C., "Computationally convincing proofs of knowledge", *Proceedings of 8th Symposium on Theoretical Aspects of Computer Science*, February 1991, Springer – Verlag, pp. 251 – 262.

[83] BRASSARD, G., CRÉPEAU, C., ROBERT, J.–M., "All-or-nothing disclosure of secrets", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 234 – 238.

[84] BRASSARD, G., CRÉPEAU, C., ROBERT, J.–M., "Information theoretic reductions among disclosure problems", *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, October 1986, pp. 168 – 173.

[85] BRASSARD, G., CRÉPEAU, C. YUNG, M., "Constant-round perfect zero-knowledge computationally convincing protocols", *Theoretical Computer Science*, vol. 84, 1991, pp. 23–52.

[86] BRASSARD, G., MONET, S., ZUFFELLATO, D., "Algorithmes pour l'arithmétique des très grands entiers", *TSI: Technique et Science Informatiques*, vol. 5, 1986, pp. 89–102.

[87] BRASSARD, G. YUNG, M., "One-way group actions", *Advances in Cryptology — Crypto '90 Proceedings*, August 1990, Springer–Verlag, pp. 94–107.

[88] BRICKELL, E. F., "Solving low density knapsacks", *Advances in Cryptology: Proceedings of Crypto 83*, August 1983, Plenum Press, pp. 25–37.

[89] BRICKELL, E. F., "Breaking iterated knapsacks", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer–Verlag, pp. 342–358.

[90] BRICKELL, E. F., "The cryptanalysis of knapsack cryptosystems", в: *Applications of Discrete Mathematics*, R. D. Ringeisen, F. S. Roberts (editors), SIAM, Philadelphia, PA, 1988, pp. 3–23.

[91] BRICKELL, E. F., "A survey of hardware implementations of RSA", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer–Verlag, pp. 368–370.

[92] BRICKELL, E. F., DAVENPORT, D. M., "On the classification of ideal secret sharing schemes", *Journal of Cryptology*, vol. 4, 1991, pp. 123–134.

[93] BRICKELL, E. F., ODLYZKO, A. M., "Cryptanalysis: A survey of recent results", в: [333], 1992, pp. 501–540; (есть русский перевод журнальной версии в: ТИИЭР, т. 76(1988), № 5, стр. 75–94).

[94] BURMESTER, M. V. D., DESMEDT, Y. G., "Remarks on the soundness of proofs", *Electronics Letters*, vol. 25, 26 October 1989, pp. 1509–1511.

[95] CARROLL, J. M., *Computer Security*, Butterworth Publishers, Stoneham, MA, 1987.

[96] CARROLL, J. M., ROBBINS, L. E., "The automated cryptanalysis of polyalphabetic ciphers", *Cryptologia*, vol. XI, 1987, pp. 193–205.

[97] CARROLL, J. M., ROBBINS, L. E., "Computer cryptanalysis of product ciphers", *Cryptologia*, vol. XIII, 1989, pp. 303–326.

[98] CARTER, J. L., WEGMAN, M. N., "Universal classes of hash functions", *Journal of Computer and System Sciences*, vol. 18, 1979, pp. 143 – 154.

[99] CEILLIER, R., *La cryptographie*, Que sais-je?, vol. 116 (sic!), Presses Universitaires de France, Paris, 1945.

[100] CHAUM, D., "Untraceable electronic mail, return addresses and digital pseudonyms", *Communications of the ACM*, vol. 24, 1981, pp. 84 – 88.

[101] CHAUM, D., "Security without identification: Transaction system to make Big Brother obsolete", *Communications of the ACM*, vol. 28, 1985, pp. 1030 – 1044.

[102] CHAUM, D., "Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms", *Advances in Cryptology — Eurocrypt '85 Proceedings*, April 1985, Springer – Verlag, pp. 241 – 244.

[103] CHAUM, D., "Demonstrating that a public predicate can be satisfied without revealing any information about how", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 195 – 199.

[104] CHAUM, D., "The dining cryptographers problem: Unconditional sender and recipient untraceability", *Journal of Cryptology*, vol. 1, 1988, pp. 65 – 75.

[105] CHAUM, D., "Online cash checks", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 288 – 293.

[106] CHAUM, D., "The spymasters double-agent problem: Multiparty computation secure unconditionally from minorities and cryptographically from majorities", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 591 – 602.

[107] CHAUM, D., "Zero-knowledge undeniable signatures", *Advances in Cryptology — Eurocrypt '90 Proceedings*, May 1990, Springer – Verlag, pp. 458 – 464.

[108] CHAUM, D., "Some weaknesses of 'Weaknesses of undeniable signatures'", *Advances in Cryptology — Eurocrypt '91 Proceedings*, April 1991, Springer – Verlag, pp. 554 – 556.

[109] CHAUM, D., VAN ANTWERPEN, H., "Undeniable signatures", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 212 – 216.

[110] CHAUM, D., DEN BOER, B., VAN HEYST, E., MJØLSNES, S., STEENBEEK, A., "Efficient offline electronic checks", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 294 – 301.

[111] CHAUM, D., CRÉPEAU, C., DAMGÅRD, I. B., "Multiparty unconditionally secure protocols", *Proceedings of 20th ACM Symposium on Theory of Computing*, May 1988, pp. 11 – 19.

[112] CHAUM, D., DAMGÅRD, I. B., VAN DE GRAAF, J., "Multiparty computations ensuring privacy of each party's input and correctness of the result", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 87 – 119.

[113] CHAUM, D., EVERTSE, J.–H., "Cryptanalysis of DES with a reduced number of rounds", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 192 – 211.

[114] CHAUM, D., EVERTSE, J.–H., VAN DE GRAAF, J., PERALTA, R., "Demonstrating possession of a discrete logarithm without revealing it", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 200 – 212.

[115] CHAUM, D., FIAT, A., NAOR, M., "Untraceable electronic cash", *Advances in Cryptology — Crypto '88 Proceedings*, August 1988, Springer – Verlag, pp. 319 – 327.

[116] CHAUM, D., SCHAUMÜLLER–BILCH, I. (editors), *Smart Card 2000*, North–Holland, Amsterdam, Pays–Bas, 1988.

[117] CHOR, B.–Z., *Two Issues in Public Key Cryptography*, ACM Distinguished Doctoral Dissertation Series, MIT Press, Cambridge, MA, 1986.

[118] CHOR, B.–Z., GOLDREICH, O., HÅSTAD, J., FREIDMANN, J., RUDICH, S., SMOLENSKY, R., "The bit extraction problem or t-resilient functions", *Proceedings of 26th IEEE Symposium on Foundations of Computer Science*, October 1985, pp. 396 – 407.

[119] COHEN (BENALOH), J. D., FISHER, M., "A robust and verifiable cryptographically secure election scheme", *Proceedings of 26th IEEE Symposium on Foundations of Computer Science*, October 1985, pp. 372 – 382.

