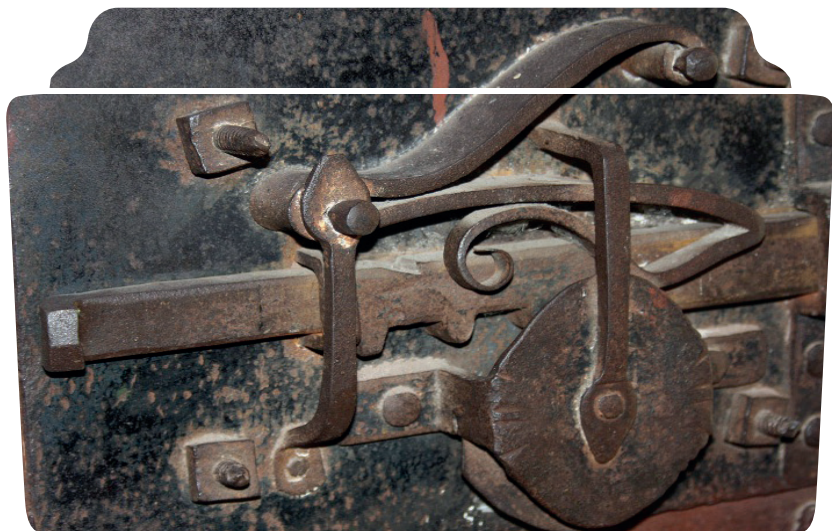
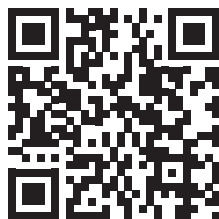


ПРО КРИПТОГРАФИЮ



www.symbol-sight.com



Виктор де Касто



ПРО КРИПТОГРАФИЮ



Автор идеи
и научный редактор серии
СЕРГЕЙ ДЕМЕНОК

НАУЧНО-ПОПУЛЯРНОЕ
ИЗДАТЕЛЬСТВО
«СТРСТ»
Санкт-Петербург. 2020

УДК 001, 501, 510

ББК 22.1

К 28

К 28 ПРО КРИПТОГРАФИЮ (Символ — машина — квант) — СПб.: Страта, 2020. — 240 с., с илл. — (серия «Просто»)

ISBN 978-5-907314-15-3

Чем больше одни стремятся что-то скрыть, тем больше другие хотят это «что-то» узнать. Когда люди только научились писать, их тайны материализовались, представ в образе символов, иероглифов, букв, цифр. Но в таком виде они стали доступны другим. С этого времени началось извечное соревнование между шифровальщиками, пытающимися скрыть информацию, и криптоаналитиками, стремящимися расшифровать ее.

Криптография сегодня — это область научных, прикладных, инженерно-технических исследований, основанная на фундаментальных понятиях математики, физики, теории информации и сложности вычислений.

В книге рассказывается об истории криптографии: от примитивных систем шифрования и дешифровки, придуманных людьми еще в древние времена, до современных компьютерных алгоритмов — как существующих, так и тех, над которыми работают нынешние ученые-криптографы.

Книга предназначена для широкого круга читателей.

Все права защищены. Никакая часть настоящей книги не может быть воспроизведена или передана в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, а также размещение в Интернете, если на то нет письменного разрешения владельцев.

All rights reserved. No parts of this publication can be reproduced, sold or transmitted by any means without permission of the publisher.

УДК 001, 501, 510

ББК 22.1

ISBN 978-5-907314-15-3

© Деменок С. Л., текст, 2020

© ООО «Страта», 2020

КОНВЕРТ БЕЗ ОБРАТНОГО АДРЕСА

Эта книга появилась у нас в редакции довольно неожиданно. Но не совсем случайно. Дело в том, что более десяти лет назад я познакомился с Виктором де Касто в Лиме. Тогда в центре Лимы у президентского дворца дежурили бронетранспортеры. Выходя вечером на улицы центральных кварталов, женщины снимали и прятали сережки, кольца и сумочки.

Однако в пределах комплекса гостиницы Sheraton было и безопасно, и уютно. Мы с моим коллегой Хорхе Артуро коротали вечер в лобби-баре. За соседним столиком в одиночестве смаковал *pisco sour* седой испанец лет пятидесяти. Мы как-то естественно разговорились. Нашего нового знакомого звали Виктор. Он — частый гость в Лиме. Будучи профессором в одном из испанских университетов, он консультировал перуанское правительство по вопросам безопасности и финансовой политики.

Мы общались на английском, но как только возникала заминка, Виктор и Хорхе тут же переходили на испанский. Уже позже Хорхе рассказывал мне, что язык нашего консультанта выдавал испанского аристократа начала прошлого века. Каким-то чудесным образом его испанский избежал новояза, вульгарности и сокращений. В тот вечер Виктор удивил меня познаниями во фрактальной геометрии. В этой области я полагал себя человеком сведущим. Обычно в моем окружении мало кто слышал о фракталах. Виктор знал и глубоко понимал тему. Он высказался в том смысле, словно был лично знаком с Гастоном Жулиа и Пьером Фату.

Впрочем, это вполне могло быть списано на искажение при переводе с испанского на английский или на русский. Уже

в том первом разговоре Виктор горячо поддержал идею публикации серии научно-популярных книг. Как бы к слову он заметил, что в старинных библиотеках и в археологических экспедициях совсем недавно были обнаружены тексты, содержание которых будет в высшей степени интересным для наших читателей. По всей видимости, Виктор принимал участие в расшифровке этих текстов и был достаточно осведомлен в вопросах криптографии. Уже в ту первую встречу он пообещал принять участие в написании некоторых книг для нашей серии, начав с книги о криптографии.

С тех пор мы встречались два или три раза в самых разных местах. Всякий раз Виктор обещал прислать тот или иной текст. Но ничего не присылал. Я не считал удобным настаивать или напоминать о старых обещаниях. И вдруг мы получили по почте диск с текстом давно обещанной книги. Ее заголовок *Just so cryptography* ясно указывал на то, что книга написана Виктором, хотя никакого сопроводительного письма мы не получили. Все наши попытки связаться с Виктором завершились ничем. Он пропал. Такое случалось и раньше. Случится и в будущем.

Мы печатаем полученный текст с надеждой, что книга доставит удовольствие многим нашим читателям, и на то, что когда-нибудь ее увидит и автор.

Редактор

ГЛАВА I. КОДИРОВАНИЕ И ШИФРОВАНИЕ

«На свете множество неразгаданных шифров, непонятых языков, загадочных тайнописей и нерасшифрованных карт... Карты, языки, коды, шифры разгадываются и декодируются каждый день, порой этому предшествуют мучительные годы исследований и расчетов. Последние разработки позволяют расшифровывать ранее непонятные и неразборчивые языки при помощи компьютера».

Livejournal.com

ОТ ОСКОЛКА — К КУБИТУ

Искусство создания посланий, которые могут быть поняты только отправителем и получателем, а любому другому покажутся абсолютно бессмысленными, известно давно. В Древней Греции разбивали горшок и два соседних осколка передавали незнакомым людям, чтобы они, встретившись, могли распознать друг друга, чтобы отличать своих от чужих.

Историки знают о существовании ряда «нестандартных» иероглифов, которым более четырех с половиной тысяч лет. Хотя, конечно, вряд ли возможно с полной уверенностью сказать, представляют ли они попытку скрыть информацию или просто использовались в некоем религиозном ритуале... Зато хорошо известно о табличке из Вавилона, датированной примерно 2500 годом до н. э. На ней есть слова, в которых удален первый согласный и использован целый ряд необычных вариантов обозначения звуков. Исследования показали, что в тексте описан способ изготовления глазури для гончарных изделий, а это приводит нас к выводу: текст был написан купцом или мастером-гончаром, пожелавшим защитить от конкурентов секреты своего мастерства.

По мере распространения письменности и торговли возникли великие империи, которые часто вступали в пограничные конфликты со своими географическими соседями. В результате криптография и безопасная передача информации стали делом особой важности не только для купцов, но и для правительств.

Схема шифрования в самом общем виде определялась следующими элементами: отправитель послания, получатель послания, алгоритм шифрования и определенный ключ, который позволяет отправителю шифровать послание, а получателю

расшифровывать его. Природа и функция ключей изменилась, но пока будем придерживаться этой схемы.

Изначальная цель кодирования — техническое обеспечение связи. Например, текст конвертируется в бинарный (или двоичный) язык (систему счисления, использующую только цифры 0 и 1). После кодирования большая часть этой информации должна быть защищена от любого, кто может ее перехватить. Другими словами, кодированное послание требуется зашифровать. Наконец, законный получатель должен быть способен расшифровать полученное послание.

Кодирование, шифрование и дешифровка — это основные па в «танце информации», который повторяется миллионы раз в секунду, каждую минуту, каждый час каждого дня.

А «музыка», сопровождающая этот танец, — математика.

КОД И ШИФР

Шифровальщики и специалисты по криптографии используют термин «кодировать» в несколько ином смысле. Для них кодирование — это метод написания с использованием кода, который состоит из замены одного слова другим. С другой стороны, использование шифра, или шифрование, включает замену букв или каких-то других отдельных знаков. С течением времени в широком сознании последняя форма сделалась превалирующей, причем в такой степени, что стала синонимом «написания с использованием кода», или «закодированного письма». Однако если мы возьмем более строгое научное определение, то для второго метода правильным термином будет «шифровать» (или «расшифровывать», в случае обратного процесса) послание.

Давайте представим, что мы отправляем защищённое послание «АТАКОВАТЬ». Мы можем сделать это двумя основными путями: заменить слово целиком (кодирование), заменить некоторые или все буквы, которые составляют это слово (шифрование). Простой способ кодирования слова — перевести его на язык, который не знают потенциальные любители подслушать или подсмотреть. В случае шифрования будет достаточно, например, заменить каждую букву другой (то есть стоящей в другой части алфавита). В этом случае необходимо, чтобы получатель знал использовавшуюся процедуру для того, чтобы декодировать или дешифровать текст, или послание потеряет смысл. Если мы уже договорились с получателем, что будем использовать тот или иной способ — переводить на другой язык или заменять каждую букву, — то всё, что от нас



Табличка, найденная на Крите, на которой используется так называемое «линейное письмо Б»

требуется, — это сообщить нашему получателю о выбранном языке или количестве позиций, на которые мы продвинулись в алфавите для замены каждой буквы.

В приводимом примере, если получатель получает зашифрованное послание «ВФВЙРДВФЮ» и знает, что при замене каждой буквы мы сдвигались на две позиции вперёд в алфавите русского языка, то он сможет с лёгкостью повторить процесс, двигаясь в обратном направлении, и успешно расшифровать послание.

Установленное нами разграничение между правилом шифрования (применяемая система) и параметром шифрования [меняющееся указание (инструкция), которое является специфическим для каждого послания или набора посланий] очень полезно, потому что потенциальному шпиону для расшифровки нужно знать и то, и другое.

Таким образом, шпион может знать, что ключ к шифру — это замена каждой буквы другой, находящейся далее в алфавите через определённое количество позиций (x). Однако если он не знает, какому числу соответствует x , то ему потребуется перепробовать все возможные комбинации для каждой буквы алфавита. В этом примере шифр очень простой, и испробовать все возможности — для чего требуется просто усердие — не так уж и сложно.

Эта техника дешифровки называется *методом тотального перебора*. Однако в более сложных случаях такой тип взлома кода (криптоанализ) практически невозможен — по крайней мере, вручную. Более того, на перехват и расшифровку посланий обычно накладываются жёсткие временные ограничения. Ведь информацию нужно получить и понять прежде, чем она станет бесполезной или широко известной другим.

Общее правило шифрования обычно называется *алгоритмом шифрования*, в то время как специфический параметр, используемый для шифрования или кодирования послания, называется *ключом* (в примере шифрования, приведённом выше, ключ — 2. Каждая буква исходного слова заменяется другой, которая расположена через две позиции в алфавите русского языка).

СКОЛЬКО НУЖНО КЛЮЧЕЙ?

Какое минимальное количество ключей необходимо в системе с двумя пользователями? Тремя? Четырьмя? Чтобы два пользователя могли тайно общаться друг с другом, необходим только один ключ. В случае трёх пользователей (А, В и С) необходимы три ключа: один для общения А и В, ещё один для пары А и С, а третий — для пары В и С. Точно так же четырём пользователям потребуется шесть ключей. Таким образом, если обобщить, то для n пользователей потребуется столько ключей, сколько существует комбинаций пар из n , то есть:

$$\frac{n}{2} = \frac{n(n-1)}{2}$$

В результате для относительно небольшой системы из 10000 связанных между собой пользователей потребуется 49 995 000 ключей. Если взять население земного шара, составляющее шесть миллиардов человек, то от количества ключей голова пойдёт кругом:

17 999 999 997 000 000 000.

Очевидно, что для каждого алгоритма шифрования возможно огромное количество ключей, поэтому знание одного алгоритма может быть бесполезным, если мы не имеем представления, какой ключ нужен для расшифровки. Поскольку ключи обычно легче заменить и распространить, кажется логичным для обеспечения безопасности системы шифрования сосредоточиться на том, чтобы хранить ключи в тайне и уделять именно этому максимальное внимание. Такой принцип был установлен в конце XIX столетия голландским лингвистом Огюстом Керкгоффсом фон Ниевенхофом и поэтому известен как принцип Керкгоффса.

ПРИНЦИП КЕРКГОФФА

В соответствии с принципом Керкгоффа, ключ — это основной элемент, обеспечивающий безопасность криптографической системы. До относительно недавнего времени ключи отправителя и получателя во всех возможных криптографических системах должны были быть идентичными или по крайней мере симметричными, то есть их необходимо было использовать и для шифрования, и для расшифровки послания. Поэтому ключ являлся общей тайной отправителя и получателя, и, таким образом, используемая криптографическая система была уязвимой с обеих сторон. Этот тип криптографии, который зависит от ключа, имеющегося как у отправителя, так и у получателя, называется «шифрование закрытым ключом».

Это было свойством всех криптографических систем, изобретённых людьми с начала времён, независимо от используемого алгоритма и сложности. Сделать ключ одним и тем же для получателя и отправителя кажется единственно разумным и полностью соответствующим здравому смыслу.

Как мы видели выше, для классической криптографии требовалось огромное количество ключей. Однако в случае открытой (общедоступной) криптографической системы любым двум пользователям, которые обмениваются посланиями, требуются только четыре ключа: их соответствующие открытые и закрытые ключи. В этом случае количеству пользователей n требуется $2n$ ключей.

В конце концов, разве может один человек кодировать послание в соответствии с одним кодом, а второй расшифровывать его в соответствии с другим и надеяться понять полученный текст? Тысячи лет это считалось полным абсурдом. Однако, как мы увидим ниже, всего пять десятилетий назад абсурд стал абсолютно возможным и теперь используется повсеместно.

В наши дни алгоритмы шифрования, которые используются в большинстве коммуникационных связей, состоят, как правило, из двух ключей: закрытого ключа, который уже



Доктор Огюст Керкгоффс

стал обычным делом, и открытого ключа, который знают все. Механизм передачи состоит в следующем: отправитель получает открытый ключ получателя, которому хочет отправить послание, и использует его для шифровки послания. Получатель использует свой закрытый ключ для расшифровки полученного послания.

Более того, эта система имеет очень важное дополнительное преимущество: ни отправителю, ни получателю не нужно заранее встречаться и договариваться ни о каких используемых ключах, поэтому безопасность системы гораздо выше, чем было возможно ранее.

Эта полностью революционная форма известна как «шифрование открытым ключом» и сегодня составляет основу безопасности в коммуникационных сетях. Фактически, как мы подробно выясним ниже, современная криптография держится на двух китах. Первый — модульная арифметика, а второй — теория чисел и в особенности та ее часть, которая занимается изучением простых чисел.

ТЕЛЕГРАММА ГЕРМАНСКОМУ ПОСЛУ

Криптография — это одна из областей прикладной математики, в которой наиболее очевиден контраст между первоначальной четкостью, лежащей в основе теории, и туманными последствиями ее внедрения и применения на практике.

А ведь иногда от успеха или провала в обеспечении защиты связи и коммуникаций зависит судьба целых наций. Одним из самых впечатляющих примеров того, как криптография изменила курс истории почти сто лет назад, является так называемое «дело о телеграмме Циммермана».

7 мая 1915 года, когда половина Европы была вовлечена в кровавый конфликт Первой мировой войны, немецкая подводная лодка торпедировала трансатлантический пассажирский лайнер «Лузитания», шедший под британским флагом недалеко от берегов Ирландии. Результатом стала одна из наиболее ужасных трагедий в истории: погибли 1198 гражданских лиц, 124 из которых были американцами.

Новость вызвала ярость в общественном мнении Соединенных Штатов Америки, и администрация президента Вудро Вильсона предупредила немецкое правительство, что если подобное повторится, США немедленно вступят в войну на стороне союзников. В дополнение к этому Вильсон потребовал, чтобы немецкие подводные лодки придерживались правил ведения морской войны, установленных Гаагскими конвенциями 1899 и 1907 годов, что ставило под угрозу преимущество немецкого флота, применяющего по отношению к гражданским судам тактику неограниченной подводной войны.

В ноябре 1916 года Германия назначила новым министром иностранных дел Артура Циммермана, имевшего репутацию прекрасного дипломата. Новость была положительно принята прессой США, которая посчитала это назначение благоприятным знаком для американо-германских отношений.

В январе 1917 года, менее чем через два года после трагедии с «Лузитанией», когда война была в самом разгаре, посол Германии в Вашингтоне Иоганн фон Бернсторф получил от Циммермана следующую зашифрованную телеграмму с указанием тайно передать ее коллеге, германскому послу в Мексике, Генриху фон Эккардту:

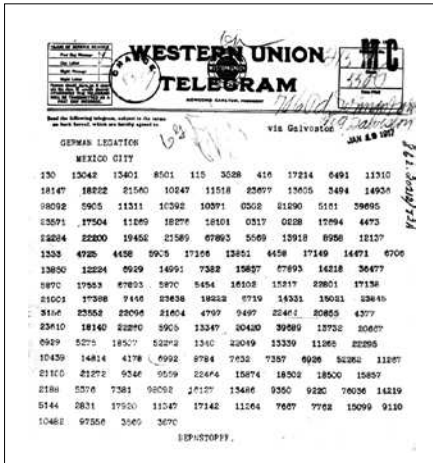
«Мы намерены с первого февраля возобновить неограниченную подводную войну. Тем не менее, следует предпринять все попытки к тому, чтобы США и дальние сохраняли нейтралитет. Однако, если такие попытки окажутся безуспешными, мы предложим Мексике заключить союз на следующих условиях: совместное ведение боевых действий и совместное заключение мира; серьезная финансовая поддержка с нашей стороны и понимание нами стремления Мексики по возвращению утраченных территорий в Техасе, Нью-Мексико и Аризоне. Детали соглашения оставляются на ваше усмотрение [фон Эккардта].

Вы должны довести до сведения Президента [Мексики] о вышеуказанном с соблюдением максимальной степени секретности, как только станет точно известно о начале войны с США, и в дополнение к этому предложить ему по собственной инициативе пригласить Японию для немедленного присоединения к союзу и стать посредником между Японией и нами.

Пожалуйста, обратите внимание Президента на тот факт, что использование наших подводных лодок в полной мере открывает перспективу заставить Англию в течение нескольких месяцев заключить мир».

Если бы содержание телеграммы сделалось достоянием общественности, это наверняка бы привело к началу войны между Германией и США.

Кайзер Вильгельм II, разумеется, понимал, что после того как немецкие подводные лодки начнут действовать с нарушением Гаагских конвенций, война делается неизбежной, однако он надеялся, что к тому времени Великобритания капитулирует, и США попросту не успеют включиться в завершившийся военный конфликт. К тому же активная угроза со стороны Мексики у южных рубежей США заставит американцев сто раз подумать, прежде чем вступить в конфликт, происходящий вдали от их границ.



Телеграмма Циммермана, отправленная послом Германии в Вашингтоне своему коллеге в Мексике, Генриху фон Эккардту

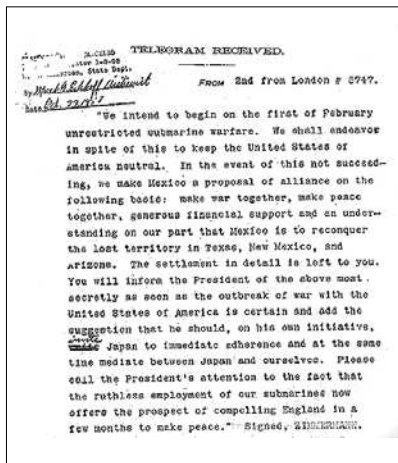
Но Мексике требовалось некоторое время для подготовки своих вооруженных сил. Поэтому было жизненно необходимо, чтобы тайные намерения Германии оставались неизвестными американцам достаточно долго.

Однако у британского правительства были другие планы. Вскоре после начала войны британцы перерезали подводные телеграфные кабели, которые соединяли Германию с западным полушарием напрямую. Таким образом связь должна была осуществляться через другие кабели — те, где британцы могли перехватывать сообщения. США пытались добиться переговоров об окончании войны и поэтому позволяли Германии продолжать передавать дипломатические послания. В результате послание Циммермана было получено немецким посольством в Вашингтоне в целости и сохранности.

Британское правительство отправило перехваченное послание в отдел, занимавшийся дешифровкой и взломом кодов, который назывался «комната № 40».

Немцы использовали свой обычный алгоритм шифрования, которым пользовалось Министерство иностранных дел, а также шифр, известный, как 0075, который эксперты из комнаты № 40 уже частично взломали. Указанный алгоритм включал замену слов (кодирование), а также букв (шифрование). Эта практика была подобна той, которая использовалась

*Та же телеграмма,
но в расшифрованном виде*



немцами в еще одном шифровальном инструменте того времени, шифре ADFGVX, который мы более подробно рассмотрим ниже.

Британцам не потребовалось много времени для расшифровки телеграммы. Правда, они не хотели сразу же доводить ее содержание до американцев. Для этого имелись две причины.

Во-первых, секретная телеграмма была отправлена под дипломатическим прикрытием, которое США обеспечивали немецким посланиям, а британцы эту привилегию напрочь проигнорировали. Во-вторых, если бы телеграмму сделали достоянием общественности, то немецкое правительство сразу же узнало бы, что его коды взломаны, и немедленно изменило систему шифровки.

Поэтому британцы решили сообщить американцам, будто бы раздобыли содержание телеграммы, полученной фон Эккардтом, чтобы таким образом убедить немцев, что телеграмма перехвачена уже расшифрованной, в Мексике.

В конце февраля правительство Вильсона передало содержание телеграммы прессе. Некоторые представители прессы, в частности газеты, принадлежащие издательскому дому «Херст» (Hearst), который был настроен против возможной войны и прогермански, вначале отнеслись к ней весьма скептически. Однако к середине марта Циммерман публично признал

авторство противоречивого послания. Чуть более двух недель спустя, 6 апреля 1917 года, Конгресс США объявил войну Германии.

Это решение имело далеко идущие последствия для Европы и мира...

В заключение отметим, что, хотя телеграмма Циммермана и стала чрезвычайным событием своего времени, но это всего лишь одна историческая страничка, в которой важную роль сыграла криптография.

На протяжении этой книги мы увидим немало других примеров, разбросанных по столетиям и культурам.

Тем не менее, мы почти наверняка можем утверждать, что далеко не все знаем о самых важных исторических событиях.

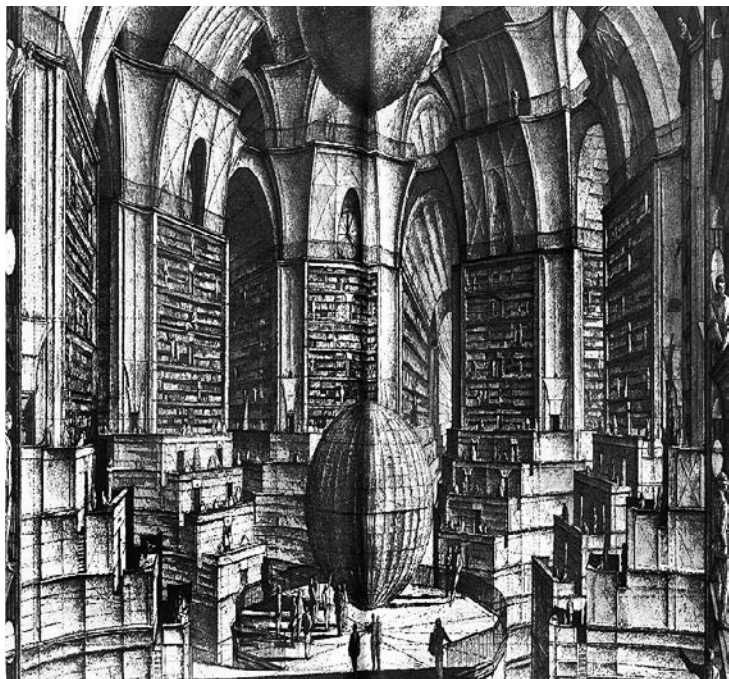
**Ведь благодаря своей природе история
криптографии — это история тайн.**



ПОСЛАНИЕ ИЗ ВАВИЛОНА

Аргентинский писатель Хорхе Луис Борхес представил в коротком рассказе «Вавилонская библиотека» такое огромное книгохранилище, что на полках стояли все возможные книги: романы, поэмы и диссертации, а также опровержения этих диссертаций, и опровержения опровержений, и так далее, до бесконечности.

Криптоаналитик, путём проб и ошибок пытающийся расшифровать сообщение, зашифрованное с помощью «одноразового блокнота», столкнётся с подобной же ситуацией. Поскольку шифр полностью произволен, возможная расшифровка будет содержать все возможные тексты той же длины: истинное послание и короткое опровержение послания, и то же самое послание, в котором все правильные существительные заменены другими существительными той же длины, и так далее до бесконечности...



ГЛАВА 2.

КРИПТОГРАФИЯ ОТ АНТИЧНЫХ ВРЕМЕН...

«История кодов и шифров — это многовековая история поединка между создателями шифров и теми, кто их взламывает... Трагическая казнь Марии Стюарт, королевы Шотландии, явилась драматической иллюстрацией слабостей одноалфавитной замены; очевидно, что в поединке между криптографами и криптоаналитиками последние одержали верх. Любой, кто отправлял зашифрованное сообщение, должен был отдавать себе отчет, что опытный дешифровальщик противника может перехватить и раскрыть самые ценные секреты».

Сингх Саймон. «Книга шифров»

СКРЫТЫЕ ПОСЛАНИЯ

Древнегреческий учёный Геродот упоминает в своей знаменитой «Истории», посвящённой описанию греко-персидских войн в V веке до н. э., два любопытных случая применения стеганографии, которые свидетельствуют о большой находчивости людей того времени.

В первом примере, который содержится в «Талии», третьей книге «Истории», Гистией, тиран города Милет, приказал одному человеку побрить голову. Затем он написал на бритой голове послание и стал ждать, пока у мужчины снова вырастут волосы. После того как они отросли, посыльного отправили в лагерь Аристагора. Добравшись туда, посыльный объяснил суть дела Аристагору, и волосы снова сбрили, открыв, таким образом, сообщение, которого здесь давно ждали.

Второй пример, если это, конечно, происходило в действительности, имеет гораздо большую историческую важность, поскольку позволил Демарату, царю Спарты, находящемуся в ссылке в Персии, предупредить своих соотечественников о грядущем вторжении персидского царя Ксеркса. Эту историю Геродот рассказывает в «Полигимнии», седьмой книге «Истории»:

«Демарат не мог открыто предупредить их, поэтому ему пришла в голову такая мысль: он взял пару табличек [для письма], соскоблил с них воск и написал о планах царя на деревянной поверхности табличек. Затем он снова покрыл их расплавленным воском и таким образом скрыл сообщение. В результате казавшиеся пустыми таблички не вызывали никаких подозрений у стражников в дороге.

Когда таблички наконец оказались в Лакедемонe (Спарта), тамошние жители не могли понять,

в чём тут секрет, пока, насколько я понимаю, Горго [...] не предложила соскоблить воск с табличек, потому что под ним — как она подсказала — найдут написанное на дереве послание».

Распространенное стеганографическое средство, которое выдержало испытание временем, — это симпатические (невидимые) чернила. Их применение описано в тысячах рассказов и фильмов. Используемые материалы — лимонный сок, сок растений и даже человеческая моча — обычно органического происхождения и с высоким содержанием углерода. Поэтому высохшие чернила имеют склонность к потемнению, когда оказываются под воздействием умеренно высоких температур, как, например, жар от пламени свечи. Полезность стеганографии нет смысла оспаривать, хотя она делается совершенно непригодной, когда речь идет о больших количествах посланий. Более того, если ее использовать напрямую, без дополнительных ухищрений, у нее обнаруживается существенный недостаток: когда послание однажды все-таки перехватят, содержание его сразу же станет известным. По этой причине стеганография в основном используется как дополнение к криптографии, как средство усиления безопасности сверхсекретных передач.

Во время холодной войны в полных драматизма шпионских триллерах герои часто отправляли послания при помощи средства, на котором буквы оказывались слишком мелкими для чтения невооружённым глазом, — микрофильма. Техника родилась за несколько лет до начала холодной войны, в годы Второй мировой, когда немецкие агенты использовали стеганографическую технику, известную как микрофотоснимок, то есть снимок с очень большим уменьшением. Он состоял из фотографии короткого текста, которая сводилась до размера точки, которую затем включали в виде одного из многочисленных символов в безобидный текст.

Вооруженные конфликты являлись мощным стимулом и побудительным мотивом для развития безопасности информационных сообщений. Поэтому неудивительно, что такие воинственные люди, как спартанцы (если верить Геродоту, они являлись мастерами стеганографии), также стали первопроходцами и в развитии криптографии.

СПАРТА ПРОТИВ АФИН

В период вооружённых конфликтов между спартанцами и афинянами за контроль над Пелопоннесом часто использовались длинные полоски пергамента, обёрнутые вокруг цилиндрической палки, которую называли скитала. Послание писали на пергаментной полосе, которой оборачивали цилиндр. Когда пергаментную полосу разматывали, послание становилось неподдающимся прочтению. Даже если противник знал, какая использовалась техника шифрования (то есть алгоритм), а точные размеры скиталы не были известны, то любой, кто перехватывал сообщение, сталкивался с огромными трудностями при его расшифровке. Толщина и длина скиталы по сути являлись ключом шифровальной системы.

Скитала позволяет полностью менять порядок букв в послании. Чтобы получить представление об этом методе, который называется транспозицией, рассмотрим простой пример с перестановкой всего трёх букв: А, О и Р. Не требуется никаких расчётов, чтобы выяснить, что эти буквы могут быть представлены шестью возможными способами: АОР, АРО; ОАР, ОРА; РОА, РАО.

А = СООБЩЕНИЕ С ПОМОЩЬЮ СКИТАЛЫ
Б = СИМКЕООИО ЩТЬБЬЩЦ ЮЛЕП ЫНОС

*А — послание, которое предстоит передать,
Б — но если полосу бумаги развернуть, то получается
полная чушь, набор ничего не значащих букв*

Если говорить абстрактно, то процесс следующий: после того как одна из трёх возможных букв ставится на первое место, позволяя три различных варианта расстановки, мы остаёмся с двумя буквами, которые, в свою очередь, можно расставить двумя различными способами, чтобы получить новый общий результат: $3 \times 2 = 6$ вариантов расстановки. В случае более длинного послания, например, из 10 букв, количество возможных расстановок будет равняться:

$$10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1.$$

Такая операция называется факториал, выражается математическим обозначением $10!$ и даёт общий результат **3628800**. Если говорить в общем, о количестве букв n , то есть $n!$ различных способов их перестановки. Таким образом, послание, которое состоит всего из 40 букв, может дать такое количество возможных способов перестановки букв, что расшифровать его вручную становится практически невозможно.

Неужели мы нашли идеальный криптографический метод?..

Увы, не совсем. По сути алгоритм перестановки (транспозиции) наугад даёт наиболее высокий уровень безопасности, но что будет представлять собой ключ, который позволит расшифровать послание?!

Случайность процесса — это и его сила, и слабость.

ОТЕЦ АНАЛИТИЧЕСКОЙ КРИПТОГРАФИИ

Евклид (325–265 гг. до н. э.) — древнегреческий математик, автор первого из дошедших до нас трактатов по математике, считается отцом аналитической криптографии. Основная его работа «Начала» (в латинизированной форме — «Элементы») содержит изложение планиметрии, стереометрии и ряда вопросов теории чисел; в ней он подвёл итог предшествующему развитию математики и создал фундамент дальнейшего развития этой науки. Хотя в общественном сознании его работы больше всего ассоциируются с геометрией пространства, она связана и с арифметическими операциями с конечным количеством чисел, или модулями, и представляет собой один из основных трудов, изучаемых современной криптографией. Арабские учёные давно знали и восхищались работами Евклида, но первое издание его работ в Европе появилось в Венеции только в 1482 году. Думается, нельзя считать совпадением то, что и арабы, и венецианцы были великими мастерами криптографии.



ШИФР ПОЛИБИЯ

Один из первых известных алгоритмов подстановки — это так называемый шифр Полибия, названный в честь греческого историка Полибия (около 203 — около 120 гг. до н. э.), который оставил нам его описание. Этот шифр — один из старейших среди тех, о которых мы имеем подробную информацию. Он основан на выборе пяти букв алфавита, чтобы те служили в виде «шапки» для столбцов и стояли первыми в строках таблицы пять на пять, затем ячейки таблицы заполняются буквами алфавита. Шифр сконструирован так, что каждая буква соответствует паре букв, в зависимости от строки и столбца таблицы, по которым они и определяются. Изначально использовался греческий алфавит, который включает 24 буквы, поэтому I и J из английского алфавита, состоящего из 26 букв, обычно объединяются в одной ячейке. Таблица заполняется в порядке, о котором договариваются отправитель и получатель. Обратите внимание, что в шифровальном алфавите должно быть 25 букв (5×5). Шифровальный алфавит также может быть размещён в таблице, где «шапкой» служат цифровые значения (например, цифры 1, 2, 3, 4, и 5). В таком случае получится следующая таблица:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I-J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Давайте рассмотрим пример двух версий использования шифра Полибия. Исходное послание — BLANKS. Из таблицы получаем:

В будет заменяться парой цифр 12

L будет заменяться парой 31

A будет заменяться парой 11

N будет заменяться парой 33

K будет заменяться парой 25

S будет заменяться парой 43

Получается зашифрованное послание **123111332543**.

ШИФР ЦЕЗАРЯ

*«Veni, vidi, vici»
(Пришел, увидел, победил).*

Юлий Цезарь

Шифры, в которых используется замена (то есть шифры замены или шифры подстановки), развивались параллельно с шифрами, в которых используется перестановка (транспозиционными шифрами). В отличие от транспозиции, при строгой замене одна буква заменяется другой или символом любого типа. В отличие от транспозиции, при замене необязательно пользоваться только буквами, из которых состоит послание. При транспозиции буква меняет позицию, но сохраняет свою роль. Та же самая буква имеет то же самое значение в исходном и в зашифрованном послании. При замене буква остаётся на своей позиции, но меняет свою роль (та же самая буква или символ имеет одно значение в исходном послании и совсем другое — в зашифрованном послании).

В первом веке до н. э. появился один шифр, в котором используется принцип замены. Он известен под названием «шифр Цезаря», поскольку Юлий Цезарь был одним из самых известных людей, которые его использовали. Шифр Цезаря хорошо изучен, он является исключительно полезным, потому что иллюстрирует принципы модульной арифметики, одной из математических основ написания закодированных посланий.

Шифр Цезаря основан на принципе, когда каждый символ в начальном тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 3 А была бы заменена на Г, Б станет Д, и так далее.

По словам великого историка Светония, автора «Жизни двенадцати Цезарей», Юлий Цезарь кодировал свои личные письма при помощи алгоритма замены следующего типа: каждая буква оригинала заменялась на другую, которая стояла в алфавите через три позиции после неё: А заменялась на Д, В

на E, и т. д. W заменялась на Z, и, в конечном счёте, X, Y и Z заменяются на A, B и C.

Кодирование и декодирование послания, зашифрованного таким методом, можно провести с использованием простой таблицы:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Верхний ряд букв представляет собой изначальный алфавит, а нижний ряд — трансформированный, полученный в результате использования шифра Цезаря.

Когда два алфавита, исходный и шифровальный, располагаются таким образом, то шифрование послания — это просто вопрос замены букв одного буквами другого. Ключ к шифру называется по букве, которая соответствует зашифрованной A (первой букве исходного алфавита). В данном случае это D. Классическое выражение AVE CAESAR (Здравствуй, Цезарь) будет зашифровано как DYN FDHVDU. И наоборот, зашифрованное послание WUHH после расшифровки даёт TREE (дерево).

АЛЬ-КИНДИ: ВЗЛОМ ШИФРА

Священная книга мусульман Коран состоит из ста сорока двух глав (сур), каждая из которых излагает откровения пророка Магомета. Эти откровения были записаны при жизни пророка его сподвижниками и в дальнейшем собраны первым халифом Абу Бакром. Умар и Усман, второй и третий халифы соответственно, завершили его начинание.

Фрагментарная природа изначальных текстов привела к рождению ветви теологии, занимающейся датированием различных откровений. Среди других техник датирования учёные, занимающиеся изучением Корана, вычисляли частоту появления определённых слов, которые считаются возникшими только в течение периода написания. Если в откровении встречалось достаточно таких новых слов, то было разумно сделать вывод, что это относительно позднее откровение.

Эта инициатива оказалась первым специфическим инструментом криптоанализа, или дешифровки, из когда-либо изобретённых. Он называется «частотным анализом».

Первым человеком, который оставил письменное свидетельство об этой революционной технике, стал философ Абу Юсуф Якуб ибн Исхак ибн Саббах аль-Кинди, употребительное сокращение имени — Аль-Кинди (около 801–873) — арабский философ, математик, теоретик музыки, астроном. Родился, по разным данным, в Куфе или в Басре, детство провёл в Басре, работал в Доме мудрости в Багдаде. Являлся фаворитом халифов ал-Мамуна (813–833)



*Рукопись Корана,
Египет, возможно,
Фатимиды, X век;
выполнена восточным
куфическим почерком
коричневыми
чернилами
на пергаменте*

и ал-Мутасима (833–842), которые были покровителями представителей раннего калама — мутазилитов.

В 1987 году экземпляра трактата Аль-Кинди, озаглавленного «О расшифровке криптографических посланий», всплыл в архиве в Стамбуле. В нём содержится очень краткое изложение первопрородческой техники.



«Один из способов расшифровки зашифрованного послания, если мы знаем, на каком языке оно написано, — это найти обычный незашифрованный текст, написанный на том же языке, причём достаточно длинный, потом сосчитать, сколько раз в нём появляется каждая буква. Букву, которая встречается наиболее часто, мы называем «первой»; букву, частота появления которой следует за первой, мы называем «второй»... и так далее, пока мы не охватим все встречающиеся в тексте буквы. Затем мы рассматриваем текст, расшифровкой которого занимаемся, и классифицируем встречающиеся в нём символы таким же образом. Находим символ, который встречается наиболее часто, и заменяем его «первой» буквой из нашего текста, делаем то же самое со «второй» и так далее, пока не охватим все символы в криптограмме, которую расшифровываем».

Выше в том же тексте Аль-Кинди упоминает, что при использовании способа шифрования, где применяется замена, каждая буква исходного послания «остаётся на своём месте, но меняет свою роль», и именно это постоянство, «сохранение позиции», делает возможным частотный криптоанализ. Гениальность Аль-Кинди изменила равновесие в среде шифровальщиков и дешифровщиков, и чаша весов, по крайней мере, на какое-то время, склонилась в пользу тех, кто подслушивает и подсматривает.

ШИФРОВАНИЕ СЛОВА БОЖЬЕГО

Средневековые криптоаналитики присутствуют в Ветхом Завете, и они несут несколько фрагментов священных текстов с помощью шифра замены, который шифр состоит из замененных любой буквой находится на том же расстоянии от п находится от его начала. Например, в языке А меняется на Z, В на Y и т. д. В Ветхом Завете замены производятся буквами ского языка. Таким образом в Книге Babel (Вавилон) в зашифрованном виде

Надежность и простота алгоритма для замены ключевого слова, на многие столетия сделали эту систему предпочитаемой системой шифрования.



Страница из еврейской библии (ранее XVIII века)

В то время, по общему мнению, шифровальщики побеждали дешифровщиков.

ЧАСТОТНЫЙ АНАЛИЗ НА ПРАКТИКЕ

Если идти от наиболее часто встречающейся к наименее часто встречающейся, то буквы в английских текстах используются следующим образом:

E T A O I N S H R D L C U M W F G Y P B V K J X Q Z.

Процент появления каждой буквы представлен в следующей частотной таблице:

A 8,17%	H 6,09%	O 7,51%	V 0,98%
B 1,49%	I 6,97%	P 1,93%	W 2,36%
C 2,78%	J 0,15%	Q 0,10%	X 0,15%
D 4,25%	K 0,77%	R 5,99%	Y 1,97%
E 12,70%	L 4,03%	S 6,33%	Z 0,07%
F 2,29%	M 2,41%	T 9,06%	G 2,02%
N 6,75%	U 2,76%		

Если послание зашифровано при помощи алгоритма замены, аналогичного тому, что обсуждался выше, оно может быть расшифровано в соответствии с частотой использования букв в исходном послании. Достаточно посчитать, сколько раз встречается каждая из букв в зашифрованном тексте, и сравнить их с частотной таблицей для языка, на котором писалось послание. Таким образом, если текст написан на английском языке и наиболее часто встречающаяся в зашифрованном тексте буква — это J, то она, вероятнее всего, соответствует букве E. Если второй по частоте появления является буква Z, то те же рассуждения приводят нас к выводу, что ей наиболее вероятно соответствует буква T. Процесс повторяется для всех букв зашифрованного текста.

Дешифровка с помощью частотного анализа — это история, полная драматизма, привлекавшая внимание немалого количества писателей. Возможно, самым известным произведением, основанным на крипто-анализе послания, является «Золотой жук» Эдгара По. Другие авторы, такие как Жюль Верн и Артур Конан Дойл, использовали подобные приёмы, чтобы добавить напряжения в сюжетные линии.

РУКОВОДСТВО ДЛЯ ЮНЫХ ЛЕДИ

Камасутра — это пространное руководство, которое посвящено, среди прочего, тому, что необходимо знать каждой женщине, чтобы быть хорошей женой, созданное около IV в. до н. э. брамином Малланага Ватсьяна. В нем рекомендуется до 64 различных навыков, в том числе по музыке, кулинарии и шахматах. Особый интерес для нас имеет навык под номером 45, потому что он посвящен искусству тайного письма, или «млекчита-викалпа». Древний мудрец рекомендует несколько методов, в том числе такой: разделить алфавит на две части и распределить буквы по парам случайным образом. В этой системе каждое соответствие пар представляет собой ключ. Например, один из вариантов для латиницы может быть следующим:

A B C D E F G M L K J I H
T U V W X Y Z S R Q P O N

Чтобы написать тайное послание, требуется всего лишь заменять каждую букву A в изначальном тексте на T, G на Z, H на N и т. д., и наоборот при расшифровке.

Требовался способ шифрования, который дал бы ключи, являющиеся одновременно простыми, лёгкими для запоминания и передачи, но в то же время не требующие жертв, связанных с безопасностью. В результате начался поиск идеального алгоритма, и первых успехов в этом достигли римские императоры.

ШИФРОВКА ИЗ "ЗОЛОТОГО ЖУКА"

Уильям Легран, герой рассказа «Золотой жук» (1843) Эдгара Алана По, нашел место, где зарыт сундук с сокровищами, после расшифровки надписи на клочке пергамента. Легран использовал статистический метод, основанный на частоте появления букв, которые составляют текст на английском языке. Зашифрованное послание было следующим:

53‡††305)) 6*;4826) 4‡.) 4‡);806*;48†8¶ 60)) 85;1‡
 (‡;‡*8†83 (88) 5*†;46 (;88*96*‡;8) *‡ (;485);5*†2.*‡
 (;4956*2 (5*_4) 8¶ 8*;4069285);) 6†8) 4‡‡;1
 (‡9;48081;8:8‡ 1;48†85;4) 485†528806*81 (‡9;48;
 (88;4 (‡‡34;48) 4‡;161‡;188;‡‡;

Легран начал с предположения, что исходный текст был написан на английском языке. В английском чаще всего встречается буква «е». Затем он составил список букв, основываясь на частоте использования, от наиболее часто встречающейся к наименее часто встречающейся: а, о, l, d, h, n, r, s, t, ц, у, с, f, g, i, m, w, b, k, p, q, x, z.

Герой составил на основании криптограммы таблицу. В первом ряду — знаки зашифрованного послания, во втором ряду — частота их появления.

8	;	4	‡)	*	5	6	(†	1	0	9	2	:	3	?	¶	_
33	26	19	16	16	13	12	11	10	8	8	6	5	5	4	4	3	2	1

Поэтому «8» — скорее всего буква «е». Затем герой нашел три знака, которые могли бы обозначать the, определенный артикль, который также очень часто встречается в английском языке, что позволило Леграну перевести знаки ";", «4» и «8». Он догадался, что означает сочетание «; (88», поскольку знал, что обозначают три знака из четырех. И, получив сочетание «t(е», он сделал вывод о том, какую букву обозначает «(». Это может быть только «г», а в результате получилось слово «tree» (дерево). Наконец, используя далее аналогичную криптоаналитическую технику и проявив немалое терпение, Легран составил следующий частичный шифровальный алфавит:

5	†	8	3	4	6	*	‡	(;	?
a	d	e	g	h	i	n	o	r	t	u

Этого достаточно для расшифровки послания:

«A good glass in the bishop`s hostel in the devil`s seat forty-one degrees and thirteen minutes northeast and by north main branch seventh limb east side shoot from the left eye of the death`s-head a bee line from the tree through the shot fifty feet out».

Русский перевод:

«Хорошее стекло в трактире епископа на четвертом стуле сорок один градус тринадцать минут север-северо-восток главный сук седьмая ветвь восточная сторона стреляй из левого глаза мертвой головы прямая от дерева через выстрел на пятьдесят футов».

Очевидно, что частотный метод не всегда может применяться столь непосредственным образом. Частоты в таблице, представленной выше, — это только средние показатели.

К примеру, в коротких текстах типа «Visit the zoo kiosk for quiz tickets» относительная частота появления букв очень сильно отличается от языковых характеристик в целом.

Поэтому для текстов, включающих менее 100 знаков, этот простой анализ используется крайне редко.

Однако частотный анализ не ограничивается изучением букв самих по себе. Хотя мы соглашаемся с тем, что маловероятным является наиболее частое появление буквы Е в коротком зашифрованном тексте, мы с большей уверенностью можем считать, что пять наиболее часто встречающихся букв — это, вероятнее всего, А, Е, I, О и Т (пусть даже не знаем, какая из них которой соответствует). А и I никогда не встречаются в паре в английском языке, в то время, как другие буквы могут. Более того, также вероятно, что каким бы коротким ни был текст, гласные имеют тенденцию появляться перед и после скоплений других букв, в то время как согласные имеют тенденцию группироваться с гласными или с небольшим числом согласных. Таким образом, мы, возможно, сумеем отделить Т от А, Е, I и О.

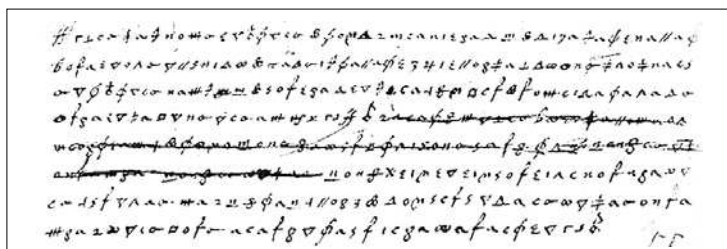
По ходу успешной расшифровки писем станут появляться слова, в которых нам нужно расшифровать всего один или два знака, что позволит строить гипотезы о том, что это за буквы. Скорость дешифровки увеличивается по мере дешифровки большего количества писем.

ШРИФТ МАРИИ СТЮАРТ

8 февраля 1587 года шотландская королева Мария Стюарт была обезглавлена в замке Фотерингей после того, как её признали виновной в государственной измене.

Расследование, которое привело к столь жестокому приговору, показало вне всяких сомнений: Мария и в самом деле действовала в сговоре с группой аристократов-католиков, которую возглавлял молодой Энтони Бабингтон. Заговорщики планировали покушение на английскую королеву Елизавету I с целью возвести Марию на трон католического государства, включающего и Англию, и Шотландию.

Решающие доказательства представила служба контрразведки Елизаветы, которую возглавлял лорд Уолсингем. Доказательства включали целый ряд писем, которыми обменивались Мария и Бабингтон. Из них было совершенно ясно, что молодая шотландская королева знала о планирующемся убийстве и одобряла его. Упомянутые письма были зашифрованы с помощью алгоритма, который соединял в себе шифры и коды. Иными словами, буквы оригинала не только заменялись другими знаками, но использовались также и уникальные символы, которые обозначали определённые, часто встречающиеся слова.



Фрагмент одного из писем шотландской королевы Марии Стюарт
заговорщику Энтони Бабингтону — тех писем, которые в конце
концов приведут к смертному приговору

Шифровальный алфавит Марии Стюарт приводится ниже:

a b c d e f g h i k l m n o p q r s t u x y z
 0 1 2 3 4 5 6 7 8 9

Если не считать того, что в шифровальном алфавите Марии Стюарт вместо букв используются символы, он ничем не отличается от других алфавитов, которыми на протяжении столетий пользовались криптографы по всему миру.

Конспираторы были уверены в надёжности шифра, но, к сожалению для Марии, лучший криптоаналитик Елизаветы, Томас Фелиппес, был специалистом по частотному анализу и без особого труда сумел расшифровать письма. Таким образом был разоблачён так называемый заговор Бабингтона, а правительства и секретные агенты по всей Европе получили сигнал: традиционный алгорит подстановки (замены) больше не является безопасным и надёжным.

Криптографы оказались беспомощны перед силой нового дешифровочного инструмента.

ПРОРЫВ АЛЬБЕРТИ

Однако решение проблемы, которую поставил перед шифровщиками частотный анализ, было найдено ещё за век с лишним до того, как Мария Стюарт сложила голову на плахе.

Создателем нового шифра стал учёный эпохи Возрождения Леон Баттиста Альберти, обладавший самыми разнообразными талантами. Альберти первым изложил математические основы учения о перспективе. Также он внес существенный вклад в развитие криптографии, предложив в книге 1466 г. «Трактат о шифрах» идею многоалфавитного шифра, который состоял из добавления второго шифровального алфавита к первому, как показано в следующей таблице:

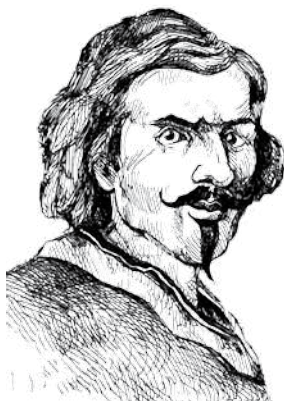
(1)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
(2)	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
(3)	M	N	B	V	C	X	Z	L	K	J	H	G	F	D	S	A	P	O	I	U	Y	T	R	E	W	Q

Строка (1) — обычный алфавит.

Строка (2) — первый алфавит, который используется для шифрования.

Строка (3) — второй алфавит, используемый для шифрования.

Можно попеременно использовать два шифровальных алфавита. Например, в случае слова SHEEP (овца) для первой буквы используется первый шифровальный алфавит (V),



Леон Баттиста Альберти

для второй буквы — второй шифровальный алфавит (L) и т. д. В нашем примере SHEEP будет зашифрована как VLHCS.

Преимущество полиалфавитного шифровального алгоритма, в сравнении с предыдущими, очевидно: буквы двойного ЕЕ из исходного текста шифруются двумя различными путями, Н и С. Чтобы ещё больше запутать любого криптоаналитика, которому предстоит работа с зашифрованным текстом, одна и та же буква в шифровке представляет две различные буквы в исходном тексте. Поэтому частотный анализ теряет большую часть своей полезности.

Альберти никогда не представлял свою идею официально, а в дальнейшем такой шифр был разработан примерно в одно и то же время, но независимо друг от друга, двумя академиками, немцем Иоганном Тритемием и французом Блезом де Виженером.

ДИСК АЛЬБЕРТИ

Практический способ применения полиалфавитного шифра — это использование приспособления, известного как шифровальный диск Альберти. Это портативное шифровальное приспособление состоит из двух концентрических дисков, один из них зафиксирован и на нем выгравирован обычный алфавит. Второй диск вращается, на нем алфавит повторен.

Отправитель может, вращая диск, совместить обычный алфавит с таким количеством шифровальных алфавитов, сколько раз можно сдвинуть подвижный круг, меняя положение букв. Максимальное число равняется количеству букв используемого алфавита.

Шифр, получаемый с помощью диска Альберти, очень стойк к частотному анализу. Для расшифровки послания получатель должен только повернуть диск такое же количество раз, как сделал отправитель. Надежность шифра, как и всегда, зависит от сохранения в тайне кодов, то есть установки алфавита на вращающемся диске и количества сдвигов. Диск Альберти с единственным подвижным кругом, на который нанесен обычный алфавит, позволяет использовать шифр Цезаря при каждом вращении. Подобные приспособления использовались не в таком уж отдаленном прошлом, например, во время гражданской войны в Америке (1861–1865).



*Диск Альберти,
которым пользовались
сторонники Конфедерации
во время гражданской войны
в Америке*

КВАДРАТ ВИЖЕНЕРА

В шифре Цезаря используется моноалфавитная система: там присутствует один шифровальный алфавит, который соответствует алфавиту исходного текста таким образом, что зашифрованная буква всегда соответствует одной и той же букве исходного текста (в классическом шифре Цезаря D — это всегда A, E — всегда B и т.д.). С другой стороны, в полиалфавитном шифре определённая буква в послании может заменяться таким количеством букв, сколько используется шифровальных алфавитов. Для шифрования текста, по мере того как шифровальщик переходит от одной буквы исходного текста к другой, каждый раз используется другой шифровальный алфавит.

Первая и самая известная система полиалфавитного шифра известна под названием «квадрат Виженера».

Виженер — французский дипломат, криптограф и алхимик. В 1549 году французское правительство отправило его с дипломатической миссией в Рим, где он заинтересовался криптографией. В 1585 году Виженер написал основополагающую работу «Трактат о цифрах и тайнописи», в которой описывается система шифрования, получившая его имя в 1854 году.



Блез де Виженер

Квадрат Виженера

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
2	B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
3	C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
4	D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
5	E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
6	F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
7	G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
8	H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
9	I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
10	J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
11	K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
12	L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
13	M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
14	N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
15	O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
16	P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
17	Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
18	R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
19	S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
20	T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
21	U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
22	V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
23	W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
24	X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
25	Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
26	Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Таблица алфавитов Виженера состояла из алфавита исходного текста из n букв, под которыми появлялись шифровальные алфавиты в количестве n .

Перемещение (сдвиг) проходило циклично, по одной букве, влево, в сравнении с предыдущим действием. Иными словами, это была квадратная матрица из 26 строк и 26 столбцов. Она представлена в таблице. Обратите внимание на симметричность в соответствии букв. Пару составляют $(A, R) = (R, A)$, то же самое отношение применимо ко всем буквам. Не трудно заметить, что квадрат Виженера состоит из алфавита исходного текста из n букв, каждая из которых трансформируется в соответствии с увеличивающимися параметрами. Таким образом, первый шифровальный алфавит служит для применения шифра Цезаря с параметрами $a = 1$ и $b = 2$; второй эквивалентен шифру Цезаря с $b = 3$ и т. д.

Ключ к квадрату Виженера состоит из информации о том, какие буквы послания зашифрованы и на сколько строк вниз мы спускаемся, чтобы найти соответствующую букву для шифрования. Самый простой ключ состоит в сдвиге на одну строку вниз для каждой буквы исходного послания. Таким образом, классическая фраза VENI VIDI VICI («Пришёл, увидел, победил») будет шифроваться следующим образом.

Чтобы зашифровать первую букву V, находим соответствующую букву в строке 2: W. Чтобы зашифровать E, находим соответствующую букву в строке 3: G. Чтобы зашифровать N, находим соответствующую букву в строке 4: Q.

I (строка 5): M
V (строка 6): A
I (строка 7): O
D (строка 8): K
I (строка 9): Q
V (строка 10): E
I (строка 11): S
C (строка 12): N
I (строка 13): U

Из исходной фразы после шифрования получается: WGQM AOKQESNU

Сразу же видно, что повторяющиеся в исходном послании буквы исчезают.

Однако каждый криптограф желает изобрести такой код, который легко запоминать, сообщать партнёру по переписке и периодически обновлять.

Чтобы создать меньшие, более лёгкие для использования квадраты Виженера, использовались ключевые слова с тем же или меньшим количеством букв, что и расшифровываемое послание. Ключевое слово формировало первые буквы в каждой строке (см. следующую таблицу), за каждой из них следовал основной алфавит (так, как он представлен в полном квадрате). Затем слово писали под исходным текстом, повторяя столько раз, сколько требовалось. Затем буква ключевого слова под каждым из знаков исходного текста направляла криптографа к строке в квадрате, из которой следовало взять букву для шифрования.

Квадрат Виженера со строками, определяемыми ключевым словом JACKSON

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m

Например, мы хотим зашифровать послание BUY MILK TODAY (купи молока сегодня) при помощи ключевого слова JACKSON:

Исходное послание	B	U	Y	M	I	L	K	T	O	D	A	Y
Ключевое слово	J	A	C	K	S	O	N	J	A	C	K	S
Зашифрованное послание	k	u	a	w	a	z	x	c	o	f	k	q

Получается зашифрованное послание: **kuawazxcofkq.**

Как и в случае всех классических шифровальных систем, зашифрованное послание, которое было обработано с помощью квадрата Виженера, симметрично исходному тексту. Например, мы хотим расшифровать послание WZPKGIMQHQ при помощи ключевого слова WINDY:

Исходное послание	?	?	?	?	?	?	?	?	?
Ключевое слово	W	I	N	D	Y	W	I	N	D
Зашифрованное послание	W	Z	P	K	G	I	M	Q	H

Давайте посмотрим на первый столбец. Мы хотим найти неизвестное «?» при условии, что $(?, W) = W$. Для этого мы ищем строку W в квадрате Виженера, представленном в таблице, приведённой выше, пока не найдём W, и смотрим, какому столбцу соответствует найденная буква. Ответ — A. Затем мы ищем букву «?», подтверждающую, что $(?, I) = Z$, то есть строку I, а в ней букву Z, и получаем R, и так далее.

Получается исходное послание: ARCHIMEDES.

Историческая важность квадрата Виженера, которую он делит с другими полиалфавитными шифрами, как, например, шифром Гронсфельда (который был разработан примерно в то же время и подробно описывается ниже), — это противодействие частотному анализу.

Если одну и ту же букву можно зашифровать несколькими способами и в дальнейшем послание можно без труда расшифровать, как в таком случае проводить криптоанализ? С этой системой шифрования никто не мог справиться в течение двух с половиной столетий, пока в ее расшифровке не преуспел англичанин Чарльз Бэббидж в 1854 году.

В Европе XVII века широко распространился менее надежный, но более простой метод шифрования — шифр Гронсфельда. Этот шифр придумал голландец Джозт Максимилиан Бронкхорст, граф Гронсфельд. Это полиалфавитный шифр, аналогичный квадрату Виженера, но менее трудный.

ШИФР ГРОНСФЕЛЬДА

Чтобы зашифровать послание, начинаем со следующей таблицы:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
1	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
2	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
3	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
4	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
5	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
6	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
7	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
8	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
9	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

Следующий шаг — выбрать наугад число от 0 до 9, чтобы заменить каждую букву в послании, которое мы хотим зашифровать. Если исходное послание — MATHEMATICAL, то мы выбираем наугад 12 чисел, например, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2. Этот ряд чисел будет служить ключом к шифру. Затем мы заменяем каждую букву послания буквой, которая соответствует ей в выбранной строке в таблице.

Послание	M	A	T	H	E	M	A	T	I	C	A	L
Ключ	1	2	3	4	5	6	7	8	9	0	1	2
Зашифрованное послание	P	F	A	S	R	D	T	O	F	E	D	Q

М шифруется как Р (берётся буква из строки 1 в столбце «М») и так далее. Всё послание шифруется как

«PFASRDTOKFDQ».

Буква А в послании шифруется как F, Т и D.

Как и обычно в случае полиалфавитных шрифтов, эта система шифрования отличается стойкостью к методу тотального перебора и частотному анализу. Количество ключей для шифра Гронсфельда в случае алфавита из 26 букв составляет

$$26! \times 10 = 4,03 \times 10^{26} \text{ ключей.}$$

«ЧЕРНЫЕ КАБИНЕТЫ»

Хотя процесс и занял почти восемь столетий, полиалфавитные шифры типа квадрата Виженера наконец одержали победу над частотным анализом. Любопытно, что моноалфавитные системы, несмотря на их слабость, имели определённое преимущество — они были очень простыми для внедрения. Криптографы занимались оттачиванием методов, наполняя разрабатываемые алгоритмы всяческими хитростями, но в основном использовали те же концепции, что и в самых простых шифрах.

Одним из наиболее успешных вариантов моноалфавитной системы являлся гомофонный шифр подстановки.

С его помощью была предпринята попытка сбить с толку потенциальных дешифровщиков, которые используют статистический криптоанализ. Буквы, которые встречаются наиболее часто, заменяли по иным системам. Например, если буква Е составляет, в среднем, около десяти процентов текста на любом языке, гомофонный шифр подстановки пытался изменить частоту появления в тексте этой буквы, заменяя Е десятью попеременно меняющимися знаками. Такие методы использовались ещё в XVIII столетии.

Однако дело постепенно двигалось вперёд. Появление великих держав и сопровождающее этот процесс развитие дипломатии вызвали заметное увеличение спроса на надёжную связь. Эта тенденция ещё более усилилась с появлением новых технологий, таких как телеграф, что привело к резкому увеличению объёма коммуникаций. Европейские государства оборудовали так называемые «чёрные кабинеты», нервные центры деятельности, где кодировались самые секретные сообщения и расшифровывались перехваченные вражеские телеграммы. Профессиональная работа «чёрных кабинетов» вскоре привела к тому, что любая форма моноалфавитного шифра подстановки сделалась ненадёжной независимо от модификации. Постепенно участники игры «Обмен информацией» стали применять только полиалфавитные алгоритмы. Частотный анализ утратил свою силу, и дешифровщики опять стали беззащитны перед лицом наступающих шифровальщиков.

КРИПТОГРАФЫ ПРИ ДВОРЕ «КОРОЛЯ СОЛНЦЕ»

Хотя за пределами дворца Людовика XIV мало кто знал о существовании братьев Антуана и Бонавентуры Россиньолей, эти два человека входили в список тех людей Европы, которых больше всего боялись в период потрясений XVII века. Способность расшифровывать послания противников Франции (и личных врагов монарха) не уступала их изобретательности как криптографов. Они разработали так называемый Grand Chiffre (великий шифр), сложный алгоритм замены слогов, который использовался для шифрования самых важных посланий короля. Однако, когда братья умерли, шифр вышел из употребления и стал неподдающимся взлому. И только в 1890 году специалист по криптографии, вышедший в отставку военный служащий Этьен Базери взялся за трудное дело дешифровки зашифрованных документов и после многих лет напряженной работы расшифровал тайные послания «Короля Солнце».



«Король Солнце» Людовик XIV



Этьен
Базери

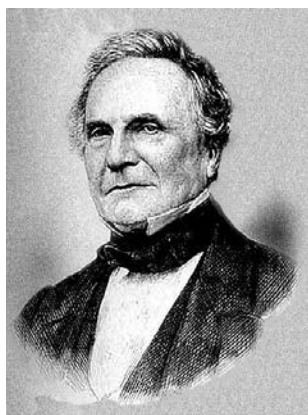
НЕИЗВЕСТНЫЙ КРИПТОАНАЛИТИК

Британский математик Чарльз Бэббидж (1791-1871) — выдающийся учёный XIX столетия. Он изобрёл один из первых механических компьютеров, который назывался «разностной машиной», и далеко опередил своё время. Его интересы охватывали все разделы математики и технологии того времени.

Бэббидж решил заняться дешифровкой полиалфавитных алгоритмов, используя квадрат Виженера. Для этого он сосредоточил внимание на одном из свойств шифра. Вспомним, что при использовании шифра Виженера длина выбранного ключевого слова определяет количество используемых шифровальных алфавитов. Таким образом, если используется ключевое слово WALK (прогулка), то каждая буква исходного послания может быть зашифрована четырьмя различными способами. То же самое относится и к словам. Это свойство станет точкой опоры, с помощью которой Бэббидж попытается вернуть мир полиалфавитного шифра.

Рассмотрим следующий пример послания, зашифрованного при помощи квадрата Виженера.

Исходное послание	B	Y	L	A	N	D	O	R	B	Y	S	E	A
Ключевое слово	W	A	L	K	W	A	L	K	W	A	L	K	W
Зашифрованное послание	X	Y	W	K	J	D	Z	B	X	Y	D	O	W



Чарльз Бэббидж — английский математик, изобретатель первой вычислительной машины. Иностраннный член-корреспондент Императорской академии наук в Санкт-Петербурге (1832). Имеет труды по теории функций, механизации счёта в экономике. Сконструировал и построил (1820-1822) машину для табулирования. С 1822 года работал над постройкой разностной машины. В 1833 году разработал проект универсальной цифровой вычислительной машины — прообраза современной ЭВМ.

Наше внимание сразу же привлекает тот факт, что слово ВУ из исходного послания зашифровано одними и теми же буквами в обоих случаях — ХУ. Это произошло потому, что начало второго ВУ в исходном послании появляется через восемь знаков после начала первого, а восемь — это число, кратное количеству букв (четыре) в ключевом слове (WALK).

С такой информацией и при условии достаточно длинного исходного текста вполне можно догадаться о длине ключевого слова. Процедура следующая: вы записываете все повторяющиеся знаки и отмечаете, через какое количество знаков они повторяются. Затем ищете целые делители этих последних чисел. Общие делители — это числа, которые являются кандидатами на то, чтобы представлять длину ключевого слова.

Предположим, что наиболее вероятный кандидат — это 5, потому что это общий делитель, который появляется наиболее часто. Теперь нам нужно догадаться, каким буквам соответствует каждая из пяти букв ключевого слова. Если мы вспомним процесс шифрования, то каждая буква ключевого слова в квадрате Виженера устанавливает моноалфавитный шифр для соответствующей буквы исходного послания.

В случае нашего гипотетического ключевого слова из пяти букв (C1, C2, C3, C4, C5) шестая буква (C6) шифруется с помощью того же алфавита, что и первая буква (C1), седьмая буква (C7) — с помощью того же алфавита, что и вторая (C2), и т. д. Поэтому в действительности криптоаналитик занимается пятью отдельными моноалфавитными шифрами, каждый из которых уязвим для традиционного криптоанализа.

Процесс завершается составлением частотной таблицы для каждой из букв зашифрованного текста, соответствующих одним и тем же буквам ключевого слова (C1, C6, C11 ... и C2, C7, C12 ...), пока вы не получите пять групп букв, которые в целом составляют общую длину послания. Затем сравните эти таблицы с помощью частотной таблицы для языка исходного послания, чтобы расшифровать ключевое слово. Если две группы данных не совпадают, анализ начинается сначала со второй наиболее вероятной длиной ключевого слова. В конце концов мы определим, по крайней мере, одно вероятное ключевое слово, и нам останется только расшифровать послание.

С помощью этого метода и был взломан полиалфавитный код. Однако поразительное открытие Бэббиджа, которое он совершил где-то около 1854 года, осталось неизвестным. Экцентричный британский интеллектуал никогда не публиковал сведений о нём, и только исследование его заметок позволило узнать, что именно он был первопроходцем в расшифровке полиалфавитных ключевых слов.

К счастью для криптоаналитиков всего мира, через несколько лет, в 1863 году, прусский офицер Фридрих Касиски опубликовал подобный метод.

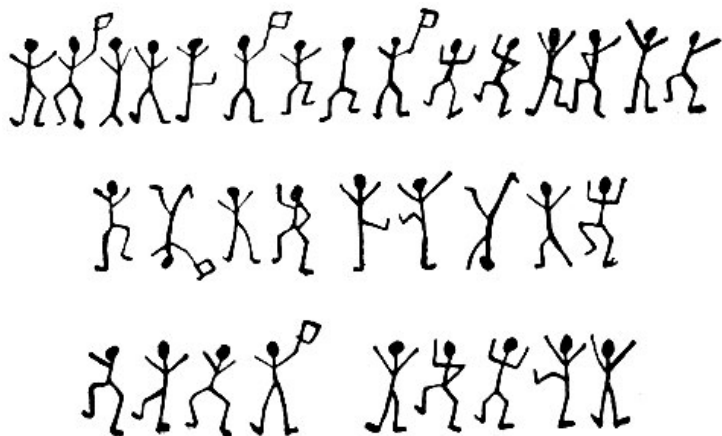
Независимо от того, кто первым взломал полиалфавитный шифр, он перестал быть неуязвимым. С этого момента надёжность шифра стала в меньшей степени зависеть от великих алгоритмических инноваций и в большей степени — от увеличения количества потенциальных шифровальных алфавитов. Их требовалось настолько много, чтобы частотный анализ и его вариации не справлялись со взломом.

Другой важной целью стал поиск путей ускорения криптоанализа. Оба направления исследований шли на сближение, сходясь к одной точке, и породили один и тот же процесс: **компьютеризацию.**

КРИПТОАНАЛИТИК ШЕРЛОК ХОЛМС И МЕТОД ПОДБОРА

Помните удивительную историю с пляшущими человечками, рассказанную знаменитым Шерлоком Холмсом своему другу доктору Ватсону? Мы приведем странные записки, которые преступник посылал своей жертве, а потом напомним конец рассказа.

Вот эти записки:



На первый взгляд пляшущие человечки могут показаться просто забавой. Но Шерлок Холмс, хорошо знакомый с различными видами тайнописи, сразу определил, что перед ним шифр, и понял: фигурки обозначают буквы. Необходимо было только найти ключ. Когда ему это удалось, знаменитый сыщик смог не только прочесть записки, но и послать преступнику строчку пляшущих человечков,



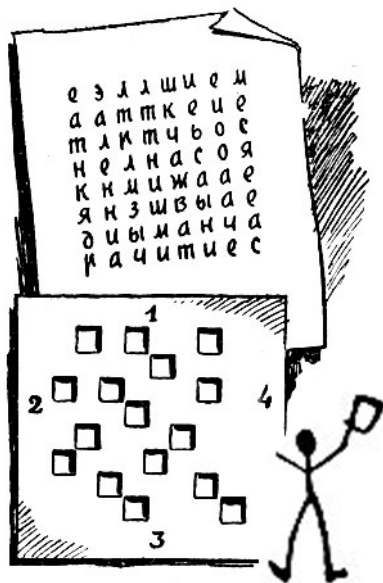
после чего преступник попал в руки правосудия. Пляшущие фигурки означали слова: «Приходите сюда сейчас».

УДИВИТЕЛЬНАЯ РЕШЕТКА

На картинке представлен один из примеров тайнописи, тесно связанный с математикой. Понять такой шифр без ключа невозможно. Чтобы прочесть послание на картинке, нужно проделать следующее. Возьмите листок бумаги. Нарисуйте 64 шахматные клетки. Теперь вырежьте отверстия точно по рисунку, получится решетка. Наложите ее цифрой 1 вверх на беспорядочно написанные буквы квадрата. Посмотрите: в отверстиях появился текст — электронная вычис... Теперь поверните решетку по часовой стрелке на четверть оборота. Получите следующую часть фразы: лительная машина р... Еще такой поворот: ешает сложные мате... И, наконец, при последнем повороте: матические задачи.

Ошибкой было бы думать, что существует только один тип решетки, именно с 16 отверстиями, с расположением как у нас в примере. В таком случае каждый легко бы мог читать зашифрованный текст. Число окошек и их расположение можно менять. Даже если оставить в решетке 16 окошек, то и тогда

*С помощью этой решетки
вы прочтете, что здесь
написано*



можно набрать невероятно большое число способов их расположения — более чем четыре миллиарда. Разгадать такой шифр без ключа (в нашем случае решетки) было бы крайне сложно.

Для того чтобы скрыть решетку от потенциальных шпионов, были придуманы различные способы хранения решетки в памяти. И тут на помощь пришла математика. Если поставить в клетках решетки, где нет выреза, ноли, а в местах выреза — единицы, то получим восемь рядов цифр:

Первый — 01010010

Второй — 00001000

Третий — 10100010

Четвертый — 00010000

Пятый — 01000100

Шестой — 10001000

Седьмой — 00100010

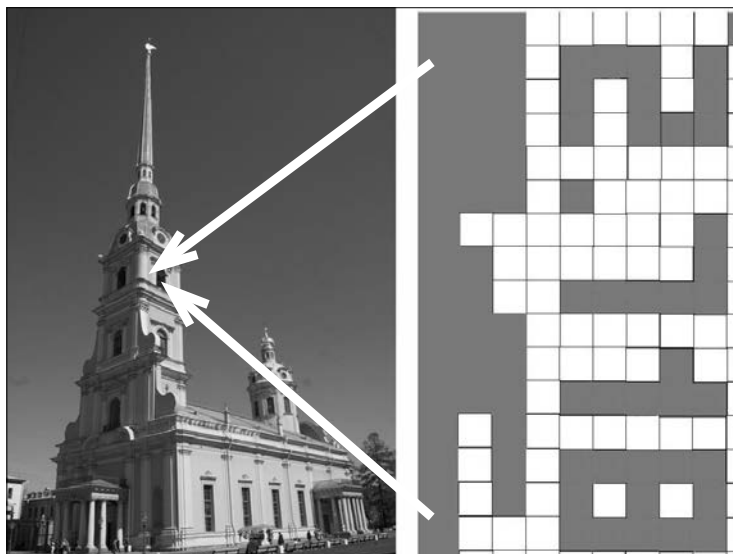
Восьмой — 00010001

Существует много разных систем шифрования. К ним прибегают в военном деле, на дипломатической службе и в любых других случаях, когда нужно сохранить в тайне содержание переписки. Шифрами пользовались и наши революционеры-подпольщики, вынужденные вести переписку так, чтобы царские жандармы не могли ее читать. Из рассмотренных выше примеров видно, что в первом случае Шерлоку Холмсу пришлось трудно, он имел дело со множеством символов — с пляшущими человечками. Революционеры же шифровали группы букв, меняя их расположение, а сама решетка кодировалась только с помощью двух символов.

ОТ КРИПТОГРАФИИ — К СТЕНОГРАФИИ

Криптография — это весьма древняя наука, вероятно, столь же древняя, как и сама письменность. Однако это далеко не единственный возможный способ тайной передачи информации. В конце концов, любой текст при прочитывании должен восприниматься как текст, а если возможность прочитывания скрыта ото всех, кроме получателя, то мы достигли цели. Наука о скрытой передаче информации путём сохранения в тайне самого факта передачи называется стеганографией (от греч. *Στεγανος* — скрытый и греч. *Γραφω* — пишу, буквально «тайнопись»).

Хотя это и может показаться парадоксальным, но развитие новых технологий привело к возрождению стеганографии. К примеру, секретное послание можно запрятать в традиционный аудиофайл, причём слушатель не заметит никакой разницы в звучании. Файлы с изображениями также могут быть использованы для передачи скрытой информации. Пример цифровой стеганографии: число «е», основание натурального логарифма, до трёх знаков после запятой скрыто в крошечном фрагменте более крупного изображения. Слева — кажущаяся обычной фотография, справа — пиксели, вычлененные из одного маленького участка, в котором скрывается число 2,718.



КИНО И КОДИРОВАНИЕ



В классическом научно-фантастическом фильме «2001: Космическая Одиссея» (1968) Стэнли Кубрика на основе романа Артура Кларка наделённый сознанием суперкомпьютер космического корабля, который называется HAL 9000, сходит с ума и пытается убить экипаж, состоящий из людей. А теперь проанализируйте слово HAL так, будто перед нами послание, зашифрованное при помощи шифра Цезаря с ключом В. В этом случае буква Н будет соответствовать букве I, А соответствует В, а L соответствует М. Получится аббревиатура IBM, в то время — крупнейший производитель компьютеров в мире. А может, это просто совпадение?

В случае использования описанного шифра Цезаря, при перехвате послания, если дешифровщик знает лишь, какой алгоритм использовался, но не знает ключа, ему потребуется использовать все возможные варианты, пока не получится послание, имеющее смысл. Для этого, самое большее, ему потребуется проверить все ключи. Если алфавит состоит из n букв, то количество возможных замен даст n кодов.

ШИФРОВКИ В ТРАНШЕЯХ

Самый известный сигнал, передаваемый с помощью кода Морзе, — это сигнал SOS. Он был учреждён в качестве призыва о помощи группой европейских стран в 1906 году из-за простоты передачи (три точки, три тире, три точки — передаётся без межбуквенных интервалов), и ему тогда не приписывалось никакого словесного смысла. Однако люди вскоре дали сигналу разные значения.

Наиболее известным стало *Save our Souls* (спасите наши души). В дальнейшем, по мере частого использования в море, популярной сделалась и другая версия расшифровки SOS — *Save Our Ship* (спасите наш корабль). В русском языке пользуются значением «Спасите от смерти».



ГЛАВА 3.

ИСТОРИЯ ШИФРОВАНИЯ НА РУСИ

*«Даже большой мастер не может обучить
другого мастера: различные ключи подходят
к различным замкам».*

Илья Григорьевич Эренбург

*«В нашем тексте много цифр,
не смущайся — это шифр».*

Михаил Константинович Щербаков

САМОЕ ПРОСТОЕ — ИСПОЛЬЗОВАТЬ МАЛОИЗВЕСТНЫЙ АЛФАВИТ

Русь географически находилась далеко от форпостов западной цивилизации. И понятно, что в иноземных азбуках большинство наших предков совершенно не разбиралось. Однако то же самое можно сказать и о людях, которых ныне принято называть элитой. Русские князья и их разного рода советники тоже были не шибко грамотны, чтобы «уразуметь чуждые письмена».

Но на самой Руси в давние времена существовали способы письменности, довольно быстро забытые большинством восточных славян. Одной из первых славянских азбук стала глаголица. Предполагается, что именно её создал славянский просветитель св. Константин (Кирилл) Философ для записи церковных текстов на старославянском языке.

Исторически глаголица имела две формы: округлая, известная под названием «болгарская», и угловатая («хорватская»).

В русской письменности использовалась только «болгарская». По всей видимости, вычурность знаков привела к тому, что глаголица не выдержала конкуренции с более удобной для повседневных нужд кириллицей и довольно быстро отошла на второй план, а потом вовсе осталась лишь в воспоминаниях.

Во всяком случае историки утверждают, что к концу XV века было уже так. Этот факт и позволил использовать глаголицу в роли шифра. Заметим, что особой защищённостью сей способ тайнописи не обладал, а потому и большого распространения не получил. Известно всего несколько книг, написанных глаголицей.

Помимо глаголицы на территории Руси существовала из собственных алфавитов ещё и древнепермская письменность (известная также под названием «абур» или «анбур»). Эту азбуку использовали комизыряне, комипермяки, русские и некоторые другие народы, жившие на северо-востоке Восточно-Европейской равнины в XIV–XVII веках. Древнепермский алфавит был создан проповедником и просветителем коми Стефаном Пермским в 1372 году на основе кириллицы, греческого алфавита и древнепермских рунических символов.

Как и прочие не слишком известные алфавиты, абур иногда использовался для шифрования.

Разумеется, древнерусские мастера шифровального дела не могли пройти мимо греческого алфавита. Поскольку некоторые русские буквы (ч, ж, ц, ю, я) не имеют аналогов в греческом алфавите, вместо них использовались либо изменённые кириллические буквы, либо полностью придуманные знаки.

Появление этого способа тайнописи, по-видимому, вызвано оживлением начавшихся с конца XIV века сношений Московской Руси с греками. Кроме того, его связывают с определённой модой, которая прошла к концу XVI века.

Несколько позже на Руси в качестве метода шифровки стали использовать латинский алфавит. Распространение западноевропейской культуры и литературы происходило, главным образом, в конце XVI и в течение XVII веков. Поэтому среди русских тайнописей латиница стала появляться в XVII и ещё чаще в XVIII веках. Этот метод обычно внедряли в жизнь представители киевского духовенства. Отметим также, что латинская азбука в качестве шифра использовалась в переписке Посольского приказа.

В наше время мысль использовать в качестве шифра латиницу может вызвать улыбку. О какой защите информации может идти речь, если эти значки на компьютерной клавиатуре известны любому школьнику? С распространением латинского алфавита по Руси он как средство шифрования умер сам по себе.

И немедленно возник вопрос: каким образом ещё можно скрыть смысл послания от нежелательных глаз?

Мы не знаем точно, кому пришла в голову идея построить шифр на основе родной кириллицы, просто-напросто изменив начертание букв, но по русским рукописным памятникам ясно, что такая тайнопись существовала уже в XIV веке. В строгой науке она называется системой «изменённых знаков». Выделяют две её разновидности:

а) систему знаков, изменённых «путём прибавок» к обычным начертаниям;

б) систему, построенную на принципе, сходном с греческой тахиграфией, когда вместо буквы пишется лишь её часть.

Первую разновидность такой тайнописи известный славист М.Н. Сперанский открыл в Смоленской Псалтыри

1395 года. Эта Псалтырь Онежского Крестного монастыря хранилась в своё время в Архангельском местном отделении Церковно-Археологического комитета. Её писец, смолянин, инок Лука, прекрасно владевший искусством письма, любил, видимо, и тайнопись. При создании тайнописного алфавита инок Лука использовал следующие приёмы:

перевороты букв

П → П → П → П
 М → М → W → W → W
 е → е → у → у

деформацию букв

Г → Г → Г
 З → З → W → W → W
 А → ОI → ОI → ОI → ОI
 Б → Б → Б → Б → Б
 Л → Л → Л
 В → В → В

урезание части буквы

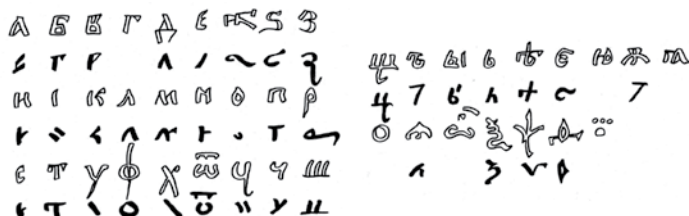
В → З → З
 Р → Р → Р

особые начертания

О → 8
 У → Ç
 С → ∇
 Н → IГ → IГ → U

Ко второй разновидности относится так называемая полусловица. Характерные построения знаков полусловицы:

- вместо целой буквы пишется её характерная часть, чтобы разные буквы не совпадали своими знаками;
- знаки переворачиваются в обратную сторону;
- в виде вариантов встречаются знаки, полученные деформацией исходных букв.



Существовала также «гласная полусловица» — разновидность «полусловицы», где меняются только гласные буквы, а согласные остаются на своих местах без изменения.

В качестве примера — небольшая запись на «Поморских ответах» в рукописи № 3006 Исторического музея (XVIII век).