[120] COOK, S. A., "The complexity of theorem proving procedures", *Proceedings of 3rd ACM Symposium on Theory of Computing*, May 1971, pp. 151 – 158; (есть русский перевод в: *Кибернетический сборник*, нов. сер., вып. 12, М., Мир, 1975, стр. 5 – 15).

[121] COPPERSMITH, D., "Cheating at mental poker", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 104 – 107.

[122] COPPERSMITH, D., "Cryptography", *IBM Journal of Research and Development*, vol. 31, 1987, pp. 244 – 248.

[123] COPPERSMITH, D., "DES designed to withstand differential cryptanalysis", *Internet*, Newsgroup `sci.crypt`, 23 March 1992.

[124] COPPERSMITH, D., ODLYZKO, A. M., SCHROEPPEL, R., "Discrete logarithms in GF(p)", *Algorithmica*, vol. 1, 1986, pp. 1 – 15.

[125] COUVREUR, C., QUISQUATER, J.–J., "An introduction to fast generation of large prime numbers", *Philips Journal of Research*, vol. 37, 1982, pp. 231 – 264; *errata* в: *ibid*, vol. 38, 1983, p. 77.

[126] CRÉPEAU, C., "A secure poker protocol that minimizes the effect of player coalitions", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 73 – 86.

[127] CRÉPEAU, C., "A zero-knowledge poker protocol that achieves confidentiality of the players' strategy, *or* How to achieve an electronic poker face", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 239 – 247.

[128] CRÉPEAU, C., "Equivalence between two flavours of oblivious transfers", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 350 – 354.

[129] CRÉPEAU, C., "Verifiable disclosure of secrets and application", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 150 – 154.

[130] CRÉPEAU, C., *Correct and Private Reductions among Oblivious Transfers*, thèse de doctorat, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 1990.

[131] CRÉPEAU, C., SÁNTHA, M., "Efficient reduction among oblivious transfer protocols based on new self-intersecting codes", *Proceedings of Sequences 91: Methods in Communication, Security and Computer Science*, June 1991, Springer – Verlag.

[132] CSISZÁR, I., KÖRNER, J., "Broadcast channels with confidential messages", *IEEE Transactions on Information Theory*, vol. IT–24, 1978, pp. 339 – 348.

[133] DAMGÅRD, I. B., *The Application of Claw Free Functions in Cryptography; Unconditional Protection in Cryptographic Protocols*, тезисы докторской диссертации, Matematisk Institut, Aarhus Universitet, Århus, Danemark, 1988.

[134] DAVIDA, G., "Chosen signature cryptanalysis of the RSA (MIT) public key cryptosystem", технический отчет TR–CS–82–2, Department of EECS, University of Wisconsin, Milwaukee, WI, 1982.

[135] DAVIES, D. W., PARKIN, G. I. P., "The average cycle size of the key stream in output feedback encipherment", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 97–98.

[136] DAVIES, D. W., PRICE, W. L., *Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer*, John Wiley & Sons, New York, NY, 1984.

[137] DAVIO, M., DESMEDT, Y. G., FOSSEPREZ, M., GOVAERTS, R., HULSBOSCH, J., NEUTJENS, P., PIRET, P., QUISQUATER, J.–J., VANDEWALLE, J., WOUTERS, P., "Analytical characteristics of the DES", *Advances in Cryptology: Proceedings of Crypto 83*, August 1983, Plenum Press, pp. 171–202.

[138] DAVIO, M., DESMEDT, Y. G., GOUBERT, J., HOORNAERT, F., QUISQUATER, J.–J., "Efficient hardware and software implementations of the DES", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 144 – 146.

[139] DEAVOURS, C. A., "Unicity points in cryptanalysis", *Cryptologia*, vol. 1, 1977, pp. 46–68.

[140] DEAVOURS, C. A., KRUH, L., *Machine Cryptography and Modern Cryptanalysis*, Artech House, Dedham, MA, 1985.

[141] DELORME, C., QUISQUATER, J.–J., "Networks, graphs and security: I. Secrecy on identity of sender and addressee; II. Key exchanges", представлено на *First International Conference on Industrial and Applied Mathematics*, Paris, June 1987; существует в виде рукописи Manuscrit M210, Philips Research Laboratory, Bruxelles, Belgique, 1987.

[142] DENNING, D. E. R., *Cryptography and Data Security*, Addison–Wesley, Reading, MA, 1983.

[143] DENNING, D. E. R., "Digital signatures with RSA and other public-key cryptosystems", *Communications of the ACM*, vol. 27, 1984, pp. 388–392.

[144] De Santis, A., Micali, S., Persiano, G., "Non-interactive zero-knowledge proof systems", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 52 – 72.

[145] Desmedt, Y. G., *Analysis of the Security and New Algorithms for Modern Industrial Cryptography*, тезисы докторской диссертации, Katholieke Universiteit Leuven, Belgique, 1984.

[146] Desmedt, Y. G., "Unconditionally secure authentication schemes and practical and theoretical consequences", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 42 – 55.

[147] Desmedt, Y. G., "Major security problems with the 'unforgeable' (Feige–)Fiat–Shamir proofs of identity and how to overcome them", *Proceedings of Securicom 88*, March 1988, pp. 147 – 159.

[148] Desmedt, Y. G., Goutier, C., Bengio, S., "Special uses and abuses of the Fiat – Shamir passport protocol", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 21 – 39.

[149] Desmedt, Y. G., Odlyzko, A. M., "A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 516 – 522.

[150] Desmedt, Y. G., Vandewalle, J., Govaerts, R., "Critical analysis of the security of knapsack public key algorithms", *IEEE Transactions on Information Theory*, vol. IT–30, 1984, pp. 601 – 611.

[151] Desmedt, Y. G., Yung, M., "Weaknesses of undeniable signature schemes", *Advances in Cryptology — Eurocrypt '91 Proceedings*, April 1991, Springer – Verlag, pp. 205 – 220.

[152] Deutsch, D., "Quantum theory, the Church – Turing principle and the universal quantum computer", *Proceedings of the Royal Society*, Londres, vol. A400, 1985, pp. 97 – 117.

[153] Deutsch, D., "Three connections between Everett's interpretation and experiment", в: *Quantum Concepts in Space and Time*, R. Penrose, C. Isham (editors), Clarendon Press, Oxford, 1986, pp. 215 – 225.

[154] Deutsch, D., "Quantum communication thwarts eavesdroppers", *New Scientist*, 9 December 1989, pp. 25 – 26.

[155] Deutsch, D., Jozsa, R., "Rapid solution of problems by quantum computation", рукопись находится у авторов, December 1991.

[156] DIFFIE, W., "The first ten years of public-key cryptography", в: [333], 1992, pp. 135–175; (есть русский перевод журнальной версии в: ТИИЭР, т. 76(1988), №5, стр. 54–74).

[157] DIFFIE, W., HELLMAN, M. E., "New directions in cryptography", *IEEE Transactions on Information Theory*, vol. IT–22, 1976, pp. 644–654.

[158] DIFFIE, W., HELLMAN, M. E., "Exhaustive cryptanalysis of the NBS Data Encryption Standard", *Computer*, vol. 10, June 1977, pp. 74–84.

[159] DIFFIE, W., HELLMAN, M. E., "Privacy and authentication: An introduction to cryptography", *Proceedings of the IEEE*, vol. 67, 1979, pp. 397–427; (есть русский перевод в: ТИИЭР, т. 67(1979), №3, стр. 71–109).

[160] DUSSE, S. R., KALISKI, B. S., JR., "A cryptographic library for the Motorola DSP 56000", *Advances in Cryptology — Eurocrypt '90 Proceedings*, May 1990, Springer–Verlag, pp. 230–244.

[161] EIER, R., LAGGER, H., "Trapdoors in knapsack cryptosystems", *Cryptography — Proceedings, Burg Feuerstein 1982*, Lecture Notes in Computer Science, vol. 149, Springer–Verlag, 1983, pp. 316–322.

[162] EINSTEIN, A., PODOLSKY, B., ROSEN, N., "Can quantum-mechanical description of physical reality be considered complete", *Physical Review*, vol. 47, 1935, p. 777.

[163] EKERT, A. K., "La mécanique quantique au secours des agents secrets", *La Recherche*, June 1991, pp. 790–791.

[164] EKERT, A. K., "Quantum cryptography based on Bell's theorem", *Physical Review Letters*, vol. 67, 5 August 1991, pp. 661–663.

[165] EKERT, A. K., RARITY, J., TAPSTER, P., PALMA, G. M., "Practical quantum cryptography based on two-photon interferometry", *Physical Review Letters*, vol. 69, 31 August 1992, pp. 1293–1296.

[166] EL GAMAL, T., "A public key cryptosystem and a signature scheme based on discrete logarithm", *IEEE Transactions on Information Theory*, vol. IT–31, 1985, pp. 469–472.

[167] EVANS, A., KANTROWITZ, W., WEISS, E., "A user authentication scheme not requiring secrecy in the computer", *Communications of the ACM*, vol. 17, 1974, pp. 437–442.

[168] EVEN, S., GOLDREICH, O., LEMPEL, A., "A randomized protocol for signing contracts", *Communications of the ACM*, vol. 28, 1985, pp. 637–647.

[169] EVEN, S., YACOBI, Y. G., "Cryptography and $\mathcal{NP}$-completeness", *Proceedings of 7th International Colloquium on Automata, Languages, and Programming*, Springer – Verlag, 1980, pp. 195 – 207.

[170] EVERTSE, J.-H., VAN HEYST, E., "Which new RSA signatures can be computed from some given RSA signatures?", *Journal of Cryptology*, vol. 5, 1992, pp. 41 – 52.

[171] FEIGE, U., FIAT, A., SHAMIR, A., "Zero-knowledge proofs of identity", *Journal of Cryptology*, vol. 1, 1988, pp. 77 – 94.

[172] FEIGE, U., LAPIDOT, D., SHAMIR, A., "Multiple non-interactive zero knowledge proofs based on a single random string", *Proceedings of 31st IEEE Symposium on Foundations of Computer Science*, October 1990, pp. 308 – 317.

[173] FEIGE, U., SHAMIR, A., "Zero knowledge proofs of knowledge in two rounds", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 526 – 544.

[174] FEIGENBAUM, J., "Encrypting problem instances, or, ..., can you take advantage of someone without having to trust him?", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 477 – 488.

[175] FEISTEL, H., "Cryptography and computer privacy", *Scientific American*, May 1973, pp. 15 – 23.

[176] FELDMEIER, D. C., KARN, P. R., "UNIX password security — Ten years later", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 44 – 63.

[177] FORTNOW, L., "The complexity of perfect zero-knowledge", *Proceedings of 19th ACM Symposium on Theory of Computing*, May 1987, pp. 204 – 209.

[178] FORTUNE, S., MERRIT, M., "Poker protocols", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 454 – 464.

[179] FRANKLIN, M., *Mathematical Investigations of the Data Encryption Standard*, mémoire de maîtrise, Department of EECS, University of California, Berkeley, CA, 1985.

[180] FREIZE, A. M., HÅSTAD, J., KANNAN, R., LAGARIAS, J. C., SHAMIR, A., "Reconstructing truncated integer variables satisfying linear congruences", *SIAM Journal on Computing*, vol. 17, 1988, pp. 262 – 280.

[181] FRIEDMAN, W. F., *Elements of Cryptanalysis*, Aegean Park Press, Laguna Hills, CA, 1976 (написана намного раньше).

[182] GALIL, Z., HABER, S., YUNG, M., "Cryptographic computation: Secure fault-tolerant protocols and the public-key model", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer-Verlag, pp. 135-155.

[183] GALIL, Z., HABER, S., YUNG, M., "Minimum knowledge interactive proofs for decision problems", *SIAM Journal on Computing*, vol. 18, 1989, pp. 711-739.

[184] GALLAGER, R. G., *Information Theory and Reliable Communications*, John Wiley & Sons, New York, NY, 1968. (есть русский перевод: Р. ГАЛЛАГЕР, *Теория информации и надежная связь*, М., Сов. Радио, 1974).

[185] GARDNER, M., "A new kind of cipher that would take millions of years to break", *Scientific American*, August 1977, pp. 120-124; (есть русский перевод в сборнике: М. ГАРДНЕР, *От мозаик Пенроуза к надежным шифрам*, М., Мир, 1993, стр. 231-245, см. также стр. 246-254).

[186] GAREY, M. R., JOHNSON, D. S., *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*, W. H. Freeman and Company, San Francisco, CA, 1979; (есть русский перевод: М. ГЭРИ, Д. ДЖОНСОН, *Вычислительные машины и труднорешаемые задачи*, М., Мир, 1982).

[187] GARLIŃSKI, J., *Intercept: The Enigma War*, J. M. Dent and Sons, Ltd, Londres, 1979.

[188] GLASS, A. S., "Could the smart card be dumb?", *Abstracts of Papers from Eurocrypt 86*, ISBN 91-7870-077-9, 1986, pp. 1.5A-1.5B.

[189] GLEIK, J., "A new approach to protecting secrecy is discovered", *New York Times*, 17 February 1987, p. C1.

[190] GOLDREICH, O., "Two remarks concerning the Goldwasser-Micali-Rivest signature scheme", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer-Verlag, pp. 104-110.

[191] GOLDREICH, O., "Towards a theory of software protection and simulation by oblivious RAM's", *Proceedings of 19th ACM Symposium on Theory of Computing*, May 1987, pp. 182-194.