ВШЛГЛ. БЛГ.р.А.П.К.
рНИ. (АГІОА.НЛЦТІ
ВІ.П-КТІНІІ
ЖІТЛІ

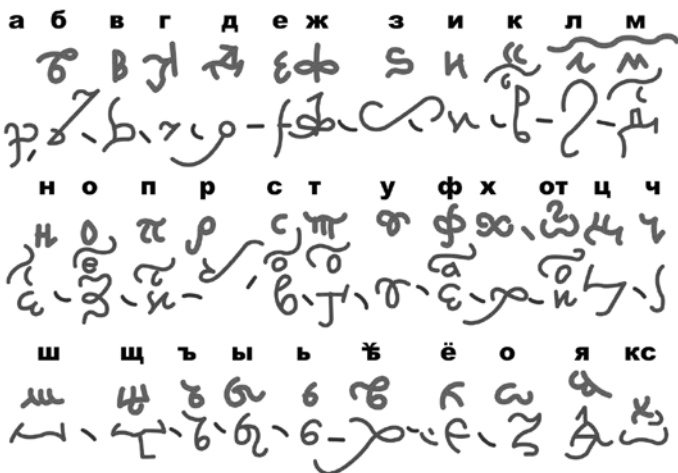
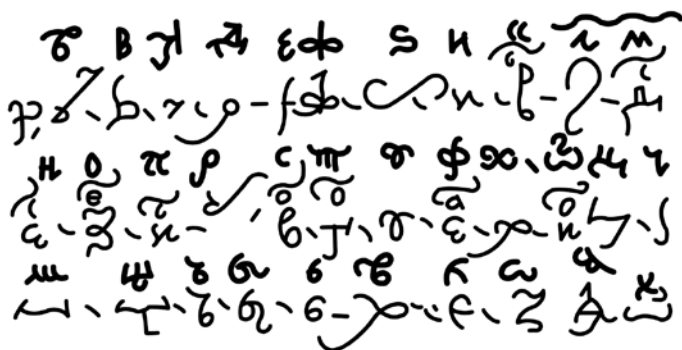
ВШЛГЛ. БЛГ.р.А.П.К.
рНИ. (АГІОА.НЛЦТІ
ВІ.П-КТІНІІ
ЖІТЛІ

а
е
о
и
у

вашего благородна поко
рни. слуги олонцети
ви.пустинии
жители

НО ВЕДЬ ЗНАКИ ДЛЯ ЗАМЕНЫ БУКВ МОЖНО И ПРИДУМАТЬ!

Если представить себе логику размышлений человека, стремящегося защитить своё послание от посторонних, то вполне понятно возникновение идеи о том, что кириллические буквы при шифровании можно не просто менять по форме, но и попросту заменить специально придуманными знаками. Реализация этой идеи и стала следующим шагом в развитии шифровального дела на Руси. Судя по исследованиям историков, это произошло в XVII веке.



В 1619 году из польского плена возвратился Фёдор Романов, отец царя Михаила, постриженный Борисом Годуновым в монахи под именем Филарет. Став патриархом, он фактически был соправителем страны. Филарет лично заведовал внешней политикой государства и, столкнувшись с необходимостью защиты дипломатической переписки, в 1633 году разработал «для своих государевых и посольских тайных дел» особый «склад затейным письмом».

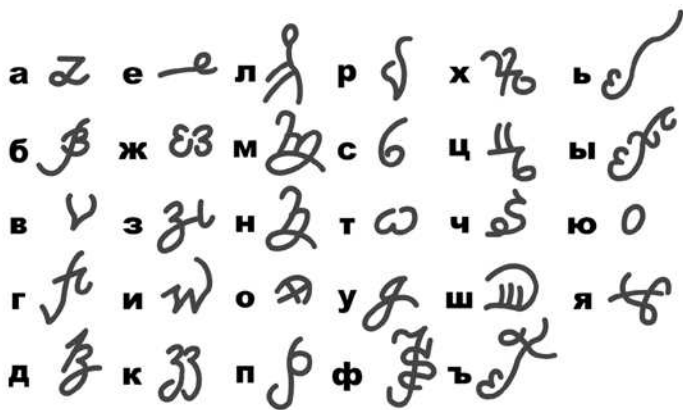
Весной того же 1633 года с дипломатической миссией в Швецию был послан воевода Дмитрий Францбеков. В сохранившихся документах той поры имеется приказ Францбекову, который заканчивается повелением: «Да что он, Дмитрий, будучи в Свее, по сему тайному наказу о тех или иных о наших тайных делах и наших тайных вестях проводит и ему о всём писати ко государю царю и великому князю Михайлу Федоровичу всея Руси к Москве по сему государеву тайному наказу закрытым письмом».

В черновике этого наказа, находящегося в архивах Посольского приказа, слово «затейное» везде зачёркнуто и заменено другим — «закрытое».

Таким образом, 1633 год вполне можно считать годом рождения шифровального дела в России. Тайнопись перестала быть забавой писцов-умников и сделалась одним из важнейших средств сохранения государственной тайны.

Вообще, в XVII веке дипломатическая тайнопись стала употребляться довольно широко. Тайнописями писали и «памяти», то есть инструкции подъячим Приказа тайных дел, ездившим за границу с разнообразными, иногда с очень важными поручениями. Так, например, подъячий Приказа тайных дел Г. Никифоров отвозил думному дворянину А. Л. Ордин-Нащокину, руководителю русской делегации, ведшей переговоры в Польше, написанное «хитрым» письмом предписание царя Алексея Михайловича, «указную азбуку и против той азбуки тайное письмо». Согласно повелению царя, Ордин-Нащокин должен был «по той азбуке о... тайных и всяких делах» писать в Приказ тайных дел.

А. Л. Ордин-Нащокин, в это время заведовавший Посольским приказом, уже и до того применял особую тайнопись. Однако подобные шифры использовались не только при дипломатической переписке.



Ключ тайнописи А. Л. Ордин-Нащокина

В 1667 году в честь приезда в Москву Вселенских Патриархов государев пушечных и колокольных дел мастер Александр Григорьев отлил самый знаменитый колокол Саввино-Сторожевского монастыря, что под Звенигородом, — Большой Благовестный — весом в 2125 пудов. Колокол обладал необычайно глубоким и красивым звоном, равного которому в России не было. По внешней стороне колокола шли надписи в девять рядов: верхние шесть на старославянском языке, а нижние три — тайнописи, состоящие из 425 знаков. Текст надписей предположительно составлен лично царём Алексеем Михайловичем.

А вот и ещё один пример шифра, разработанного по рассматриваемому методу. Это так называемая «решётка». Её учёные отыскали на отдельном листе Патриаршей библиотеки № 93 (вторая половина XVII века).

Здесь метод шифрования заключается в замене обычных букв фигурами, взятыми из решётки, составленной из двух параллельных линий, пересечённых под прямым углом двумя такими же линиями. В полученных клетках помещено по четыре и по три буквы в порядке азбуки: в тайнописи буквы заменяются, при этом первая — чистой фигурой, а следующие — той же фигурой с одной, двумя или тремя точками. Поскольку в решётке поместились не все буквы, буквы *С, і, Ч, Ъ* и фита писались «в открытую».

«ФЛОПЯЦЕВСКАЯ АЗБУКА», «АЗБУКА КОПЦЕВА» И ДРУГИЕ

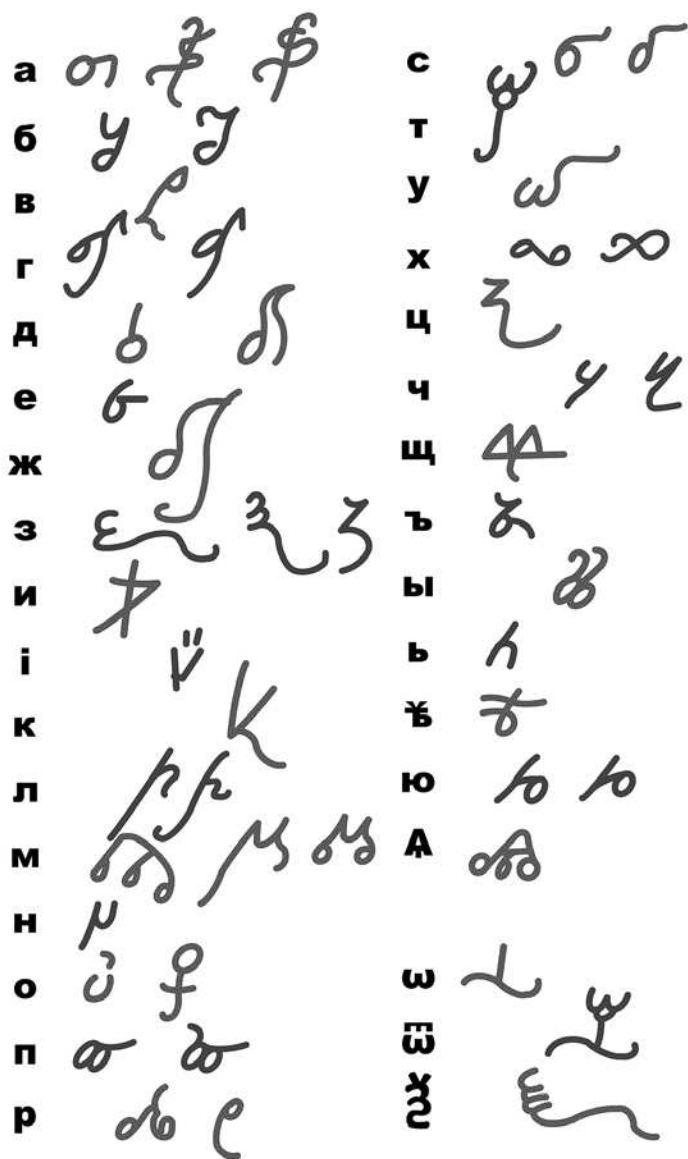
Методы шифрования, основанные на замене кириллических букв специально придуманными знаками, долгое время были весьма популярны. Специальные алфавиты разрабатывались неоднократно. По сравнению с уже рассмотренными примерами их делали более сложными. Кроме того, научные исследования доказывают, что авторы таких шифров были знакомы с работами друг друга. Чтобы убедиться в этом, достаточно ознакомиться с несколькими шифрами, которые историки датируют XVII веком.

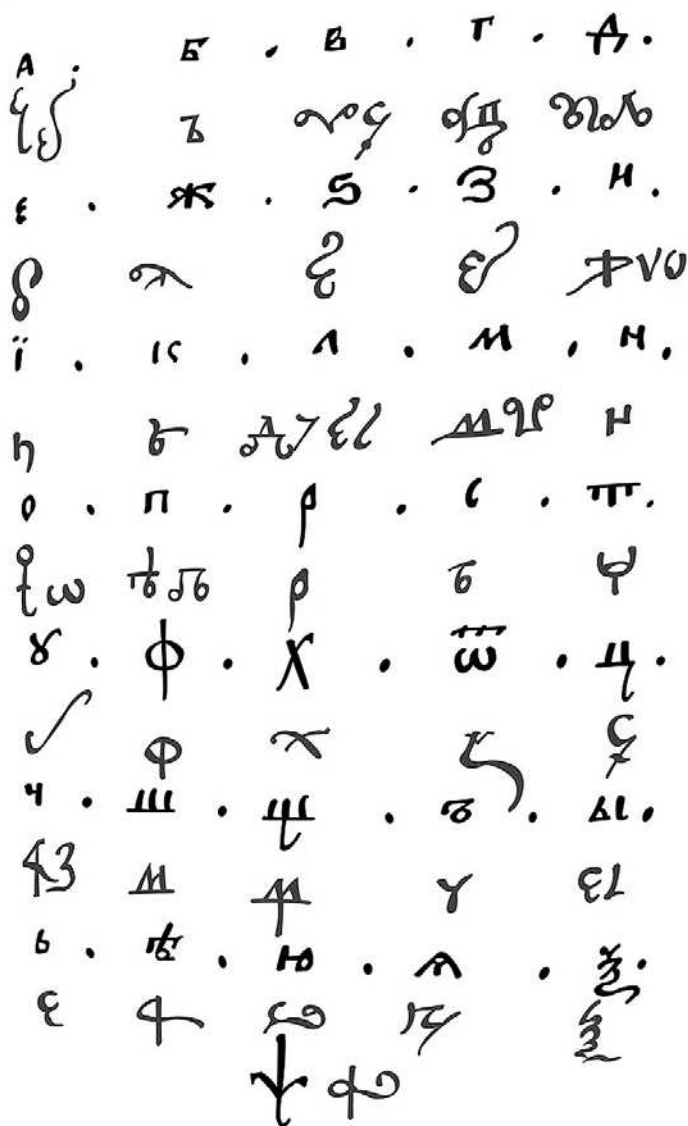
На полях рукописи Румянцевского музея № 460 («хронограф XVII века») обнаружили массу заметок тайнописной азбукой, частично обрезанных при позднейшем переплетении рукописи. Изучив эти заметки, известный филолог А. Х. Востоков составил алфавит этой тайнописи. Листок с этим ключом был вложен в саму рукопись. Знаки этой тайнописи основаны:

- на двенадцати изменённых кириллических буквах — М, З, П, Х, Щ, Ъ, Ы, Ю, Я, К, Ч и Ц;
- на придуманных знаках для девяти букв — А, Г, Д, Ж, Р, Т, У, омега и М;
- на греческих скорописных начертаниях для шести букв — Д, Е, М, Н, П и С;
- на полусловице для двух букв — Ь и В;
- а также на двух перевёрнутых буквах — Б и З.

Кроме того, для четырнадцати букв имеются неединичные (двойные и тройные) начертания знаков.

Явно придуманную тайнописную азбуку обнаружили на листе 129 в рукописи библиотеки б. Московской Синодской типографии. Некоторые буквы этого шифра имеют двойные или тройные соответствующие знаки. Часть знаков является вариантами друг друга, но большинство — смесь из полусловицы, греческих и глаголических букв и деформированной кириллицы.



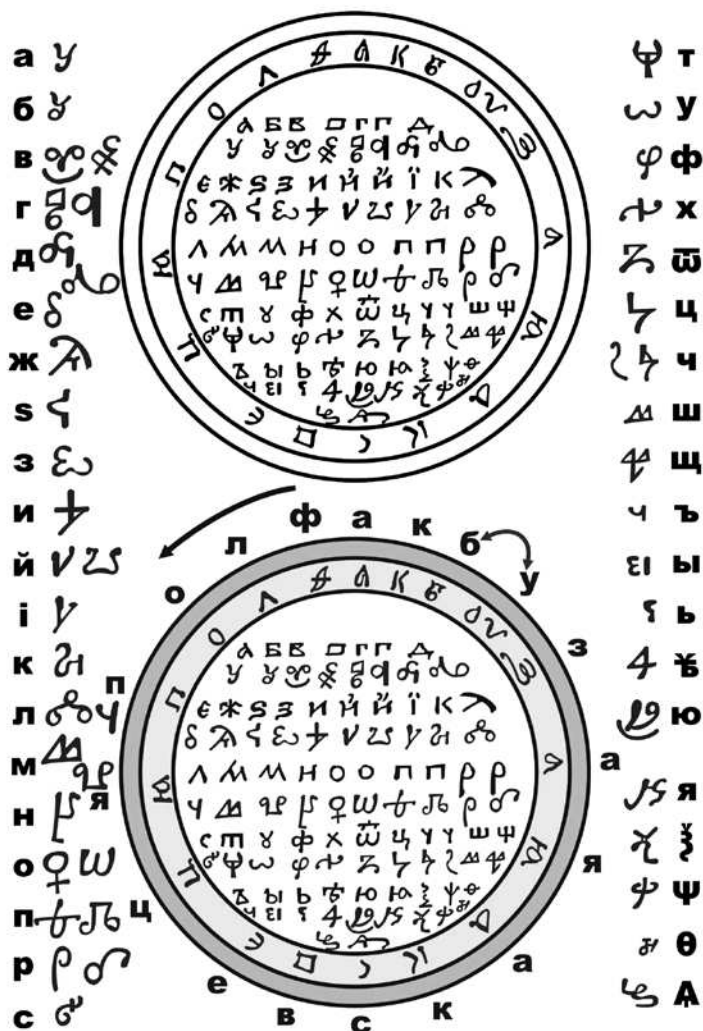


Тайнописная азбука из рукописи библиотеки
 б. Московской Синодской типографии

[illegible]

Handwritten symbols and characters, possibly representing a cipher or shorthand, arranged in two columns. The symbols include various letters, numbers, and special characters, some with superscripts or subscripts.

И, наконец, рукопись, известная как «флопяцевская азбука» из Российской национальной библиотеки [О. XVI. 2 (Толст. III, 27) л. 94 об.].



Здесь шифр состоит из:

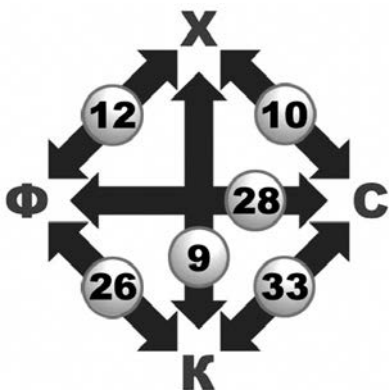
- тринадцати специально придуманных знаков;

- десяти греческих букв — «дельта» для буквы Е, «ипсилон» для букв А, Б и И, «омега» для букв И, У и О, «ро» для буквы Р, «сигма» для буквы С, «фи» для буквы Ф, «хи» для русской буквы «кси» и «пси» для русской буквы «пси»;
- девяти случаев замены кириллицы — буква Ы заменяет букву К, буква Я — Ж, буква Ч меняет букву Л и букву Ъ, буква З меняет букву О, М — Ш, варианты буквы Е — Ы и Ь, буква К меняет Ю;
- шести глаголических — для букв В, Г, Д, Л, М и П.

Кроме того, для десяти букв — В, Г, Д, И, Л, М, О, П, Р и Ч — азбука даёт по несколько начертаний. Из них пять — В, И, Л, О и Р — из первого десятка самых часто используемых букв современного русского языка. Если принять, что в древнерусском языке частотные характеристики употребления букв отличались не намного, то выбор десятка «усложнённых» букв не представляется случайным.

Чтобы обнаружить связь между всеми четырьмя шифрами, достаточно сравнить их между собой. На приведённых ниже рисунках шифры обозначены следующим образом:

- из хронографа XVII века — Х;
- из рукописи библиотеки Синодской типографии № 1028 — С;
- из «азбуки копцева» — К;
- из «флопяцевской азбуки» — Ф.



На этом рисунке дано количество общих знаков разных шифров

[illegible]

На этом рисунке все шифры сведены в одну таблицу, в которой выделены буквы-аналоги. Общих знаков много.

Наибольшее количество общих знаков (71) имеет шифр из рукописи библиотеки Синодской типографии № 1028. Наиболее оригинальный комплект знаков (меньше связей с другими азбуками — 31) имеет шифр из хронографа XVII века.

Промежуточные значения (68 и 66 соответственно) имеют «азбука копцева» и «флопяцевская азбука». Поскольку во всех шифрах наряду с общими имеются и оригинальные

Неординарные знаки в алфавитах																			
а б в г д ж з и к л м о п р с х ч																	*	**	
х а б		г д		з		л м о п р с х ч											13	5	
с а		в г д		и		л м о п											9	5	
к а		в г д ж		и к		л м о п р											ч	13	6
ф		в г д		и		л м о п р											ч	10	5

*На этом рисунке показаны буквы,
которые имеют неординарное соответствие знаков.*

** Количество букв с неординарными знаками.*

*** Количество частных букв с неординарными знаками.*

знаки, шифры нельзя считать вариантами одной разработки. Но большое количество сходных знаков позволяет предположить, что авторы были знакомы с работой друг друга.

Представление букв несколькими знаками может быть как результатом компиляции знаков из других азбук, с которыми был знаком разработчик, так и попыткой намеренного усложнения азбуки — чтобы зашифрованный текст было труднее прочесть тому, для кого послание не предназначалось. В этом можно усмотреть элемент развития шифровального дела в целом.

А ПОЧЕМУ БЫ КИРИЛЛИЦУ НЕ ЗАМЕНИТЬ... КИРИЛЛИЦЕЙ?

Итак, мы рассмотрели, каким образом древние отечественные шифровальщики использовали для сокрытия смысла посланий от нежелательных глаз чужие алфавиты («чуждые письмена») и специально придуманные для этой цели знаки. Но ведь можно попросту заменить одну кириллическую букву другой. Так родился шифр, известный под названием «литорея» (от лат. *Littera* — буква).

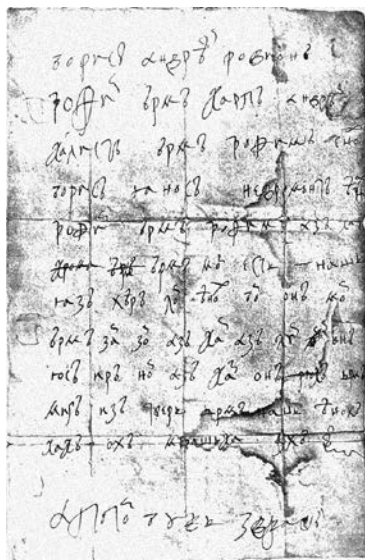
При шифровании с помощью простой литореи все согласные буквы сначала разбиваются на два ряда:

б	в	г	д	ж	з	к	л	м	н
щ	ш	ч	ц	х	ф	т	с	р	п

Потом верхние буквы используют вместо нижних, и наоборот. Гласные остаются без замен.

К примеру, слово «наступление» будет выглядеть как «палкунсепие». Похожий ключ для латинского алфавита мы рассматривали в главе 2 (раздел «Руководство для юных леди»). Этот метод шифровки (известный также как «тарабарская грамота») был распространен в конце XIV–XV веков. Однако мода на использование его в быту не прекращалась до XVIII века включительно.

Для примера приведем письмо царя Алексея Михайловича своему двоюродному брату стольнику Афанасию Матюшкину, зашифрованное именно с помощью тарабарской грамоты. Конечно, простая литорея была



Письмо царя
Алексея Михайловича

не слишком сложным шрифтом. Более сложной считалась ли-
терей мудрая, ключ к которой заключается в следующем: имею-
щие численное значение буквы церковнославянского алфавита
обозначались точками, черточками и кружками. Буквы, не упо-
требляющиеся в значении цифровых знаков, а именно: б, ж, ъ, ь,
ы, ю, я, оставались на своих местах без замен.

<i>а</i>	<i>в</i>	<i>г</i>	<i>д</i>	<i>е</i>	<i>ѕ</i>	<i>ѕ</i>	<i>и</i>	<i>Ѡ</i>	<i>і</i>
.

	
		
			
				
					
							.	.	.
							.	.	.

<i>к</i>	<i>л</i>	<i>м</i>	<i>н</i>	<i>ξ</i>	<i>о</i>	<i>и</i>	<i>ч</i>
—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—
		—	—	—	—	—	—
			—	—	—	—	—
				—	—	—	—
					—	—	—
						—	—
							—

<i>р</i>	<i>с</i>	<i>т</i>	<i>у</i>	<i>ф</i>	<i>х</i>	<i>ψ</i>	<i>ω</i>	<i>ц</i>
о	о	о	о	о	о	о	о	о
	о	о	о	о	о	о	о	о
		о	о	о	о	о	о	о
			о	о	о	о	о	о
				о	о	о	о	о
					о	о	о	о
						о	о	о
							о	о
								о

Ǻ Ψ	Б Ѥ	В Ѧ	Г Ѧ	Д Б
Є В	Ж Г	С Д	З Є	Н Ж
İ С	К Ѥ	Л Н	М І	Н К
О Л	П И	Р Н	С О	Т П
У Р	Ф С	Х Т	Ѡ У	Ц Ф
Ч Х	Ш Ѡ	Щ Ц	Ъ У	Ы Ш
Ь Ψ	Ѧ Ъ	Ѡ Ы	Ю Ь	Ѧ Ѧ
Ѧ Ѡ	О Ю	Ѧ Ѧ	Ѥ Ѧ	Ѧ Ѧ

Одной из разновидностей мудрой литореи является шифрование в «квадратах», известное с XVII века. Ключ для этого метода представляет собой таблицу из сорока квадратов. В каждом из квадратов изображались две разные буквы алфавита. Причем одни буквы были окрашены киноварью (красные), а другие просто написаны чернилами. Красные буквы (на рисунке — черные) — это буквы обычного алфавита, серые — их замена при шифровании.

Азбука-шифр начинается с четвертой позиции обычной. Иными словами, сдвинута на три буквы вправо: буква А соответствует букве Г обычного алфавита (вспомним шифр Цезаря).

Проще ключ для этого шифрования можно представить в виде двух строк соответствия.



Весьма похожий метод шифрования использовал для своей переписки Пётр Первый. Однако у него есть и существенное отличие — некоторые буквы изначального послания при шифровке заменялись на целые слоги (см. с. 81).

Это чередование слогов и одинарных букв в шифровке используется и в наши дни, поскольку существенно затрудняет взлом шифра посторонним. К тому же шифрованный текст не имел пробелов. Но шифробозначения подобраны таким образом, что при расшифровке сведущим человеком это не вызывало трудностей.

а мѣ	б лн	в но	г нн	д зѣ	е ѡѣ	ж нѹ
з о	и пб	к ра	л сѣ	м пн	н ѣ	о хн
п ѡ	р ца	с ѣѣ	т шѣ	у ам	ф з	х ѣ
ѡ ѡ	ц б	ч тѣ	ш ю	щ я	ъ ѣ	ы а
ь ѣѣ	ѣ ва	ю тѣ	я дн			

Метод шифрования Петра Первого

В заключение заметим, что по сути своей литорея является шифром простой замены, который легко дешифруется современными методами.

В период русского Средневековья нашими предками ещё не использовались арабские цифры. Числовое значение имели некоторые буквы кириллицы, такая система называлась «цифирь». Она вполне позволяет использовать при шифровании метод, заключающийся в сложении цифрового значения букв.

Пример: буква Л (люди) означала число 30, буква М (мыслете) — 40. Если в шифровке написано ЛМ, то, сложив 30 и 40, получим 70. Число 70 обозначалось буквой О (он). Иными словами, сочетание в шифровке двух букв ЛМ означало в исходном тексте букву О. Таким же образом зашифровывались и остальные буквы послания.

ВОСПОЛЬЗУЕМСЯ ЦИФИРЬЮ

Самый старый из известных образцов такой тайнописи находится в псковском Апостоле 1307 г. (Собрание Большой Патриаршей библиотеки, № 722). Такая система тайнописи была популярна на Руси долгое время: с XIV по XVII век.

Существовал и другой метод использования цифири — так называемый описательный разряд. Здесь, как явствует из названия, шифровка велась с помощью описания, словами.

Примером может служить текст из рукописного собрания Кирилло-Белозерского монастыря XV в.: «Аще хощеши увѣдати имя писавшаго книгу сию, и то ти напишу: «Десятерица сугубая ($10+10=20$) и пятерица четверицею ($5 \times 4 = 20$, сумма 40) и единъ (1); десятирица дващи ($10 \times 2 = 20$) и един (1); десятая четыре сугубо и четырежды по пяти ($10 \times 2 \times 4 + 4 \times 5 = 100$); дващи два съ единомъ ($2 \times 2 + 1 = 5$); единица четверицею сугубо ($1 \times 4 \times 2 = 8$); в семь имени словъ седмерица, три столпы и три души, царь. И всего же числа в семь имени РОЕ (175)».

Тут зашифровано имя «Макарей».

Сумма букв-цифр действительно 175 и семь букв, из которых три гласные и три согласные и одна (й) полугласная. Используются количественные числительные и сумма. Последняя часть служит как бы проверкой для всей шифровки.

Наконец, существовал ещё один метод шифрования — так называемый значковый разряд. При его использовании некоторые буквы-цифры зашифровывались условными значками. Например, цифири из разряда единиц заменялись соответствующим числом точек. Цифири из разряда десятков заменялись соответствующим числом вертикальных чёрточек, сотни заменялись кружками. Например, четыре точки соответствовали цифре «четыре» и букве Д, семь вертикальных чёрточек — цифре 70 и, следовательно, букве О, три кружка — цифре 300 и, следовательно, её буквенному обозначению — Т.

НЕ СВЯЗАТЬ ЛИ НАМ ШИФРОВОЧКУ?

Помимо обычного написания русских слов существовала так называемая «вязь». Она представляет собой тип письма, в котором буквы сближаются или соединяются одна с другой и связываются в непрерывный орнамент. Такой метод тоже иногда применяли с целью зашифровать текст.

Для примера приведём «вязь знамени Ермака» (она украшает одно из знамён, под которыми Ермак в 1582 году покорил Сибирское ханство Кучума).

Помимо этого примера известны также: вязь знамени Дмитрия Пожарского, поморская вязь, вязь молитвы «Достойно есть яко», монокондил и др.



грозный

ахтрати́гия



воевода

миха

Грозный воевода

а(р)х(ис)трати́гия Мих(аил)а

«Вязь знамени Ермака»

В качестве метода шифрования нельзя ещё не упомянуть акrostих. Он представляет собой стихотворение, по первым буквам строк которого можно получить определённую информацию.

Для примера приведём следующий акростих Г. Державина:

Родясь от пламени, на небо возвышаюсь;

Оттуда на землю водою возвращаюсь!

С земли меня влечёт планет

всех князь к звездам;

А без меня тоска смертельная цветам.

Как видно, здесь зашифровано слово «роса». Однако практическая полезность этого метода для криптографии мизерна. Скорее он является занимательным упражнением для поэтов.

Рассмотренные в этой главе исторические материалы убедительно доказывают, что наши предки не остались в стороне от процесса развития мировой криптографии, хотя, в силу исторических и географических причин, обретя государственность позже Западной Европы, соответственно позже и пришли к необходимости использовать шифрование в государственных делах.

ГЛАВА 4.

ШИФРОВАЛЬНЫЕ МАШИНЫ

«Если бы шифровальные машины использовались как положено — без повторяющихся разовых ключей, без силей, без ограничений по установкам на штепсельной коммутационной панели и расположениям шифраторов и без шаблонных сообщений, то вполне вероятно, что их вообще никогда бы не смогли взломать».

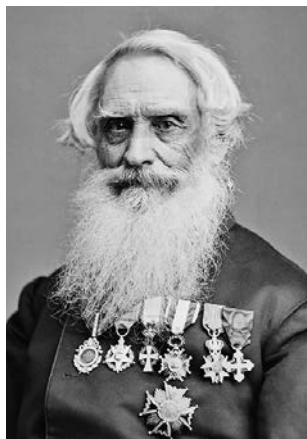
Сингх Саймон. «Книга шрифтов»

АЗБУКА МОРЗЕ

В XIX веке применение кодов расширилось и вышло за пределы обмена тайными посланиями.

Развитие телеграфа в первой трети столетия и изобретение многоканального телеграфа Томасом Алва Эдисоном (1847-1931) произвели настоящую революцию в средствах связи и кардинально изменили мир. Поскольку работа телеграфа основана на передаче электрических импульсов, требовалось внедрить систему, которая переводила бы содержание посланий на язык, доступный такой передаче. Другими словами, потребовался код.

Среди различных предложений выиграла система из точек и тире, которую изобрёл американский художник и изобретатель Сэмюэл Финли Бриз Морзе.



Сэмюэл Морзе

Морзе учился в Йельском университете, где прослушал курс лекций по новой тогда области физики — электричеству. Интерес к электричеству и телеграфии возник у Морзе, как полагают, в 1832 году, когда он возвращался из Европы в США. На борту судна зашел разговор об опытах Фарадея по электромагнетизму — «извлечению искр из магнита». Морзе пришло в голову, что сочетание искр можно использовать как код для передачи сообщений. Во время месячного плавания он сделал

несколько предварительных чертежей, а по прибытии в Америку построил электромагнитный телеграфный аппарат. В 1837 году он продемонстрировал изобретение в Нью-Йоркском университете. В 1838 году Морзе разработал специальный код (азбука Морзе) и послал первое телеграфное сообщение:

«Чудны дела твои, Господи!»

Код (азбука) Морзе может считаться предшественником тех кодов, которые много десятилетий спустя станут использоваться всеми нами для введения данных в компьютеры и получения информации от них.

Азбука Морзе представляет буквы алфавита, числа и другие знаки в виде комбинаций точек, тире и пробелов. Таким образом она переводит алфавит в то, что может быть выражено набором простых световых, звуковых или электрических сигналов.

Каждая точка представляет собой короткий посыл, продолжительность которого составляет примерно $1/25$ секунды; тире имеет утроенную продолжительность. Пробелы между буквами также имеют утроенную продолжительность, а для пробелов между словами используются послы пятикратной продолжительности.

Вначале Морзе отказали в patente на его код в США и Европе. Наконец, в 1843 году правительство профинансировало строительство телеграфной линии между Вашингтоном и Балтимором. В 1844 году была проведена первая закодированная передача, а вскоре после этого учреждена компания, перед которой поставили цель: провести телеграфные линии по всей Северной Америке.

К 1860 году, когда Наполеон III наградил Морзе орденом Почётного Легиона, и США, и Европа уже были покрыты телеграфными проводами.

К моменту смерти Морзе в 1872 году в Америке насчитывалось свыше 300000 километров кабелей для связи.

Изначально для отправки и получения телеграфных сообщений использовался простой аппарат, изобретённый самим Морзе в 1844 году. Аппарат состоял из телеграфного ключа,

который служил для включения и выключения электрического тока, и электромагнита, который принимал входящие сигналы. Каждый раз при нажатии ключа — обычно это делалось указательным или средним пальцем — устанавливался электрический контакт. Прерывающиеся импульсы, производимые воздействием на телеграфный ключ, передавались на кабель, который состоял из двух медных проводов. Эти провода, прикрепленные к высоким деревянным «телеграфным» столбам, соединяли различные телеграфные станции в стране и часто тянулись на сотни километров, не прерываясь.

В приёмнике главной деталью был электромагнит, изготовленный из мотка медной проволоки, обёрнутого вокруг железной сердцевины. Как только проволока получала импульсы электрического тока, соответствующие точкам и тире, железная сердцевина намагничивалась и притягивала движущуюся часть, также сделанную из железа. При ударе о магнит получался отчётливый звук. При получении точки он представлял собой короткий «щелчок», в случае тире звук был более длинным.

Первоначально при отправке телеграммы с помощью такого аппарата требовались оператор-человек для «набивания» кодированной версии послания на одном конце линии и другой оператор для получения и расшифровки на другом конце. Традиционные знаки переводятся в азбуку Морзе в соответствии со следующей таблицей:

Русский алфавит	Латинский алфавит	Код Морзе	Цифры и знаки препинания	Код Морзе
А	A	• —	1	• — — — —
Б	B	— • • •	2	• • — — —
В	W	• — —	3	• • • — —
Г	G	— — •	4	• • • • —
Д	D	— • •	5	• • • • •

Русский алфавит	Латинский алфавит	Код Морзе	Цифры и знаки препинания	Код Морзе
Е	E	•	6	— • • • •
Ж	V	• • • —	7	— — • • •
З	Z	— • • •	8	— — — • •
И	I	• •	9	— — — — •
Й	J	• — — —	0	— — — — —
К	K	— • —	,	• — • — —
Л	L	• — • •	•	• • • • •
М	M	— —	;	— • — • —
Н	N	— •	:	— — — • •
О	O	— — —	?	• • — • •
П	P	• — — •	№	— • • • •
Р	R	• • •	"	• — • • •
С	S	• • •	'	• — — — •
Т	T	—	()	— • — — —
У	U	• • —	!	— — — • —
Ф	F	• • — •	—	— • • • —
Х	H	• • • •	ждать	• — • • •
Ц	C	— • — •	понял	• • • — •
Ч		— — — •	/	— • • • •
Ш		— — — —	знак раздела	— • • • —
Щ	Q	— — • —	исправление	• • • • • •
Ы	Y	— • • —	Начало передачи	— • • — — •
Ь	X	— • • —	Готов к приёму	• — — — — —
Э		• • • • •	Начало действия	• — • • — — — —
Ю		• • • —	Окончание	• — • • •
Я		• — • —		

Таким образом, послание «Учим код Морзе» будет закодировано следующим образом:

У	Ч	И	М		К	О	Д		М	О	Р	З	Е
...	---	..	--		--	---	...		--	---	...	---	.

.. — .. . — .. . — . — — . — . — — — — .

На этой картинке написано слово «электро-» с помощью азбуки Морзе. Если внимательно рассмотреть азбуку Морзе, можно заметить, что некоторые буквы передаются одним-двумя знаками: Е — точкой, Т — тире, А — точкой, тире, И — две точки и т. д., а другие — четырьмя и даже пятью: Ж — четыре тире, Х — четыре точки, Ч — три тире, точка, Э — две точки, тире, две точки.

Давайте определим, из скольких знаков должна состоять кодовая группа, чтобы можно было записывать все буквы одинаковым количеством, нолей и единиц.

Вначале возьмем только два знака — 0 и 1. Получим:

А=00

Б=01

В=10

Г=11

На этом наши возможности оказались исчерпаны: 22=4.

Возьмем три знака:

А=000 Д=100

Б=001 Е = 101

В=010 Ж=110

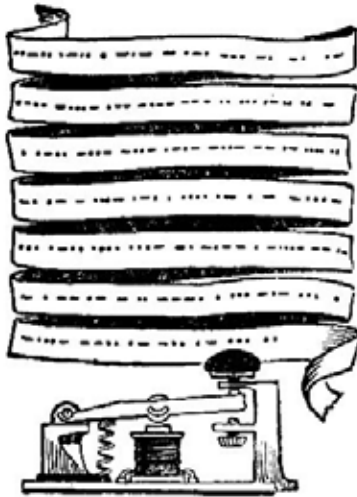
Г=011 З=111

И опять подошли к пределу — 23=8.

Возьмем четыре знака, получим 24=16. При пяти знаках, наконец, достигнем нужного результата — 25=32, а это уже дает то число комбинаций, которое необходимо для кодирования всех букв в нашем алфавите.

Вот один из примеров такого кода:

А = 00000	Ж = 00110	О = 01101	Х = 10100	Ю = 11010
Б = 00001	З = 00111	П = 01110	Ц = 10101	Я = 11011
В = 00010	И = 01000	Р = 01111	Ч = 10110	Й = 11100



С помощью двух символов — точки и тире можно записать любой текст

Г = 00011	К = 01001	С = 10000	Ш = 10111	Ь = 11101
Д = 00100	Л = 01010	Т = 10001	Щ = 11000	Ъ = 11110
Е = 00101	М = 01011	У = 10010	Ы = 11001	Э = 11111
	Н = 01100	Ф = 10011		

Теперь подсчитаем, из скольких знаков должна состоять кодовая группа, чтобы все цифры от 0 до 9 записывать тоже одинаковым количеством полей и единиц. И здесь оказывается, двух и трех знаков мало. А вот четырех достаточно. Они дают 16 комбинаций, а нам надо всего 10.

Вот как будут выглядеть цифры в закодированном виде:

0=0000 5=0101

1=0001 6=0110

2=0010 7=0111

3=0011 8=1000

4=0100 9=1001

Эти коды алфавита и цифр не единственны, их может быть очень много.

Воспользуемся нашими кодами и закодируем такое слово, как «кибернетика», и число «13».

Слово будет выглядеть как группы нолей и единиц: 01001 01000 00001 00101 10000 01100 00101 10010 01000 01001 00000.

Число 13 запишется так: 0001 0011.

Для записи двузначного числа понадобилось восемь знаков. Можно ли обойтись меньшим количеством знаков? Можно. Для этого только нужно изобразить число опять с помощью 0 и 1, но уже не кодовыми группами, а в двоичной системе счисления.

Как упоминалось выше, код Морзе был, в определенном смысле, первой версией цифровых систем связи будущего.

Для подтверждения этой мысли достаточно перевести азбуку Морзе в цифры: точка станет 1, а тире — 0. Эти группы из 1 и 0 будут встречаться нам в последующих главах.

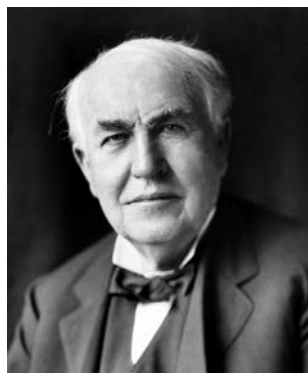
В XX веке традиционную телеграфию заменила беспроводная связь, которая вышла на первый план благодаря изобретению радио. Вчерашние телеграфисты сделали радистами. Новая технология означала, что послания можно передавать на более высокой скорости. Однако послания, отправляемые в виде электромагнитных волн, относительно легко перехватывать. Это обеспечило криптоаналитикам большой объем зашифрованного материала для работы и помогло укрепить их доминирующее положение в битве с криптографами, тем более что большая часть шифров, которыми пользовались правительства и частные агентства, основывались на уже известных алгоритмах.

Это, например, относится к шифру Плейфера, который изобрели британцы барон Лайон Плейфер и сэр Чарльз Уитстон. Шифр Плейфера был очень хитроумной вариацией шифра Полибия — однако, в конце концов, только вариацией.

НЕВЕРБАЛЬНАЯ СВЯЗЬ

У Томаса Алвы Эдисона, всемирно известного американского изобретателя и предпринимателя, имелись проблемы со слухом, и он общался с женой, Мэри Стилвелл, при помощи азбуки Морзе. Когда Эдисон ещё ухаживал за Мэри, он сделал ей предложение, выстукивая слова легкими касаниями ее ладони, а Мэри ответила таким же образом. Затем телеграфный код стал обычным способом общения этой пары.

При посещении театра Эдисон клал руку Мэри себе на колено, чтобы она могла «телеграфировать» ему диалоги актеров.



Бетховен — ещё один человек, имевший проблемы со слухом и ассоциируемый с телеграфом, хотя и несколько косвенно: первые четыре ноты его выдающейся Пятой симфонии имеют ритм,

напоминающий послание азбукой Морзе: точка, точка, точка, тире. В азбуке Морзе «точка точка точка тире» соответствует английской букве V, первой букве в слове victory (победа). Именно поэтому радиостанция Би-Би-Си во время Второй мировой войны использовала Пятую симфонию Бетховена во всех своих передачах для оккупированной Европы.



ШИФР ПЛЕЙФЕРА

Создатели этого шифра, барон Лайон Плейфер и сэр Чарльз Уитстон (являющийся также создателем телеграфного аппарата, названного его именем), были друзьями и соседями, и оба увлекались криптографией. Этот метод напоминает своего знаменитого предшественника, шифр Полибия. В нём также используется таблица из пяти столбцов и пяти строк. Первый шаг — это замена каждого знака исходного текста на пару букв в соответствии с шифром из 5 различных букв. В нашем примере шифром будет служить JAMES. В случае алфавита из 26 букв у нас получается следующая шифровальная таблица:

J	A	M	E	S
B	C	D	F	G
H	I-K	L	N	O
P	Q	R	T	U
V	W	X	Y	Z

Следующий шаг — это разделение исходного текста на пары букв, или диграфы. Две буквы, из которых состоит каждый диграф, должны быть разными, а для того чтобы избежать возможных совпадений, мы используем букву X. Мы также используем эту букву, чтобы завершить диграф в случае, если остается одна последняя буква слова. Например, если исходное послание THRILL, то деление на диграфы даст:

TR IL LX

Слово TOY делится на диграфы как:

TO YX

После того, как мы получили исходное послание в форме диграфов, мы можем начать его шифровать, учитывая три условия:

- (а) Две буквы диграфа находятся в одной строке
- (б) Две буквы диграфа находятся в одном столбце
- (в) не выполняется ни одно из вышеуказанных условий

Если у нас случай (а), то знаки в диграфе заменяются буквой, расположенной справа от каждой («следующей» в естественном порядке в таблице). Таким образом пара JE заменяется на AS:

J	A	M	E	S
----------	----------	----------	----------	----------

В случае (б) знаки диграфа заменяются буквой, которая расположена сразу же под ней в таблице. Например, диграф ET шифруется как FY, а TY — YE:

E
F
N
T
Y

В случае (в), чтобы зашифровать первую букву диграфа, мы смотрим на её строку, пока не доходим до столбца, который содержит вторую букву. Шифр для исходного текста — это буква, которая находится на пересечении строки и столбца. Чтобы зашифровать вторую букву, мы смотрим на строку, пока не доходим до столбца, в котором содержится первая буква.