[192] GOLDREICH, O., GOLDWASSER, S., MICALI, S., "On the cryptographic applications of random functions", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer-Verlag, pp. 276-288.

[193] GOLDREICH, O., GOLDWASSER, S., MICALI, S., "How to construct random functions", *Journal of the ACM*, vol. 33, 1986, pp. 792–807.

[194] GOLDREICH, O., KRAWCZYK, H., "On the composition of zero-knowledge proof systems", *Proceedings of 17th International Conference on Automata, Languages and Programming*, 1990, Springer–Verlag, pp. 268–282.

[195] GOLDREICH, O., MICALI, S., WIGDERSON, A., "How to play any mental game, or: A completeness theorem for protocols with honest majority", *Proceedings of 19th ACM Symposium on Theory of Computing*, May 1987, pp. 218–229.

[196] GOLDREICH, O., MICALI, S., WIGDERSON, A., "Proofs that yield nothing but their validity, or All languages in $\mathcal{NP}$ have zero-knowledge proof systems", *Journal of the ACM*, vol. 38, 1991, pp. 691–729.

[197] GOLDWASSER, S., MICALI, S., "Probabilistic encryption and how to play mental poker keeping secret all partial information", *Proceedings of 14th ACM Symposium on Theory of Computing*, May 1982, pp. 365–377.

[198] GOLDWASSER, S., MICALI, S., "Probabilistic encryption", *Journal of Computer and System Sciences*, vol. 28, 1984, pp. 270–299.

[199] GOLDWASSER, S., MICALI, S., RACKOFF, C., "The knowledge complexity of interactive proof-systems", *SIAM Journal on Computing*, vol. 18, 1989, pp. 186–208.

[200] GOLDWASSER, S., MICALI, S., RIVEST, R., "A digital signature scheme secure against adaptive chosen-message attacks", *SIAM Journal on Computing*, vol. 17, 1988, pp. 281–308.

[201] GOLDWASSER, S., MICALI, S., YAO, A. C.–C., "On signatures and authentication", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 211–215.

[202] GOOD, I. J., "Pioneering work on computers at Bletchley", в: *A History of Computing in the Twentieth Century*, N. Metropolis, J. Howlett, G.–C. Rota (editors), Academic Press, New York, NY, 1980, pp. 31–46; а также в: *Annals of the History of Computing*, vol. 1, 1979, pp. 38–48.

[203] GORDON, J. A., "Strong primes are easy to find", *Advances in Cryptology: Proceedings of Eurocrypt 84*, April 1984, Springer–Verlag, pp. 216–223.

[204] GOTTLIEB, A., "Conjugal secrets — The untappable quantum telephone", *The Economist*, vol. 311, 22 April 1989, p. 81.

[205] GOVAERTS, R., VANDEWALLE, J., BOSSELAERS, A., PRENEEL, B., "Fast software implementation of the Data Encryption Standard (DES)", труды отчета ESAT/COSMIC, Katholieke Universiteit Leuven, Belgique, 1990.

[206] VAN DE GRAAF, J., PERALTA, R., "A simple and secure way to show the validity of your public key", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 128 – 134.

[207] GROLLMANN, J., SELMAN, A. L., "Complexity measures for public-key cryptosystems", *SIAM Journal on Computing*, vol. 17, 1988, pp. 309 – 335.

[208] GUILLOU, L. C., DAVIO, M., QUISQUATER, J.–J., "Public-key techniques: Randomness and redundancy", *Cryptologia*, vol. XIII, 1989, pp. 167 – 189.

[209] GUILLOU, L. C., UGON, M., QUISQUATER, J.–J., "The smart card: A standardized security device dedicated to public cryptography", в: [333], 1992, pp. 561 – 613.

[210] HÅSTAD, J., "On using RSA with low exponent in a public key network", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 403 – 408.

[211] HÅSTAD, J., "Pseudo-random generation under uniform assumptions", *Proceedings of 22nd ACM Symposium on Theory of Computing*, May 1990, pp. 395 – 404.

[212] HELLMAN, M. E., "An extension of the Shannon theory approach to cryptography", *IEEE Transactions on Information Theory*, vol. IT–23, 1977, pp. 289 – 294.

[213] HELLMAN, M. E., "DES will be totally insecure within ten years", *IEEE Spectrum*, vol. 16, July 1979, pp. 32 – 39.

[214] HELLMAN, M. E., "The mathematics of public-key cryptography", *Scientific American*, August 1979, pp. 146 – 157.

[215] HELLMAN, M. E., MERKLE, R. C., SCHROEPPEL, R., WASHINGTON, L., DIFFIE, W., POHLIG, S., SCHWEITZER, P., "Results on an initial attempt to cryptanalyze the NBS Data Encryption Standard", технический отчет SEL 76–042, Stanford University, 1976.

[216] HERZBERG, A., PINTER, S. S., "Public protection of software", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 158 – 179.

[217] HOORNAERT, F., GOUBERT, J., DESMEDT, Y. G., "Efficient hardware implementations of the DES", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 147 – 173.

[218] HUBER, K., "Some considerations concerning the selection of RSA moduli", *Advances in Cryptology — Eurocrypt '91 Proceedings*, April 1991, Springer – Verlag, pp. 294 – 301.

[219] IMPAGLIAZZO, R., LEVIN, L. A., LUBY, M., "Pseudo-random generation from one-way functions", *Proceedings of 21st ACM Symposium on Theory of Computing*, May 1989, pp. 12 – 24.

[220] IMPAGLIAZZO, R., LUBY, M., "One-way functions are essential for complexity based cryptography", *Proceedings of 30th IEEE Symposium on Foundations of Computer Science*, October 1989, pp. 230 – 235.

[221] IMPAGLIAZZO, R., YUNG, M., "Direct minimum-knowledge computations", *Advances in Cryptology — Crypto '87 Proceedings*, August 1987, Springer – Verlag, pp. 40 – 51.

[222] JIANG YINGKE, "L'homme qui ne voulait pas avoir tort", *Fables de la Chine antique*, 15th Century, издание на разных языках, Beijing, 1980, pp. 122 – 123.

[223] DE JONGE, W., CHAUM, D., "Attacks on some RSA signatures", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 18 – 27.

[224] DE JONGE, W., CHAUM, D., "Some variations on RSA signatures & their security", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 49 – 59.

[225] JOZSA, R., "Characterizing classes of functions computable by quantum parallelism", *Proceedings of the Royal Society*, Londres, vol. A435, 1991, pp. 563 – 574.

[226] JUENEMAN, R. R., "Analysis of certain aspects of output feedback mode", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 99 – 127.

[227] KAHN, D., "Modern cryptology", *Scientific American*, July 1966, pp. 38 – 46.

[228] KAHN, D., *The Codebreakers: The Story of Secret Writing*, Macmillan Publishing Co., New York, NY, 1967.

[229] KAHN, D., "Cryptology goes public", *Foreign Affairs*, осень 1979; перепечатано в: [230], pp. 186 – 203.