Шифр для исходного текста снова находим на пересечении строки и столбца.

Например, в случае диграфа CO C шифруется как G, а O — как I или K.

J	A	M	E	S
B	C	D	F	G
H	I-K	L	N	O
P	Q	R	T	U
V	W	X	Y	Z

Чтобы зашифровать послание ТЕА с ключевым словом JAMES, мы действуем следующим образом:

- Выражаем послание в виде диграфа: ТЕ АХ
- Т шифруется как Y

- Е шифруется как F
- А как M
- X как W

Зашифрованное послание: YFMW.

Несмотря на изобретательность создателей, дешифровка этих пусть и усовершенствованных, но фактически используемых повторно шифров в конечном счёте была вопросом времени и вычислительных возможностей. Криптографическая история Первой мировой войны прекрасно это иллюстрирует.

Немецкий дипломатический шифр не отличался особой сложностью. Однако сами немцы не подозревали, что ещё один применяемый ими шифр, известный как ADFGVX и используемый для шифровки самых важных сообщений, предназначенных для отправки на фронт, несмотря на его предполагаемую неуязвимость, также может быть взломан криптоаналитиками противника. Двойной провал немецких кодов Первой мировой войны заставил всех понять необходимость более надёжного шифрования. Этой цели можно было достичь, затруднив работу криптоаналитиков, то есть сделав дешифровку более трудной.

НЕДАЛЕКО ОТ ПАРИЖА

В июне 1918 года немецкие войска готовились к наступлению на французскую столицу. Союзникам было очень важно перехватить сообщения противника, чтобы выяснить направление удара. Немецкие сообщения, предназначенные для фронтовых частей, шифровались с помощью шифра ADFGVX, который в Берлине считали не поддающимся взлому.

Наш интерес к этому шифру исходит из того, что в нём сочетаются алгоритм подстановки и алгоритм транспозиции. Это один из самых сложных методов классической криптографии. Немцы начали применять его в марте 1918 года. Узнав о существовании этого шифра, французы тут же попытались его взломать. К счастью, в центральном шифровальном бюро работал талантливый криптоаналитик Жорж Пэйнвин. Упорная работа шла день и ночь.

Вечером 2 июня 1918 года Пэйнвину удалось расшифровать первое послание. Это был приказ фронтовым частям:

«Поторопитесь с доставкой боеприпасов. Даже днём, если противник не сможет заметить».

Также удалось понять, что сообщение отправлено из какого-то местечка между Монтдидье и Компьенем, примерно в 80 километрах к северу от Парижа.

Этот успех Пэйнвина позволил французам сорвать атаку противника и остановить наступление немцев.

Как уже говорилось, шифр ADFGVX состоит из двух частей: подстановка и транспозиция. Первый этап — подстановка. Мы используем таблицу, в первой строке и первом столбце которой стоят буквы ADFGVX. Оставшиеся ячейки таблицы наугад заполняются тридцатью шестью знаками: двадцать шесть букв алфавита и цифры от 0 до 9. Расположение знаков составляет ключ к шифру, и получателю явно необходима эта информация для понимания содержания послания.

	A	D	F	G	V	X
A	O	P	F	0	Z	C
D	G	3	B	H	4	K
F	A	1	7	J	R	2
G	5	6	L	D	E	T
V	V	M	S	N	Q	I
X	U	W	9	X	Y	8

Шифр состоит из перевода каждого знака послания в координаты, используя буквы из группы ADFGVX. Первая координата — это буква, которая соответствует строке, вторая соответствует столбцу. Например, если мы хотим зашифровать число 9, то получим XF. Послание Target is Paris («цель — Париж») будет зашифровано следующим образом:

T	a	r	g	e	t	i	s	P	a	r	i	s
GX	FA	FV	DA	GV	GX	VX	VF	AD	FA	FV	VX	VF

До этого момента мы имели дело с простой подстановкой, и для расшифровки этого послания было бы достаточно частотного анализа. Однако шифрование включает и второй этап — транспозицию.

Транспозиция зависит от ключевого слова, о котором предварительно договорились отправитель и получатель.

Этот этап шифрования производится следующим образом. Во-первых, мы составляем таблицу с таким количеством столбцов, сколько букв насчитывает ключевое слово, и заполняем ячейки зашифрованным текстом. Буквы ключевого слова пишутся в верхней строке новой таблицы. В нашем примере ключевым словом будет ВЕТА.

Мы создаём новую таблицу, в которой первая строка состоит из ключевого слова, а следующие строки содержат буквы, полученные путём кодирования послания при помощи

подстановки (замены). Любая пустая ячейка заполняется цифрой ноль, которая, как мы видим из первой таблицы, обозначается AG.

Таким образом, чтобы применить второй этап к нашему посланию *Target in Paris*, мы вначале вспоминаем, что шифрование путём подстановки (замены) дало нам:

GX	FA	FV	DA	GV	GX	VX	VF	AD	FA	FV	VX	VF
----	----	----	----	----	----	----	----	----	----	----	----	----

Затем мы используем ВЕТА в качестве ключевого слова, и у нас появляется новая таблица.

B	E	T	A
G	X	F	A
F	V	D	A
G	V	G	X
V	X	V	F
A	D	F	A
F	V	V	X
V	F	A	G

Мы продолжаем, используя транспозиционный шифр, и изменяем положение столбцов, чтобы буквы ключевого слова расположились в алфавитном порядке.

Это даёт нам следующую таблицу.

A	B	E	T
A	G	X	F
A	F	V	D
X	G	V	G
F	V	X	V
A	A	D	F
X	F	V	V
G	V	F	A

Получается зашифрованное послание, которое пишется по столбцам — первый столбец сверху вниз, второй столбец сверху вниз и т. д. В нашем примере получается:

A A X F A X G G F G V A F V X V V X D V F F D G V F V A

Как мы видим, послание состоит из очевидно взятых наугад букв A, D, F, G, V и X. Немцы выбрали эти шесть букв, потому что они сильно отличаются друг от друга по звуку при отправке с помощью азбуки Морзе. Это помогало получателю с гораздо большей лёгкостью определять возможные ошибки при передаче. Более того, поскольку послания состояли всего из шести букв, передача по телеграфу была несложной даже для неопытных телеграфистов. Если мы обратимся к таблице с азбукой Морзе, то увидим следующие коды для каждой из букв шифра ADFGVX:

A = • —
D = — • •
F = • • — •
G = — — •
V = • • • —
X = — • • —

Получателю требуется только изначальная базовая таблица, в которой наугад расставлены буквы и цифры, и ключевое слово для того, чтобы расшифровать послание.

МАШИНА ЭНИГМА

В 1919 году немецкий изобретатель и предприниматель Артур Шербиус запатентовал машину, специально разработанную для обеспечения абсолютно надежной связи.

Ее название — Enigma (загадка) — с тех пор сделалось синонимом военных секретов. Как мы увидим ниже, «Энигма», несмотря на очевидную сложность, по сути представляла собой усовершенствованную версию диска Альберти. Из-за относительной легкости в использовании и из-за сложности получающегося в результате шифра немецкое правительство выбрало «Энигму» для шифрования большей части военных сообщений в период Второй мировой войны.

Неудивительно, что расшифровка кода, используемого в «Энигме», сделалась приоритетной задачей для союзников, воюющих против нацистской Германии. И когда успех наконец был достигнут, перехваченные и расшифрованные союзными разведками послания сыграли решающую роль в приближении конца войны.

История взлома кода «Энигма» очень увлекательна. В этом деле участвовали разведывательные службы Польши и Великобритании, среди героев можно назвать гениального математика Алана Тьюринга, человека, который считается отцом информатики и теории искусственного интеллекта.

Сражение с кодом «Энигма» также привело к созданию первого цифрового компьютера и может считаться самым впечатляющим эпизодом в долгой и яркой истории военного криптоанализа.

Сама по себе «Энигма» представляла электромагнитный аппарат, внешне напоминающий пишущую машинку. Особенным его делали роторы, которые меняли свое положение при каждом нажатии клавиши, причем таким образом, что даже если последовательно вводилась одна и та же буква исходного текста, то каждый раз она кодировалась по-другому. Физический процесс шифрования был относительно простым.

Отправитель устанавливал штекеры и роторы машины в соответствии с исходным положением, которое указывалось в специальной инструкции по шифрованию, действовавшей в то или иное время (эти инструкции регулярно менялись).

Затем он печатал первую букву исходного текста, и машина автоматически выдавала альтернативную букву, которая появлялась на световой панели, — это была первая буква зашифрованного послания.

Переключатель первого ротора обеспечивал поворот (сдвиг), который переставлял ротор в одно из двадцати шести возможных положений. Измененное положение ротора изменяло шифр, и тогда оператор вводил вторую букву и так далее.

Для расшифровки послания было достаточно ввести знаки полученного зашифрованного текста в другую машину «Энигма» при условии, что исходные параметры на второй машине были такими же, как на шифрующей.

Используя представленную схему-рисунок, мы можем представить в очень упрощенном виде схему шифровального механизма «Энигмы». Как мы видим, когда ротор машины «Энигма» находится в исходном положении, каждая буква исходного послания заменяется другой, кроме А, которая остается неизменной. После шифрования первой буквы ротор делает поворот на одну треть. В этом новом положении буквы теперь заменяются другими, отличными от первого шифрования. Процесс завершается на третьей букве, после чего ротор возвращается в исходное положение, и последовательность шифрования повторяется. В стандартной машине «Энигма» переключения роторов дают двадцать шесть положений для каждой из букв алфавита. Следовательно, один ротор может провести двадцать шесть различных шифровок.

Исходное положение ротора — это ключ. Для увеличения количества возможных ключей «Энигма» изначально имела до трёх роторов, механически соединённых друг с другом.

Поэтому, когда первый ротор завершал оборот, в дело вступал следующий, и так далее, пока не совершали полный оборот все роторы. В целом это давало $26 \times 26 \times 26 = 17576$ возможных шифров. В дополнение к этому разработка Шербиуса позволяла менять порядок переключений, и таким образом количество кодов, как мы увидим ниже, ещё больше увеличивалось.

Помимо роторов в механизме «Энигмы» имелась ещё и коммутационная панель, расположенная между первым ротором и клавиатурой. Коммутационная панель позволяла менять местами пары букв перед подключением к ротору,

и таким образом к шифру добавлялось ещё значительное количество кодов.

У стандартной машины «Энигма» было шесть кабелей, которые могли менять местами до шести пар букв. Таким образом А меняется местами с С, В с А, а С с В.

С добавлением коммутационной панели упрощённая версия «Энигмы» с тремя буквами будет функционировать следующим образом. Сколько же дополнительных кодов обеспечивало кажущееся тривиальным добавление коммутационной панели? Нам нужно рассмотреть количество способов соединения шести пар букв, выбранных из группы в двадцать шесть букв.

Возможное количество трансформаций n пар букв из алфавита из N букв определяется по следующей формуле:

$$\frac{N!}{(N-2n)!n!2}$$

В нашем примере $N = 26$ и $n = 6$.

Это дает нам 100391791500 комбинаций!



Восьмирольная версия
«Энигмы»

Таким образом, общее количество шифров, которые обеспечивает машина «Энигма» с тремя роторами на двадцать шесть букв каждый и коммутационной панелью с шестью кабелями, следующее:

Если считать вращения переключателей роторов, $263 = 26 \times 26 \times 26 = 17576$ комбинаций.

Точно так же три ротора (1, 2, 3) могут заменять друг друга, и переключения с одного на другой могут происходить в следующем порядке: 1—2—3, 1—3—2, 2—1—3, 2—3—1, 3—1—2, 3—2—1. Это даёт нам ещё шесть возможных дополнительных комбинаций.

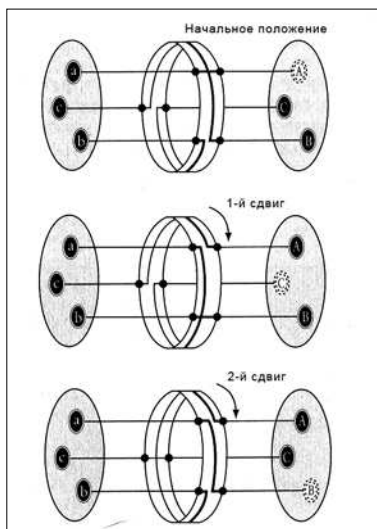
Наконец, мы подсчитали, что изменение установки шести кабелей на коммутационной панели по сравнению с исходной даёт нам 100391791500 дополнительных шифров.

Общее количество шифров, получаемых в результате различных специфических комбинаций, составляет

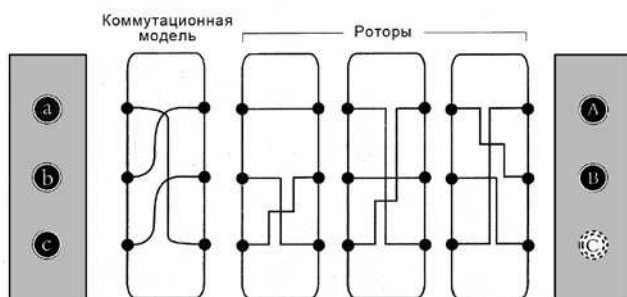
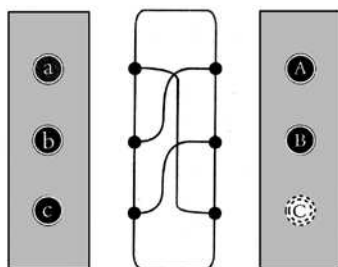
$$6 \times 17576 \times 100391791500 = 10586916764424000$$

Поэтому машины «Энигма» были способны шифровать текст, используя более десяти тысяч триллионов различных

Для простоты на этом рисунке используются роторы с алфавитом всего из трёх букв. В результате у каждого ротора имеется только три возможных положения вместо двадцати шести в реальной машине



На иллюстрации
показана работа
перестановочной
коммутационной панели,
опять в упрощённой
форме — только
с тремя буквами и тремя
кабелями



комбинаций. Правители Германии верили в то, что их связь имеет защиту самого высокого уровня и абсолютно надёжна.

Это была гигантская ошибка.

ВЗЛОМ ШИФРА МАШИНЫ «ЭНИГМА»

Любой ключ машины «Энигма» вначале указывал на конфигурацию коммутационной панели для каждой из возможных перестановок букв — например, В/З, F/Y, R/C, Т/Н, Е/О и L/J. Это означало, что первый кабель переставляет буквы В и Z и так далее.

Во-вторых, ключ показывал порядок роторов (например, 2–3—1), и, наконец, ключ включал исходное положение роторов (например, R, V, В, что означало, какая буква стоит на исходной позиции). Эти установки собирались в справочниках шифров или инструкциях по использованию кодов, и сами инструкции тоже передавались в зашифрованном виде. Они могли меняться каждый день или когда того требовали обстоятельства.

К примеру, определённые ключи предназначались для определённых видов посланий. Чтобы избежать повторения одного и того же кода в течение дня, на протяжении которого могли отправляться тысячи сообщений, операторы «Энигмы» использовали ряд хитроумных уловок для передачи новых кодов ограниченного использования — без необходимости менять всю инструкцию с общими кодами.

Таким образом, диспетчер отправлял сообщение из шести букв, закодированное в соответствии с применимым в тот день кодом, которое в действительности представляло собой новую группу исходных положений для роторов, например, Т — Y — J (для большей надёжности отправитель кодировал эти указания дважды, отсюда и получается шесть букв). Затем он кодировал настоящее сообщение в соответствии с этим новым указанием.

Получатель получал сообщение, которое не мог расшифровать, пользуясь кодом того дня, но знал, что первые шесть букв — это фактически инструкция для того, чтобы поставить роторы в другое положение. Получатель делал это, причём коммутационная панель и порядок использования роторов оставались неизменными. Затем он мог правильно расшифровать сообщение.

Союзники получили первую ценную информацию об «Энигме» в 1931 году от немецкого шпиона Ганса-Тило Шмидта. Она состояла из различных руководств по практическому



Мариан Адам Реевский

использованию машины. Контакт с Шмидтом установила французская разведслужба, в дальнейшем французы поделились информацией с польскими коллегами. Польский отдел криптоанализа, *Biuro Szyfrów* (шифровальное бюро), принялся за работу с документами Шмидта и также заполучил несколько машин «Энигма», похищенных у немцев.

В польскую команду, занимающуюся взломом кодов, входила большая группа математиков, что для того времени было весьма необычным.

Среди них был талантливый и скромный молодой человек, которого звали Мариан Реевский. Он сразу же сосредоточил свои усилия на кодах из шести букв, предшествовавших в течение дня многим сообщениям, которыми обменивались немцы. Реевский высказал предложение о том, что вторые три буквы кода являлись новым шифром первых трёх, и поэтому четвёртая, пятая и шестая буквы могут дать ключ к вращению роторов.

На основании этого открытия, хотя оно и может показаться очень незначительным, Реевский сделал несколько весьма важных выводов, которые в конечном итоге привели к разгадке механизма «Энигмы». Подробности этого процесса слишком сложны, мы не станем на них здесь останавливаться, но нельзя не отметить, что через несколько месяцев Реевский снизил

количество возможных кодов, которые требовали расшифровки, с десяти тысяч триллионов до всего лишь 105456. Это было результатом анализа различных комбинаций роторов и их вращений.

Чтобы выполнить задачу, Реевский построил аппарат, который известен как «Криптологическая бомба».

Он работал подобно «Энигме» и мог копировать любое из возможных положений трёх роторов в поисках дневного кода. Уже в 1934 году Шифровальное бюро взломало «Энигму» и могло расшифровать любое послание в течение 24 часов.

Хотя немцы и не догадывались, что поляки проникли в тайну «Энигмы», они всё равно продолжали совершенствовать систему — ведь она работала уже более десяти лет. В 1938 году операторы «Энигмы» получили в добавление к трём стандартным два дополнительных ротора, а вскоре после этого появились новые модели машины с коммутационными панелями на десять кабелей.

Внезапно количество возможных кодов возросло примерно до 159 квинтильонов.

Только одно добавление двух дополнительных роторов увеличило количество возможных комбинаций установки порядка переключения с шести до 60. То есть любой из пяти роторов в первом положении (пять вариантов) умножается на один из четырёх оставшихся роторов в положении два (четыре варианта), умножается на один из трёх роторов в положении три (три варианта) $= 5 \times 4 \times 3 = 60$. Хотя в Шифровальном бюро знали, как расшифровать код, у них не было возможностей, необходимых для анализа новых конфигураций роторов.

ЭСТАФЕТУ ПРИНИМАЮТ АНГЛИЧАНЕ

Усовершенствование системы «Энигма» происходило не случайно: Германия начала экспансию в Европе с аннексии Чехословакии и Австрии и планировала вторжение в Польшу.

В 1939 году пламя войны уже полыхало в центре Европы, и Польша была завоёвана. Однако поляки успели передать имеющиеся у них машины «Энигма» и все свои наработки по теме британским союзникам. В августе британцы решили объединить свои криптоаналитические подразделения, которые до этого работали разобщённо. В результате была создана Правительственная школа кодов и шифров (англ.: Government Code and Cypher School, GC&CS).

Для размещения новой организации выбрали особняк, расположенный в пригороде Лондона, в поместье Блетчли-Парк. В состав группы, работавшей в Блетчли-Парк, вошёл и новый криптоаналитик, блестящий молодой математик из Кембриджа Алан Тьюринг. Тьюринг считался мировым авторитетом в области вычислительной техники, которая в то время ещё пребывала в зародышевом состоянии, и интересовался всеми новыми и революционными разработками.

Расшифровка кода усовершенствованных машин «Энигма» оказалась толчком для нескольких гигантских шагов в развитии вычислительной техники.

Специалисты, собранные в Блетчли-Парк, сосредоточились на коротких фрагментах зашифрованного текста, которые, как они подозревали, соответствовали сегментам исходного текста. Например, благодаря английским разведчикам, которые работали на фронтах, они знали, что примерно в шесть часов вечера каждый день немцы имеют привычку передавать кодированные сообщения о метеорологических условиях в различных местах вдоль линии фронта. Поэтому имелись основания считать, что сообщение, перехваченное вскоре после этого времени, содержит зашифрованную версию таких исходных слов, как «погода» и «дождь».

*Алан Мэтисон Тьюринг*

Тьюринг изобрёл электрическую систему, которая позволяла воспроизводить каждую из 1054650 возможных комбинаций взаимного расположения трёх роторов менее, чем за пять часов. В эту систему вводили зашифрованные слова, которые, судя по длине знаков и другим подсказкам, можно было подозревать в соответствии некоторым фрагментам исходного текста — таким, как упомянутые выше «погода» и «дождь».

Предположим, специалисты заподозрили, что обнаруженный в шифровке текст FRTGY — это зашифрованная версия слова **Regen** («Дождь»). Шифр вводили в машину, и если находилась комбинация роторов, которая в результате давала слово *regen*, то криптоаналитики понимали, что нашли коды, которые соответствуют заданной конфигурации роторов. Затем оператор вводил зашифрованный текст в реальную машину «Энигма» с роторами, установленными в соответствии с кодом. Если машина демонстрировала, например, расшифрованный фрагмент в виде слова *neger*, было ясно, что часть кода, относящегося к положению кабелей коммутационной панели, включает перестановку букв R и N. Таким образом, специалисты получали весь код.

Секреты «Энигмы» постепенно раскрывались.

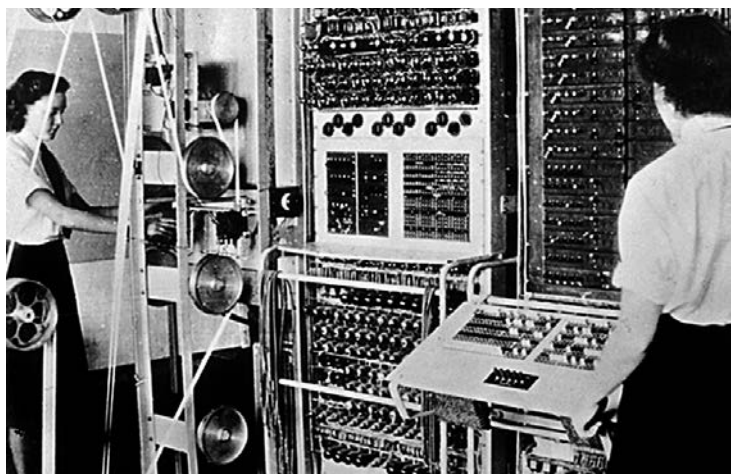
В процессе разработки и оттачивания указанных выше аналитических механизмов команда из Блетчли-Парк построила первый в истории цифровой программируемый компьютер, который называли «Колосс».

Шифры других стран

Япония разработала две свои собственные системы кодирования, известные как Purple и JN-25. Первая использовалась для дипломатической связи, а вторая — для отправки военных сообщений. В обеих системах использовались механические приспособления. Например, JN-25 включал алгоритм подстановки, который переводил знаки японского языка (до 30000 знаков) в ряд чисел в соответствии с составленными наугад таблицами из групп по пять чисел.

Несмотря на предпринятые японцами предосторожности, британцы и американцы взломали и код Purple, и код JN-25. Разведанные, полученные благодаря перехвату шифровок на Purple и JN-25, имели кодовое название Magic («Магия») и существенно повлияли на исход важнейших столкновений во время боевых действий в Тихом океане, в частности, сражение в Коралловом море и сражение у атолла Мидуэй, которые произошли в 1942 году.

Разведанные Magic использовались и для планирования стратегических миссий. К примеру, для организации перехвата и уничтожения в апреле 1943 года самолёта, на котором летел главнокомандующий японскими ВМФ адмирал Ямамото.



«Колосс», предшественник современного компьютера, созданный специалистами из Блетчли-Парк

Закодированные разговоры индейцев-навахо

В то время как США успешно воспользовались информацией, перехваченной у противника на Тихоокеанском театре военных действий, сами американские военнослужащие использовали несколько весьма необычных «кодов» для передачи сообщений. Алгоритмы шифровки работали с реально существующими словами. Эти коды — чокто, команчи, мескуаки и, наиболее часто, навахо — не были детально представлены в сложных руководствах, они вообще не были результатом разработки отдела криптографов — это просто были языки североамериканских индейских племён.

Армия США назначала радистами людей из этих племён, отправляла их в различные подразделения на фронте и приказывала передавать сообщения на их родных языках, которых не знали не только японцы, но и все остальные солдаты американских войск.

Радисты использовали неизвестность языка за пределами своих этнических групп, дополнительно заменяя значимые слова на основе официально или неофициально разработанного кода. Эти индейцы, «ведущие закодированные разговоры», служили в американских подразделениях вплоть до войны в Корее.

ШИФР ХИЛЛА

Шифры, о которых мы рассказывали до этого момента — где один знак каким-то предварительным установленным образом заменяется на другой, — как мы уже видели, всегда уязвимы и могут быть взломаны криптоаналитиками.



*Немного
нелинейной
алгебры*

В 1929 году американский математик Лестер Хилл изобрёл, запатентовал и выставил на продажу (правда, безуспешно) новую шифровальную систему, в которой использовалась модульная арифметика совместно с линейной алгеброй.

Как мы убедимся ниже, очень полезным инструментом для шифрования послания может быть матрица, использующая составление пар букв из текста и ассоциацию каждой буквы с определенным числовым значением.

Для зашифровки послания используем матрицу:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

с ограничением, что ее определителем (или детерминантом) будет 1, то есть $ad - bc = 1$. Для расшифровки послания используется обратная матрица:

$$A^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Ограничение значения детерминанта устанавливается таким образом, чтобы обратная матрица функционировала в качестве инструмента для расшифровки. Как правило, для алфавита из n букв необходимо, чтобы НОД (детерминант A , n) = 1.

Если истинно противоположное, то существование обратного числа в модульной арифметике не может быть гарантировано.

Продолжая этот пример, мы берем алфавит из 26 букв с «пробелом», для обозначения которого в данном примере используем символ @. Мы приписываем каждой букве числовое значение, как показано в следующей таблице:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	@
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Чтобы получить значения между 0 и 26, мы будем работать по модулю 27.

Процесс шифрования и расшифровки текста следующий. Вначале мы определяем шифровальную матрицу A с детерминантом 1.

Например,
$$A = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix}.$$

Дешифровочной матрицей будет обратная матрица

$$A^{-1} = \begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix}.$$

Таким образом, A будет ключом к шифру, а A^{-1} — ключом к расшифровке.

Для примера ниже мы покажем, как зашифровать послание «BOY». Буквы послания группируются парами: BO Y@. Их числовые эквиваленты в соответствии с таблицей — это пары чисел (1, 14) и (24, 26).

Затем мы умножаем матрицу A на каждую из пар чисел.

Шифрование «BO»:

$$BO = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 1 \\ 14 \end{pmatrix} = \begin{pmatrix} 43 \\ 100 \end{pmatrix} \equiv \begin{pmatrix} 16 \\ 19 \end{pmatrix} \pmod{27},$$

что, в соответствии с таблицей, соответствует буквам (Q, T).

Шифрование «Y@»:

$$Y@ = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 24 \\ 27 \end{pmatrix} = \begin{pmatrix} 102 \\ 230 \end{pmatrix} \equiv \begin{pmatrix} 21 \\ 14 \end{pmatrix} \pmod{27},$$

что соответствует буквам (V, O).

Послание «VOY» шифруется как «QTVO».

Для расшифровки осуществляется обратная операция с использованием матрицы:

$$A^{-1} = \begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix}.$$

Мы берем пару букв (Q, T) и ищем их числовые эквиваленты в таблице (16, 19). Затем умножаем их на A^{-1} и получаем:

$$\begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 16 \\ 19 \end{pmatrix} = \begin{pmatrix} 55 \\ -13 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 14 \end{pmatrix} \pmod{27},$$

что соответствует буквам (B, O).

То же самое производим со второй парой (V, O) и их числовыми значениями (21, 14), и получаем:

$$\begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 21 \\ 14 \end{pmatrix} = \begin{pmatrix} 105 \\ -28 \end{pmatrix} \equiv \begin{pmatrix} 24 \\ 26 \end{pmatrix} \pmod{27},$$

что соответствует буквам (Y, @).

Мы доказали, что ключ расшифровки работает.

Для этого примера мы рассматривали пары из двух символов. Процесс шифрования будет более надёжным, если сгруппируем буквы по три или даже по четыре. В таких случаях расчёты будут делаться с матрицами 3×3 и 4×4 соответственно. Это весьма трудоёмко, если расчётами заниматься вручную. Однако современные компьютеры способны работать с большими матрицами и соответствующими им обратными матрицами.

У шифра Хилла имеется слабое место: если у получателя есть маленький фрагмент исходного текста, можно расшифровать и всё послание. Так что поиск идеального шифра ещё далеко не закончен.

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

Задача передачи секретного сообщения между двумя пользователями — главная, но не единственная задача криптографии. Существует ещё ряд важных задач, близких по технологиям их решения. Согласованные действия пользователей, приводящие к решению подобной задачи, называются криптографическим протоколом. Приведём здесь несколько примеров подобных задач.

Протокол распределения ключей ставит своей целью генерацию общего случайного ключа между двумя пользователями с условием того, чтобы он был известен только им и никому другому. Наличие подобного ключа нужной длины практически означает возможность гарантированно секретной передачи данных.

Таким образом, задачу генерации ключа можно считать эквивалентной задаче передачи секретного сообщения.

Протокол подписания контракта решает задачу, возникающую при подписании соглашений удалёнными абонентами: два не доверяющих друг другу человека при подписании контракта не хотят допустить ситуацию, при которой один из абонентов получил подпись другого, а сам не подписался.

Протокол аутентификации работает со следующей задачей: при взаимодействии двух человек у каждого из них могут возникнуть опасения, что их собеседник не тот, за кого себя выдаёт. Задача аутентификации состоит в том, чтобы убедить собеседника в собственной личности.

Наиболее распространённой криптографической задачей является пересылка секретных данных. При этом задача передающей стороны состоит в пересылке сообщения принимающей стороне таким образом, чтобы шпион (предпринимающий определённый набор отведённых ему действий) не мог получить достаточно информации об исходном тексте сообщения. В каждом конкретном случае могут допускаться разные наборы действий шпиона, так как возможности предполагаемых перехватчиков различны: это могут быть и хулиганы, и мощные государственные структуры.

ГЛАВА 5.

ОБЩЕНИЕ ПРИ ПОМОЩИ НОЛЕЙ И ЕДИНИЦ

*«Мы почитаем всех нулями, а единицами —
себя».*

А. С. Пушкин «Евгений Онегин»

ДВОИЧНЫЙ БИНАРНЫЙ КОД

Чтобы компьютер понял и обработал информацию, она должна быть переведена с языка, на котором записана, на так называемый бинарный, или двоичный, язык. Этот язык состоит всего из двух цифр: 0 и 1.

Изобретение компьютера «Колосс» и взлом кода машины «Энигма» привели к самой масштабной коммуникационной революции в истории человечества.

Этот гигантский шаг вперёд в огромной степени опирался на развитие системы шифрования, позволившей осуществлять надёжную, эффективную и быструю связь внутри сети, которой управляют два основных оператора: компьютеры и их пользователи. Кстати, нельзя не заметить, что, используя слово «безопасность», мы имеем в виду сегодня не только криптографию и секретность. Слово это теперь имеет более широкое значение, которое охватывает ещё и понятия надёжности и эффективности.

Основу технологической революции составляет двоичная (бинарная) система. Этот суперпростой код формируется двумя знаками, нолём и единицей, и используется в вычислительной технике из-за своей способности представлять взаимодействие электронных цепей в компьютере («есть сигнал в цепи» соответствует 1, а «нет сигнала в цепи» соответствует 0).

Байты и терабайты	
Возможности памяти и объёма сохраняемых в компьютере данных измеряются множествами байтов:	
Килобайт (kB):	1024 байта
Мегабайт (MB):	1048576 байтов
Гигабайт (GB):	1073741824 байта
Терабайт (TB):	1099511627776 байтов

Каждый 0 и каждая 1 называются «бит» [термин произошёл от английского словосочетания «binary digit» (двоичное число)].

КОД ASCII

Одним из многочисленных применений двоичной системы является специфическая группа знаков, длина каждого из которых составляет восемь бит, — известная под термином «байт».

Эти алфавитно-цифровые знаки представляют собой базовые символы, которые используются при современной связи. Они называются «код ASCII» (англ.: American Standard Code for Information Interchange — американский стандартный код для обмена информацией). Из восьми нолей и единиц можно составить $2^8 = 256$ различных цифровых групп.

Код ASCII позволяет пользователям вводить текст в компьютер. Когда мы печатаем буквенно-цифровой знак, компьютер превращает его в байт — цепочку из восьми битов. К примеру, если вы наберёте на клавиатуре букву А, то компьютер превратит её в двоичное число 01000001.

Двоичные значения в ASCII даются всем обычно употребляемым знакам: 26 заглавным буквам английского алфавита, 26 строчным буквам, 10 цифрам, 7 знакам пунктуации и несколькими особым знакам. Для каждого двоичного кода знака даётся соответствующее десятичное число:

Если вы набираете, к примеру, «GOTO 2», фразу на языке программирования BASIC, то компьютер переведёт знаки в соответствующую двоичную последовательность:

В результате компьютер выдаст следующую последовательность цифр:

Символы	G	O	T	O	Пробел	2
Перевод на компьютерный язык	01000111	01001111	01010100	01001111	00100000	00110011

010001110100111101010100010011110010000000110011

**Американский стандартный код для обмена информацией
(наиболее часто используемые символы)**

Символ	Двоичн. код	Десятичн. число	Символ	Двоичн. код	Десятичн. число	Символ	Двоичн. код	Десятичн. число
Пробел	00100000	32	@	01000000	64	'	01100000	96
!	00100001	33	A	01000001	65	a	01100001	97
"	00100010	34	B	01000010	66	b	01100010	98
#	00100011	35	C	01000011	67	c	01100011	99
\$	00100100	36	D	01000100	68	d	01100100	100
%	00100101	37	E	01000101	69	e	01100101	101
&	00100110	38	F	01000110	70	f	01100110	102
'	00100111	39	G	01000111	71	g	01100111	103
(00101000	40	H	01001000	72	h	01101000	104
)	00101001	41	I	01001001	73	i	01101001	105
*	00101010	42	J	01001010	74	j	01101010	106
+	00101011	43	K	01001011	75	k	01101011	107
,	00101100	44	L	01001100	76	l	01101100	108
-	00101101	45	M	01001101	77	m	01101101	109
.	00101110	46	N	01001110	78	n	01101110	110
/	00101111	47	O	01001111	79	o	01101111	111
.	00110000	48	P	01010000	80	p	01110000	112
0	00110001	49	Q	01010001	81	q	01110001	113
1	00110010	50	R	01010010	82	r	01110010	114
2	00110011	51	S	01010011	83	s	01110011	115
3	00110100	52	T	01010100	84	t	01110100	116
4	00110101	53	U	01010101	85	u	01110101	117
5	00110110	54	V	01010110	86	v	01110110	118
6	00110111	55	W	01010111	87	w	01110111	119
7	00111000	56	X	01011000	88	x	01111000	120
8	00111001	57	Y	01011001	89	y	01111001	121
9	00111010	58	Z	01011010	90	z	01111010	122
:	00111011	59	[01011011	91	{	01111011	123
;	00111100	60	\	01011100	92		01111100	124
<	00111101	61]	01011101	93	}	01111101	125
>	00111110	62	^	01011110	94	~	01111110	126
?	00111111	63	_	01011111	95	DEL	01111111	127

ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА

В вычислительной технике используется ещё один достойный внимания код — шестнадцатеричная система. Эта система счисления оперирует с шестнадцатью уникальными числами (отсюда и название), в отличие от обычной, где используется десять цифр. Шестнадцатеричная система — второй компьютерный язык после двоичного.

В чём её примечательность?

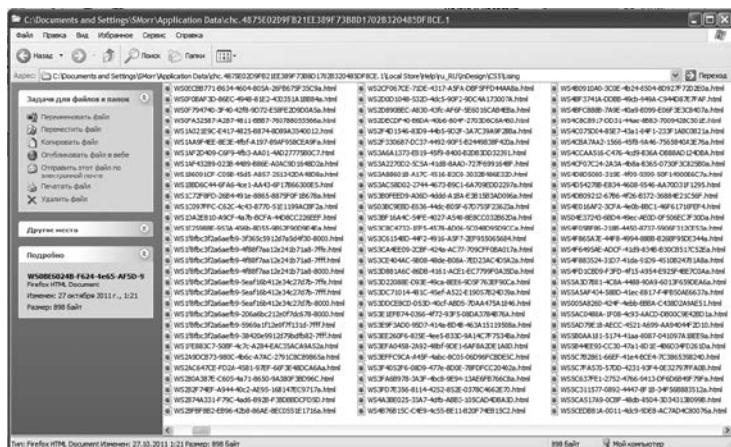
Вспомним, что базовая единица в работе компьютера, байт, состоит из восьми битов, что дает вплоть до $2^8 = 256$ различных комбинаций 0 и 1.

$$2^8 = 2^4 \times 2^4 = 16 \times 16.$$

Другими словами, одному байту равна комбинация двух шестнадцатеричных цифр.

Шестнадцать цифр шестнадцатеричной системы — это традиционные 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 и ещё шесть, установленные по соглашению A, B, C, D, E, F.

Чтобы считать в шестнадцатеричной системе, мы делаем следующие замены:



В именах служебных файлов, автоматически генерируемых операционной системой современного компьютера, используется шестнадцатеричная система

От 0 до 15: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

От 16 до 31: 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F

От 32 и далее: 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C...

В именах служебных файлов, автоматически генерируемых операционной системой современного компьютера, используется шестнадцатеричная система.

Шестнадцатеричные цифры не различают прописные и строчные буквы (1E означает то же самое, что и 1e). В приводимой ниже таблице показаны первые 16 двоичных чисел и их шестнадцатеричные эквиваленты:

Перевод двоичных чисел в шестнадцатеричные

Двоичные	Шестнадцатеричные
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Чтобы перейти от двоичного кода к шестнадцатеричному, мы группируем биты в четыре группы по четыре справа и завершаем перевод в соответствии с предыдущей таблицей. Если количество двоичных чисел не является множеством из четырёх, мы заполняем разницу 0 слева. Чтобы перейти от шестнадцатеричного кода к двоичному, мы превращаем каждое шестнадцатеричное число в его двоичный эквивалент, как в приводимом ниже примере.

$9F2_{16}$ — это формальное обозначение шестнадцатеричного числа (выражаемого подстрочным индексом 16). Вспомните, что соответствующее двоичное число — это:

9	F	2
1001	1111	0010

Таким образом $9F2_{16} = 100111110010_2$ (обратите внимание, что подстрочный индекс 2 указывает на то, что число выражено в двоичной системе).

Давайте проведём обратный процесс: в 11101001102 десять цифр. Поэтому мы завершаем число двумя нолями слева, чтобы было 12 цифр, которые мы можем сгруппировать по четыре.

Мы преобразовываем:

$$1110100110_2 = 0011\ 1010\ 0110_2 = 3A6_{16}.$$

Какие взаимоотношения существуют между шестнадцатеричными знаками и кодами ASCII? Каждый код ASCII включает восемь битов (один байт) информации, поэтому пять ASCII знаков включают 40 битов (пять байтов), и, поскольку шестнадцатеричный знак включает четыре бита, мы делаем вывод, что пять ASCII знаков — это 10 шестнадцатеричных знаков.

Давайте рассмотрим пример с кодированием фразы шестнадцатеричным кодом. К примеру, попробуем проделать это с названием компании «NotRealCo Ltd», последовательно выполняя следующие шаги:

1. Переводим «NotRealCo Ltd» в двоичную версию по стандарту ASCII.
2. Группируем цифры по четыре (если длина двоичного ряда не кратна четырём, то добавляем слева ноли).
3. Изменяем код с помощью таблицы преобразования двоичных и шестнадцатеричных чисел.

Результат работы представлен в таблице:

PRO криптографию (Символ — машина — квант)

Сообщение	N	o	t	R	e	a	l
Эквивалент в ASCII	01001110	01101111	01110100	01110010	01100101	01100001	01101100
Перевод в 16-чный код	4E	6F	74	72	65	61	6C

Сообщение (продолжение)	C	o		L	t	d
Эквивалент в ASCII	01100011	01101111	00100000	01001011	01110100	01100100
Перевод в 16-чный код	63	6F	20	4B	74	64

Таким образом, название компании «NotRealCo Ltd», зашифрованное в шестнадцатеричной системе, выглядит следующим образом:

4E 6F 74 72 65 61 6C 63 6F 20 4B 74 64

СИСТЕМЫ СЧИСЛЕНИЯ И ЗАМЕНА ОСНОВАНИЯ

Говорят, что система счисления из n чисел также имеет основание n . На человеческих руках десять пальцев, и очень вероятно, что именно поэтому была изобретена десятичная система: считали при помощи пальцев.

Десятичное число типа 7392 представляет количество, равное 7 тысячам, 3 сотням, 9 десяткам и 2 единицам. Тысячи, сотни, десятки, единицы — это порядки (степени) основания системы счисления, в данном случае 10. Поэтому число 7392 может быть выражено ещё и следующим образом:

$$7392 = 7 \cdot 10^3 + 3 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0.$$

Однако принято писать только коэффициенты при степенях (7, 3, 9 и 2).