[230] KAHN, D., *Kahn on Codes*, Macmillan Publishing Co., New York, NY, 1983.

[231] KALISKI, B. S., JR., RIVEST, R. L., SHERMAN, A. T., "Is the Data Encryption Standard a group? (Results of cycling experiments on DES)", *Journal of Cryptology*, vol. 1, 1988, pp. 3–36.

[232] KILIAN, J., "Founding cryptography on oblivious transfer", *Proceedings of 20th ACM Symposium on Theory of Computing*, May 1988, pp. 20–31.

[233] KILIAN, J., *Uses of Randomness in Algorithms and Protocols*, ACM Distinguished Doctoral Dissertation Series, MIT Press, Cambridge, MA, 1990.

[234] KILIAN, J., "A note on efficient zero-knowledge proofs and arguments", *Proceedings of 24th ACM Symposium on Theory of Computing*, May 1992, pp. 723–732.

[235] KILIAN, J., MICALI, S., OSTROVSKY, R., "Minimum resource zero-knowledge proofs", *Proceedings of 30th IEEE Symposium on Foundations of Computer Science*, November 1989, pp. 474–479.

[236] KNUTH, D. E., *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms*, second edition, Addison-Wesley, Reading, MA, 1981; (есть русский перевод первого издания: Д. КНУТ, *Искусство программирования для ЭВМ, т. 2: Получисленные алгоритмы*, М., Мир, 1977).

[237] KOBLITZ, N., *A Course in Number Theory and Cryptography*, Springer-Verlag, New York, NY, 1987.

[238] KOCHANSKI, M., "Developing an RSA chip", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer-Verlag, pp. 350–357.

[239] KOLATA, G. B., "Cryptography: On the brink of a revolution?", *Science Magazine*, vol. 197, 1977, pp. 747–748.

[240] KOLATA, G. B., "New codes coming into use", *Science Magazine*, vol. 208, 1980, p. 694.

[241] KOLATA, G. B., "Cryptographers gather to discuss research", *Science Magazine*, vol. 214, 1981, pp. 646–647.

[242] KONHEIM, A. G., *Cryptography: A Primer*, John Wiley & Sons, New York, NY, 1981; (есть русский перевод: А. Г. КОНХЕЙМ, *Основы криптографии*, М., Радио и связь, 1987, тираж 500 экз.).

[243] KOTHARI, S., "Generalized linear threshold scheme", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer-Verlag, pp. 231–241.

[244] KRANAKIS, E., *Primality and Cryptography*, Wiley–Teubner Series in Computer Science, 1986.

[245] LAGARIAS, J. C., ODLYZKO, A. M., "Solving low-density subset sum problems", *Journal of the ACM*, vol. 32, 1985, pp. 229 – 246.

[246] LAKSHMIVARAHAN, S., "Algorithms for public-key cryptosystems: Theory and application", *Advances in Computers*, vol. 22, 1983, pp. 45 – 108.

[247] LAMACCHIA. B. A., ODLYZKO, A. M., "Computation of discrete logarithms in prime fields" *Designs, Codes and Cryptography*, vol. 1, 1991, pp. 47 – 62.

[248] LANDAU, S., "Zero knowledge and the department of defense", *Notices of the American Mathematical Society*, vol. 35, 1988, pp. 5 – 12.

[249] LEMPEL, A., "Cryptology in transition", *ACM Computing Surveys*, vol. 11, 1979, pp. 285 – 303.

[250] LEVIN, L. A., "One-way functions and pseudo-random generators", *Proceedings of 17th ACM Symposium on Theory of Computing*, May 1985, pp. 363 – 365.

[251] LIPTON, R., "How to cheat at mental poker", *Proceedings of American Mathematical Society Short Course on Cryptography*, 1981.

[252] LONGPRÉ, L., "The use of public-key cryptography for signing checks", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 187 – 197.

[253] LUBY, M., MICALI, S., RACKOFF, C., "How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin", *Proceedings of 24th IEEE Symposium on Foundations of Computer Science*, November 1983, pp. 11 – 21.

[254] LUBY, M., RACKOFF, C., "How to construct pseudorandom permutations from pseudorandom functions", *SIAM Journal on Computing*, vol. 17, 1988, pp. 373 – 386.

[255] LUND, C., FORTNOW, L., KARLOFF, H., NISAN, N., "Algebraic methods for interactive proof systems", *Proceedings of 31st IEEE Symposium on Foundations of Computer Science*, October 1990, pp. 2 – 10.

[256] MACMILLAN, D., "Single chip encrypts data at 14 Mb/s", *Electronics*, 16 June 1981, pp. 161 – 165.

[257] MASSEY, J. L., "Contemporary cryptology: An introduction", в: [333], 1992, pp. 1 – 39; (есть русский перевод журнальной версии в: ТИИЭР, т. 76 (май 1988), № 5, стр. 24 – 42).

[258] MATYAS, S., "Digital signatures — an overview", *Computer Networks*, vol. 3, 1979, pp. 87–94.

[259] MAURER, U. M., "Conditionally-perfect secrecy and a provably-secure randomized cipher", *Journal of Cryptology*, vol. 5, 1992, pp. 53–66.

[260] MAURER, U. M., "Fast generation of prime numbers and secure public-key cryptographic parameters", будет опубликовано в *Journal of Cryptology*, предварительная версия в: *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer–Verlag, pp. 636–647.

[261] MAURER, U. M., "Secret key agreement by public discussion based on common information", будет опубликовано в *IEEE Transactions on Information Theory*, предварительная версия в: *Proceedings of 23rd ACM Symposium on Theory of Computing*, May 1991, pp. 561–571.

[262] MCELIECE, R. J., "A public-key cryptosystem based on algebraic coding theory", DSN Progress Report 42-44, Jet Propulsion Laboratory, 1978, pp. 114–116.

[263] MENEZES, A. J., OORSCHOT, VAN, P. C., VANSTONE, S. A., *Handbook of Applied Cryptology*, CRC Press, N. W., Boca Raton, Florida, 1996.

[264] MCIVOR, R., "Smart cards", *Scientific American*, November 1985, pp. 130–137.

[265] MERKLE, R. C., "Secure communications over insecure channels", *Communications of the ACM*, vol. 21, 1978, pp. 294–299.

[266] MERKLE, R. C., HELLMAN, M. E., "Hiding information and signatures in trapdoor knapsacks", *IEEE Transactions on Information Theory*, vol. IT-24, 1978, pp. 525–530.

[267] MERKLE, R. C., HELLMAN, M. E., "On the security of multiple encryption", *Communications of the ACM*, vol. 24, 1981, pp. 465–467.

[268] MEYER, C. H., MATYAS, S. M., *Cryptography: A New Dimension in Computer Data Security*, John Wiley & Sons, New York, NY, 1982.