Кроме десятичной системы существует много других систем счисления (на самом деле их общее количество попросту бесконечно). В этой книге мы уделяем особое внимание двум системам: двоичной (основанной на числе 2) и шестнадцатеричной (основанной на числе 16). В двоичной системе счисления у коэффициентов могут быть только два значения: 0 и 1. Цифры двоичных чисел — это коэффициенты при степенях по основанию 2. Таким образом, число 11011_2 также может быть записано и следующим образом:

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

Если мы вычисляем выражение справа от знака равенства, то получим 27, что является десятичной формой двоичного числа 11011_2 . Для обратного процесса мы последовательно делим десятичное число на 2 (основу двоичной системы) и отмечаем остатки, пока не получим коэффициент 0. Двоичное число будет иметь конечный коэффициент в виде своей первой цифры, за ней будут следовать остатки, начиная с последнего в списке.

Чтобы визуализировать процесс, запишем число 76 в двоичной системе:

76 при делении на 2 имеет коэффициент 38 и остаток 0

38 при делении на 2 имеет коэффициент 19 и остаток 0

19 при делении на 2 имеет коэффициент 9 и остаток 1

9 при делении на 2 имеет коэффициент 4 и остаток 1

4 при делении на 2 имеет коэффициент 2 и остаток 0

2 при делении на 2 имеет коэффициент 1 и остаток 0

Поэтому число 76, записанное в двоичной системе, будет выглядеть как 1001100_2 .

Конвертирование числа, выраженного в одной системе счисления, в другую называется изменением основания.

КАК ИЗМЕРИТЬ ИНФОРМАЦИЮ

Коды, о которых вкратце рассказано выше, делают возможным безопасную и эффективную связь между компьютерами, программами и пользователями. Однако этот онлайн-язык основывается на общей теории информации, которая является фундаментом самой связи.

Первый шаг в формулировании этой теории является настолько элементарным, что иногда и в голову не придёт: как можно измерить информацию?

Тем не менее простая фраза «Приложение объёмом в два килобайта» рождена целым рядом блестящих открытий, которые начинаются со статьи, опубликованной в 1948 году американцем Клодом Э. Шенноном и озаглавленной «Математическая теория связи». В этой основополагающей новаторской статье Шеннон предложил пользоваться единицей измерения количества информации, которую он назвал бит.

Работа Шеннона была порождена проблемой, знакомой многим современным читателям. Какой способ шифровки послания лучше всего использовать, чтобы оно не пострадало во время пересылки? Шеннон пришёл к выводу, что код, который всегда будет защищать от потери информации, определить невозможно. Иными словами, когда передаётся информация, ошибки неизбежны.



*Клод Шеннон:
гений без
«нобелевки»*

Однако этот вывод не отменил поиск стандартов кодификации, которые, даже если и не предотвращают все возможные проблемы, могут, по крайней мере, обеспечить самый высокий уровень надёжности.

При цифровой передаче информации, после того как сообщение было создано отправителем (это, кстати, может быть и не человек, а, к примеру, компьютер), оно шифруется в двоичной системе и поступает в канал связи. Последний состоит из компьютера-отправителя, компьютера-получателя и физического переносчика сигнала, который может быть и кабелем, и беспроводным соединением (радиоволны, инфракрасное излучение и т. п.).

Путешествие по каналу — самый чувствительный процесс, потому что в нём сообщение может быть подвержено всем

видам помех, включая смешивание с другими сигналами, отрицательное влияние температур в среде, ослабление сигнала по мере прохождения сквозь среду проводника.

Эти источники помех называются шумом.

Чтобы свести к минимуму влияние шума, нужно не только защитить соединение, но также и установить способ определения ошибок и исправления их, когда они возникают.

Один из способов называется дублированием.

Дублирование состоит из повторения, по определённым критериям, определённых характеристик послания. Вот пример, который поможет прояснить процесс.

Давайте представим текст, в котором каждое слово состоит из четырёх битов, в целом 16 слов ($2^4 = 16$), каждое имеет вид $a_1 a_2 a_3 a_4$. Перед тем как отправить сообщение, мы добавляем к слову три дополнительных бита $c_1 c_2 c_3$, чтобы закодированное сообщение, поступающее в канал связи, имело форму $a_1 a_2 a_3 a_4 c_1 c_2 c_3$.

Элементы $c_1 c_2 c_3$ обеспечат безопасность сообщения — они называются кодами с контролем четности — и образуются следующим образом:

$c_1 =$	0, если $a_1 + a_2 + a_3$ чётное
	1, если $a_1 + a_2 + a_3$ нечётное
$c_2 =$	0, если $a_1 + a_2 + a_4$ чётное
	1, если $a_1 + a_2 + a_4$ нечётное
$c_3 =$	0, если $a_2 + a_3 + a_4$ чётное
	1, если $a_2 + a_3 + a_4$ нечётное

Мы припишем следующие коды сообщению 0111:

Поскольку $0 + 1 + 1 = 2$ чётное, число $c_1 = 0$

Поскольку $0 + 1 + 1 = 2$ чётное, число $c_2 = 0$

Поскольку $0 + 1 + 1 = 2$ чётное, число $c_3 = 0$

Следовательно, сообщение 0111 будет передаваться в виде 0111001. Из следующих 16 «слов» мы таким образом получаем таблицу:

Исходное сообщение	Отправленное сообщение
0000	0000000
0001	0001011
0010	0010111
0100	0100101
1000	1000110
1100	1100011
1010	1010001
1001	1001101
0110	0110010
0101	0101110
0011	0011100
1110	1110100
1101	1101000
1011	1011010
0111	0111001
1111	1111111

Предположим, что принимающая система получает сообщение 1010110. Предположим, что принимающая система получает сообщение 1010110. Обратите внимание, что этой комбинации 0 и 1 нет в ряду возможных сообщений, поэтому она должна являться ошибкой передачи. Чтобы попытаться исправить ошибку, система сравнивает каждую цифру с рядами цифр возможных сообщений, чтобы найти наиболее вероятную альтернативу. Для этого система проверяет, сколько цифр кажутся неправильными, как показано ниже:

Возможное сообщение	0000000	0001011	0010111	0100101	1000110
Полученное сообщение	1010110	1010110	1010110	1010110	1010110
Количество различных цифр на каждой позиции	4	5	2	5	1

Возможное сообщение	1100011	1010001	1001101	0110010	0101110
Полученное сообщение	1010110	1010110	1010110	1010110	1010110
Количество различных цифр на каждой позиции	4	3	4	3	4

Возможное сообщение	0011100	1110100	1101000	1011010	0111001	1111111
Полученное сообщение	1010110	1010110	1010110	1010110	1010110	1010110
Количество различных цифр на каждой позиции	3	2	5	2	6	3

Ошибочное слово (1010110) отличается от другого слова (1000110) одной цифрой. Поскольку разница наименьшая, система предложит получателю эту вторую, исправленную версию.

Аналогично работает программа в текстовом процессоре, проверяющая орфографию.

Когда программа замечает слово, отсутствующее во внутреннем словаре, она предлагает ряд ближайших альтернатив. Количество позиций, которыми сообщение, понимаемое как последовательность знаков, отличается от другого, известно как «расстояние между двумя последовательностями». Этот специфический механизм обнаружения и исправления ошибок был предложен американцем Ричардом Хэммингом, современником Клода Шеннона.

В информационных технологиях, как и в любой другой области человеческой деятельности, мало обнаружить возможные ошибки — их требуется ещё и исправить. Если, как в последнем примере, в коде имеется только один кандидат на минимальное расстояние, то проблема достаточно проста. Если мы обозначим символом t минимальное количество раз, когда 1 появляется в последовательности (выпуская последовательность, которая полностью состоит из 0), то мы можем подтвердить, что:

Если t чётное, то мы можем исправить $\frac{t-1}{2}$ ошибок.

Если t нечётное, мы можем исправить $\frac{t-2}{2}$ ошибок.

В случае, когда обнаружение ошибок — наша единственная цель, максимальное их количество, которое мы можем определить, составит $t - 1$.

В 16-знаковом языке, который рассматривался выше, $t = 3$, из чего мы делаем вывод, что механизм способен обнаружить $3 - 1 = 2$ ошибки и исправить $(3 - 1) : 2 = 1$ ошибку.

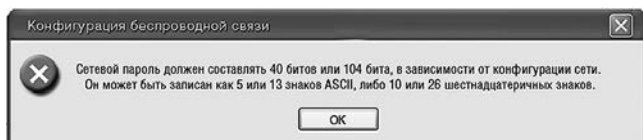
ПРОТОКОЛ ДЛЯ БЕЗОПАСНОЙ ПЕРЕДАЧИ

В 1997 году был введён протокол для безопасной передачи информации через беспроводные сети, который называли WEP (от англ. «Wired Equivalent Privacy» — означает «Обеспечение безопасности кабельного эквивалента»). В обиходе — протокол защиты данных WEP, или просто WEP. Этот протокол включает алгоритм шифровки, который называется RC4, и два типа кодов из 5 и 13 знаков ASCII соответственно. Таким образом, мы имеем дело с кодами из 40 или 104 битов или, в качестве альтернативы, из 10 или 26 шестнадцатеричных знаков.

**5 буквенно-цифровых знаков = 40 бит =
10 шестнадцатеричных знаков**

**13 буквенно-цифровых знаков = 104 бита =
26 шестнадцатеричных знаков**

Провайдер, обеспечивающий соединение, поставяет коды, хотя пользователь может их менять. Перед установлением связи компьютер спрашивает ключ. В следующем диалоговом окне мы видим сообщение об ошибке и о ключе WEP с указанием его длины в битах, знаках ASCII и шестнадцатеричных знаках:



На самом деле реальные ключи (пароли) длиннее. Начав с тех, которые предоставляет пользователь, RC4 образует новый ключ с большим количеством битов, именно он и используется для шифрования передачи. Это криптография открытого ключа, более подробно мы объясним её в главе 6. Пользователь, который желает заменить ключ, должен помнить, что ключ из десяти шестнадцатеричных знаков будет более надёжным, чем ключ из пяти буквенно-цифровых знаков, хотя размер в битах один и тот же.

ГЛАВА 6.

КОДИРОВАНИЕ В ПРОМЫШЛЕННЫХ И ТОРГОВЫХ МАСШТАБАХ

«Только атаки дилетантов нацелены на машины; атаки профессионалов нацелены на людей».

Брюс Шнайер

«Хотя сухие коды банков, супермаркетов и других крупных игроков в экономике вызывают в душе человека гораздо меньше очарования, чем криптография или бинарная математика, но именно они являются столпами, поддерживающими устойчивость современного общества. Главной целью использования этих кодов является обеспечение уникальности и точной идентификации продукции, будь то банковские счета, книги или яблоки».

Виктор де Касто

КРЕДИТНЫЕ КАРТЫ

Дебетовые и кредитные карты, которые предлагают крупные банки и торговые сети, по сути идентичны заданным группам чисел и обрабатываются при помощи определённого алгоритма и системы подтверждения, все они базируются на нашей старой подруге, модульной арифметике.

Большинство карт имеют 16 цифр, которые состоят из числового ряда между 0 и 9. Числа сгруппированы по 4 цифры так, чтобы их было легче считать. Для наших целей мы зададим их следующим образом:

ABCD EFGH IJKL MNOP

Каждая группа цифр кодирует какую-то часть информации: первая группа (ABCD) соответствует ID (идентификационному номеру) банка (или другой организации, которая предоставляет услугу). У каждого банка имеется собственный номер, который может варьироваться в соответствии с континентом и который также связан с маркой и условиями карты. Например, в случае VISA и некоторых известных банков первые четыре цифры следующие:

А В С D	Провайдер
4940	Ситибанк
4024	Банк Америки
4128	Ситибанк (США)
4302	HSBC («Эйч-эс-би-си»)

Пятая цифра (Е) соответствует типу карты и указывает, какое финансовое учреждение управляет счётом:

Тип	Провайдер
3	American Express
4, 0, 2	Visa
5, 0	MasterCard
6	Discover

Как мы видим, это не жёсткое правило.

Следующие десять цифр (FGH IJKL MNO) — уникальный идентификатор для каждой карты. Эта идентификация не только даёт номер для ссылок на счёт каждого клиента, но также связана и с видом карты — классическая, золотая, платиновая и т. д., а также указывает на кредитный лимит, процентные ставки и дату окончания срока действия.

Наконец, есть контрольная цифра (Р), которая связана с предыдущими цифрами в соответствии с алгоритмом Луна, названным в честь Ханса Петера Луна (1896-1964), инженера, который его разработал.



Алгоритм
Луна

Для карты с 16 цифрами этот алгоритм работает следующим образом:

1. Для каждой цифры, занимающей нечётную позицию, начиная с первой цифры слева, мы рассчитываем новую цифру путём умножения на 2. Если результат этого умножения больше 9, мы складываем две цифры нового числа (или совершаем операцию, эквивалентную вычитанию 9). Например, если мы получаем 18, то складываем $1 + 8 = 9$ (или производим вычитание $18 - 9 = 9$).
2. Затем складываем все числа, которые были рассчитаны таким образом, и цифры, расположенные на чётных позициях (включая последнюю контрольную цифру).

3. Если общая сумма кратна 10 (то есть её значение 0 по модулю 10), числа на карте действительны. Обратите внимание, что именно последняя контрольная цифра в конце концов даёт сумму, кратную 10.

К примеру, в случае карты, имеющей номер

1234 5678 9012 3452

в соответствии с алгоритмом Луна:

Шаг первый:

$$1 \cdot 2 = 2$$

$$3 \cdot 2 = 6$$

$$5 \cdot 2 = 10 \Rightarrow 1 + 0 = 1$$

$$7 \cdot 2 = 14 \Rightarrow 1 + 4 = 5$$

$$9 \cdot 2 = 18 \Rightarrow 1 + 8 = 9$$

$$1 \cdot 2 = 2$$

$$3 \cdot 2 = 6$$

$$5 \cdot 2 = 10 \Rightarrow 1 + 0 = 1$$

Шаг второй:

$$2 + 6 + 1 + 5 + 9 + 2 + 6 + 1 = 32$$

$$2 + 4 + 6 + 8 + 0 + 2 + 4 + 2 = 28$$

$$2 + 28 = 60$$

В результате получилось 60, что кратно 10. Поэтому карта действительна.

Ещё одним способом применения алгоритма Луна является следующий: номер карты ABCD EFGH IJKL MNOP правильный, если двойное значение суммы цифр на нечётной позиции и сумма цифр на чётной позиции плюс количество цифр на нечётной позиции, которые больше 4, кратны 10.

$$2 \cdot (A+C+E+G+I+K+M+O) + (B+D+F+H+J+L+N+P) + \text{количество цифр на нечётной позиции, которые больше 4} \equiv 0 \pmod{10}.$$

Применим вторую версию алгоритма к рассмотренному выше примеру:

1234567890123452

$$2(1 + 3 + 5 + 7 + 9 + 1 + 3 + 5) + (2 + 4 + 6 + 8 + 0 + 2 + 4 + 2) + 4 = \\ = 100 \equiv 0 \pmod{10}$$

Убеждаемся, что число, взятое для примера, — действующий номер кредитной карты и что кажущиеся взятыми наугад кредитные коды следуют строгому математическому стандарту.

А возможно ли восстановить цифру, которой не хватает в коде карты? Да, если мы имеем дело с действующей кредитной картой. Давайте определим значение Z в номере 4539 4512 03Z8 7356.



Diner's Club

Начнём процесс, умножая на 2 числа на нечётных позициях ($4-3-4-1-0-Z-7-5$), сведя их к единичной цифре.

$$4 \cdot 2 = 8$$

$$3 \cdot 2 = 6$$

$$4 \cdot 2 = 8$$

$$1 \cdot 2 = 2$$

$$0 \cdot 2 = 0$$

$$Z \cdot 2 = 2 \cdot Z$$

$$7 \cdot 2 = 14; 14 - 9 = 5$$

$$5 \cdot 2 = 10; 10 - 9 = 1.$$

Теперь складываем цифры на чётных позициях и новые цифры из нечётных позиций и получаем:

$$30 + 41 + 2Z = 71 + 2Z.$$

При этом $71 + 2Z$, как мы знаем, должно быть кратно 10.

Если бы значение Z было больше 4 (и меньше 10), $2Z$ было бы числом между 10 и 18. Значение $2Z$, сведённое к единичной цифре, — это $2Z - 9$, таким образом, предыдущая сумма будет $71 + 2Z - 9$. Единственное значение Z , которое даст выражение кратного 10 числа, — это 9. Если, наоборот, Z меньше или равно 4, мы видим, что нет значения, которое доказывало бы, что $71 + 2Z$ — это кратное 10 число. Следовательно, потерянная цифра — это 9, и полный номер кредитной карты: 4539 4512 0398 7356.

ПРИМЕНЕНИЕ ПРОГРАММЫ EXCEL ДЛЯ РАСЧЕТА КОНТРОЛЬНОЙ ЦИФРЫ КРЕДИТНОЙ КАРТЫ

Номер, ассоциируемый с кредитной картой, состоит из 15 цифр плюс контрольный код. Числа сгруппированы в четыре группы по четыре цифры. Контрольная цифра рассчитывается в соответствии с приводимым ниже алгоритмом.

Microsoft Excel - Применение EXCEL.xls																												
Файл Правка Вид Вставка Формат Сервис Данные Служб. Опции Справка Adobe PDF																												
Arial Cyr = 10 = Ж А У Обычный = [шрифт] [размер] [жир] [курсив] [подчеркнут] [выровнено] [выделено] [цвет] [фон] [свойства]																												
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V							
2	Итого																											
3	Номер кредитной карты:	5	5	2	1		4	5	7	2		6	1	6	2		3	6	2	4								
4																												
5	Используемые цифры:			5	2	1		4	5	7	2		6	1	6	2		3	6	2								
6	Сумма цифр на четной позиции:			2				4			7		6		6			3		2								
7	Сумма цифр на четной позиции:																											
8	Количество цифр на четной позиции больше 4:																											
9	Сумма двух предыдущих значений:																											
10	Цифра на нечетной позиции:	5			1				5			2			1		2			6								
11	Сумма цифр на нечетной позиции:																											
12	Сумма двух предыдущих результатов плюс 1:																											
13	Остаток деления предыдущего результата на 10:																											
14	Контрольная цифра - это 0, если предыдущий результат 0, в противном случае это 10 минус предыдущий результат:																											
15																												

ПЕРВЫЕ ШТРИХКОДЫ

Первая система штрихкодов была запатентована 7 октября 1952 года американцами Норманом Вудландом и Бернардом Силвером. Первые коды сильно отличались от современных. Вместо знакомых штрихов Вудланд и Силвер предлагали концентрические круги. Это была фигура, похожая на мишень. Первое официальное использование штрихкода в магазине произошло в 1974 году в Трое, штат Огайо.

Современный штрихкод состоит из ряда чёрных штрихов (которые кодируются как 1 в двоичной системе) и пробелов между ними (которые кодируются как 0). Штрихкоды используются для идентификации продаваемых предметов. Коды обычно печатаются на этикетках и считываются с помощью оптического приспособления.



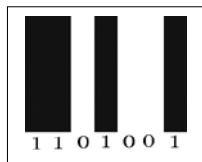
*Норман
Вудланд*

Это приспособление, подобно сканеру, ловит отражающийся свет и превращает чёрные и белые полосы в буквенно-цифровой ключ, который затем отправляется на компьютер.

Существуют многочисленные стандарты штрихкодов: Код 128, Код 39, Codabar, EAN (Европейский номер товара — он появился в 1976 году в версиях из 8 и 13 цифр) и UPC (Универсальный код продукции, который, в основном, используется в США и имеет версии из 12 и 8 цифр). Наиболее часто встречающийся код — это версия EAN из 13 цифр.

Несмотря на разнообразие стандартов, штрихкод позволяет идентифицировать любую продукцию в любой части земного шара, быстро и без большой погрешности.

*Вот так толщина штрихов и пробелов
в штрихкоде соответствует числам
двоичной системы*



ШТРИХКОД EAN-13

Код EAN — это аббревиатура от European Article Number (Европейский номер товара). Так его называли при создании, в 1976 году. Теперь он известен как Международный номер товара. Это самый распространённый стандарт штрихкода, он используется по всему миру. Коды EAN обычно состоят из тринадцати цифр, представленных чёрными штрихами и белыми пробелами, которые вместе составляют легко считываемый двоичный код.

EAN-13 представляет эти тринадцать цифр при помощи тридцати штрихов и пробелов. Цифры распределены на три части: первая состоит из двух или трёх цифр и указывает на код страны; вторая состоит из девяти или десяти цифр и идентифицирует компанию и продукцию; третья состоит только из одной цифры, она называется контрольной цифрой и предназначена для проверки правильности считывания кода сканирующим устройством.

Для кода AB CDEFG HIJKL M эти части разделяются следующим образом:

- Первые две (AB) формируют код страны, в которой зарегистрировано предприятие, выпустившее продукцию. Оно может находиться и не на территории этой страны. Например, код Великобритании — 50, код Ирландии — 539, код России — 460.
- Следующие пять (CDEFG) идентифицируют компанию, которая произвела продукцию.
- Следующие пять (HIJKL) указывают код продукции, который приписан компании.
- Последняя (M) — это контрольная цифра. Чтобы рассчитать её, нужно сложить цифры на нечётных позициях, начиная слева и не считая контрольной. К получившемуся значению затем добавляется умноженная на три сумма цифр на чётных позициях. Контрольная цифра — это значение, которое делает общую сумму, которая только что была вычислена, кратной 10.

Как видно, система контроля штрихкодов очень сильно напоминает ту, что используется для кредитных карт.



Давайте убедимся, что этот штрих-код действителен:
8413871003049

$$8 + 1 + 8 + 1 + 0 + 0 + 3(4 + 3 + 7 + 0 + 3 + 4) = 18 + 3(21) = 18 + 63 = 81$$

Правильная контрольная цифра должна быть $90 - 81 = 9$.

Математическая модель алгоритма основывается на модульной арифметике (по модулю 10) следующим образом:

Значение, рассчитанное для кода ABCDEFGHIJKLM, назовем N

$$A + C + E + G + I + K + 3(B + D + F + H + J + L) = N$$

Значение N по модулю 10 обозначим как n.

Контрольная цифра M определяется как $M = 10 - n$. В нашем примере мы имеем $81 \equiv 1 \pmod{10}$, поэтому контрольная цифра на самом деле будет $10 - 1 = 9$.

Предыдущий алгоритм может быть сформулирован и как эквивалентный способ, использующий контрольную цифру в расчетах. Следующая техника позволяет нам подтвердить действенность контрольного кода без необходимости вначале вычислять его.

$$A + C + E + G + I + K + 3(B + D + F + H + J + L) + M \equiv 0 \pmod{10}$$

Для кода образца

5701263900544

Получим

$$5 + 0 + 2 + 3 + 0 + 5 + 3 (7 + 1 + 6 + 9 + 0 + 4) + 4 = 100$$

$$100 \equiv 0 \pmod{10}$$

поэтому код действителен.

Из любопытства мы попробуем определить значение утраченного числа в штрихкоде. В приводимом ниже коде оно обозначено Z:

401332003Z497

Подставим цифры в соответствии с заданным алгоритмом:

$$4 + 1 + 3 + 0 + 3 + 4 (0 + 3 + 2 + 0 + Z + 9) + 7 = 64 + 3Z \equiv 0 \pmod{10}.$$

По модулю 10 мы получаем следующее уравнение:

$$4 + 3Z \equiv 0 \pmod{10}$$

$$3Z \equiv -4 + 0 \equiv -4 + 10 \times 1 = 6 \pmod{10}$$

Обратите внимание, что 3 имеет обратную величину, так как $\text{НОД}(3, 10) = 1$.

Поэтому мы находим, что Z должен быть 2. Поэтому правильный код будет

4013320032497.

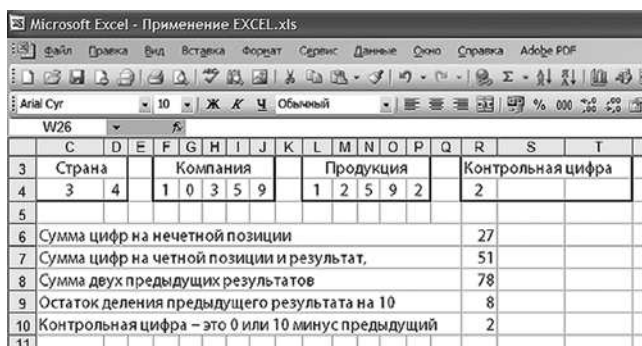
ПРИМЕНЕНИЕ ПРОГРАММЫ EXCEL ДЛЯ РАСЧЁТА КОНТРОЛЬНОЙ ЦИФРЫ КОДА EAN-13

Штрихкод типа EAN-13 — это номер, который состоит из 12 цифр плюс 13-я, контрольная цифра (КЦ).

13 цифр распределяются по четырём группам:

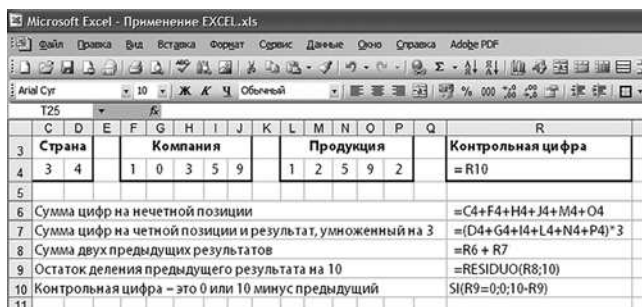
Страна		Компания					Продукция					КЦ
8	4	1	1	3	4	9	0	4	5	1	2	6

Алгоритм расчёта состоит из следующих шагов:



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
3	Страна		Компания					Продукция					Контрольная цифра					
4	3	4	1	0	3	5	9	1	2	5	9	2	2					
5																		
6	Сумма цифр на нечетной позиции															27		
7	Сумма цифр на четной позиции и результат,															51		
8	Сумма двух предыдущих результатов															78		
9	Остаток деления предыдущего результата на 10															8		
10	Контрольная цифра – это 0 или 10 минус предыдущий															2		
11																		

При использовании Excel этот алгоритм будет записан следующим образом:



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
3	Страна		Компания					Продукция					Контрольная цифра				
4	3	4	1	0	3	5	9	1	2	5	9	2	=R10				
5																	
6	Сумма цифр на нечетной позиции															=C4+F4+H4+J4+M4+O4	
7	Сумма цифр на четной позиции и результат, умноженный на 3															=(D4+G4+I4+L4+N4+P4)*3	
8	Сумма двух предыдущих результатов															=R6 + R7	
9	Остаток деления предыдущего результата на 10															=RESIDUO(R8;10)	
10	Контрольная цифра – это 0 или 10 минус предыдущий															SI(R9=0;0;10-R9)	
11																	

КОДЫ QR

В 1994 году японская компания «Denso-Wave» разработала графическую систему шифрования для идентификации деталей автомобилей на сборочном конвейере.

Система, которую назвали QR (аббревиатура от англ. Quick response, быстрое реагирование), благодаря скорости считывания сканерами, разработанными для этой цели, распространилась далеко за пределы автомобилестроительных заводов. Всего через несколько лет большая часть японских мобильных телефонов могла мгновенно считывать информацию, содержащуюся в коде.

QR представляет собой матричный код, который формируется с помощью меняющегося числа чёрных и белых квадратиков, которые, в свою очередь, расположены в форме большого квадрата. Квадратики представляют собой двоичное значение (0 или 1) и поэтому похожи на штрихкоды, хотя добавление второго измерения даёт коду способность хранить большее количество информации.



Пример QR-кода

ПРОСТЫЕ ЧИСЛА И МАЛАЯ ТЕОРЕМА ФЕРМА

Простые числа — это подгруппа натуральных чисел, которая включает все элементы большего множества, они больше 1 и не имеют других делителей, кроме себя самих и единицы. Фундаментальная теорема арифметики устанавливает, что любое натуральное число, которое больше единицы, может быть представлено как результат степеней простых чисел, и это представление (факторизация) является уникальным. Например:

$$\begin{aligned}20 &= 2^2 \times 5 \\63 &= 3^2 \times 7 \\1050 &= 2 \times 3 \times 5^2 \times 7\end{aligned}$$

Все простые числа, за исключением 2, являются нечетными. Единственные два последовательных простых числа — это 2 и 3. Нечетные, следующие друг за другом числа, то есть числа, разность между которыми равна 2 (например, 17 и 19), называются простыми числами-близнецами. Простые числа Мерсенна и Ферма тоже представляют особый интерес.

Простое число является простым числом Мерсенна, если при сложении его с 1 результатом является степень 2. Например, 7 — это простое число Мерсенна, поскольку $(7+1 = 8 = 2^3)$.

Поэтому первыми восемью простыми числами Мерсенна являются:

$$3; 7; 31; 127; 8191; 131071; 524287; 2147483647$$

Сегодня мы знаем только 40 или около того простых чисел Мерсенна. Самое большое — это гигантское число: 243112609—1, вычисленное в 2008 году. Для сравнения: количество элементарных частиц во всей Вселенной оценивается менее чем в 2300.

С другой стороны, простое число Ферма — это число в следующем виде:

$$F_n = 2^{2^n} + 1, \text{ где } n \text{ — натуральное число}$$

Последовательность простых чисел Ферма начинается так:

3 ($n=0$), 5 ($n=1$), 17 ($n=2$), 257 ($n=3$), 65537 ($n=4$)...

Простые числа Ферма названы в честь блестящего французского юриста и математика, который их открыл, Пьера де Ферма (1601—1665). Француз сделал многочисленные и важные дополнительные открытия, связанные с простыми числами. Выделяется так называемая малая теорема Ферма, которая устанавливает:

если p — простое число, то для любого целого числа $a^p = a$ по модулю p .

Результат имеет большую важность для современной криптографии.

ГЛАВА 7.

КРИПТОГРАФИЯ С ИСПОЛЬЗОВАНИЕМ КОМПЬЮТЕРА

«Цифровые данные можно сделать не копируемыми настолько, насколько воду можно сделать сухой».

Брюс Шнайер

ПЕРВЫЕ ШАГИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Во время быстрого развития компьютерных технологий о криптографии не забывали. И неудивительно — использование компьютера для шифровки послания — это в большей или меньшей степени тот же самый процесс, что и шифрование в докомпьютерную эпоху. Однако тут имеются три фундаментальных отличия.

Во-первых, компьютер можно запрограммировать на имитацию работы традиционной шифровальной машины (например, 1000-роторной) без необходимости строить реальную такую машину. Во-вторых, компьютер работает только с двоичными числами, и потому процесс шифрования будет происходить на этом уровне (даже если при последующей расшифровке цифровая информация преобразуется обратно в текст). И в-третьих, компьютеры производят шифровку и расшифровку послания чрезвычайно быстро.

Первые шифры, предусматривающие использование преимуществ потенциальных вычислительных машин, были разработаны в 1970-е годы.

Примером является «Люцифер», шифр, который разделял текст на блоки объёмом в 64 бита и при помощи сложной замены шифровал некоторые из них, затем снова их группировал в только что зашифрованный блок битов и повторял процесс.

Система требовала наличия и у отправителя, и у получателя компьютера с одинаковой шифровальной программой и общего числового ключа.

Версия «Люцифера» из 56 битов под названием DES была представлена в 1976 году. DES (от англ. Data encryption standard, стандарт шифровки данных) используется и сегодня, хотя этот алгоритм был взломан в 1999 году и почти везде заменён версией AES (Advanced encryption standard, усовершенствованный стандарт шифровки) из 128 битов в 2002 году.

Несомненно, это шифрование максимально использовало способность компьютера к обработке данных, но точно так же, как и их предшественники тысячу лет назад, компьютеризованные коды по-прежнему оставались уязвимыми — опасность того, что неуполномоченный получатель сможет раздобыть коды и, зная алгоритм, расшифрует послание, никуда не делась.

Эта основная слабость любой «классической» системы криптографии известна как проблема распределения ключей.

Никто не спорит: чтобы обеспечить секретность кода, ключи шифрования должны быть защищены в большей степени, чем алгоритм.

КАК БЕЗОПАСНО РАСПРЕДЕЛИТЬ КЛЮЧИ?

Даже в самых простых случаях это может привести к сложнейшим логистическим проблемам — как, например, обеспечить большую армию тысячами шифровальных справочников? Или каким образом снабдить ими центры мобильной связи, которые работают в экстремальных условиях (к примеру, экипажи подводных лодок или подразделения в разгар сражения)?

Не имеет особого значения, насколько сложна классическая система шифрования, — все они уязвимы для перехвата соответствующих ключей.

Идея безопасного обмена ключами может казаться содержащей внутреннее противоречие: какой смысл посылать ключ к посланию, которое уже зашифровано — ключом, которым ранее обменялись обычным образом? Однако, если обмен организуется как связь с многочисленными обменами, вполне можно представить решение проблемы — по крайней мере в теории.

Давайте предположим, что отправителя зовут Ольга. Она зашифровывает послание с помощью ключа и отправляет его получателю Павлу. Тот заново зашифровывает уже зашифрованное послание с помощью своего ключа и возвращает его отправителю. Ольга расшифровывает послание с помощью своего ключа и отправляет это новое послание, которое теперь зашифровано только с помощью ключа Павла, а тот его расшифровывает.

Старая, как мир, проблема безопасного обмена ключами внезапно решена! Неужели это может быть правдой? К сожалению, нет.

В любом сложном алгоритме шифрования порядок, в соответствии с которым применяются ключи, является самым важным звеном, а мы увидели в нашем теоретическом примере, что Ольга должна расшифровывать послание, которое уже было зашифровано с помощью другого ключа.

Когда порядок шифров меняется на противоположный, результатом будет бессмыслица.

Теория на самом деле не объясняется приведённым выше сценарием, но проливает свет на способ решения проблемы.

В 1976 году два молодых американских учёных, Бейли Уитфилд Диффи и Мартин Хеллман, нашли способ, которым два человека могут перекидываться зашифрованными посланиями без какой-либо необходимости обмениваться секретным ключом. Этот метод использует модульную арифметику, а также свойства простых чисел.

Идея состоит в следующем:

1. Ольга выбирает число и хранит его в тайне. Назовём это число N_{j1} .

2. Павел выбирает наугад другое число, которое он тоже оставляет в секрете. Мы назовём это число N_{p1} .

3. Затем Ольга и Павел рассчитывают функцию типа $f(x) = ax \pmod{p}$ по своим числам соответственно, причём p — простое число, которое знают они оба.

- В результате этой операции Ольга получает новое число N_{j2} , которое она отправляет Павлу.

- Таким же путём Павел получает новое число N_{p2} , которое отправляет Ольге.

4. Ольга решает уравнение в форме $(\text{mod } p)$ и получает новое число, C_j .

5. Павел решает уравнение в форме $(\text{mod } p)$ и получает новое число, C_p .

Хотя такое кажется невозможным, C_j и C_p — это одно и то же.

И теперь у нас есть ключ.

Обратите внимание, что Ольга и Павел обменивались информацией всего два раза — когда они договаривались о расчёте функции $f(x) = ax \pmod{p}$ и когда отправляли друг другу N_{j2} и N_{p2} .

В обоих случаях информация не являлась ключом, и потому посторонний перехват не будет угрожать безопасности системы шифрования. Ключ этой системы будет иметь общую форму:

$$a^{N_{j1} \cdot N_{p1}} \text{ по модулю } p.$$

Также важно учитывать, что исходная функция имеет особое свойство — она не является обратимой, то есть, зная и функцию, и результат применения ее к переменной

x , невозможно (или, по крайней мере, очень трудно) получить исходную переменную x .

Особое свойство этого типа уравнений состоит в том, что их трудно сделать обратными — они асимметричны.

Для значений p , превышающих 300, и a , превышающих 100, решение — и, соответственно, взлом ключа — становятся чрезвычайно трудными.

Этот алгоритм является основой современной криптографии.

А теперь, чтобы подчеркнуть суть, мы повторим процесс с определенными числовыми значениями. Используемая функция — это:

$$f(x) = 7^x \pmod{11}$$

1. Ольга выбирает число, N_{j1} , например 3, и вычисляет $f(x) = 7^x \pmod{11}$, получая $f(3) = 7^3 \equiv 2 \pmod{11}$.
2. Павел выбирает число, N_{p1} , например 6, и вычисляет $f(x) = 7^x \pmod{11}$, получая $f(6) = 7^6 \equiv 4 \pmod{11}$.
3. Ольга отправляет Павлу свой результат, 2, а Питер отправляет Джеймсу свой, 4.
4. Ольга вычисляет $4^3 \equiv 9 \pmod{11}$.
5. Павел вычисляет $2^6 \equiv 9 \pmod{11}$.

Это значение, 9, и будет ключом системы.

Джеймс и Питер обменялись функцией $f(x)$ и числами 2 и 4.

Полезна ли эта информация для того, кто ее перехватит?

Давайте предположим, что наш нежелательный получатель знает и функцию, и числа. Его проблема теперь заключается в решении N_{j1} и N_{p1} по модулю 11, где N_{j1} и N_{p1} — это числа, которые и Джеймс, и Питер хранят в секрете — даже друг

от друга. Если шпиону удастся их обнаружить, то у него будет

ключ только для решения $a^{N_{J1} \cdot N_{P1}}$ по модулю p .

Кстати, решение этой задачи называется в математике **дискретным логарифмом**. Например, в случае

$$f(x) = 3^x \pmod{17}$$

мы видим, что $3^x = 15 \pmod{17}$, и, пробуя различные значения x , мы находим, что $x = 6$, и подтверждаем отношение $3^x = 15$.

Алгоритмы такого типа и проблема дискретного логарифма не удостоивались большого внимания до начала 1990-х годов, и специалисты только в последние годы занялись их проработкой. В приводимом выше примере мы говорим, что 6 — это дискретный логарифм 15 с основанием (базисом) 3, по модулю 17.

Диффи и Хеллман представили свою идею во время работы Национальной компьютерной конференции, на семинаре, который нельзя не назвать новаторским и разрушающим привычные каноны. Их работу можно изучить в полном объёме на сайте www.cs.berkeley.edu/~christos/classics/diffiehellman.pdf, она называется «New Directions in Cryptography» (Новые направления в криптографии).



За алгоритмом — люди

Алгоритм Диффи — Хеллмана продемонстрировал возможность создания криптографического метода, который, с одной стороны, не требует обмена ключами, а с другой, как это ни парадоксально, для части процесса основывается на общедоступной связи — именно так передаётся исходная пара чисел, которые служат для определения ключа.

Иными словами, алгоритм сделал возможным иметь безопасную систему шифрования между отправителями и получателями, которым совершенно не требуется встречаться, чтобы тайно договориться о ключе.

Однако кое-какая проблема всё же имеется... если Ольга отправила Павлу послание, когда Павел спит, ей придётся ждать, пока

**тот проснётся, чтобы включиться в процесс
создания ключа.**

Отыскивая новые, более эффективные алгоритмы, Диффи разработал в теории систему, где ключ шифрования будет отличаться от ключа расшифровки, и поэтому вывести один из другого будет просто невозможно. В этой теоретической системе у Ольги будет два ключа: ключ шифрования и ключ расшифровки. Из этих двух ключей Ольга сделает открытым только первый, чтобы любой, кто захочет отправить ей послание, мог его зашифровать. Получив послание, Ольга его расшифрует, используя ключ расшифровки, который никому, кроме неё, не известен.

Но возможно ли применение этой системы на практике?



*Вирусы
и бэкдоры*

НА ПОМОЩЬ ПРИХОДЯТ ПРОСТЫЕ ЧИСЛА

В августе 1977 года известный американский писатель и популяризатор науки Мартин Гарднер, ведущий рубрику о занимательной математике в журнале «Scientific American», озаглавив свою очередную статью «Новый тип шифра, на взлом которого потребуются миллионы лет». Он объяснил принципы новой системы с открытым ключом, представил зашифрованное послание и открытый ключ N , который использован для создания шифра:

$$N = 114.381.625.757.888.867.669.235.779.076.146.612.010. \\ 218.296.721.242.362.562.561.842.935.706.935.245.733.897.83 \\ 0.597.123.563.958.705.058.989.075. \\ 147.599.290.026.879.543.541.$$

Гарднер предложил своим читателям расшифровать послание на основании данной информации и даже дал подсказку — решение требует факторизации N в простые числа p и q . Ко всему в придачу, Гарднер пообещал приз в 100 долларов (приличная сумма в то время) тому, кто первым найдет правильный ответ.

Гарднер написал, что всякий, кто захочет получить более подробную информацию о шифре, должен отправить запрос создателям шифра Рону Ривесту, Ади Шамиру и Лену Эйделману из лаборатории информации Массачусетского технологического института.

Правильный ответ был получен только через семнадцать лет, и для его нахождения потребовались усилия и совместная работа более шести сотен человек.

Ключ оказался: $p = 32.769.132.993.266.709.549.961.988.19$
 $0.834.461.413.177.642.967.992.942.539.798.288.533$ и $q = 3.49$
 $0.529.510.847.650.949.147.849.619.903.898.133.417.764.638.49$
 $3.387.843.990.820577$, а зашифрованное послание было:

«Магические слова — это беззливый гриф».

Представленный алгоритм Гарднера известен как RSA — аббревиатура из первых букв фамилий Rivest, Shamir и Adelman. Это первое практическое применение модели открытого

ключа, предложенной Диффи, и сегодня ею регулярно пользуются. Она обеспечивает почти абсолютные безопасность и надежность, потому что процесс расшифровки представляет собой невероятно трудную работу, хотя и возможен.

Далее мы рассмотрим основы системы в упрощенной форме.

НАДЁЖНЫЙ АЛГОРИТМ RSA

Алгоритм RSA, названный так в честь его авторов — американских математиков Ривеста, Шамира и Адельмана, — основывается на определённых свойствах простых чисел. Мы здесь ограничимся представлением базовых предположений, которые лежат в его основе.

- Группа чисел меньше, чем n , которые также являются простыми вместе с n , называется функцией Эйлера и выражается $\varphi(n)$.

- Если $n = pq$, при том, что p и q — простые числа, то $\varphi(n) = (p - 1)(q - 1)$.

- Из малой теоремы Ферма мы знаем, что любое натуральное число, которое больше единицы, может быть представлено как результат степеней простых чисел, и это представление (факторизация) является уникальным. Например:

$$\begin{aligned}20 &= 2^2 \times 5 \\63 &= 3^2 \times 7 \\1050 &= 2 \times 3 \times 5^2 \times 7\end{aligned}$$

Все простые числа за исключением 2 являются нечётными. Единственные два последовательных простых числа — это 2 и 3. Нечётные, следующие друг за другом числа, то есть числа, разность между которыми равна 2 (например, 17 и 19), называются простыми числами-близнецами.

Если a — целое число больше 0, а p — простое число, то мы имеем

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

В соответствии с теоремой Эйлера, если $\text{НОД}(n, a) = 1$, то $a^{(n)} \equiv 1 \pmod{1}$.

Как упоминалось выше, при системе с «открытым ключом» ключ шифрования даётся любому отправителю, заинтересованному в пересылке сообщений. Каждый получатель имеет свой собственный открытый ключ. Послания всегда будут передаваться переведёнными в числа, как при использовании кода ASCII, так и любой другой системы.

Вернёмся к переписке Ольги и Павла.

Первым делом Ольга создаёт значение n как результат двух простых чисел p и q ($n = p q$), и мы выбираем значение e таким образом, что $\text{НОД}((n), e) = 1$. Вспомним, что $\varphi(n) = (p - 1)(q - 1)$.

Данные, которые делаются общедоступными, — это значение n и значение e (мы ни при каких обстоятельствах не откроем значения p и q).

Пара (n, e) — это открытый ключ к системе, а значения p и q известны как числа RSA. Параллельно этому Ольга вычисляет единственное значение d по модулю $\varphi(n)$, которое соответствует тому, что $d \cdot e = 1$, то есть обратную величину e по модулю $\varphi(n)$. Мы знаем, что эта обратная величина существует, потому что $\text{НОД}((n), e) = 1$. Значение d — это закрытый ключ к системе. Со своей стороны, Павел использует открытый ключ (n, e) , чтобы расшифровать послание M при помощи функции $M = me \pmod{n}$.

Получив послание, Ольга проводит операцию $Md = (me)d \pmod{n}$.

Это выражение эквивалентно $Md = (me)d = m \pmod{n}$, что доказывает: послание может быть расшифровано.

Каким же образом обеспечивается надёжность алгоритма RSA?

Потенциальный шпион знает значение n и e , потому что они являются общедоступными. Чтобы расшифровать послание, ему требуется d , закрытый ключ.

Значение d образуется из n и e .

Вспомним, что для построения d необходимо знать $\varphi(n) = (p - 1)(q - 1)$, в частности p и q . Для этого «достаточно» разложить на составные части n , как результат двух простых чисел p и q . Проблема для шпиона состоит в том, чтобы факторизовать большое число как результат двух простых чисел, а это медленный и трудоёмкий процесс. Если n достаточно большое (более сотни цифр), нет известного способа нахождения p и q за разумный период времени. А в наше время простые числа, используемые при шифровании посланий высшей степени секретности, превышают 200 цифр.

РАЗУМНАЯ СЕКРЕТНОСТЬ

Алгоритм RSA требует много времени на вычисления, а для таких вычислений нужны быстродействующие процессоры. До 1980-х годов только правительства, армии и крупные компании имели достаточно мощную технику, чтобы работать с RSA. В результате они фактически обладали монополией на эффективное шифрование.

Летом 1991 года Филипп Циммерман, американский программист и активный борец за права человека, в частности, за право на тайну частной жизни, бесплатно предложил систему PGP (от англ. Pretty Good Privacy — надёжная конфиденциальность), алгоритм шифрования, способный работать на домашних компьютерах. PGP использует классическую симметричную кодификацию — и обеспечивает ей большую скорость на домашних компьютерах, — но шифрует ключи с помощью асимметричного RSA.

Филипп Циммерман (род. 1954) — американский инженер-программист, ставший инициатором и возглавивший движение, цель которого — сделать современную криптографию доступной для всех.

Кроме изобретения системы PGP он создал в 2006 году Zfone, программу для обеспечения защиты голосовой связи через Интернет. Является президентом Открытого Альянса PGP, группы, лоббирующей программное обеспечение с открытым кодом. В настоящее время Циммерман предоставляет консультации по поводу криптографии большому количеству компаний и промышленных организаций, а также является членом центра по делам сети Интернет и Общества Стэнфордского юридического университета.

Циммерман объяснил причины этой меры в открытом письме, которое стоит здесь процитировать, хотя бы частично, поскольку программист предвидел, как мы будем жить, работать и общаться два десятилетия спустя:

«Личное. Тайное. И никого не касается, кроме вас. Вы можете планировать политическую кампанию, обсуждать размер своих налогов или иметь тайную

любовную связь. Или можете заниматься тем, что, по вашему мнению, не нарушает закон, хотя, с точки зрения власти, незаконно. Что бы это ни было, вам совсем не хочется, чтобы вашу личную переписку по электронной почте или конфиденциальные документы читал кто-то ещё. Нет ничего плохого в обеспечении тайны вашей частной жизни. Тайна частной жизни — это, как и Конституция, традиционная американская ценность...

Мы идём в будущее, когда страна будет вдоль и поперёк пересечена мощными оптоволоконными сетями, которые соединят вместе все наши персональные компьютеры, а они получают всё большее распространение. Электронная почта станет не новшеством, как сегодня, а нормой для всех. Правительство примется защищать наши электронные письма придуманными им же, правительством, протоколами шифрования. Вероятно, большинство людей на это молча согласятся. Однако некоторые люди, возможно, предпочтут свои собственные меры защиты... Если секретность станет вне закона, то только те, кто вне закона, будут защищены секретностью.

Разведкагентства имеют доступ к хорошим криптографическим технологиям. Точно так же ими владеют и те, кто торгует в крупных масштабах оружием и наркотиками. То же самое можно сказать и про тех, кто работает на оборонную промышленность, нефтяные компании и прочие гигантские корпорации. Но простые люди и низовые политические организации, по большей части, не имели доступа к криптографическим технологиям с открытым ключом того уровня, который используют военные. До сегодняшнего дня не имели. PGP позволяет людям защитить свою частную жизнь своими руками. На такие защитные средства в обществе растёт спрос. Именно поэтому я их и создал».

Из размышлений Циммермана мы видим, что век информации несёт угрозу нашим традиционным представлениям о защите частной жизни.

Следовательно, хорошее понимание механизмов кодификации и шифрования, которые нас окружают, не только сделает нас образованнее, но и может оказаться невероятно полезным, когда дело доходит до защиты того, что для нас ценно.

PGP начала распространяться с момента её создания и представляет собой самый важный инструмент частной криптографии, который доступен сегодня.

УДОСТОВЕРЕНИЕ ПОДЛИННОСТИ СООБЩЕНИЙ И КЛЮЧЕЙ

Различные системы шифрования с открытым ключом — или с объединением открытого и закрытого ключа типа PGP — обеспечивают высокий уровень конфиденциальности при передаче информации. Однако безопасность комплексных систем связи типа Интернета не основывается только на конфиденциальности.

До появления современных коммуникационных технологий подавляющее большинство писем приходило к получателю из хорошо известных источников (члены семьи, друзья или коллеги). Однако сегодня на каждого человека обрушивается лавина посланий от тысяч отправителей. Просто прочитывая эти послания, часто невозможно определить их подлинность, а из-за этого вполне могут возникнуть проблемы. К примеру, как нам предотвратить совершённое кем-либо искажение адреса, с которого пришло электронное письмо?

Диффи и Хеллман предложили весьма хитроумный способ: как использовать шифрование с помощью открытого ключа для удостоверения подлинности послания. В системе криптографии такого типа отправитель шифрует послание с помощью открытого ключа получателя, который, в свою очередь, использует свой собственный закрытый ключ для расшифровки послания. Диффи и Хеллман обратили внимание на то, что RSA и другие подобные алгоритмы демонстрируют интересную симметричность.

Закрытый ключ может использоваться и для шифрования послания, а открытый ключ — для его расшифровки.

Эта операция совершенно не обеспечивает безопасности — открытый ключ легко доступен для всех, но убеждает получателя послания в том, что оно пришло от определённого отправителя, владельца закрытого ключа. Чтобы удостовериться подлинность отправителя достаточно, в теории, добавить дополнительное шифрование к обычному, используя следующий процесс:

1. Отправитель (Ольга) шифрует послание с помощью открытого ключа получателя. Этот первый шаг обеспечивает конфиденциальность.
2. Ольга снова шифрует послание, на сей раз с помощью своего закрытого ключа. Таким образом, послание удостоверяется, или «подписывается».
3. Получатель (Павел) использует открытый ключ отправителя, чтобы «взломать» шифровку из шага 2. Таким образом подтверждается происхождение послания.
4. Затем Павел использует свой секретный ключ, чтобы «взломать» шифровку из шага 1.

ХЭШ-ПОДПИСЬ

Одна из проблем, связанных с теоретическим планом, представленным выше, состоит в том, что для шифрования с помощью открытого ключа требуется компьютер значительной мощности. Кроме того, повторение процесса с целью подписания и подтверждения каждого послания займёт очень много времени.

Именно поэтому на практике подписание послания выполняется с помощью математических ресурсов, известных как хэш-функции. Начиная с исходного послания, эти алгоритмы создают простую цепь битов (обычно 160), которая называется хэш, и производят это таким образом, что вероятность, что различные послания связаны с одним и тем же хэшем, почти равна нулю. Также практически невозможно отменить процесс и получить исходное послание, если начать только с его хэша.

Хэш любого послания зашифровывается Ольгой с помощью её закрытого ключа и отправляется вместе с зашифрованным посланием традиционным образом. Павел расшифровывает послание, которое содержит хэш, с помощью открытого ключа Ольги. Затем, при условии, что ему известна хэш-функция, которую использовала Ольга, он применяет эту функцию к посланию и сравнивает два хэша. Если они совпадают, то личность отправителя подтверждена.

Более того, становится определённо известно, что к исходному посланию никто посторонний не прикасался.

		Хэш-функция					
Послание							
Красный	=>	DKJD	1242	AACB	788B	761A	
		696C	24D9	7009	CA99	2D17	
Красный цвет соответствует самой низкой частоте	=>	0896	56BB	ZC7D	CBE2	823C	
		ADD7	8CD1	9AB2	JJ6J	8ABC	

Красный цвет соответствует самой низкой частоте	=>	FCD3	7FDB	D588	4C75	4BF4
		1799	7D88	ACDE	92B9	6A6C
Красный цвет соответствует самой низкой частоте	=>	D401	C0A9	7D9A	46AF	FB45
		76B1	79A9	0DA4	AEFE	4819

Небольшие изменения в содержании послания создают совершенно другие «хэши». Таким образом получатель может быть уверен, что с текстом никто не манипулировал.

СЕРТИФИКАТЫ ОТКРЫТЫХ КЛЮЧЕЙ

Однако самая важная проблема, с которой приходится сталкиваться, используя криптографическую систему с открытым ключом, — это не удостоверение подлинности посланий, а сами открытые ключи.

Как Ольге и Павлу узнать, что открытый ключ другого действителен?

Давайте предположим, что шпион обманывает Ольгу, дав ей свой собственный открытый ключ, одновременно заставляя её поверить, что это ключ получателя. Если шпиону удаётся перехватить послание, он может использовать свой закрытый ключ для его расшифровки. Чтобы избежать раскрытия, шпион использует открытый ключ получателя для новой шифровки послания и отправки его в исходное место назначения.

Именно поэтому и государственные учреждения, и частные компании занимаются независимой сертификацией открытых ключей. Сертификат такого рода содержит, кроме соответствующего ключа, информацию о получателе и дату окончания действия. Держатели этих типов ключей делают их общедоступными, и с определённой степенью надёжности их можно использовать и ими можно обмениваться.

Большинство шпионов и хакеров, которые действуют онлайн, мало интересуют послания, которыми обмениваются обычные люди,

за одним важным исключением — номеров их кредитных карт.

Система криптографии, которая защищает передачу такой секретной информации, известна как TLS (от англ. Transport Layer Security — безопасность транспортного узла). Она была разработана корпорацией Netscape, занимающейся программным обеспечением для Интернета в 1994 году, и принята в качестве мирового стандарта два года спустя.

Протокол TLS объединяет открытые и симметричные ключи в довольно сложный процесс, который здесь представляется в сжатой форме.

Во-первых, браузер онлайн-покупателя подтверждает, что онлайн-продавец имеет действующий сертификат на открытый ключ. Если это так, то покупатель использует общедоступный ключ для шифрования второго ключа, который является симметричным, и отправляет продавцу. Затем продавец использует свой секретный ключ для расшифровки послания и получает симметричный ключ, который будет использоваться для шифрования всей обрабатываемой информации.

В результате, чтобы получить номер кредитной карты, используемой в любой сделке онлайн, шпиону потребуется проникнуть не в одну, а в две шифровальные системы.

ШИФРОВАНИЕ ВО ВРЕД

Сегодня всё большее распространение получают вредоносные программы, создаваемые злоумышленниками с целью личного обогащения. Дело заключается в шифровании всех файлов на компьютере-жертве с последующим требованием выкупа. Формально такие программы относятся к виду Trojan-Ransom, существует и другой общеупотребимый и ёмкий термин — шифровальщики. Шифровальщики стали весьма серьёзной проблемой для пользователей, особенно корпоративных.

В ближайшее время лёгкой победы над шифровальщиками ожидать не стоит. На это есть как минимум две весомые причины:

1. Шифровальщики непрерывно эволюционируют. Это битва «меча и щита»: совершенствуются средства защиты — совершенствуются средства нападения.
2. Атаке подвергается не компьютер пользователя, а система компьютер + пользователь, то есть одним из компонентов вектора атаки является человек. Человеку свойственно поддаваться эмоциям и действовать нерационально. Человек способен проигнорировать предупреждение защитных средств или вовсе отключить их. Именно на это и рассчитывают злоумышленники.

Рассмотрим эволюцию шифровальщиков в аспекте применяемых ими методов шифрования и криптосхем. В зависимости от используемой криптосхемы и способа получения ключа, в одних случаях можно легко дешифровать зашифрованные данные, в других случаях за разумное время этого сделать нельзя.

Вирусописатели регулярно меняют криптографические схемы, средства обфускации и форматы исполняемых файлов.

ШИФРОВАНИЕ С ПОМОЩЬЮ ОПЕРАЦИИ «XOR»

Начнём с программ, осуществляющих самое примитивное шифрование. Ярким представителем таких вредоносных программ является семейство Trojan-Ransom.Win32.Xorist. Оно обладает следующими характерными особенностями:

- Xorist — один из немногих шифровальщиков, который выполняет свою угрозу и портит файлы пользователя при многократном неправильном введении пароля.
- Для шифрования используется операция «XOR». Уязвимостью этой криптосхемы является то, что возможно лёгкое дешифрование файлов за счёт известных стандартных заголовков файлов. Чтобы противостоять этой атаке, Xorist шифрует файлы не с самого начала, а с некоторым отступом. По умолчанию этот отступ составляет 104 h байт, но может быть изменён при компиляции сэмпла.
- Для усложнения алгоритма шифрования используется рандомизация ключа с помощью первой буквы названия файла.

В целом, несмотря на все ухищрения создателей Xorist, зашифрованные им файлы вполне успешно поддавались сравнительно лёгкой дешифровке. Возможно, поэтому в настоящий момент злореды семейства Xorist практически не встречаются.

СИММЕТРИЧНОЕ ШИФРОВАНИЕ

Симметричной схемой шифрования называется схема, при которой для шифрования и расшифровки используется пара ключей, связанных соотношением симметрии. В подавляющем большинстве случаев в таких схемах для шифрования и расшифровки используется один и тот же ключ.

Если ключ «вшит» в тело шифровальщика, при наличии тела зловреда можно достать из него ключ и создать эффективную утилиту для расшифровки файлов. Такие вредоносные программы обычно стараются удалить себя после шифрования файлов. Примером программ этого типа могут служить некоторые модификации семейства Rakhni.

Если же ключ получается с сервера злоумышленников либо генерируется и отправляется на него, то наличие образца вредоносной программы мало что даёт — необходим экземпляр ключа, находящегося на сервере злоумышленников. Если такой ключ удаётся восстановить (вредоносная программа, по понятным причинам, старается удалить такой ключ после использования), то создать утилиту для дешифровки возможно. В этом случае также могут оказаться полезными системы, кэширующие интернет-трафик пользователей.

АСИММЕТРИЧНОЕ ШИФРОВАНИЕ

Асимметричной схемой шифрования называется схема, при которой ключи шифрования и расшифровки не связаны очевидным соотношением симметрии. Ключ для шифрования называют открытым (или публичным), ключ для расшифровки называют закрытым (или приватным). Вычисление закрытого ключа по известному открытому — очень сложная математическая задача, не решаемая за разумный срок на современных вычислительных мощностях.

В основе асимметричных криптосхем лежит так называемая односторонняя функция с секретом. Говоря простым языком — это некая математическая функция, зависящая от параметра (секрета). Если секретный параметр неизвестен, значение функции относительно легко вычисляется в «прямом» направлении (по известному значению аргумента вычисляется значение функции) и чрезвычайно трудно в «обратном» (по значению функции вычисляется значение аргумента). Однако всё меняет знание секретного параметра — с его помощью «инвертировать» функцию не составляет особого труда.

Асимметричное шифрование с одной ключевой парой

Если открытый ключ зашит в тело вредоносной программы, то наличие тела вредоносной программы в отсутствие закрытого ключа практически никак не помогает в деле расшифровки файлов (но помогает детектированию этой программы, а также похожих на неё программ в дальнейшем).

Однако если становится известен закрытый ключ (а он должен, как минимум, содержаться в дешифраторе, который предлагают купить злоумышленники), то появляется возможность расшифровать данные у всех пользователей, пострадавших от модификации программы, содержащей соответствующий открытый ключ.

Пример вредоносных программ этого вида — Trojan-Ransom.Win32.Rector. Вот характерные особенности этого семейства:

- Используется асимметричное шифрование, публичный ключ хранится внутри тела шифровальщика.
- Для ускорения шифрования файлы зашифровываются не целиком, а небольшими кусками. Зашифрованные куски дописываются в конец файла, а их место заполняется последовательностями периодичностью в один байт. Из-за этого зашифрованный файл приобретает характерный «поцарапанный» вид.
- Для злоумышленников недостатком этой схемы является то, что для дешифрования файлов необходимо раскрытие закрытого ключа, с помощью которого впоследствии можно расшифровать все файлы, зашифрованные одной модификацией зловреда.

Соответственно, хотя прямое дешифрование файлов невозможно, несколько пользователей, пострадавших от одной и той же модификации зловреда, могут объединяться и покупать один дешифратор на всех.

Если открытый ключ получается с сервера злоумышленников (что позволяет обеспечить уникальность открытого ключа для каждого пользователя), то наличие тела вредоносной программы также не помогает в деле расшифровки данных — необходимо знание закрытого ключа. Однако тело программы помогает вычислить и заблокировать вредоносный сервер и тем самым защитить других пользователей.

ШИФРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ НЕСКОЛЬКИХ КЛЮЧЕЙ

Чтобы обеспечить уникальность дешифратора для каждого пользователя, применяются схемы с несколькими ключами. При этом ключ для шифрования данных генерируется на компьютере жертвы. Это может быть как симметричный ключ, так и асимметричная ключевая пара. Алгоритм для генерации ключа выбирается таким, чтобы получившийся ключ был уникален для каждого пострадавшего пользователя. Иными словами, вероятность того, что эти ключи совпадут в двух разных случаях шифрования, должна быть крайне мала. Однако случается так, что злоумышленники допускают ошибки, и ключ генерируется из относительно небольшого диапазона возможных значений. В этом случае данные пользователя удаётся расшифровать, перебрав все возможные значения ключа. Но в последнее время такие случаи достаточно редки.

С помощью сгенерированного ключа шифруются данные пользователя. Затем ключ, который необходим для расшифрования данных, зашифровывается на другом открытом ключе. Этот открытый ключ сгенерирован заранее, и соответствующего ему закрытого ключа в теле шифровальщика нет, зато этот закрытый ключ известен злоумышленнику. После этого оригинал ключа, необходимого для расшифрования данных, удаляется, и на компьютере пользователя остаётся только его зашифрованная версия.

Теперь, получив зашифрованную копию ключа, злоумышленник может извлечь из неё ключ, необходимый для расшифрования данных пользователя, и встроить его в дешифратор. При этом дешифратор будет бесполезен для других пострадавших пользователей. Что, с точки зрения злоумышленника, выгодно отличает схему с двумя ключами от перечисленных выше.

Невозможно за приемлемое время расшифровать файлы, зашифрованные алгоритмом RSA с длиной ключа 1024 бит.

Пример вредоносных программ, использующих схему с несколькими ключами, — Trojan-Ransom.BAT.Scatter. Семейство Scatter обладает несколькими существенными особенностями:

- Применяется более совершенная криптосхема с двумя парами асимметричных ключей, что позволяет злоумышленникам расшифровывать файлы жертвы, не раскрывая свой закрытый ключ.
- Сэмплы из этого семейства написаны на скриптовых языках, это позволяет легко менять вредоносный функционал. Скрипты лучше поддаются обфускации, и этот процесс проще автоматизировать.
- Сэмплы имеют модульную структуру. Модули скачиваются с сайтов злоумышленников во время исполнения скрипта.
- Для шифрования и удаления файлов ключей используются переименованные легитимные утилиты.
- Достигнут высокий уровень автоматизации процесса. Автоматизации подвержено практически всё: автоматически генерируются вредоносные объекты, автоматически рассылаются письма. Более того, по заверениям злоумышленников, автоматизирован процесс обработки писем от жертв и дальнейших контактов с пострадавшими. Автоматически происходит расшифровка тестовых файлов жертвы, оценка стоимости информации, выставление счёта, проверка оплаты, отсылка дешифратора. Нам трудно проверить правдивость этой информации, но с учётом данных, полученных при изучении модулей Trojan-Ransom.BAT.Scatter, нет причин не верить этим заявлениям. Косвенным их подтверждением также служит вот этот аккаунт в Twitter.

Семейство Scatter появилось относительно недавно: первые сэмплы были обнаружены в конце июля 2014 года. За короткое время оно успело значительно эволюционировать, обзаведясь функциональностью Email-Worm и Trojan-PSW.

ГЛАВА 8.

КВАНТОВАЯ КРИПТОГРАФИЯ

«То, что мы пока не можем телеграфировать схему человека из одного места в другое, связано, в основном, с техническими трудностями...»

Норберт Винер

«Заявление, что квантовая криптография является стойкой, качественно отличается от всех прежних заявлений. Квантовая криптография является не просто практически нераскрываемой, она нераскрываема совершенно. Квантовая теория — самая удачная теория в истории физики — подразумевает, что Ева никогда не сможет безошибочно перехватить криптографический ключ однократного использования, который был создан Алисой и Бобом. Ева не сможет даже попытаться перехватить криптографический ключ однократного использования без того, чтобы Алиса и Боб не были предупреждены о ее действиях».

Сингх Саймон. «Книга шифров»

НЕМНОГО КВАНТОВОЙ ТЕОРИИ

Кажется очевидным, что электроны дискретны, они — воплощение понятия «частицы». О свете, напротив, наиболее естественно думать как о волне, и уже более 200 лет известно, что он способен проявлять себя в волновых эффектах, таких как интерференция и дифракция.

Квантовая теория информации находится на стыке двух наиболее значительных теорий XX века: квантовой механики и теории информации. Она имеет дело с квантово-механическими состояниями и рассматривает их способность участвовать в переносе и обработке информации. Эта наука появилась в 60-е годы прошлого века, во времена бурного развития вычислительной техники, как следствие того, что при постоянном уменьшении размеров вычислительных устройств со временем неизбежно возникнет необходимость использовать одиночные квантовые состояния в качестве информационного ресурса. В то время подобная перспектива означала новые сложности, в первую очередь сильное влияние квантового шума, который считался однозначно разрушающим фактором. Однако при более подробном изучении этого явления выяснилось, что квантовый шум может оказывать существенную помощь при передаче и обработке информации: так, явление квантового «размазывания» частицы по нескольким точкам пространства обладает свойством интерференции, способным в ряде случаев принести большую пользу.

Квантовая теория информации как новая наука работает с квантовыми явлениями. Оказывается, что применение квантовых состояний способно вывести скорость вычислений на новый уровень благодаря идее квантового компьютера, а также

гарантировать абсолютную секретность распространения ключей в квантовой криптографии.

Детектирование и квант света

При любом детектировании первоначально рассеянный свет ведёт себя так, как если бы он сжимался до мельчайших размеров, появляясь моментально в одном месте. Всё это может свидетельствовать о дискретной природе света. Первое определение световых лучей, данное Исааком Ньютоном, соответствует этим современным представлениям:

«Под лучами света я понимаю его мельчайшие частицы, которые следуют друг за другом вдоль одних и тех же линий и одновременно нескольких линий».

С тех пор как американский химик Ж. Н. Льюис в 1926 году дал им название, принято отождествлять эти «мельчайшие частицы» с фотонами.

Определение Ньютона выглядит как пророческое предвидение квантовой теории.

Принцип неопределённости Гейзенберга

Часто, когда мы что-то измеряем, то воздействуем на объект измерения. Чтобы посмотреть ночью на спящего ребёнка, нам придётся открыть дверь и включить свет, возможно, разбудив малыша. Аналогично, чтобы измерить положение атома, нужно, чтобы на атоме рассеялся свет (или что-то ещё), что, очевидно, повлияет на него. Всё это чисто классические рассуждения.

В рамках классической физики существует два способа обойти эту проблему. С одной стороны, мы всегда можем уменьшить величину воздействия настолько, насколько захотим, уменьшив интенсивность света, например. С другой стороны, поскольку



Автор
неопределённости

любое воздействие управляется детерминистическими законами классической физики, мы можем внести соответствующую поправку. Игра на бильярде, например, зависит от того, насколько хорошо мы можем предсказать влияние воздействия разных факторов. Анализируя конечную энергию и импульс, мы можем определить, как переместились бильярдные шары, даже не наблюдая за самими столкновениями. Мы можем поэтому сделать обратный расчёт и скорректировать наши измерения с любой точностью, какая нам нужна. До сих пор не было ничего необычного. Основная идея Гейзенберга состояла не в том, что измерения влияют на измеряемую систему. Речь шла о признании новых принципиальных измерительных ограничений, связанных с «квантом действия». Существуют два таких предела.

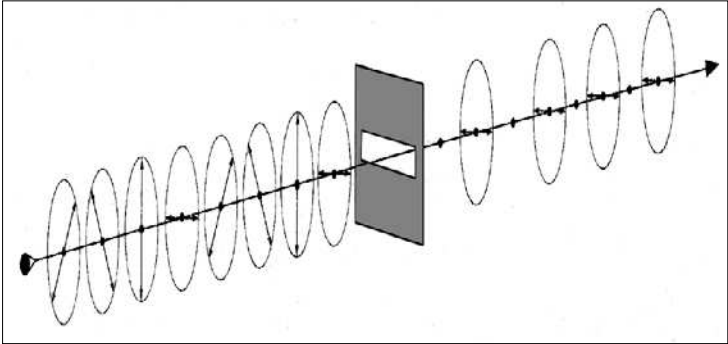
Во-первых, в то время как, согласно классической физике, мы можем сделать воздействие сколь угодно малым, согласно квантовой механике, это невозможно.

Воздействие света, например, квантовано, поэтому фотон не может не повлиять на частицу, с которой он сталкивается.

Второе ограничение, налагаемое квантовой механикой, состоит в том, что это воздействие неуправляемо и непредсказуемо.

Если говорить более конкретно, то невозможно определить с какой-либо степенью точности определённые дополнительные свойства частицы в любой данный момент. Давайте, например, рассмотрим случай с фотонами.

Одно из их фундаментальных свойств — это поляризация, технический термин, который относится к колебаниям или вибрации электромагнитной волны. Хотя фотоны вибрируют во всех направлениях, для нашего короткого примера предположим, что они вибрируют в четырёх: вертикально, горизонтально, диагонально влево, диагонально вправо. В таком случае, в соответствии с принципом Гейзенберга, единственный способ подтвердить поляризацию определённого фотона — это провести его через фильтр или «щель», которая, в свою



Если мы пропустим ряд фотонов различной поляризации через горизонтальный фильтр, то увидим, что половина из тех, которые ориентированы диагонально, пройдёт сквозь фильтр с изменением поляризации на горизонтальную

очередь, может быть и горизонтальной, и вертикальной, или диагональной с направлением влево или вправо.

Поляризованные горизонтально фотоны будут проходить горизонтальный фильтр без изменений, в то время как поляризованные вертикально фотоны будут заблокированы. Что касается фотонов, которые поляризованы диагонально, половина из них пройдёт через фильтр с изменением поляризации на горизонтальную, а вторая половина будет отскакивать совершенно произвольно. Более того, после того как фотон выйдет из фильтра, с определённой вероятностью узнать его исходную поляризацию будет невозможно.

Какое отношение поляризация фотонов имеет к криптографии? Очень существенное, как мы увидим ниже. Для начала возьмём на себя роль исследователя, который хочет узнать поляризацию ряда фотонов. Для этого он может выбрать фильтр только с фиксированной ориентацией (к примеру, горизонтальной). Предположим, что фотон проходит сквозь фильтр. Какую информацию из этого получит исследователь? Конечно, он может предположить, что исходная поляризация фотона была не вертикальной. Может ли он сделать ещё какое-либо предположение? Нет.

Вначале можно было бы подумать, что существует большая вероятность того, что исходный фотон был ориентирован

горизонтально, а не диагонально, потому что половина «диагональных» не пройдёт сквозь фильтр. Однако количество диагонально ориентированных фотонов также в два раза больше количества горизонтальных. Важно подчеркнуть, что трудность определения поляризации фотона — это не результат какого-то технологического или теоретического недостатка, который может быть исправлен в будущем.

Это следствие природы самой субатомной реальности.

Если это свойство должным образом использовать, то оно может быть применено к созданию совершенно не поддающегося взлому кода, такого Святого Грааля криптографии.

Странная кошка

На семинаре по квантовой физике, который проводился в 1958 году, Нильс Бор высказал своё мнение по поводу предложения одного из выступавших следующим образом:

«Мы все согласны с тем, что эта теория безумна. Вопрос, по которому мы не можем прийти к соглашению, заключается в том, достаточно ли она безумна для того, чтобы иметь шанс оказаться правильной».

И в самом деле, насколько же безумна квантовая механика? В качестве примера давайте возьмём принцип суперпозиции состояний. Частица представляет собой суперпозицию состояний, если в один и тот же момент занимает более одной позиции или если одновременно обладает различными количествами энергии. Однако когда наблюдатель измеряет частицу, всегда будет казаться, что она занимает одну позицию или обладает совершенно определённым количеством энергии.

Сам Шрёдингер придумал мысленный эксперимент, «кошку Шрёдингера», чтобы проиллюстрировать эту кажущуюся абсурдной идею.

Представьте кошку, которую заточили в стальную изолированную камеру. Внутри камеры находятся колба с синильной кислотой и счётчик Гейгера, с помощью реле управляющий молотком. Если происходит распад частицы, реле срабатывает, молоток опускается на колбу и разбивает её. В результате кошка отравлена. Распад частицы, о которой идёт речь, за период,

Кошка Шрёдингера — мысленный эксперимент, который иллюстрирует концепцию квантовой теории о суперпозиции состояний



определённый для эксперимента, может произойти с вероятностью 50%.

Вся эта конструкция, зависящая от поведения одной-единственной частицы, работает в соответствии с законами квантовой физики.

Давайте предположим, что определённый для эксперимента период времени истёк. Вопрос: жива кошка или мертва? Или, если говорить на жаргоне квантовой механики, в каком состоянии находится система «камера–кошка»? Ответ таков: пока наблюдатель не открыл камеру и не «измерил» состояние системы, частица могла распасться, а могла и не распасться. Иными словами, система отражает принцип суперпозиции состояний: строго говоря, кошка не жива и не мертва, а одновременно и жива, и мертва.

Для всех, кто посчитает суперпозицию состояний притянутой за уши гипотезой, нужно отметить, что уважаемые господа физики предлагали альтернативные интерпретации. Например, теория о множестве параллельных вселенных утверждает, что идея о суперпозиции состояний — это неподкреплённый доказательствами тезис, а в реальности дело обстоит так, что для каждого

из возможных состояний, в которых может оказаться частица, — позиция, количество энергии и т. д. — существует альтернативная вселенная, где частица принимает другое состояние.

Иными словами, в одной вселенной кошка в камере жива, а в другой — мертва.

Когда наблюдатель открывает камеру и убеждается, что кошка жива, его действие является неотъемлемой частью только одной из возможных вселенных. В параллельной вселенной — полностью заполненной своими собственными звёздами, мирами, городами и людьми, — тот же самый наблюдатель смотрит в камеру и с великой скорбью убеждается, что кошка отравлена.

Сторонники теории параллельных вселенных, правда, всё ещё не объяснили, как эти вселенные взаимодействуют друг с другом... Тем не менее, теория объясняет, почему квантовые реальности ведут себя странным образом, а само это поведение было подтверждено в ходе многочисленных убедительных экспериментов.

Квантовые неразрушающие измерения

В классической физике мы можем несколько раз измерить одну величину. Мы можем измерить длину стола и, измерив её ещё раз, получить тот же результат. Мы можем посчитать и затем пересчитать количество яиц в упаковке. Но квантовая механика налагает ограничения на возможность повторного измерения той же величины. Действительно, измерения, согласно квантовой теории, могут иметь разрушающее действие. Повторные измерения координаты могут не совпадать из-за принципа неопределённости. Повторная регистрация фотона может оказаться невозможной из-за того, что он был уничтожен при первом измерении. Однако были разработаны способы преодоления этих ограничений.

По словам Филиппа Циммермана, которые Саймон Сингх приводит в своей «Книге о кодах»:

«Современная криптография может создавать шифры, которые на самом деле находятся вне пределов всех известных форм криптоанализа».

Как мы отмечали, взлом алгоритмов шифрования типа RSA или DES и даже смешанных систем типа PGP, используя метод тотального перебора, находится вне вычислительных возможностей даже самых быстрых компьютеров. Можно ли предполагать, что какой-то вид математического сокращения процесса позволит шпионам будущего снизить сложность криптоанализа? Хотя исключать эту возможность нельзя, очень вероятной её никто не считает.



Саймон
Лехна Сингх

Протоколы квантового состояния

К 1984 году накопленных результатов оказалось достаточно, чтобы сформулировать принципы квантовой криптографии и представить, хотя на тот момент и не строгие, но интуитивно понятные доводы в пользу секретности подобного способа распределения ключей. Затем пришло время для развития собственно формализма квантовой криптографии: были описаны требуемые действия легитимных пользователей, формализованы действия шпиона, а также была доказана секретность первого протокола распределения ключей, названного BB84.

Основные факты квантовой теории информации, на которых основывается квантовая криптография, — это связанные между собой утверждения о невозможности копирования произвольных квантовых состояний и о невозможности достоверно различить неортогональные состояния. В сочетании эти факты дают то, что попытки различения квантовых состояний из неортогонального набора ведут к помехам, а значит, действия перехватчика могут быть детектированы по величине ошибки на приёмной стороне.

Важно отметить, что квантовая криптография не делает никаких предположений о характере действий подслушивателя и величине доступных ему ресурсов: предполагается,

что перехватчик обладает любыми ресурсами и делает все возможные действия в рамках известных на сегодня законов природы. Это существенным образом отличает квантовую криптографию от классической, которая опирается на ограничения в вычислительной мощности подслушивателя.

Неформально принцип действия всех протоколов квантовой криптографии можно описать так: передающая сторона (Ольга) на каждом шаге посылает одно из состояний из их неортогонального набора, а принимающая сторона (Павел) производит такое измерение, что после дополнительного обмена классической информацией между сторонами они должны иметь битовые строки, полностью совпадающие в случае идеального канала и отсутствия перехватчика. Ошибки же в этих строках могут говорить как о неидеальности канала, так и о действиях подслушивателя. При величине ошибки, превышающей некоторый предел, действие протокола прерывается, иначе легитимные пользователи могут извлечь полностью секретный ключ из их частично совпадающих битовых строк.

Коллапс волновой функции

Важным законом квантовой механики является редукция, или коллапс волновой функции. Это свойство называется также редукцией фон Неймана и означает переход состояния после измерения в одно из собственных состояний оператора измерения. Так, при измерении $\{M\}$ и получении результата i исходное состояние будет преобразовано в

$$\rho'_i = \frac{\sqrt{M_i} \rho \sqrt{M_i}}{\text{Tr} M_i \rho}.$$

Это одно из важнейших для квантовой криптографии свойств. Поскольку оно говорит о том, что попытки измерить систему влекут помехи, из чего следует, что попытки перехвата информации всегда можно выявить по дополнительным ошибкам на приёмной стороне. При обнаружении попыток подслушивания по их количеству даются оценки возможной утечки информации к перехватчику.

Невозможность клонирования

Покажем частный результат из теории составных квантовых систем, важный для квантовой криптографии. Как известно, неортогональные квантовые состояния нельзя достоверно различить. Здесь будет показано, что их нельзя клонировать — например, чтобы собрать более полную статистику результатов измерений.

Преобразование U , клонирующее произвольное чистое квантовое состояние $|\psi\rangle$, можно описать так:

$$U|\psi\rangle \otimes |A\rangle = |\psi\rangle \otimes |\psi\rangle,$$

где $|A\rangle$ — исходное состояние вспомогательной системы.

Чтобы показать невозможность такого преобразования, достаточно рассмотреть его действие на базисные состояния $|0\rangle$ и $|1\rangle$:

$$\begin{aligned} U|0\rangle \otimes |A\rangle &= |0\rangle \otimes |0\rangle, \\ U|1\rangle \otimes |A\rangle &= |1\rangle \otimes |1\rangle, \end{aligned}$$

а также на состояние $1/\sqrt{2} (|0\rangle + |1\rangle)$. В силу линейности оператора U и приведённых чуть выше соотношений должно выполняться

$$U\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes |A\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

С другой стороны, по определению U , должно получаться

$$U\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes |A\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle).$$

Полученное противоречие доказывает невозможность клонирования произвольных квантовых состояний.

Отметим, что клонировать состояния из ортогонального набора можно: для этого достаточно, например, измерить их и приготовить состояние, соответствующее результату измерения, — в этом случае он будет безошибочным.

Составные квантовые системы

Рассмотрение квантовых систем, состоящих из нескольких частиц (их называют составными системами), может порой привести к интересным свойствам, не встречающимся в классическом случае. Ещё в 1935 году в переписке Эйнштейна, Подольского и Розена были отмечены очень интересные свойства составных квантовых систем, противоречащие локальности: выходило, что действия над одной из подсистем могут мгновенно оказать влияние на другую подсистему, каково бы ни было расстояние между ними. Описание этого свойства привело к возникновению формализма составных квантовых систем и свойств совершаемых над ними действий.

Тензорное произведение

Определим то, в каком пространстве обитают составные квантовые системы. Рассмотрим для начала наиболее элементарный случай двух кубитов. На интуитивном уровне очевидно, что возможны 4 варианта их совместного состояния:

- оба кубита в состоянии $|0\rangle$;
- первый кубит в состоянии $|0\rangle$, второй в состоянии $|1\rangle$;
- первый кубит в состоянии $|1\rangle$, второй в состоянии $|0\rangle$;
- оба кубита в состоянии $|1\rangle$.

Именно эти 4 вектора будут являться базисными в пространстве двух указанных кубитов.

БИТЫ И КУБИТЫ

Однако какое же отношение суперпозиция состояний частиц имеет к вычислениям — даже если не трогать криптографию? До 1984 года никто не думал высказывать предположений о взаимоотношениях между этими двумя областями. Примерно в то время британский физик Дэвид Дойч выступил с революционной идеей: какими бы стали компьютеры, если бы, вместо того чтобы работать по законам классической физики, они начали подчиняться законам квантовой механики? Как вычисления могли бы выиграть от суперпозиции состояний частиц?

Вспомним, что традиционные компьютеры работают с минимальными единицами информации, которые называются биты и способны принимать значения 0 и 1.

С другой стороны, для квантового компьютера можно задать в виде наименьшей единицы информации элементарную частицу, которая способна находиться в двух возможных состояниях. К примеру, спин электрона может иметь два направления, вверх или вниз. В результате эту частицу можно использовать для представления значения 0 (спин вниз) или значения 1 (спин вверх). Но если вспомнить о суперпозиции состояний, то электрон способен представлять оба значения одновременно.



Дэвид Дойч

Эту новую единицу информации назвали «кубит» (сокращённо от «квантовый бит», или q-бит). Манипуляции с ней могут открыть двери в мир супермощных компьютеров.

В результате поиск нужного числа будет проводиться по всем до единой возможным опциям одновременно. Если мы увеличим количество кубитов до, скажем, 250, количество одновременных операций, которые могут быть при этом выполнены, составит около 1075, немногим меньше, чем, теоретически, насчитывается атомов в нашей Вселенной.

Традиционный компьютер выполняет вычисления последовательно. Давайте в качестве примера возьмём числовую

информацию, содержащуюся в 32 битах. С этим количеством битов мы можем зашифровать числа от 0 до 4292967295. Если традиционному компьютеру нужно найти в этой группе определённое число, ему придётся проводить поиск бит за битом. Однако квантовый компьютер может выполнить такую задачу гораздо быстрее. Для иллюстрации этой возможности представьте себе, что мы помещаем 32 электрона в специальный контейнер и загоняем их в суперпозицию состояний. Затем воздействуем на них электрическими импульсами, достаточно мощными для того, чтобы изменить спин электрона с «вверх» на «вниз». Эти 32 электрона — кубиты нашего квантового компьютера — будут представлять все возможные комбинации спина «вверх» (1) и спина «вниз» (0) одновременно.

Рассматривали и другие типы простых систем с двумя возможными состояниями:

- электроны со спином вверх или вниз;
- атомы в основном или возбуждённом состоянии;
- фотоны, отражённые или прошедшие через делитель луча;
- фотоны горизонтальной или вертикальной поляризации.

Каждое из этих состояний можно рассматривать как запись бита и использовать для передачи информации. Мы могли бы, например, сделать машину, в которой электрон со спином вверх обозначает 0, а со спином вниз — 1. Таким способом мы можем обозначить все буквы алфавита набором значений проекций спинов электронов.