[269] MICALI, S., SCHNORR, C. P., "Efficient, perfect polynomial random number generators", *Journal of Cryptology*, vol. 3, 1991, pp. 157–172.

[270] MONTGOMERY, P. L., "Modular multiplication without trial division", *Mathematics of Computation*, vol. 44, 1985, pp. 519–521.

[271] MORAIN, F., "Atkin's test: News from the front", *Advances in Cryptology — Eurocrypt '89 Proceedings*, April 1989, Springer – Verlag, pp. 626 – 635.

[272] MORRIS, R., THOMPSON, K., "Password security: A case history", *Communications of the ACM*, vol. 22, 1979, pp. 594 – 597.

[273] NAOR, M., "Bit commitment using pseudo-randomness", *Journal of Cryptology*, vol. 4, 1991, pp. 151 – 158.

[274] NAOR, M., OSTROVSKY, R., VENKATESAN, R., YUNG, M., "Perfect zero-knowledge arguments for $\mathcal{NP}$ can be based on general complexity assumptions", *Advances in Cryptology — Crypto '92 Proceedings*, August 1992, Springer – Verlag.

[275] NAOR, M., YUNG, M., "Universal one-way hash functions and their cryptographic applications", *Proceedings of 21st ACM Symposium on Theory of Computing*, May 1989, pp. 33 – 43.

[276] NAOR, M., YUNG, M., "Public-key cryptosystems provably secure against chosen ciphertext attacks", *Proceedings of 22nd ACM Symposium on Theory of Computing*, May 1990, pp. 427 – 437.

[277] NATIONAL BUREAU OF STANDARDS, "Data Encryption Standard", *Federal Information Processing Standard*, U. S. Department of Commerce, FIPS PUB 46, Washington, DC, 1977.

[278] NATIONAL BUREAU OF STANDARDS, "DES modes of operation", *Federal Information Processing Standard*, U. S. Department of Commerce, FIPS PUB 81, Washington, DC, 1980.

[279] NECHVATAL, J., "Public key cryptography", в: [333], 1992, pp. 177 – 288.

[280] NEWMAN, D. B., JR., PICKHOLTZ, R. L., "Cryptography in the private sector", *IEEE Communications Magazine*, vol. 24, 1986, pp. 7 – 10.

[281] ODLYZKO, A. M., "Discrete logarithms in finite fields and their cryptographic significance", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 225 – 314.

[282] ORTON, G. A., ROY, M. P., SCOTT, P. A., PEPPARD, L. E., TAVARES, S. E., "VLSI implementation of public-key encryption algorithms", *Advances in Cryptology — Crypto '86 Proceedings*, August 1986, Springer – Verlag, pp. 277 – 301.

[283] OSTROVSKY, R., "An efficient software protection scheme", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 610 – 611.

[284] PELEG, S., ROSENFELD, A., "Breaking substitution ciphers using a relaxation algorithm", *Communications of the ACM*, vol. 22, 1979, pp. 598 – 605.

[285] PERALTA, R., "Simultaneous security of bits in the discrete log", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer – Verlag, pp. 62 – 72.

[286] PETERSON, I., "Bits of uncertainty: Quantum security", *Science News*, vol. 137, 2 June 1990, pp. 342 – 343.

[287] POE, E. A., "The Gold-Bug", *Dollar Newspaper*, 21 & 28 June 1843; напечатано в различных сборниках рассказов Эдгара По; имеются также многочисленные русские издания, например, в: ЭДГАР ПО, *Избранные произведения в двух томах*, М., Художественная литература, 1972, т. 2, стр. 157 – 191.

[288] POHLIG, S., HELLMAN, M. E., "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Transactions on Information Theory*, vol. IT–24, 1978, pp. 106 – 110.

[289] POMERANCE, C., "Recent developments in primality testing", *Mathematical Intelligencer*, vol. 3, 1981, pp. 97 – 105.

[290] POMERANCE, C., SMITH, J. W., TULER, R., "A pipe-line architecture for factoring large integers with the quadratic sieve algorithm", *SIAM Journal on Computing*, vol. 17, 1988, pp. 387 – 403.

[291] PURDY, G. B., "A high security log-in procedure", *Communications of the ACM*, vol. 17, 1974, pp. 442 – 445.

[292] QUISQUATER, J.–J., COUVREUR, C., "Fast decipherment algorithm for RSA public-key cryptosystem", *Electronic Letters*, vol. 18, 1982, pp. 905 – 907.

[293] QUISQUATER, J.–J., DELVAUX, Y., *Notes sur la cryptographie et la sécurité*, Philips Research Laboratory, Bruxelles, Belgique, 1988.

[294] QUISQUATER, J.–J., DESMEDT, Y. G., "Chinese lotto as an exhaustive code-breaking machine", *Computer*, vol. 24, 1991, pp. 14 – 22.

[295] QUISQUATER, J.–J., M., M. & M. and GUILLOU, L. C., M. A., G., A., G. & S., "How to explain zero-knowledge protocols to your children", *Advances in Cryptology — Crypto '89 Proceedings*, August 1989, Springer – Verlag, pp. 628 – 631.

[296] RABIN, M. O., "Digital signatures", в: *Foundations of Secure Computation*, R. A. DeMillo, D. P. Dobkin, A. K. Jones, R. J. Lipton (editors), Academic Press, New York, NY, 1978, pp. 155 – 168.

[297] RABIN, M. O., "Digital signatures and public-key functions as intractable as factorization", технический отчет MIT/LCS/TR-212, Massachusetts Institute of Technology, 1979.

[298] RABIN, M. O., "Probabilistic algorithm for testing primality", *Journal of Number Theory*, vol. 12, 1980, pp. 128 – 138.

[299] RABIN, M. O., "How to exchange secrets by oblivious transfer", технический отчет TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[300] RABIN, M. O., "Transaction protection by beacons", *Journal of Computer and System Sciences*, vol. 27, 1983, pp. 256 – 267.

[301] RABIN, T., BEN-OR, M., "Verifiable secret sharing and multiparty protocols with honest majority", *Proceedings of 21st ACM Symposium on Theory of Computing*, May 1989, pp. 73 – 85.

[302] RANDELL, B., "The COLOSSUS", в: *A History of Computing in the Twentieth Century*, N. Metropolis, J. Howlett, G.-C. Rota (editors), Academic Press, New York, NY, 1980, pp. 47 – 92.

[303] REJEWSKI, M., "How Polish mathematicians deciphered the Enigma", *Annals of the History of Computing*, vol. 3, 1981, pp. 213 – 234.

[304] RIVEST, R. L., "A description of a single-chip implementation of the RSA cipher", *Lambda Magazine*, vol. 1, 1980, pp. 14 – 18.

[305] RIVEST, R. L., "A short report on the RSA chip", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, 1983, p. 327.