Такая машина будет квантовой, а не классической. Квантовое кодирование отличается от классического возможностью суперпозиции. Кроме двух состояний, возможных для классического бита, любая из вышеупомянутых систем может находиться в суперпозиции $q = \alpha (0) + \beta (1)$, где α и β — комплексные коэффициенты. Это есть кубит, термин, введённый Беном Шумахером в 1995 году.

С помощью простых физических воздействий мы можем изменять состояние кубитов. Например, если пропустить фотон с горизонтальной поляризацией H через четвертьволновую пластину, она сдвинет фазу на 90 градусов, что изменит поляризацию фотона. Если мы скажем, что фотон горизонтальной поляризации H обозначает 0, а с вертикальной поляризацией V обозначает 1, то это изменение состояния фотона

$$q \text{ initial} = (0) \rightarrow q \text{ final} = 1/\sqrt{2} [(0) + i(1)]$$

можно рассматривать как изменение состояния кубита.

В качестве кубита можно использовать не только состояния с разной поляризацией, но и с разным направлением движения. Состояние фотона, распространяющегося в некотором направлении (0), становится суперпозицией состояний после прохождения делителя луча:

$$q \text{ initial} = (0) \rightarrow q \text{ final} = 1/\sqrt{2} [(0) + (1)],$$

когда фотон имеет конечную вероятность двигаться как в первоначальном направлении, обозначаемом (0), так и в перпендикулярном к нему направлении, обозначаемом (1).

Работа Дойча доказала, что квантовые компьютеры теоретически возможны.

Идея кубита, как и в целом квантовых вычислений, может быть проиллюстрирована множеством примеров. Одна и та же теория и одни и те же алгоритмы могут быть применены ко многим квантовым системам.

Дюжины институтов и исследовательских групп по всему миру стремятся превратить когда-нибудь теорию в практическую реальность. Однако пока они не смогли преодолеть технические трудности разработки жизнеспособного квантового компьютера. Некоторые эксперты полагают, что потребуются ещё около четверти века для достижения такой цели. Другие вообще сомневаются, что это возможно.

ВЫЧИСЛЯЕМ КВАНТАМИ

В последние десятилетия XX века появились квантовые вычисления, новый и революционный способ проектирования и управления компьютерами. Хотя квантовый компьютер всё ещё находится на этапе теории, он может обладать вычислительной мощностью, способной методом проб и ошибок расшифровать сегодняшние алгоритмы шифрования. И когда-нибудь криптоаналитики смогут вновь включиться в игру. Эта находящаяся пока в зачаточном состоянии технологическая революция основывается на квантовой механике, теоретическом здании, построенном в начале прошлого столетия учёными, в число которых, помимо многих других, входили датчанин Нильс Бор, британец Поль Дирак и немцы Макс Планк, Вернер Гейзенберг и Эрвин Шрёдингер.

Видение Вселенной в соответствии с постулатами квантовой механики настолько противоречит человеческой интуиции, что знаменитой стала цитата из Альберта Эйнштейна:

«Бог не играет в кости».

Несмотря на сомнения и сдержанное отношение Эйнштейна, теория квантовой механики была успешно протестирована бесконечное количество раз, и её работоспособность теперь не вызывает сомнений. Научное сообщество в целом принимает, что на макроскопическом уровне — то есть Вселенная звёзд — придерживается законов классической физики. Однако в квантовом мире — фантастически малом царстве частиц, которые меньше атомов, таких, как кварки, фотоны, электроны и т.д., работает совсем другой набор правил, который ведёт к поразительным парадоксам. Без этой теории не было бы таких вещей, как атомные реакторы или лазерные считывающие устройства. Нельзя было бы объяснить яркое сияние солнца или функционирование ДНК.

ПЕРЕДАЧА ИНФОРМАЦИИ ПО КВАНТОВЫМ КАНАЛАМ

При исследовании передачи информации с помощью квантовых состояний возникает ряд вопросов, связанных с характеристиками использующих квантовые объекты каналов. Основной из этих вопросов связан с пропускной способностью таких каналов, то есть с максимальной скоростью безошибочной передачи данных.

Самый общий случай квантового канала — это отображение квантовых состояний во множество квантовых состояний. Такое отображение можно расширить на случай произвольных операторов в гильбертовом пространстве:

$$\Phi : \mathfrak{T}(\mathcal{H}) \rightarrow \mathfrak{T}(\mathcal{H}).$$

Под $\mathfrak{T}(\mathcal{H})$ понимается пространство операторов со следовой нормой:

$$\|T\|_1 = \text{Tr}|T|, \quad |T| = \sqrt{T^*T}, T \in \mathfrak{T}(\mathcal{H}).$$

Подобный подход автоматически даёт два требования, связанных с тем, что оператор плотности должен переходить в оператор плотности:

- положительные операторы должны переходить в положительные;
- должен сохраняться след оператора;
- также естественно требовать аффинность отображения Φ : статистические ансамбли состояний должны переходить также в статистические ансамбли их образов, то есть для набора вероятностей $\{p_i\}$

$$\Phi\left[\sum_i p_i \rho_i\right] = \sum_i p_i \Phi[\rho_i], \quad p_i \geq 0, \quad \sum_i p_i = 1.$$

Более формально требования к отображению Φ на пространстве $\mathfrak{T}(\mathcal{H})$ операторов в гильбертовом пространстве таковы: линейность, положительность и сохранение следа.

Для каждого отображения Φ , действующего на множестве квантовых состояний и соответствующего картине Шрёдингера, определяется сопряжённое состояние Φ^* , соответствующее картине Гейзенберга и действующее на множестве M квантовых наблюдаемых. Эти отображения связаны соотношением:

$$\text{Tr} \Phi[\rho] M = \text{Tr} \rho \Phi^*[M].$$

Сопряжённое отображение действует на пространстве операторов с операторной нормой

$$\| B \| = \| B \|_{\infty} = \max_{\psi: \|\psi\|=1} \| B\psi \| .$$

При подобном определении сопряжённое отображение Φ^* должно обладать следующими свойствами: линейность, положительность, сохранение единицы (унитальность).

Линейные коды

Определим важное подмножество классических кодов, которое удобно тем, что его можно задать с помощью матриц сравнительно небольшого размера. Такие коды называются линейными. Исходное сообщение длины n преобразуется в кодовое слово длины k с помощью умножения на порождающую матрицу размера $n \times k$, состоящую из нулей и единиц. Так, код с повторением для входных слов длины 2 описывается с помощью матрицы размера 2×6 :

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Легко видеть, что действие этой матрицы на входные слова соответствует коду с повторением:

$$G(0,0) = (0,0,0,0,0,0), \quad G(0,1) = (0,0,0,1,1,1), \\ G(1,0) = (1,1,1,0,0,0), \quad G(1,1) = (1,1,1,1,1,1).$$

Существенное преимущество линейных кодов в том, что вся информация, необходимая для кодирования 2^n кодовых слов, содержится всего лишь в kn элементах порождающей матрицы, что позволяет сильно экономить компьютерную память.

Процесс обнаружения и исправления ошибок описывается в этом случае другой матрицей, которая называется проверочной. Это матрица H , ядром которой являются кодовые слова и только они, то есть $Hx = 0$ выполняется тогда и только тогда, когда x — кодовое слово. В этом случае проверочная матрица будет иметь размеры $(k-n) \times n$.

Очевидно, что если исходное кодовое слово x при передаче по каналу преобразовалось в ошибочное слово $y + e$, то $Hu = Hx + He = He$. Это даёт возможность, имея значения He_j для набора $\{e_j\}$ всевозможных n -мерных векторов с единицей всего лишь на одной j -й позиции, определить, в какой именно позиции произошла ошибка, и исправить её.

Линейный код, где каждое из сообщений длины n кодируется с помощью k битов данных, называется $[n, k]$ -кодом. Основное свойство линейных кодов заключается в том, что существует $[n, k]$ -код, способный при больших n исправить q ошибок в n битах исходного сообщения, если

$$\frac{n}{k} \geq 1 - h\left(\frac{2q}{n}\right).$$

Этот важный результат называется границей Варшамова — Гильберта.

Также важно отметить, что для всякого линейного кода C его проверочную матрицу H после транспонирования можно использовать как порождающую матрицу другого кода, который в этом случае называется двойственным к коду C . Его порождающая матрица H^T , а проверочная — G^T . Очевидно, что кодовые слова проверочной матрицы будут ортогональны кодовым словам исходной матрицы C .

ПЕРЕДАЧА СИГНАЛЬНЫХ СОСТОЯНИЙ

$$\begin{aligned} + : |x\rangle &= |0\rangle, \quad |y\rangle = |1\rangle, \\ \times : |u\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |v\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

Легко проверить, что эти базисы удовлетворяют условию несмещённости:

$$\begin{aligned} |\langle x|u\rangle|^2 &= |\langle x|v\rangle|^2 = \frac{1}{2}, \\ |\langle y|u\rangle|^2 &= |\langle y|v\rangle|^2 = \frac{1}{2}, \end{aligned}$$

которое неформально сводится к тому, что с точки зрения одного базиса состояния в другом расположены симметрично.

На этапе подготовки состояний Ольга случайным образом выбирает один из указанных базисов, а затем случайно выбирает значение бита: 0 или 1, и в соответствии с этим выбором посылает один из четырёх сигналов:

- $|x\rangle$, если это базис «+» и значение бита равно 0;
- $|y\rangle$ при том же базисе и значении бита 1;
- $|u\rangle$ при выпадении базиса «x» и бита 0;
- $|v\rangle$, если в базисе «x» выпал бит 1.

При посылке каждого из этих сигналов Ольга запоминает свой выбор базиса и выбор бита, что приводит к появлению двух случайных битовых строк на её стороне. Павел, получая каждый из присланных Ольгой сигналов, производит над ним случайным образом одно из двух измерений, каждое из которых способно дать достоверный результат из-за ортогональности состояний внутри каждого базиса Ольги:

$$\begin{aligned} M_0^+ &= |x\rangle\langle x|, & M_1^+ &= |y\rangle\langle y|, \\ M_0^\times &= |u\rangle\langle u|, & M_1^\times &= |v\rangle\langle v|. \end{aligned}$$

В результате у него оказывается две строки: с тем, какие из базисов были выбраны для изменения, и с исходами этих измерений.

После передачи всех состояний и проведения измерений Ольга и Павел имеют по две строки. Здесь происходит согласование базисов: по открытому каналу Ольга и Павел объявляют друг другу свои строки с выбором базисов и выбрасывают посылки, в которых базисы не совпали. При этом, если базис, используемый для посылки состояния Ольгой, совпал с измерением базиса Павла, то в случае отсутствия помех в канале связи результаты в их битовых строках на соответствующей позиции будут совпадать, поэтому после этапа согласования базисов в случае идеального канала и отсутствия действий со стороны перехватчика Ольга и Павел должны обладать одними и теми же битовыми строками.

Однако, если в канале были ошибки или шпион пытался перехватить информацию, битовые строки Ольги и Павла могут не совпадать, поэтому для проверки они должны согласованно раскрыть примерно половину своих битовых строк. Ошибка в раскрытой битовой последовательности даёт достаточно точную оценку ошибки во всей последовательности, и по ней можно довольно точно оценить вероятность ошибки в оставшихся позициях. Если величина ошибки оказывается больше некоторой величины (параметра протокола), то передача данных прекращается: это означает, что перехватчик обладает слишком большой информацией о ключе.

В противном случае перед Ольгой и Павлом стоит задача получения общего секретного ключа. Эту задачу можно разбить на два этапа: сначала производится коррекция ошибок, в результате которой в распоряжении Ольги и Павла оказываются совпадающие битовые строки. Второй этап, усиление секретности, ставит своей целью исключить информацию о ключе, которая могла попасть к шпиону в результате действий над квантовыми состояниями или в ходе коррекции ошибок. В результате этой операции у шпиона не должно оставаться информации об общей битовой строке Ольги и Павла.

КВАНТОВЫЕ КОДЫ КОРРЕКЦИИ ОШИБОК

Классические коды коррекции ошибок используются для безошибочной передачи данных по каналам с помехами. Так, если в канале допустима помеха в одном произвольном бите, то простейшим кодом для безошибочной передачи данных будет код с повторением: вместо сигнала 0 будем посылать 000, а вместо 1 — 111. На приёмной стороне принимается решение о переданном сигнале по близости в метрике Хемминга: сигнал, содержащий два или три ноля (000, 001, 010, 100), трактуется как 0, а сигнал с двумя или тремя единицами — как единица. Указанный подход невозможен для применения в квантовом случае. Первое же препятствие этому — запрет на клонирование квантовых состояний.

Коды, исправляющие ошибку в одном кубите

Отличия квантового случая восстановления информации от классического видны уже при описании ошибки: если в классическом случае единственным вариантом ошибки является смена бита $0 \leftrightarrow 1$, то в квантовом случае возможные ошибки образуют непрерывное множество. Простейшим примером квантовой ошибки является фазовая ошибка:

$$\begin{aligned} |0\rangle &\rightarrow |0\rangle \\ |1\rangle &\rightarrow -|1\rangle \end{aligned}$$

Обратим внимание, что если битовая ошибка меняет местами состояния из множества $\{|0\rangle, |1\rangle\}$, то фазовая ошибка оставляет такие состояния нетронутыми (за исключением несущественной общей фазы), но меняет местами состояния в наборе $\{1/\sqrt{2}(|0\rangle + |1\rangle), 1/\sqrt{2}(|0\rangle - |1\rangle)\}$, устойчивом к битовым ошибкам.

Заметим, что произвольное кубитовое состояние $\psi = \alpha |0\rangle + \beta |1\rangle$ можно обезопасить от битовой ошибки с помощью следующего кода (запрета на клонирование ортогональных состояний нет):

$$\begin{aligned} |0\rangle &\rightarrow |000\rangle \\ |1\rangle &\rightarrow |111\rangle \end{aligned}$$

Покажем, как это происходит. На выходе канала можно произвести измерение

$$\begin{aligned}M_0 &= |000\rangle\langle 000| + |111\rangle\langle 111|, \\M_1 &= |100\rangle\langle 100| + |011\rangle\langle 011|, \\M_2 &= |010\rangle\langle 010| + |101\rangle\langle 101|, \\M_3 &= |001\rangle\langle 001| + |110\rangle\langle 110|.\end{aligned}$$

Не трудно видеть, что такое измерение не меняет исходного состояния и что при отсутствии ошибки выпадает результат M_0 , а при ошибке в i -й позиции результат M_i .

Результат такого измерения называют синдромом ошибки. Тогда, зная, в какой позиции произошла ошибка, можно произвести корректирующее преобразование K_i , заключающееся в инверсии i -го кубита. В результате на выходе окажется исходное состояние.

Итак, целью процедуры коррекции ошибок является получение из частично совпадающих битовых строк Ольги и Павла полностью идентичных строк. Эта обычная процедура, она имеет дело лишь с классическими битами и открытыми каналами связи. Наиболее эффективная процедура коррекции ошибок сводится к использованию случайных кодов. Пропускная способность классического канала с вероятностью ошибки Q равна $1 - h(Q)$.

$$C_{\text{clas}}(Q) = 1 - h(Q),$$

где $h(Q)$ — бинарная энтропия Шеннона.

Зная вероятность ошибки в канале и имея последовательность длины n , Ольга генерирует 2^n случайных кодовых слов. Параметр δ можно сделать малым при больших значениях n . К этому списку Ольга присоединяет и свою последовательность битов, после чего открыто сообщает набор кодовых слов Павлу (а значит, они становятся известны и шпиону). Из полученного списка кодовых слов Павел выбирает ближайшее к своей последовательности в метрике Хемминга. Для канала с шумом при таком выборе кодовых слов Павел с вероятностью единица выберет битовую строку Ольги.

Отметим, однако, что полностью случайные коды трудно реализовать на практике, так что при их использовании необходимо хранить в памяти экспоненциально большое (в зависимости от битовой строки n) число кодовых слов. В реальных схемах используются другие, конструктивные коды, эффективность которых ниже.

Усиление секретности

Задача этапа усиления секретности состоит в том, чтобы получить из частично секретных общих битовых строк Ольги и Павла полностью неизвестный шпиону секретный ключ. Обычно в ходе такой процедуры длина ключа существенно уменьшается.

Основным методом на данном этапе является использование класса универсальных хэш-функций \mathcal{G} . Это функции, отображающие набор n -битовых строк A в набор m -битовых строк B таким образом, что для случайно выбранной хэш-функции $g \in \mathcal{G}$ и любых несовпадающих элементов $a^1, a^2 \in A$ вероятность совпадения их образов $g(a^1) = g(a^2)$ не превосходит $1/|B|$. То есть задача нахождения прообразов двух различных элементов в B не может решиться более эффективно, чем с помощью перебора или угадывания. Легитимные пользователи, имея оценку информации шпиона, всегда могут выбрать длину финального ключа m настолько малой, что неопределённость шпиона относительно финального ключа будет сколь угодно близка к неопределённости простого угадывания, что соответствует его полной секретности.

Таким образом, в ситуации, когда взаимная информация Ольги и Павла превосходит взаимную информацию Ольги и шпиона, всегда можно из исходного частично секретного ключа получить полностью секретный ключ, сжав его при помощи универсальной хэш-функции.

Используем описанные выше простые представления, чтобы создать нашу первую квантовую машину, квантовое устройство шифрования.

КАК ИЗБЕЖАТЬ ПОДСЛУШИВАНИЯ

Общеизвестно, как трудно защититься от подслушивания. Если отправитель (Ольга) посылает получателю (Павлу) сообщение, как могут они быть уверены, что кто-то их тайно не подслушивает? Мы опишем два способа, как можно избежать подслушивания. Оба способа используют особенности квантовой механики: первый — суперпозицию состояний одной частицы и второй — перепутанные состояния двух частиц. Предположим сначала, что сообщение написано на английском языке. Можно просто запечатать его в конверт и послать по почте или передать голосом по телефону. Но шпион может перехватить письмо и подслушать телефон.

Ольга может защититься от этого, переведя сообщение на некоторый другой язык, но в этом случае нужно иметь уверенность, что этот язык известен только ей и Павлу. Значит, они должны создать собственный специальный язык. Единственный способ сделать это — использовать то, что известно как «ключ» и «одноразовая клавиатура».

Предположим, что все буквы алфавита (английского) пронумерованы числами от 1 до 26 и, чтобы включить знаки препинания, добавлено ещё несколько чисел. Тогда любое сообщение можно записать как последовательность целых чисел. Допустим, что сообщение Ольги — это просто yes. Поскольку у — 25-я буква алфавита, е — 5-я и s — 14-я, последовательность чисел, обозначающих данное письмо, будет такой:

25, 5, 14 .

Если шпион перехватил это сообщение, он может легко догадаться, что Ольга использовала очевидную связь между буквами и числами, и таким образом расшифровать сообщение. Чтобы избежать этого, Ольга использует вторую последовательность чисел, называемую «ключ», каждый член которой добавляется к соответствующему члену первой последовательности. В результате получается криптограмма, или зашифрованное сообщение. Если, например, ключом является последовательность 1, 1, 1, 1 ... то отправитель добавил бы 1 к каждому из приведённых выше чисел и получит криптограмму

Если шпион начнёт переводить это обратно на английский язык, то получит *zft*, что является тарабарщиной: Ольга обеспечила секретность своего сообщения.

Конечно, это не очень надёжная секретность, поскольку ключ тривиален. Шпион может легко найти его и расшифровать сообщение. Для Ольги надёжнее использовать в качестве ключа случайную последовательность чисел.

Для более формального подхода представим сообщение как набор целых чисел $\{p_1, p_2, p_3, \dots, p_n\}$ и наш ключ как другой набор случайных целых чисел $\{k_1, k_2, k_3, \dots, k_m\}$. Если мы хотим полностью засекретить передачу, то необходимо, чтобы количество целых чисел в ключе было больше или равнялось их количеству в сообщении, то есть m больше или равно n .

Показано, что полностью безопасное шифрование может быть выполнено, если сложить два набора друг с другом. $c_j = p_j + k_j \bmod B$, где B — число используемых символов, отличных друг от друга (26 для букв алфавита и 2, если мы используем бинарный код). Если отправитель и получатель имеют одинаковую последовательность случайных чисел, то отправитель может кодировать сообщение, а получатель — расшифровать криптограмму. Традиционно длинная последовательность случайных чисел, являющаяся ключом, обозначается на клавиатуре. Числа использовались только один раз. После использования страница вырывалась и уничтожалась, откуда название «одноразовая клавиатура» для этого метода шифрования.

В этом методе являются два требования:

1. Ольга должна найти способ передать свой ключ Павлу, чтобы он мог расшифровать сообщение.
2. Только Павел может знать ключ: это должно быть тайной от шпиона.

Проблема состоит в том, чтобы гарантировать тайну ключа. В этом может помочь квантовое распространение ключа. Оно даёт Ольге и Павлу безопасный способ получить общий ключ без его реальной передачи, исключаяющий возможность подслушивания.

КВАНТОВЫЕ ИЗМЕРЕНИЯ

В 1984 году Чарльз Беннет и Жиль Брассарда опубликовали протокол для передачи шифровального ключа, полная безопасность которого обеспечивается особенностями квантовых измерений. Преимущество этого квантового протокола состоит в том, что получатель всегда может узнать, что шпион подслушивает, так как для передачи сообщения используются квантовые частицы. Более того, он может сообщить Ольге, подслушивает шпион или нет, прежде чем Ольга пошлёт ему какую-либо важную информацию. И, наконец, шпион никогда не будет уверен, являются ли перехваченные им данные ценной информацией или мусором.

Протокол начинается с кодирования ключа в бинарном коде и затем посылается последовательность кубитов, которой записан этот ключ. В дальнейшем мы будем использовать горизонтально поляризованный (H) фотон для обозначения «0» и вертикально поляризованный (V) фотон для обозначения «1». Ольга для передачи ключа посылает последовательность горизонтально и вертикально поляризованных фотонов.

Догадавшись о методе передачи, шпион может перехватить сообщение с помощью поляризационного делителя луча.

Он может определить поляризацию каждого фотона и таким образом узнать содержание сообщения, посланного в бинарном коде получателю. После этого шпион может переслать получателю поток фотонов в той же последовательности поляризации, который он перехватил в послании, так что ни Ольга, ни Павел не смогут догадаться, что сообщение было перехвачено. Но шпион может это сделать, только если Ольга и Павел используют только один базис поляризации.

В протоколе Беннета — Брассарда Ольга и Павел договорились использовать не один, а два базиса с линейной поляризацией. Двумя базисами могут быть, например,

1. Горизонтально и вертикально поляризованные фотоны (HV).

2. Линейная поляризация, повернутая на 45 градусов к первой ($H'V'$)

Чтобы предотвратить подслушивание, Ольга и Павел беспорядочно переключают машину между этими двумя базисами. Каждый раз, когда Ольга посылает другой фотон, она использует либо не повернутую (первую), либо повернутую (вторую) ориентацию, выбирая их беспорядочно. Так что типичное сообщение, посланное отправителем, будет $HHV'VH'VV'H'$...

Вначале получатель не имеет информации о том, как Ольга ориентировала свой поляризатор. Он просто беспорядочно ориентирует свой датчик поляризации или в первом, или во втором направлении, и записывает проекцию поляризации (H или V), которую измеряет. Тогда в среднем, в половине случаев, они повернуты друг относительно друга на 45 градусов. Но получатель не знает, для каких конкретно фотонов ориентации совпали. Он знает только, что, если ориентации совпадают, он может быть уверен в правильности полученного сообщения. Если ориентации не совпадают, то полученный, например, H фотон имеет только 50% вероятности быть зарегистрированным как H .

Сначала Павел проверяет, не перехватывались ли присылаемые сообщения. Для этого он сообщает Ольге, какое измерение с повернутым и не повернутым поляризатором он делал для каждого фотона последовательности (при этом сохраняя в тайне результаты измерений). Затем отправитель сообщает получателю, в каких измерениях его базис совпал с базисом получателя.

Павел использует эту информацию, чтобы отобрать те из своих данных, для которых ориентации совпадают, то есть правильные данные.

Чтобы проверить, не было ли подслушивания, Ольга и Павел сравнивают отобранные «правильные данные». Если эти правильные данные полностью (с учётом обычного искажения сигнала) совпадают, они могут быть уверены, что никто не перехватывал передачу сообщения. Как они это могут знать?

Подобно получателю, шпион также не знает последовательность ориентации, использованной отправителем для передачи информации. Как следствие, он получит неправильные данные для четверти случаев (в половине случаев он будет использовать неправильную ориентацию и в половине из этих случаев поляризация фотона будет измерена неправильно). Когда шпион затем передаст получателю то, что он считает копией данных отправителя, он допустит ошибку в четверти случаев. Эта ошибка обнаружится как различие между четвертью правильных данных, посланных Ольгой и полученных Павлом.

При сравнении этих данных они обнаружат, что некий шпион перехватывал их фотоны, если несовпадение превысит ожидаемую погрешность измерений. Если же отобранные данные совпадут, Ольга и Павел могут быть уверены, что никто не перехватывал сообщение, и правильные данные могут быть использованы для создания ключа, который знают только они.

Всего за несколько лет протокол Беннета — Брассарда прошёл путь от идеи и формул на доске к экспериментальной реализации и даже к коммерческой реализации в небольших масштабах. Было показано, что сигналы, кодированные с помощью поляризации, могут быть переданы на расстояние более 70 км. Одно из фундаментальных ограничений технологии связано с самим источником её преимуществ. Обычно сигналы, передаваемые по оптическому волокну, усиливаются в ретрансляторах через каждые 30—50 км, чтобы компенсировать неизбежное ослабление сигнала при передаче. Но классические ретрансляторы не могут быть использованы для передачи квантового сигнала из-за необратимой редукции суперпозиции при измерении.

Передача квантового ключа посредством перепутанных состояний

Второй метод создания безопасного ключа использует состояние ЭПР (Эйнштейна — Подольского — Розена). Самый простой способ предложили Беннет, Брассард и Мермин. Рассмотрим перепутанное состояние

$$1/\sqrt{2} [(\uparrow)\downarrow + (\downarrow)\uparrow] .$$

Для обозначения битов 0 и 1 вместо поляризации фотона мы будем использовать проекции спина

$$1/\sqrt{2} [(0)1 (1)2 + (1)1 (0)2]$$

(спин вверх обозначает 0, а спин вниз — 1).

Источник создаёт такую ЭПР-пару и посылает одну частицу отправителю, другую получателю. Затем все аналогично протоколу Беннета — Брассарда:

- Ольга и Павел каждый измеряют проекцию спина своей частицы вдоль любого из двух направлений, перпендикулярных направлению движения частицы.
- Они измеряют беспорядочно эти направления.
- Через общедоступный канал связи они сообщают друг другу ориентации своих анализаторов в последовательности своих измерений, но сохраняют в тайне результаты этих измерений.
- Зная, какие из их измерений были сделаны при одинаковой ориентации, они сохраняют результаты только этих измерений и отбрасывают другие.

Вспомним, что в ЭПР-состоянии результаты измерений, сделанные каждым из наблюдателей, совершенно случайны. Таким образом, и Ольга, и Павел обладают последовательностью 0 и 1, которая случайна, что важно для секретности ключа. Но так как антикорреляция гарантирована в ЭПР-состоянии, то когда Ольга обнаруживает 0 (спин вверх), Павел обнаруживает 1 (спин вниз), и когда Ольга обнаруживает 1, Павел обнаруживает 0. Если теперь Павел просто инвертирует свои результаты $(0) \rightarrow (1)$ и $(1) \rightarrow (0)$, каждый получит случайную, но одинаковую последовательность нулей и единиц. То есть они получают общий ключ для шифра.

Что произойдёт, если шпион перехватит один член пары, измерит его и попытается переслать Павлу поддельный фотон с теми же самыми свойствами? Поскольку состояние этого фотона больше не перепутано с состоянием второго члена ЭПР-пары, эти состояния не являются больше коррелированными. Как и прежде, если Павел и Ольга сравнят исходные

наборы своих правильных данных, они быстро обнаружат, что линия прослушивается. Таким образом, метод обеспечивает безопасную передачу ключа. Это возможно благодаря парадоксальному неклассическому поведению квантовой системы при измерении, плюс перепутанность.

Но не переносится ли информация в методе передачи ключа, использующем ЭПР-корреляцию? Это действительно происходит. Но информация исходит не от Ольги. Она исходит из источника ЭПР-пар. Ни Ольга, ни Павел не выбирают свой ключ: они просто узнали его, когда измерили свои частицы. Ключ был создан чисто случайным процессом, который не контролировался ни Ольгой, ни Павлом.

Дополнительной интересной особенностью протокола является то, что корреляция возникает только во время измерения. Предположим, что созданы 100 ЭПР-пар, и каждую из частиц пары посылают соответствующему пользователю: половину Ольге и половину Павлу. Предположим далее, что ни Ольга, ни Павел не делают никаких попыток измерить состояние своих частиц, а вместо этого решают сохранить их состояние в течение некоторого времени. Мы уже знаем, что измерения ЭПР-частицы не несут никакой информации, никаких скрытых параметров. Даже при том, что каждый фотон есть часть перепутанного состояния двух частиц, сам по себе он находится в неоднозначном состоянии и бесполезен как для Ольги, так и для Павла. Информация, которая является ключом, создаётся только в процессе измерения, при котором происходит коллапс суперпозиции и исчезает неопределённость.

Мы можем представить ситуацию, когда Ольга делает беспорядочные измерения вдоль осей X или Z набора 100 ЭПР-частиц, хранящихся у неё, только когда ключ необходим, и говорит Павлу, чтобы он провёл аналогичные измерения своих частиц. Они составляют протокол и получают ключ. В отличие от одноразовой клавиатуры, которая создаётся в одном месте и затем должна быть доставлена в другое место за достаточно продолжительное время, этот ключ даже не существовал до момента его использования.

КВАНТОВАЯ ТЕЛЕПОРТАЦИЯ

В научно-фантастических романах под телепортацией понимается возможность перемещать некоторые объекты из одного места в другое, не преодолевая расстояние между ними.

Но можно транспортировать не объект как таковой, а инструкцию по его изготовлению.

Вообразите, например, что в будущем мы будем строить автоматизированную универсальную фабрику на Марсе. Нам нужно телепортировать на эту планету нового робота, назовём его Ровер, чтобы с его помощью изучить геологическую историю планеты. Просто транспортировать Ровера — дорого. Вторая возможность — создать точное описание устройства Ровера и послать на марсианскую фабрику. Фабрика смогла бы его сделать, и, в конечном счёте, из её дверей выкатится Ровер, готовый приступить к исполнению своих обязанностей. Мы можем сказать, что робот был доставлен с Земли на Марс, не перемещаясь между этими планетами.

Нечто подобное уже применяется на практике. Чтобы послать факс, мы сканируем оригинал и посылаем информацию о результатах сканирования на удалённый приёмник, который создаёт копию. Оригинал остаётся неизменным, а новая копия изготовлена по полученному шаблону. Различие с «истинной» телепортацией здесь заключается в том, что в обоих случаях оригинал остаётся неизменным. В научной фантастике он исчезает после того, как процесс завершён.

Заметим, что необходимым первым шагом телепортации является сканирование оригинала. Оказывается, что, согласно квантовой механике, это невозможно. Таким образом, ограничения, накладываемые квантовыми измерениями, казалось бы, исключают возможность телепортации квантовых состояний. Однако это не так. Обсудим сначала ограничения, накладываемые квантовой теорией на возможность прочесть квантовую информацию, а затем найденный способ обойти эти ограничения.

При обсуждении передачи ключа с помощью суперпозиции одной частицы мы отмечали, что квантовая механика обеспечивает средство для безопасной передачи данных,

но одновременно ограничивает область использования этого метода. Это ограничение возникает вследствие того, что любой реальный сигнал неизбежно ослабляется при передаче на большие расстояния и нуждается в ретрансляции — но квантовая механика делает классическую ретрансляцию невозможной.

Чтобы это проиллюстрировать, вернёмся к нашему первому примеру, в котором ключ кодировался поляризацией фотона. Предположим, что мы хотим сделать устройство — назовём это «классическим ретранслятором» — чтобы восстановить такой сигнал, прошедший большое расстояние. Этот ретранслятор должен сделать две вещи:

1. определить поляризацию каждого фотона;
2. создать новый фотон с той же самой поляризацией и послать его дальше.

Проблема возникает с первым шагом: квантовая механика ограничивает нашу способность определить состояние поляризации фотона.

Это происходит по двум причинам. Первая связана с принципом неопределённости: любое измерение величины неизбежно изменяет эту величину случайным и непредсказуемым образом. Второе ограничение связано с процессом измерения. Предположим, что полученный фотон поляризован вдоль оси, которая наклонена относительно оси нашего ретранслятора. Это значит, что в базисе ретранслятора состояние фотона является суперпозицией. Но результат измерения ретранслятора не является суперпозицией, а определённым результатом, который получен случайным образом, с вероятностью, определяемой коэффициентами суперпозиции. Более того, нет никакой возможности узнать из результатов измерения, было ли полученное состояние суперпозицией или собственным состоянием. Как мы видели, измерение приводит к коллапсу суперпозиции, радикально изменяя состояние — или, если измерение соответствует собственному базису состояния, никакого изменения не происходит!

Это ограничение имеет место не только в случае поляризации фотона. Квантовая механика ограничивает нашу

способность определить любое состояние. Чтобы это проиллюстрировать, рассмотрим второй вид ретранслятора, предназначенный для восстановления радиосигнала. Предположим, что сигнал ЧМ (частотно модулирован), то есть его частота изменяется во времени. Мы можем представить поток фотонов, первый с частотой $\nu 1$, второй с частотой $\nu 2$, и т. д. Ретранслятор должен измерить частоту каждого принимаемого фотона перед его ретрансляцией.

Как и прежде, принцип неопределённости ограничивает возможность сделать это. Для определённости предположим, что ретранслятор — многоуровневый атом, с набором энергий перехода $\Delta E 1 = h\nu 1$, $\Delta E 2 = h\nu 2$, и т. д. Тогда, после поглощения фотона, атом остаётся в некотором возбуждённом состоянии: ретранслятор должен определить энергию этого состояния. Если для измерения требуется время Δt , то результатом является неопределённость измерения $\Delta E = h/\Delta t$, приводящая к неустранимой ошибке измерения частоты. Во-вторых, если принимаемый фотон находится в суперпозиции, а не собственном состоянии энергии, атом остаётся в суперпозиции возбуждённых состояний. Измерение энергии атома даёт только одну величину, вводя нас в заблуждение, что принятый фотон имел определённую частоту. И снова нет никакого способа узнать из результата этого измерения, было ли состояние изменено в процессе измерения.

Из этих примеров и по этим причинам мы видим, что:

- Мы не можем узнать о квантовом состоянии из результатов его измерения.
- В процессе измерения первоначальное состояние изменяется.

Физическое состояние чернил на этой странице несёт определённую информацию. Читая страницу, вы читаете эту информацию. Но если бы чернила были квантовой системой, этого нельзя было бы сделать. Квантовая информация нечитабельна в классическом смысле.

Это не значит, однако, что квантовая информация не существует. Более того, оказывается, что, если мы устояли против

желания прочесть квантовую информацию, мы можем передать её.

Мы видим, что принцип неопределённости и коллапс волновой функции подразумевают, что мы понятия не имеем, соответствуют ли результаты наших измерений фактическому состоянию системы. В 1993 году Беннетт с коллегами нашёл, как можно обойти это препятствие для телепортации, и вскоре после этого данная идея была осуществлена экспериментально двумя группами. Стратегия состоит в том, чтобы воздерживаться от измерения состояния. Оказывается, что мы можем, тем не менее, передать достаточную информацию об этом состоянии для его реконструкции в отдалённом месте. Эта стратегия также открывает возможность для квантовой ретрансляции, при которой квантовое состояние не измеряется. По идее, предложенной Беннеттом и другими, отправитель (Ольга) должен передать получателю (Павлу) состояние своей квантовой частицы для воспроизведения, послав ему квантовую частицу в состоянии, совпадающем с состоянием его частицы.

В этом процессе исходное квантовое состояние разрушается, делая его «истинным» случаем телепортации. Поскольку квантовая информация нечитабельна, ни отправитель, ни получатель не знают, в каком состоянии их частицы были или есть. Оба твёрдо уверены только в том, что в конце процедуры состояние квантовой частицы получателя является точной копией состояния, в котором находилась частица Ольги.

Ольга измеряет состояние Белла (*BSM*), определяющее отношения между неизвестным квантовым состоянием его частицы и одного из членов ЭПР-пары. Затем он использует классический канал, чтобы передать результат измерения Павлу. Это даёт Павлу достаточно информации, чтобы превратить свой член ЭПР-пары в точную копию начального состояния, используя одно из четырёх унитарных преобразований *U*.

Новым является использование вспомогательной ЭПР-пары для создания дополнительной перепутанности, без измерения квантового состояния. Кроме ЭПР-пар, Ольга и Павел используют классический канал для передачи классической информации. Ольга выполняет совместное измерение частицы, которая должна быть телепортирована вместе с одной частицей

из ЭПР-пары, — измерение, которое даёт ей информацию не о неизвестном состоянии, а об отношении между этим состоянием и состоянием её ЭПР-частицы. Затем она использует классический канал для сообщения получателю результатов этого измерения. После получения этого сообщения Павел обладает достаточной информацией, чтобы преобразовать свой член ЭПР-пары в состояние, которое является идентичным с первоначальным состоянием частицы Ольги. Это состояние частицы получателя остаётся единственным в конце процедуры, так как измерение Ольги разрушило неизвестное состояние его частицы.

В телепортации принимают участие три частицы. Первая частица находится в состоянии, которое Ольга хочет телепортировать получателю. Состояние неизвестно и останется неизвестным после телепортации. Две других частицы составляют ЭПР-пару, один член которой получает Ольга (частица 2) и другой Павел (частица 3). Состояния этих трёх частиц определяются выражениями

$$\begin{aligned}\phi_1 &= \alpha (0)1 + \beta (1)1; \\ \text{ЭПР}_{23} &= 1/\sqrt{2} [(0)2 (1)3 - (1)2 (0)3],\end{aligned}$$

где ЭПР₂₃ — состояние ЭПР-пары, индексы снизу указывают, к какой частице относится данное состояние; (0) обозначает горизонтальную поляризацию и (1) вертикальную поляризацию.

Обратите внимание, что, вследствие перепутанности, частицы 2 и 3 не имеют отдельных собственных волновых функций. Также заметьте, что состояние ЭПР₂₃ антисимметрично по отношению к перестановке частиц: ЭПР₂₃ = – ЭПР₃₂. Из рассмотрения перепутанных состояний известно, что если частица 2 проецируется измерением на состояние (0), тогда мгновенно и с полной определённости частица 3 будет найдена в состоянии (1), и наоборот.

Трюк телепортации заключается в том, что Ольга перепутывает состояние частицы 1 с состоянием частицы 2, которое уже перепутано с состоянием частицы 3, посылаемой Павлу! Не рассматривая пока, как это делается, скажем, что, в конечном счёте, Ольга будет иметь частицы 1 и 2 в одном из четырёх перепутанных состояний, так называемых состояниях Белла:

$$\Psi_{12} = 1/\sqrt{2} [(0)1 (1)2 \pm (1)1 (0)2]$$

$$\Theta_{12} = 1/\sqrt{2} [(0)1 (0)2 \pm (1)1 (1)2].$$

В 25% случаев Ольга получает антисимметричное состояние Θ_{12} . В этих случаях состояния частиц 1 и 3 идентичны друг другу, так как частица 2 перепутана с частицей 3 и с частицей 1 одинаковым образом. Поэтому, если частица 1 была в состоянии ϕ_1 , то частица 3 проецируется в состояние суперпозиции, идентичное этому состоянию частицы 1, через антисимметричную перепутанность состояний 1 и 2. Если Ольга получит любое из других трёх состояний Белла, существуют определённые преобразования, применив которые, Павел может перевести свою частицу в состояние, идентичное ϕ_1 .

Экспериментальная реализация квантовой телепортации

Эти теоретические идеи были реализованы двумя группами, в Риме и Инсбруке. Мы расскажем об эксперименте инсбрукской группы, в котором использовались поляризованные фотоны.

Перепутанная пара создавалась знакомым нам способом спонтанного параметрического понижающего преобразования. Фотоны, полученные таким способом, имеют идентичные длины волн, но линейно поляризованы в ортогональных направлениях, что гарантирует необходимую перепутанность их состояний. Один из них (фотон 2) послали Ольге, у которой был фотон, предназначенный для телепортации (фотон 1). Эти два фотона одновременно, в пределах экспериментального разрешения, направлялись в делитель луча и двигались после этого в одном направлении. При этих условиях они были неразличимы и, значит, перепутаны.

В процессе перепутывания с одинаковой вероятностью возникает одно из четырёх состояний Белла. В антисимметричном состоянии Ψ_{12} , как было написано выше, после измерения Ольгой частица 3 оказывается в первоначальном состоянии частицы 1, и от Павла не требуется дополнительных преобразований. Так как преобразования, необходимые в случае трёх остальных состояний Белла, являются сложной

экспериментальной проблемой, инсбрукская группа ограничилась демонстрацией телепортации только для случая антисимметричного состояния Ψ_{12} , возникающего в 25% случаев, отложив на будущее задачу телепортации фотонов в случаях появления симметричных перепутанных состояний.

Характер симметрии состояний Белла влияет на то, как фотоны выходят из делителя луча. Если волновая функция симметрична, фотоны выходят в одном направлении; если она является антисимметричной, они выходят в противоположных направлениях. Эта особенность была использована для обнаружения состояния Ψ_{12} , так как только оно является антисимметричным. Ольга, посылая оба фотона в свой делитель луча, выбирала только те случаи, когда на двух выходах делителя одновременно детектировались по одному фотону 1 и 2. Это гарантирует определение состояния, так как в других состояниях Белла два фотона должны детектироваться на одном выходе делителя луча.

Обнаружив фотоны на двух выходах делителя, шпион сообщает получателю через классический канал связи, что было получено состояние Ψ_{12} . Получение Павлом фотона 3 в этом случае означает реализацию телепортации, так как после измерения Ольгой состояния Ψ_{12} фотон 3 оказывается в том же самом состоянии, в котором находился фотон 1 до измерения. Инсбрукская группа не работала с тремя симметричными состояниями Белла, так как не было ещё способа обнаружить каждое из этих состояний и осуществить необходимое преобразование. Телепортация осуществлялась только в 25% случаев.

Инсбрукская группа телепортировала три различных состояния фотона 1: $+45^\circ$, -45° и круговая поляризация. Перепутанное состояние Ψ_{12} фотонов 2 и 3 было получено с помощью параметрического понижающего преобразования в нелинейном кристалле, обозначенном как «ЭПР-источник». Фотон 2 направлялся к Ольге, а фотон 3 к Павлу. Ольга «измеряла состояния Белла», перепутав фотоны 1 и 2 в делителе луча и выделяя случай совпадения сигналов в датчиках 1 и 2, расположенных на двух выходах делителя луча. При обнаружении совпадения она посылает получателю классическое сообщение об обнаружении антисимметричного состояния. С помощью

сложных трёх- и четырёхкратных схем совпадения группа показала, что в этих случаях состояние фотона 3, полученного получателем, было идентично тому, в котором был создан фотон 1.

Квантовая телепортация связана со странным раздвоением информации.

Между временем, когда Ольга разрушила своё неизвестное состояние, и временем, когда Павел восстановил его, информация, описывающая это состояние, была разделена на две части. Одна часть является просто классической: описание Ольгой результатов измерения. Это описание может быть передано в вечерних новостях, послано Павлу в письме или как угодно: это есть информация в обычном виде, с которыми мы давно знакомы.

Вторая часть есть чисто квантовая информация: абсолютная антикорреляция, присущая ЭПР-паре. Эта часть информации передаётся мгновенно на большие расстояния, но нельзя думать об этом как о переносе сообщения. Только когда эти две формы информации объединены, Павел завершает процесс телепортации.

Примечательной особенностью телепортации является то, что она возможна независимо от того, знает ли отправитель, где находится Павел. Предположим, что после получения своего члена ЭПР-пары Павел переехал на новое место, неизвестное Ольге, взяв свою частицу с собой. Если Ольга опубликует результаты своего измерения в газете, Павел может прочесть это сообщение и восстановить первоначальное квантовое состояние. В этом заключается отличие от самого простого из всех методов послать квантовое состояние получателю, когда ему посылается непосредственно частица. Чтобы сделать это, отправитель должен был бы знать местонахождение получателя, чтобы направить ему частицу.

СТРАТЕГИИ ПОДСЛУШИВАТЕЛЯ

Доказательство секретности протокола BB84 строится на том, что при величине ошибки на приёмной стороне менее 11% возможна секретная передача данных. В то же время не говорится о том, каким образом протокол теряет секретность при большей величине ошибки. Построим схему атаки, при которой достигается теоретический предел ошибки на приёмной стороне в 11%. Рассмотрим и другие стратегии подслушивателя и найдём критические величины ошибки для каждой из них. Заранее отметим, что наиболее общим случаем подслушивания является коллективная атака: при незначительном изменении протокола более общая когерентная атака не даёт дополнительной выгоды перехватчику.

Приём-перепосыл

Наиболее простым сценарием действий шпиона является измерение передаваемого по квантовому каналу состояния с дальнейшим пересылом полученного результата дальше. Именно таким образом могут прослушиваться классические каналы. В квантовом случае эта стратегия не срабатывает.

Если шпион стремится выполнить те же действия, что на своей стороне производит Павел, то, не зная исходного состояния, он неизбежно сталкивается с нерешаемой проблемой различения состояний из неортогонального набора. Так, применяя случайным образом одно из измерений

$$\begin{aligned} + : \quad M_x &= |x\rangle\langle x| & M_y &= |y\rangle\langle y|, \\ \times : \quad M_u &= |u\rangle\langle u| & M_v &= |v\rangle\langle v| \end{aligned}$$

к посланному состоянию, примерно в половине случаев шпион будет неверно угадывать базис: применять измерение «х» при посланном состоянии $|x\rangle$ или $|y\rangle$ или применять измерение «+» над состоянием из набора $\{|u\rangle, |v\rangle\}$.

В силу свойства несмещённости базисных состояний при неверно угаданном базисе вероятность ошибки шпиона

составляет 50%, то есть шпион не получает полезной информации о сигнале. Однако это ещё не все проблемы шпиона. Неверно угадав базис при измерении, вследствие свойства редукции волновой функции он неизбежно пошлёт ошибочное состояние Павлу. Так, при применении измерения «+» независимо от исходного состояния дальше будет послано одно из состояний $\{|x\rangle, |y\rangle\}$, а при применении измерения «х» — одно из состояний $\{|u\rangle, |v\rangle\}$. Измеряя эти состояния в «верном» для них базисе, Павел получит ошибку, по которой может быть обнаружено вмешательство шпиона.

Величину ошибки на приёмной стороне можно вычислить так: допустим, шпион подвергал атаке не все состояния, а некоторую их часть, атакуя каждый сигнал с вероятностью p . Тогда доля в $1 - p$ сигналов приходит к Павлу без ошибки. Шпиону же приходится просто угадывать значение бита в каждой такой посылке, а значит, в её ошибку это даст вклад, равный $(1 - p)/2$. В то же время для сигналов, атакованных шпионом, существует два равновероятных поворота событий:

- Шпион угадал базис измерения, а значит, получил точную информацию о передаваемом сигнале и не внёс какого-либо возмущения в передачу.
- Шпион ошибся в выборе базиса измерений, а значит, с вероятностью $1/2$ получил ошибочный результат и передал ошибочное состояние Павлу, что дало ошибку на его стороне с вероятностью опять же $1/2$.

Вероятность каждого из подобных сценариев равна $p/2$, и при такой стратегии доля ошибок на приёмной стороне будет равна $p/4$, а доля ошибок у шпиона составит

$$\frac{1-p}{2} + \frac{p}{4} = \frac{1}{2} - \frac{p}{4}.$$

Это значит, что при всех значениях параметра p , меньших единицы, шпион имеет больше ошибок, чем Павел, а значит, его информация о передаваемом ключе строго меньше. В то же время при $p = 1$ доля ошибок у Павла и шпиона совпадает и равна 25%. Так как ошибка Павла однозначно связана с параметром p , можно считать, что 25% — пороговая величина ошибки при

такой атаке, до которой возможно секретное распространение ключей.

Отметим, что ошибка на приёмном канале может быть вызвана не только действиями шпиона, но и неидеальностью канала или детекторов. Однако при оценке стойкости протокола предполагается, что все ошибки были вызваны перехватчиком: очевидно, это лучший для него вариант развития событий.

Таким образом, критическая ошибка на приёмной стороне — основная характеристика протоколов квантовой криптографии. В общем случае она зависит лишь от самого протокола, однако в ряде случаев атак можно вычислить каждую из них. Протокол квантовой криптографии тем лучше, чем больше его критическая ошибка: в этом случае он лучше противостоит помехам в канале связи и способен генерировать секретный ключ с большей скоростью и на больших расстояниях.

К стратегиям подслушивания относят также «прозрачное индивидуальное подслушивание», «коллективную атаку», «когерентную атаку», PNS-атаку.

Критическая длина линии связи

Важным фактором в схемах атаки является компенсация дополнительного затухания, вызванного блокировкой части импульсов шпионом: при отсутствии такой компенсации шпион может быть обнаружен по показателям затухания. Так как исходные потери в канале зависят от его длины, шпион может компенсировать блокировку всех однофотонных импульсов только при использовании достаточно длинного канала между Ольгой и Павлом. Покажем, как оценивается критическая длина канала (протокол BB84, атака PNS).

Действия шпиона при проведении PNS-атаки сводятся к следующему. Не меняя достигающих Павла посылок, шпион должен блокировать как можно больше однофотонных сигналов, оставляя у себя один из фотонов в случае многофотонного импульса. В идеальном для шпиона случае он должен блокировать все однофотонные компоненты, таким образом получая всю информацию о передаваемом ключе. Он может это сделать в том случае, когда количество испускаемых многофотонных

импульсов оказывается не меньше количества достигающих приёмной стороны сигналов.

Число фотонов в лазерном импульсе распределено по закону Пуассона.

$$p(n) = \frac{e^{-\mu} \mu^n}{n!},$$

где μ — среднее число фотонов, обычно приближённо равно 0,1.

Вероятность испускания с одним фотоном равна

$$p_1 = \mu e^{-\mu},$$

а вероятность генерации импульса с несколькими ($n \geq 2$) фотонами равна

$$p_{\geq 2} = 1 - e^{-\mu} - \mu e^{-\mu}.$$

В этих выражениях $e^{-\mu}$ — вероятность вакуумной компоненты, то есть состояния без фотонов. Доля фотонов, которые достигнут приёмной стороны в канале длины L с коэффициентом поглощения α , равна

$$(p_1 + p_{\geq 2})10^{-\alpha L/10}.$$

Для стандартных современных одномодовых волокон типа SMF-28 коэффициент поглощения составляет $\alpha = 0.18—0.2$ Дб/км. В приведённой формуле была использована консервативная в пользу шпиона оценка, так как вероятности достижения приёмной стороны отличаются для состояний с разным количеством фотонов, а чем меньше вероятность достижения приёмника, тем больше возможности шпиона по перехвату.

Так как величина в последней формуле является константой протокола и зависит от длины канала связи L , можно говорить о критической величине расстояния между Ольгой и Павлом, до которого PNS-атака является неприменимой. Таким образом, целью противодействия PNS-атаке является увеличение критической длины линии связи: чем она больше, тем более устойчивым является протокол.

ЭТОТ ШИФР НЕ ОДОЛЕТЬ

В 1984 году американец Чарльз Беннетт и канадец Жиль Брассард придумали систему шифрования, основанную на передаче поляризованных фотонов.

Изначально отправитель (Ольга) и получатель (Павел) договариваются, какой именно поляризации приписываются 0 или 1.

В приводимом здесь примере приписывание 0 и 1 будет функцией двух диаграмм или базисов поляризации: первый базис, который называется прямолинейным и представлен символом +, представляет 1 в качестве соответствия поляризации \updownarrow , а 0 — поляризации \leftrightarrow . Второй базис, который называется диагональным и представлен символом X, приписывает 1 поляризации «диагонально влево», а 0 — поляризации «диагонально вправо».

Например, сообщение 0100101011 может быть передано следующим образом:

Сообщение	0	1	0	0	1	0	1	0	1	1
Базис	×	+	+	×	+	×	×	+	×	+
Передача	\swarrow	\updownarrow	\leftrightarrow	\swarrow	\updownarrow	\swarrow	\nearrow	\leftrightarrow	\nearrow	\updownarrow

Если шпион перехватит передачу, ему потребуется использовать фильтр с фиксированной ориентацией X.

Исходное сообщение	\swarrow	\updownarrow	\leftrightarrow	\swarrow	\updownarrow	\swarrow	\nearrow	\leftrightarrow	\nearrow	\updownarrow
Фильтр	×									
Обнаруженная поляризация	\swarrow	\swarrow или \nearrow	\swarrow или \nearrow	\swarrow	\swarrow или \nearrow	\swarrow	\nearrow	\swarrow или \nearrow	\nearrow	\swarrow или \nearrow
Возможное сообщение	$\swarrow \updownarrow \leftrightarrow$	\swarrow или \updownarrow	\swarrow или \leftrightarrow	\swarrow или \updownarrow	\swarrow или \leftrightarrow	\swarrow или \updownarrow	$\nearrow \updownarrow \leftrightarrow$	\swarrow или \updownarrow	\nearrow или \leftrightarrow	\swarrow или \updownarrow

Как мы видим, не зная исходного базиса, шпион не может получить из обнаруженной поляризации вообще никакой релевантной информации. Даже зная схему соответствия 0 и 1, которую использовали Ольга и Павел, если первый поменяет

базисы наугад, то шпион будет ошибаться примерно одну треть времени (в таблице показано расчленение всех отправляемых и получаемых комбинаций, возможных при описанных условиях). Однако имеется весьма очевидная проблема: Павел оказывается в несколько не лучшем положении, чем шпион.

Добравшись до этой точки, Ольга и Павел могут обойти проблему, отправив друг другу последовательность используемых базисов с применением какого-то безопасного средства, как, например, шифрование с помощью RSA. Но тогда риску подвергнется безопасность шифра, эта угроза исходит от гипотетических квантовых компьютеров.

Чтобы преодолеть последнее препятствие, Брассард и Беннетт добавили к своему методу ещё одну тонкость.

Вспомним: ахиллесовой пятой полиалфавитных шифров из группы, использующей квадрат Виженера, было то, что использование коротких, повторяющихся ключей порождало регулярность в шифре, и повторение предоставляло криптоаналитикам определённые возможности для взлома.

Однако что случится, если используемый ключ представляет собой выбранный наугад ряд знаков, который окажется длиннее, чем само сообщение? И что, если, для большей надёжности, каждое послание, каким бы незначительным оно ни было, шифруется другим ключом?

Ответ — мы получим шифр, который взломать попросту невозможно. Одноразовый шифр-блокнот.

Первым человеком, который предложил использовать полиалфавитный шифр с уникальным ключом, был Джозеф Моборн. Вскоре после Первой мировой войны, когда он работал начальником отдела связи в Американской шифровальной службе, Моборн представил так называемый «одноразовый шифр-блокнот», состоящий из ключей, составленных из выбранных наугад рядов из более сотни знаков каждый.

Блокнот должен предоставляться Ольге и Павлу с указанием всякий раз уничтожать использованный ключ и переходить к следующему. Такая система является, как мы уже говорили, не поддающейся взлому, и это её свойство может быть продемонстрировано математически.

Заметим, что связь между главами государств, которая должна быть в высшей степени секретной, осуществляется с помощью этого способа.

Впрочем, если шифр «одноразового шифр-блокнота» настолько надёжен, почему его использование не получило широкого распространения? И почему нас беспокоит мощность квантовых компьютеров и даже упоминание о манипуляциях с фотонами?

Если не учитывать логистические трудности, связанные с составлением тысяч взятых наугад одноразовых ключей, шифр «одноразового шифр-блокнота» имеет то же самое слабое место, что и другие классические алгоритмы шифрования: распространение ключей — самый важный вопрос, который стремится решить современная криптография.

Однако передача информации поляризованными фотонами является идеальным каналом для безопасного представления уникального ключа. Чтобы это произошло, необходимы три шага до передачи послания:

1. Во-первых, Ольга отправляет получателю выбранную наугад последовательность 1 и 0 при помощи различных, точно так же выбранных наугад, фильтров вертикальных (\updownarrow), горизонтальных (\leftrightarrow) и диагональных (\nearrow, \searrow) выстраиваний.
2. Павел измеряет поляризацию полученных фотонов при помощи выбранного наугад чередования прямолинейных базисов (+) и диагональных базисов (X). Поскольку он не знает последовательность фильтров, которая использовалась Ольгой, большая часть последовательности из 0 и 1 тоже будет неправильной.
3. Наконец, Ольга и Павел устанавливают контакт любым способом, который предпочитают, не беспокоясь

о том, что канал ненадёжен, и обмениваются следующей информацией:

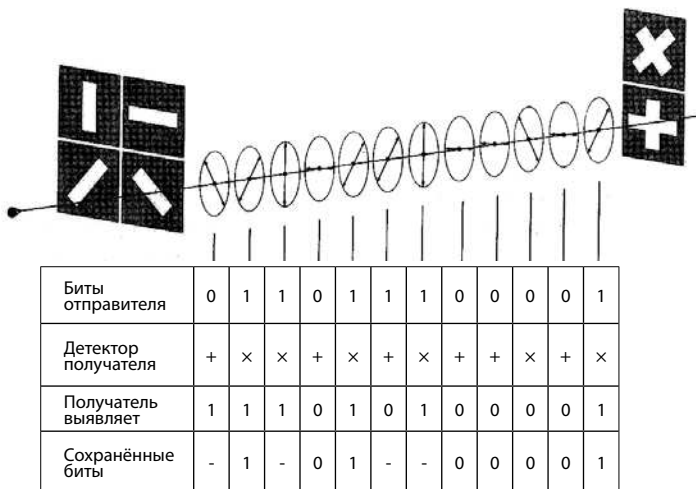
- Ольга объясняет, какой базис, прямолинейный или диагональный, должен быть использован для правильного прочтения каждого фотона, но без раскрытия его поляризации (то есть используемого фильтра);
- Павел сообщает Ольге, в каких случаях он правильно выбрал базис. Как мы видим из таблицы, если отправитель и получатель правильно получили свои соответствующие базисы, мы можем быть уверены, что передача 0 и 1 была завершена правильно;
- наконец (и тайно) каждый из них выбрасывает биты, соответствующие фотонам, которые получатель идентифицировал с ошибочным базисом.

Результат этого процесса заключается в том, что Ольга и Павел теперь имеют последовательность из 1 и 0, составленную совершенно наугад: выбор поляризационных фильтров,

Базис отправителя	Бит отправителя	Отправитель передаёт	Детектор получателя	Детектор сработал правильно?	Получатель выявляет	Бит получателя	Является ли бит получателя правильным?
Диагональные	1		+	нет		1	да
			×	да		0	нет
	0		+	нет		1	нет
			×	да		0	да
			+	да		1	да
			×	нет		0	нет
Прямолинейные	1		+	да		1	да
			×	нет		0	нет
	0		+	да		1	да
			×	нет		0	да
			+	да		1	нет
			×	нет		0	да

которые использует отправитель, произволен. Как произволен и выбор базисов, которые использует получатель.

Пример процесса, описанного выше, со всего лишь двенадцатью битами представлен на схеме:



Последовательность поляризованных фотонов на рисунке соответствует столбцам таблицы (вертикальные чёрточки)

Обратите внимание на тот факт, что из оставшихся в конце концов битов некоторые отбрасываются, даже если были правильно интерпретированы. Это происходит потому, что получатель не может быть уверен в том, что, используя неправильные базисы, выявил их правильно. Если исходная передача состоит из достаточного количества фотонов, то последовательность из 0 и 1 будет слишком длинной, чтобы составить шифр «одноразового шифр-блокнота», пригодный для шифрования посланий обычной длины.

Теперь поставим себя на место шпиона, который перехватил и отправленные фотоны, и открытые переговоры отправителя (Ольги) и получателя (Павла).

Мы уже видели, что, не зная точно, какой поляризационный фильтр использовался Ольгой, невозможно определить, где мы выявили правильную поляризацию. Не поможет тут

и информация, которой обмениваются отправитель и получатель, поскольку они никогда не передают информацию о конкретной поляризации.

Ещё больше шпион расстроится, когда после того как он не смог определить правильный базис (а из-за этого правильное изменение поляризации фотона), его вторжение ещё и обнаружат. Ведь ему никак не скрыть своё вмешательство — Ольге и Павлу достаточно проверить часть ключа довольно большой длины, чтобы выявить любую манипуляцию с поляризацией фотонов тем, кто пытался перехватить информацию.

Для этого Ольга и Павел договариваются об очень простом протоколе проверки. Завершив три предварительных шага, которые представлены выше, и при достаточном количестве сохранённых битов, Ольга при помощи какого-то традиционного средства устанавливает контакт с Павлом, и они совместно проверяют группу, скажем, из 100 битов, выбранных наугад из общей массы. Если они совпадают, то и Ольга и Павел могут быть полностью уверены, что постороннего вмешательства в передачу не было и можно считать данную последовательность хорошим одноразовым шифром. В противном случае Ольге и Павлу придётся начинать весь процесс снова.

Кратко процедуру можно описать так: передаваемое сообщение записывается в двоичном формате, а затем берётся полностью случайный ключ такой же длины и производится побитовое сложение сообщения и ключа. Получатель, зная ключ, производит на своей стороне побитовое сложение, получая в точности исходное сообщение. После выполнения этих операций ключ перестаёт использоваться, чем и объясняется название метода «одноразовый шифр-блокнот».

От сантиметров к километрам абсолютной секретности

Метод Brassarda и Bennetta безупречен с теоретической точки зрения, но когда теория перекаладывается на практические рельсы, её встречают с большой долей скептицизма.

В 1989 году, после многих месяцев напряжённой работы, Беннетт настроил систему, состоящую из двух компьютеров, установленных на расстоянии 32 см. Один из них играл роль

отправителя, а другой — получателя. Через несколько часов настроек и проб эксперимент был признан успешным.

Ольга и Павел завершили все стадии процесса и даже смогли проверить свои шифры.

Квантовая криптография оказалась возможна.

Исторический эксперимент Беннетта имел очевидный недостаток: производилась отправка секретных посланий, длина которых меньше страницы. Вероятно, не менее эффективной была бы передача информации шёпотом... Однако в последующие годы другие группы исследователей увеличили длину передачи.

В 1995 году исследователи из Женевского университета использовали оптоволоконный кабель для отправки сообщений на 23 км. В 2006 году группа из Лос-Аламосской национальной лаборатории в США, выполняя ту же процедуру, достигла расстояния в 107 км.

Хотя этот метод ещё мало полезен для общераспространённой связи, поскольку действует на небольших расстояниях, он используется в малых масштабах там, где необходима высшая степень секретности. К примеру, в правительственных зданиях и головных офисах корпораций.

Оставляя в стороне соображения, связанные с физическим ограничением отправки сообщений, нет никакой возможности саботировать передачу, даже на квантовом уровне.

Упомянутый квантовый код представляет собой окончательную победу секретности над неосторожностью и неблагоприятным, криптографии над криптоанализом. Теперь нам нужно беспокоиться только о том, каким образом применять этот мощный инструмент на практике и кто от этого выиграет.

И вы прекрасно понимаете, что это — ни в коем случае не второстепенный вопрос.

ГЛАВА 9.

И, НАКОНЕЦ, ЧТО ЖЕ ЭТО — КВАНТОВЫЙ КОМПЬЮТЕР?

«Я утверждаю, что квантовые компьютеры — устройства, потенциально способные взломать все сегодняшние шифры, — находятся на самом начальном этапе разработки, но не исключено, что такой компьютер уже кем-то создан. Те, кто единственно способен указать на мои ошибки, — это в то же время те, кто не волен этого сделать».

Сингх Саймон. «Книга шифров»

«Машина может сгенерировать сообщение, а сообщение может сгенерировать машину».

Норберт Винер

Содержание предыдущих глав убеждает нас, что плеяда современных учёных надёжно обосновала... то, чего нет. Или пока нет.

*«Полноценный квантовый компьютер является пока гипотетическим устройством, сама возможность построения которого связана с серьёзным развитием квантовой теории в области многих частиц и сложных экспериментов; эта работа лежит на переднем крае современной физики. Ограниченные (до 512 кубитов) **квантовые компьютеры уже построены**».*

(Википедия)

Теоретическое обоснование квантовых вычислений дал Дэвид Дойч в своей работе *The Fabric of Reality* («Структура реальности»):

«Для любого, кто не знаком с этим предметом, квантовое вычисление звучит как название новой технологии, возможно, самой последней в знаменитом ряду, включающем механическое вычисление, транзисторно-электронное вычисление, вычисление на кремниевых кристаллах и т. д. Но истина в том, что даже существующие компьютерные технологии зависят от микроскопических квантово-механических процессов (конечно, все физические процессы являются квантово-механическими, но здесь я имею в виду только те, для которых классическая — т. е. неквантовая — физика даёт очень неточные предсказания). Если существует тенденция к получению даже

более быстрых компьютеров с более компактным аппаратным обеспечением, технология должна стать в этом смысле даже более „квантово-механической“ просто потому, что квантово-механические эффекты доминируют во всех достаточно маленьких системах. Но если бы дело было только в этом, квантовое вычисление вряд ли смогло бы фигурировать в любом фундаментальном объяснении структуры реальности, поскольку в нём не было бы ничего фундаментально нового. Все современные компьютеры, какие бы квантово-механические процессы они ни использовали, — всего лишь различные технологические исполнения одной и той же классической идеи универсальной машины Тьюринга. Именно поэтому все существующие компьютеры имеют в сущности один и тот же репертуар вычислений: отличие состоит только в скорости, ёмкости памяти и устройствах ввода–вывода. Это всё равно что сказать, что даже самый непритязательный современный домашний компьютер можно запрограммировать для решения любой задачи или передачи любой среды, которую могут передать наши самые мощные компьютеры, при условии установки на него дополнительной памяти, достаточно долгом времени обработки и наличии аппаратного обеспечения, подходящего для демонстрации результатов работы.

Квантовое вычисление — это нечто большее, чем просто более быстрая и миниатюрная технология реализации машин Тьюринга. Квантовый компьютер — это машина, использующая уникальные квантово-механические эффекты, в особенности интерференцию, для выполнения совершенно новых видов вычислений, которые даже в принципе невозможно выполнить ни на одной машине Тьюринга, а следовательно, ни на каком классическом компьютере. Таким образом, квантовое вычисление — это не что иное, как принципиально новый способ использования природы».

ВОЗМОЖНОСТЬ СОЗДАНИЯ КВАНТОВОГО КОМПЬЮТЕРА

Из-за чрезвычайно широкой распространённости алгоритма RSA одним из важнейших предположений криптографии является сложность задачи факторизации больших чисел. И действительно, до настоящего времени не найдено алгоритма, достаточно быстро решающего эту задачу. Однако в 1994 году Шор (Shor P.W.) предложил алгоритм, с полиномиальной сложностью решающий эту задачу на квантовом компьютере. Главная причина подобного феноменального ускорения — в возможности использования так называемого «квантового параллелизма» для проведения быстрого преобразования Фурье, на котором основаны наиболее эффективные из известных алгоритмов факторизации.

Применение этого алгоритма позволяет свести задачу факторизации к технологической задаче построения квантового компьютера: если его удастся построить, схема шифрования RSA окажется ненадёжной. Это ставит возможности шифрования с открытым ключом под большую угрозу. Стоит отметить, что за последнее десятилетие не было достигнуто существенного прогресса в построении квантового компьютера. Однако протоколы квантового распределения ключей позволяют на приемлемой скорости генерировать полностью секретные ключи между удалёнными абонентами.

УСТРОЙСТВО КВАНТОВОГО КОМПЬЮТЕРА

Прежде чем рассказать, как же устроен квантовый компьютер, вспомним основные особенности квантовых частиц.

Для понимания законов квантового мира не следует прямо опираться на повседневный опыт.

Обычным образом (в житейском понимании) квантовые частицы ведут себя лишь в том случае, если мы постоянно «подглядываем» за ними, или, говоря более строго, постоянно измеряем, в каком состоянии они находятся. Но стоит нам «отвернуться» (прекратить наблюдение), как квантовые частицы тут же переходят из вполне определённого состояния сразу в несколько различных ипостасей. То есть электрон (или любой другой квантовый объект) частично будет находиться в одной точке, частично в другой, частично в третьей и т.д. Это не означает, что он делится на дольки, как апельсин. Тогда можно было бы надёжно изолировать какую-нибудь часть электрона и измерить её заряд или массу.

Но опыт показывает, что после измерения электрон всегда оказывается «целым и невредимым» в одной-единственной точке, несмотря на то, что до этого он успел побывать одновременно почти везде. Такое состояние электрона, когда он находится сразу в нескольких точках пространства, называют суперпозицией квантовых состояний и описывают обычно волновой функцией, введённой в 1926 году Э. Шрёдингером. Модуль значения волновой функции в любой точке, возведённый в квадрат, определяет вероятность найти частицу в этой точке в данный момент. После измерения положения частицы её волновая функция как бы стягивается (коллапсирует) в ту точку, где частица была обнаружена, а затем опять начинает расплываться. Свойство квантовых частиц быть одновременно во многих состояниях, называемое квантовым параллелизмом, успешно используется в квантовых вычислениях.

Сет Ллойд, профессор Массачусетского технологического института, так иллюстрирует суть квантового компьютера:

«Классическое вычисление похоже на сольную партию одного музыкального инструмента — отдельные строки чистых тонов. Квантовое вычисление похоже на симфонию, состоящую из множества тонов, интерферирующих друг с другом».

Квантовый бит

Основная ячейка квантового компьютера — квантовый бит, или, сокращённо, кубит (q-бит). Это квантовая частица, имеющая два базовых состояния. Двум значениям кубита могут соответствовать, например, основное и возбуждённое состояния атома, направления вверх и вниз спина атомного ядра, направление тока в сверхпроводящем кольце, два возможных положения электрона в полупроводнике и т. п.

Квантовый регистр

Квантовый регистр устроен почти так же, как и классический. Это цепочка квантовых битов, над которыми можно проводить одно- и двухбитовые логические операции (подобно применению операций НЕ, 2 И-НЕ и т. п. в классическом регистре).

К базовым состояниям квантового регистра, образованного L кубитами, относятся, так же как и в классическом, все возможные последовательности нулей и единиц длиной L . Всего может быть 2^L различных комбинаций. Их можно считать записью чисел в двоичной форме от 0 до $2^L - 1$. Однако эти базовые состояния не исчерпывают всех возможных значений квантового регистра (в отличие от классического), поскольку существуют ещё и состояния суперпозиции, задаваемые комплексными амплитудами, связанными условием нормировки. Классического аналога у большинства возможных значений квантового регистра (за исключением базовых) просто не существует. Состояния классического регистра — лишь жалкая тень всего богатства состояний квантового компьютера.

Представьте, что на регистр осуществляется внешнее воздействие, например, в часть пространства поданы

электрические импульсы или направлены лазерные лучи. Если это классический регистр, импульс, который можно рассматривать как вычислительную операцию, изменит L переменных. Если же это квантовый регистр, то тот же импульс может одновременно преобразовать до переменных. Таким образом, квантовый регистр, в принципе, способен обрабатывать информацию в раз быстрее по сравнению со своим классическим аналогом. Отсюда сразу видно, что маленькие квантовые регистры ($L < 20$) могут служить лишь для демонстрации отдельных узлов и принципов работы квантового компьютера, но не принесут большой практической пользы, так как не сумеют обогнать современные ЭВМ, а стоить будут заведомо дороже. В действительности квантовое ускорение обычно значительно меньше, чем приведённая грубая оценка сверху (это связано со сложностью получения большого количества амплитуд и считывания результата), поэтому практически полезный квантовый компьютер должен содержать тысячи кубитов. Но, с другой стороны, понятно, что для достижения действительного ускорения вычислений нет необходимости собирать миллионы квантовых битов.

Компьютер с памятью, измеряемой всего лишь в килокубитах, будет в некоторых задачах несоизмеримо быстрее, чем классический суперкомпьютер с терабайтами памяти.

Стоит, однако, отметить, что существует класс задач, для которых квантовые алгоритмы не дают значительного ускорения по сравнению с классическими. Работа квантового компьютера радикально отличается от работы обычного вычислительного устройства прежде всего тем, что вы не можете остановить процесс вычисления и посмотреть промежуточный результат. Любое вмешательство в работу квантового компьютера превратит «переплетение теней вероятности» в простой бит. И это губительно повлияет на конечный результат. Единственный способ узнать результат вычислений без искажений — дожидаться окончания квантовой работы.

Квантовая логика такого компьютера радикально изменяет характер вычислительного процесса. Квантовый компьютер

способен выполнять две операции одновременно, а квантовый кубит способен хранить одновременно результаты этих операций. Эта способность делать две вещи сразу присуща именно квантовой механике. В двухщелевом эксперименте фотон проходит через обе щели, эта способность связана с его волновой природой. Волновая природа квантов позволяет им интерферировать друг с другом, производя качественно новые формы и явления.

И тем не менее нет сомнения, что компьютеры, работающие по законам квантовой механики, — новый и решающий этап в эволюции вычислительных систем. Осталось только их построить.

КВАНТОВЫЕ КОМПЬЮТЕРЫ СЕГОДНЯ

Прототипы квантовых компьютеров существуют уже сегодня. Правда, пока что экспериментально удаётся собирать небольшие регистры, состоящие всего из нескольких квантовых битов. Так, недавно группа, возглавляемая американским физиком И. Чангом (IBM), объявила о сборке 5-битового квантового компьютера. Несомненно, это большой успех. К сожалению, существующие квантовые системы ещё не способны обеспечить надёжные вычисления, так как они либо недостаточно управляемы, либо очень подвержены влиянию шумов.

Однако физических запретов на построение эффективного квантового компьютера нет, необходимо лишь преодолеть технологические трудности.

Существует несколько идей и предложений, как сделать надёжные и легко управляемые квантовые биты. И. Чанг развивает идею об использовании в качестве кубитов спинов ядер некоторых органических молекул. Российский исследователь М. В. Фейгельман, работающий в Институте теоретической физики им. Л. Д. Ландау РАН, предлагает собирать квантовые регистры из миниатюрных сверхпроводниковых колец. Каждое кольцо выполняет роль кубита, а состояниям 0 и 1 соответствуют направления электрического тока в кольце — по часовой стрелке и против неё. Переключать такие кубиты можно магнитным полем.

В Физико-технологическом институте РАН группа под руководством академика К. А. Валиева предложила два варианта размещения кубитов в полупроводниковых структурах.

В первом случае роль кубита выполняет электрон в системе из двух потенциальных ям, создаваемых напряжением, приложенным к мини-электродам на поверхности полупроводника. Состояния 0 и 1 — положения электрона в одной из этих ям. Переключается кубит изменением напряжения на одном из электродов.

В другом варианте кубитом является ядро атома фосфора, внедрённого в определённую точку полупроводника.

Состояния 0 и 1 — направления спина ядра вдоль либо против внешнего магнитного поля. Управление ведётся с помощью совместного действия магнитных импульсов резонансной частоты и импульсов напряжения.

Таким образом, исследования активно ведутся и можно предположить, что в самом недалёком будущем — лет через десять — эффективный квантовый компьютер будет создан.

ВЗГЛЯД В БУДУЩЕЕ

Попробуем представить, как мог бы выглядеть будущий квантовый компьютер. Вероятно, большой (масштабируемый) компьютер будет содержать тысячи управляющих элементов, действующих локально на каждый кубит. Каким образом могло бы осуществляться это воздействие?

Скорее всего, с помощью электрических импульсов, подаваемых на микроэлектроды, подведённые к кубитам. Возможно также оптическое управление пучками света, сфокусированными на кубитах. Однако в этом случае трудно избежать паразитного воздействия на соседние кубиты дифракционных краёв сфокусированного пучка. Что касается электрических методов, то они уже давно и широко применяются в микроэлектронике для управления классическими логическими элементами. Поэтому их использование представляется наиболее перспективным и для создания масштабируемых квантовых компьютеров (возможно, конечно, что в результате какого-нибудь технологического прорыва появится ещё и третий вариант, однако революционные открытия трудно поддаются прогнозу).

Таким образом, весьма возможно, что в перспективе квантовые компьютеры будут изготавливаться с использованием традиционных методов микроэлектронной технологии и содержать множество управляющих электродов, напоминая современный микропроцессор. Чтобы снизить уровень шумов, критически важный для нормальной работы квантового компьютера, первые модели, по всей видимости, придётся охлаждать жидким гелием.

Вероятно, первые квантовые компьютеры будут громоздкими и дорогими устройствами, не уместяющимися на письменном столе и обслуживаемыми большим штатом системных программистов и наладчиков оборудования.

Доступ к ним получат сначала лишь государственные структуры, затем богатые коммерческие организации. Но примерно так же начиналась и эра обычных компьютеров.

А что же станет с классическими компьютерами? Отомрут ли они? Вряд ли. И для классических, и для квантовых компьютеров найдутся свои сферы применения. Хотя, по всей видимости, соотношение на рынке будет всё же постепенно смещаться в сторону последних.

Внедрение квантовых компьютеров не приведёт к решению принципиально нерешаемых классических задач, а лишь ускорит некоторые вычисления. Кроме того, станет возможна квантовая связь — передача кубитов на расстояние, что приведёт к возникновению своего рода квантового Интернета. Квантовая связь позволит обеспечить защищённое (законами квантовой механики) от подслушивания соединение всех желающих друг с другом. Ваша информация, хранимая в квантовых базах данных, будет надёжнее защищена от копирования, чем сейчас. Компании, производящие программы для квантовых компьютеров, смогут уберечь их от любого, в том числе и незаконного, копирования.

С другой стороны, последствия разработки реального квантового компьютера не будут ограничиваться только крахом криптографии в том виде, в котором мы её знаем. Если такие вычислительные возможности окажутся на службе чьих-либо интересов (будь то государственные или частные организации), то это может изменить равновесие между мировыми державами.

Битва за первенство в разработке такого компьютера легко приведёт к безудержной технологической спешке, которая сможет посоперничать с гонкой вооружений и борьбой за лидерство в космосе во второй половине двадцатого века.

Понятное дело, что сохранение в секрете решительных шагов вперёд в этой области исключительно соответствует интересам национальной безопасности. А потому вполне возможно, что где-нибудь в подземных тайных казематах уже появился реальный квантовый компьютер, терпеливо ждущий, пока его включат на полную мощность, чтобы навсегда изменить нашу жизнь.

Кто знает?..

СОДЕРЖАНИЕ

Глава 1. Кодирование и шифрование	5
От осколка — к кубиту.	6
Код и шифр	8
Сколько нужно ключей?.	10
Принцип Керкгоффса	11
Телеграмма германскому послу.	13
 Глава 2. Криптография от античных времен.	 19
Спарта против Афин	22
Отец аналитической криптографии	24
Аль-Кинди: взлом шифра	28
Шифрование слова Божьего.	30
Частотный анализ на практике	31
Руководство для юных леди	32
Шифровка из «Золотого жука»	33
Шрифт Марии Стюарт	35
Прорыв Альберти	37
Диск Альберти	39
Квадрат Виженера	40
Шифр Гронсфельда	45
Криптографы при дворе «Короля Солнце» . . .	47
Неизвестный криптоаналитик.	48
Криптоаналитик Шерлок Холмс и метод подбора.	51

Удивительная решетка.	52
От криптографии — к стенографии	54
Кино и кодирование	55
Шифровки в траншеях.	56

Глава 3. История шифрования на Руси 57

Самое простое — использовать малоизвестный алфавит.	59
Но ведь знаки для замены букв можно и придумать!	63
«Флопяцевская азбука», «Азбука Копцева» и другие.	67
А почему бы кириллицу не заменить... кириллицей?	75
Воспользуемся цифирью	80
Не связать ли нам шифрочку?	81

Глава 4. Шифровальные машины 83

Азбука Морзе.	84
Невербальная связь	91
Шифр Плейфера	92
Недалеко от Парижа	95
Машина «Энигма»	99
Взлом шифра машины «Энигма»	104
Эстафету принимают англичане	107
Шифр Хилла.	111
Криптографические протоколы	114

Глава 5. Общение при помощи нолей и единиц	115
Двоичный бинарный код	116
Код ASCII	117
Шестнадцатеричная система	119
Системы счисления и замена основания	123
Как измерить информацию	125
Протокол для безопасной передачи	130
 Глава 6. Кодирование в промышленных и торговых масштабах	 131
Первые штрихкоды	137
Штрихкод EAN-13	138
Коды QR	142
Простые числа и малая теорема Ферма	143
 Глава 7. Криптография с использованием компьютера	 145
Как безопасно распределить ключи?	148
На помощь приходят простые числа	153
Надёжный алгоритм RSA	155
Удостоверение подлинности сообщений и ключей.	160
Хэш-подпись	162
Сертификаты открытых ключей.	164
Шифрование во вред	166
Шифрование с помощью операции «XOR» . .	167

Симметричное шифрование	168
Асимметричное шифрование	169
Шифрование с использованием нескольких ключей	171

Глава 8. Квантовая криптография173

Немного квантовой теории	174
Биты и кубиты	185
Вычисляем квантами	188
Передача информации по квантовым каналам.	189
Передача сигнальных состояний.	192
Квантовые коды коррекции ошибок	194
Как избежать подслушивания	197
Квантовые измерения	199
Квантовая телепортация	204
Стратегии подслушателя.	212
Этот шифр не одолеть	216

Глава 9. И, наконец, что же это — квантовый компьютер?223

Возможность создания квантового компьютера.	226
Устройство квантового компьютера.	227
Квантовые компьютеры сегодня	231
Взгляд в будущее	233

ИЗДАТЕЛЬСТВО «СТРАТА»



ПРЕДСТАВЛЯЕТ ЦИКЛ ИЗДАНИЙ

О СИМВОЛАХ

symbol-sign.com

**ПРОСТО СИМВОЛ
СТРУКТУРЫ И СИМВОЛЫ
ЗНАКИ И СИМВОЛЫ
СИМВОЛ И АЛГОРИТМ
СИМВОЛИКА ЦВЕТА
СИМВОЛ И КАПИТАЛ
ОТ АБАКА ДО КУБИТА
ЗОЛОТОЙ СТАНДАРТ
ПРОСТО АРИФМЕТИКА
КАК ИСПЕЧЬ ПИ...
ПРОСТО BIG DATA
ОТ ARPANET ДО INTERNET
ПРОСТО КРИПТОГРАФИЯ**

По вопросам приобретения книг обращайтесь в издательство «СТРАТА» по адресу:

195112, Санкт-Петербург, Заневский пр., 65, корпус 5

Тел.: +7 (812) 320-56-50, 320-69-60

www.strata.spb.ru

Виктор де Касто
ПРО КРИПТОГРАФИЮ

Научно-популярное издание

Верстка и оформление Юрий Костицин
Иллюстрации Татьяна Ковалёва, Максим Ляпунов

Настоящее издание не имеет возрастных ограничений, предусмотренных Федеральным законом РФ «О защите детей от информации, причиняющей вред их здоровью и развитию» (№ 436-ФЗ).

Охраняется законом РФ об авторском праве.

Издательство «Страта»
195112, Санкт-Петербург, Заневский пр., 65, корпус 5
Тел.: +7 (812) 320-56-50, 320-69-60
www.strata.spb.ru

Подписано в печать 21.04.2020

Тираж 100 экз.

«Криптография — одна из областей прикладной математики, в которой наиболее очевиден контраст между первоначальной четкостью, лежащей в основе теории, и туманными последствиями ее внедрения и применения на практике».

Брюс Шнайер

В ваших руках совершенно новый формат издания: **BitBook** — бумажная книга с виртуальной начинкой. У книги BitBook есть собственное пространство в цифровой среде. Мы внедрили **QR-коды** в контент BitBook. Теперь можно перейти на страницу сайта www.symbol-sign.com, где размещены цветные иллюстрации, видео, программы, игры, дополнительная информация, связанная с книгой, ссылки на web-ресурсы; где указаны возможности для получения эксклюзивных данных.

«Только атаки дилетантов нацелены на машины, атаки профессионалов нацелены на людей».

Виктор де Касто



НАУЧНО-ПОПУЛЯРНОЕ
ИЗДАТЕЛЬСТВО
«СТРІСТ»