[306] RIVEST, R. L., "Cryptology", в: *Algorithms and Complexity: Handbook of Theoretical Computer Science, vol. A*, J. van Leeuwen (editor), Elsevier Science Publishers B. V. & The MIT Press, 1990, pp. 719 – 755.

[307] RIVEST, R. L., SHAMIR, A., ADLEMAN, L. M., "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol. 21, 1978, pp. 120 – 126.

[308] RIVEST, R. L., SHERMAN, A. T., "Randomized encryption techniques", *Advances in Cryptology: Proceedings of Crypto 82*, August 1982, Plenum Press, pp. 145 – 163.

[309] ROBERT, J.-M., *Détection et correction d'erreurs en cryptographie*, mémoire de maîtrise, Département d'informatique et de recherche opérationnelle, Université de Montréal, 1985.

[310] ROMPEL, J., "One-way functions are necessary and sufficient for secure signatures", *Proceedings of 22nd ACM Symposium on Theory of Computing*, May 1990, pp. 387–394.

[311] SALVAIL, L., *Le problème de réconciliation en cryptographie*, mémoire de maîtrise, Département d'informatique et de recherche opérationnelle, Université de Montréal, 1991.

[312] SCHNEIER, B, *Applied Cryptology*, second edition, John Wiley & Sons, New York, N. Y., 1995.

[313] SCHRIFT, A.W., SHAMIR, A, "The discrete log is very discreet", *Proceedings of 22nd ACM Symposium on Theory of Computing*, May 1990, pp. 405–415.

[314] SEDLAK, H., "The RSA cryptography processor: The first very high speed one-chip solution", *Advances in Cryptology — Eurocrypt '87 Proceedings*, April 1987, Springer–Verlag, pp. 95–105.

[315] SHAMIR, A., "How to share a secret", *Communications of the ACM*, vol. 24, 1979, pp. 612–613.

[316] SHAMIR, A., "On the generation of cryptographically strong pseudo-random sequences", *ACM Transactions on Computer Systems*, vol. 1, 1983, pp. 38–44.

[317] SHAMIR, A., "A polynomial time algorithm for breaking the basic Merkle–Hellman cryptosystem", *IEEE Transactions on Information Theory*, vol. IT–30, 1984, pp. 699–704.

[318] SHAMIR, A., "Identity based cryptosystems and signature schemes", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer–Verlag, pp. 47–53.

[319] SHAMIR, A., "On the security of the DES", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer–Verlag, pp. 280–281.

[320] SHAMIR, A., "$\mathcal{IP} = \mathcal{PSPACE}$", *Proceedings of 31st IEEE Symposium on Foundations of Computer Science*, October 1990, pp. 11–15.

[321] SHAMIR, A., RIVEST, R. L., ADLEMAN, L. M., "Mental poker", в: *Mathematical Gardner*, D. E. Klarner (editor), Wadsworth International, Belmont, CA, 1981, pp. 37–43; (есть русский перевод в: *Математический цветник*, Д. А. Кларнер (сост. и ред.), М., Мир, 1984, стр. 58–66).

[322] SHAND, M., BERTIN, P., VUILLEMIN, J., "Hardware speedups in long integer multiplication", *Proceedings of 2nd ACM Symposium on Parallel Algorithms and Architectures*, July 1990, pp. 138–145.

[323] SHANNON, C. E., "A mathematical theory of communications", *Bell System Technical Journal*, vol. 27, 1948, pp. 379–423, pp. 623–656; (есть русский перевод в: К. ШЕННОН, *Работы по теории информации и кибернетике*, М., ИЛ, 1963, стр. 243–332).

[324] SHANNON, C. E., "Communication theory of secrecy systems", *Bell System Technical Journal*, vol. 28, 1949, pp. 656–715; (есть русский перевод в: К. ШЕННОН, *Работы по теории информации и кибернетике*, М., ИЛ, 1963, стр. 333–402).

[325] SHANNON, C. E., "Prediction and entropy of printed English", *Bell System Technical Journal*, vol. 30, 1951, pp. 50–64; (есть русский перевод в: К. ШЕННОН, *Работы по теории информации и кибернетике*, М., ИЛ, 1963, стр. 669–686).

[326] SIMMONS, G. J., "Symmetric and asymmetric encryption", *ACM Computing Surveys*, vol. 11, 1979, pp. 305–330.

[327] SIMMONS, G. J., "Verification of treaty compliance — revisited", *Proceedings of 1983 IEEE Symposium on Security and Privacy*, 1983, pp. 61–66.

[328] SIMMONS, G. J., "The prisoners' problem and the subliminal channel", *Advances in Cryptology: Proceedings of Crypto 83*, August 1983, Plenum Press, pp. 51–67.

[329] SIMMONS, G. J., "A system for verifying user identity and authorization at the point-of-sale or access", *Cryptologia*, vol. 8, 1984, pp. 1–21.

[330] SIMMONS, G. J., "Authentication theory / Coding theory", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer–Verlag, pp. 411–431.

[331] SIMMONS, G. J., "A secure subliminal channel (?)", *Advances in Cryptology — Crypto '85 Proceedings*, August 1985, Springer–Verlag, pp. 33–41.

[332] SIMMONS, G. J., "Cryptology", *Encyclopædia Britannica*, 15th edition, Macropædia, vol. 16, Chicago, IL, 1986, pp. 913–924B.

[333] SIMMONS, G. J. (editor), *Contemporary Cryptology: The Science of Information Integrity*, IEEE Press, Piscataway, NJ, 1992; (есть русский перевод журнальных версий некоторых статей в: ТИИЭР, т. 76(1988), № 5).

[334] SIMMONS, G. J., "A survey of information authentication", в: [333], 1992, pp. 379–419; (есть русский перевод журнальной версии в: ТИИЭР, т. 76(1988), № 5, стр. 105–125).

[335] SIMMONS, G. J., "An introduction to shared secret and/or shared control schemes and their application", в: [333], 1992, pp. 441 – 497.

[336] SIMMONS, G. J., "How to insure that data acquired to verify treaty compliance are trustworthy", в: [333], 1992, pp. 615 – 630; (есть русский перевод журнальной версии в: ТИИЭР, т. 76(1988), № 5, стр. 126 – 133).

[337] SMID, M. E., BRANSTAD, D. K., "The Data Encryption Standard: Past and future", в: [333], 1992, pp. 43 – 64; (есть русский перевод журнальной версии в: ТИИЭР, т. 76(1988), № 5, стр. 43 – 54).

[338] SMITH, K., "Watch out hackers, public encryption chips are coming", *Electronic Week*, 20 May 1985, pp. 30 – 31.

[339] SOLOVAY, R., STRASSEN, V., "A fast Monte Carlo test for primality", *SIAM Journal on Computing*, vol. 6, 1977, pp. 84 – 85; *erratum* в: *ibid*, vol. 7, 1978, p. 118.

[340] STEWART, I., "Schrödinger's catflap", News and Views, *Nature*, vol. 353, 3 October 1991, pp. 384 – 385.

[341] STINSON, D. R., "Cryptography: Theory and Practice", CRC Press, 1995.

[342] TEDRICK, T., "How to exchange half a bit", *Advances in Cryptology: Proceedings of Crypto 83*, August 1983, Plenum Press, pp. 147 – 151.

[343] TEDRICK, T., "Fair exchange of secrets", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer – Verlag, pp. 434 – 438.

[344] VAN TILBORG, H. C. A., *An Introduction to Cryptology*, Kluwer Academic Publishers, Hingham, MA, 1988.

[345] TOMPA, M., "Zero knowledge interactive proofs of knowledge (a digest)", научно-исследовательский отчет RC 13282 (#59389), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY, 1987.

[346] TOMPA, M., WOLL, H., "Random self-reducibility and zero-knowledge interactive proofs of possession of information", *Proceedings of 28th IEEE Symposium on Foundations of Computer Science*, October 1987, pp. 472 – 482.

[347] TUCHMAN, W., представлено на *National Computer Conference*, Anaheim, CA, June 1978.

[348] TUCHMAN, W., "Hellman presents no shortcut solution to the DES", *IEEE Spectrum*, vol. 16, July 1979, pp. 40 – 41.

[349] VAZIRANI, U. V., VAZIRANI, V. V., "Trapdoor pseudo-random number generators with applications to protocol design", *Proceedings of 24th IEEE Symposium on Foundations of Computer Science*, November 1983, pp. 23–30.

[350] VAZIRANI, U. V., VAZIRANI, V. V., "Efficient and secure pseudo-random number generation", *Proceedings of 25th IEEE Symposium on Foundations of Computer Science*, October 1984, pp. 458–463.

[351] VERBAUWHEDE, I., HOORNAERT, F., VANDEWALLE, J., DE MAN, H., "Security considerations in the design and implementation of a new DES chip", *Advances in Cryptology — Eurocrypt'87 Proceedings*, April 1987, Springer–Verlag, pp. 287–300.

[352] VERNAM, G. S., "Cipher printing telegraph systems for secret wire and radio telegraphic communications", *Journal of the American Institute of Electrical Engineers*, vol. XLV, 1926, pp. 109–115.

[353] VOLTAIRE, *Dictionnaire philosophique*, 1769.

[354] WALLICH, P., "Quantum cryptography", *Scientific American*, May 1989, pp. 28–30.

[355] WEGMAN, M. N., CARTER, J. L., "New hash functions and their use in authentication and set equality", *Journal of Computer and System Sciences*, vol. 22, 1981, pp. 265–279.

[356] WIENER, M. J., "Cryptanalysis of short RSA secret exponents", *IEEE Transactions on Information Theory*, vol. IT–36, 1990, pp. 553–558.

[357] WIESNER, S., "Conjugate coding", *Sigact News*, vol. 15, no. 1, 1983, pp. 78–88; оригинальная рукопись, написанная примерно в 1970 году.

[358] WILKES, M. V., *Time-Sharing Computer Systems*, American Elsevier, New York, NY, 1975.

[359] WILLIAMS, D., HINDIN, H. J., "Can software do encryption job?", *Electronics*, 3 July 1980, pp. 102–103.

[360] WILLIAMS, H. C., "A modification of the RSA public-key encryption procedure", *IEEE Transactions on Information Theory*, vol. IT–26, 1980, pp. 726–729.

[361] WOOTTERS, W. K., ZUREK, W. H., "A single quantum cannot be cloned", *Nature*, vol. 299, 1982, pp. 802–803.

[362] WRIGHT, P., *SpyCatcher*, Viking Penguin Inc., New York, NY, 1987.

[363] WYNER, A. D., "The wire-tap channel", *Bell System Technical Journal*, vol. 54, 1975, pp. 1355–1387.

[364] YAO, A. C.-C., "Theory and applications of trapdoor functions", *Proceedings of 23rd IEEE Symposium on Foundations of Computer Science*, November 1982, pp. 80 - 91.

[365] YAO, A. C.-C., "Protocols for secure computations", *Proceedings of 23rd IEEE Symposium on Foundations of Computer Science*, November 1982, pp. 160 - 164.

[366] YAO, A. C.-C., "How to generate and exchange secrets", *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, October 1986, pp. 162 - 167.

[367] YARDLEY, H. O., *The American Black Chamber*, Bobbs Merrill, Indianapolis, IN, 1931; перепечатано в: Ballantine Books, New York, NY.

[368] YIU, K., PETERSON, K., "A single-chip VLSI implementation of the discrete exponential public key distribution system", *Proceedings of IEEE Global Telecommunications Conference*, vol. 1, 1982, pp. 173 - 179.

[369] YUNG, M., "Cryptoprotocols: Subscription to a public key, the secret blocking and the multi-player mental poker game", *Advances in Cryptology: Proceedings of Crypto 84*, August 1984, Springer - Verlag, pp. 439 - 453.

[370] YUNG, M., "A secure and useful keyless cryptosystem", *Information Processing Letters*, vol. 21, 1985, pp. 35 - 38.

[371] ZIMMER, C., "Perfect gibberish", *Discover*, September 1992, pp. 92 - 99.

[372] ZORPETTE, G., "Breaking the enemy's code", *IEEE Spectrum*, September 1987, pp. 47 - 51.

Именной указатель

Lecture Notes in Computer Science
Edited by G. Goos and J. Hartmanis
325

Gilles Brassard

Modern Cryptology

A Tutorial

Springer-Verlag
New York Berlin Heidelberg London Paris Tokyo

Ж. Брассар

Современная криптология

Руководство

Перевод с английского
с добавлениями
и компьютерный набор в LaTeX'е
М. П. Ветчинина

под редакцией
А. Н. Лебедева

Москва
Издательско-полиграфическая фирма ПОЛИМЕД
1999

Брассар Ж.

Современная криптология: Пер. с англ. — М., Издательско-полиграфическая фирма ПОЛИМЕД, 1999. — 176 с., илл.

ISBN 5-8832-010-2

Эта сравнительно небольшая книжка отражает многочисленные, как теоретические, так и практические аспекты современной криптологии, которые уже стали или становятся частью повседневной жизни. Информационно очень ёмкая, она написана на концептуальном уровне, неформально и с большим мастерством.

Автор книги — известнейший специалист в области криптологии, член совета директоров Международной ассоциации криптологических исследований, главный редактор журнала «Journal of Cryptology», один из основоположников квантовой криптографии и соавтор открытия квантовой телепортации профессор Монреальского университета Жиль Брассар.

Оригинальная английская версия книги была опубликована в серии «Lecture Notes in Computer Science», где печатаются труды основных ежегодных конференций по криптологии — CRYPTO и EUROCRYPT.

Для широкого круга читателей, интересующихся проблемами криптографии и ее применений.

Оглавление