СОВРЕМЕННАЯ КРИПТОГРАФИЯ «НА ПАЛЬЦАХ»

МИХАИЛ БАЛАНДИН, MICHAEL.BALANDIN@LIVE.RU
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Эта статья представляет собой запись небольшой лекции, прочитанной школьникам выпускного класса на открытом уроке. Для понимания достаточно знаний на уровне элементарной арифметики и алгебры, а также представлений о двоичной системе. В качестве основной проблемы рассматривается задача передачи информации исключительно по открытым каналам.

ПОСТАНОВКА ПРОБЛЕМЫ

История криптографии насчитывает много веков, и за все эти века наиболее часто используемой схемой было так называемое *шифрование с симметричным ключом*. Оно получило такое название из-за того, что для шифровки и расшифровки сообщений используется одинаковый (симметричный) ключ, который обе общающиеся стороны должны знать.

Основной проблемой при этом является *распространение ключа*: необходимо, чтобы все участвующие в зашифрованной переписке стороны имели ключ в своём распоряжении, и при этом он не был доступен никому другому, кроме посвящённых в тайну.

Долгое время считалось, что единственным способом распространения является лишь сугубо *личная* его передача без какого-то использования открытых каналов связи. Традиционно предусматривались способы взаимного информирования сторон о непригодности ключей к дальнейшему использованию (ключ стал известен неприятелю), и в этом случае возникает задача распространения нового ключа.

Дадим сразу определение: *открытым каналом* связи называется такой способ сообщения, при котором вся передаваемая информация может стать известной любым нежелательным третьим лицам. Кстати говоря, здесь существует и другая проблема: *достоверность* канала. Иными словами, сообщающиеся стороны должны быть уверены, что передаваемая ими информация проходит через канал связи без внесения в неё этими третьими лицами каких-либо искажений. Мы в дальнейших рассуждениях будем предполагать именно такую ситуацию: *канал является открытым, но достоверным*.

Если нужно *срочно* передать какую-либо конфиденциальную информацию по такому каналу, а ключ *предварительно не оговорён*, то традиционно прибегают к его оперативной генерации с использованием каких-либо данных, известных обеим сторонам, но заведомо неизвестных потенциальным подслушивающим лицам. Обычно употребляются даты, места, прозвища и вообще сведения личного характера — при условии, разумеется, что у *обеих* сторон *есть* такие общие воспоминания. Если же их *нету*, то задача очень сильно усложняется, либо вообще становится неразрешимой.

Впрочем, этот приём в строгом смысле не удовлетворяет условиям *открытого достоверного канала*: ведь упомянутая личная информация по каналу не передаётся и, таким образом, *не всё*

общение происходит через него. Нас же будет интересовать ситуация, когда перехватить можно *всё*, что сообщают друг другу две стороны.

КЛЮЧ ВООБЩЕ НЕ НУЖЕН?!

Первый подход выглядит довольно парадоксально, но тем не менее работает. Как оказывается, для вполне конфиденциального обмена информацией стороны вообще могут не обмениваться никакими ключами!

Сначала представим себе следующую ситуацию. Одна из сторон — а две стороны-сообщники в криптографии традиционно обозначаются именами *Алиса* и *Боб* — хочет передать другой некоторое послание. Алиса кладёт это послание в шкатулку и запирает её замкóм, ключ от которого есть только у неё. Затем шкатулка пересылается Бобу любым доступным способом (наше предположение о достоверности канала означает, что шкатулка придёт по назначению невскрытой и неподменённой, хотя её могут сколько угодно рассматривать). Боб не имеет ключа от замка Алисы, но он может навесить на шкатулку второй замóк — свой собственный, ключ от которого есть только у него. Затем шкатулка отсылается обратно Алисе. Теперь уже она не может открыть замок Боба, но может снять собственный замок, что она и делает. Шкатулка снова отправляется Бобу — теперь на ней висит лишь его замок, который он снимает и читает послание.

Если теперь заменить «материальное» навешивание и снятие замков «информационным» шифрованием и дешифрованием текста, то становится понятным, кáк можно передать сообщение без обмена ключами. Здесь, правда, есть одно «но»…

Пусть x — это исходное сообщение. Функция f будет обозначать шифрование сообщения Алисой, а функция f^{-1} — дешифрование сообщения ей же. Аналогично, функции g и g^{-1} будут означать шифрование и дешифрование сообщения Бобом. Как нетрудно видеть, общая схема передачи сообщения выглядит так:

$$\underset{\text{Алиса}}{x} \to \underset{\text{Боб}}{f(x)} \to \underset{\text{Алиса}}{g\big(f(x)\big)} \to \underset{\text{Боб}}{f^{-1}\Big(g\big(f(x)\big)\Big)} \to \underset{\text{Расшифровано Бобом}}{g^{-1}\Big(f^{-1}\Big(g\big(f(x)\big)\Big)\Big)}.$$

Достаточно очевидно, что пары функций f, f^{-1} и g, g^{-1} являются взаимно обратимыми, то есть $f^{-1}\big(f(x)\big) = x$ и $g^{-1}\big(g(x)\big) = x$. Однако для того, чтобы схема работала, этого мало. Нужна их обратимость *друг через друга*, то есть

$$f^{-1}\Big(g\big(f(x)\big)\Big) = g(x) \text{ и } g^{-1}\Big(f^{-1}\big(g(x)\big)\Big) = f^{-1}(x).$$

В общем случае эти условия вовсе не обязаны выполняться. Но методика шифрования и дешифрования, удовлетворяющая им, существует.

Пусть мы перевели секретное сообщение из текстового в цифровой вид. Если Алиса и Боб пользуются компьютерами, то это происходит само собой: каждый символ представляется своим ASCII или Unicode-номером. Для уменьшения объёма передачи данных (а это всегда затрудняет взлом шифра) существуют специальные методы устранения избыточности, но это тема отдельного разговора. В двоичной системе счисления сообщение является последовательностью битов, каждый из которых может иметь значение 0 или 1.

Введём следующую функцию двух переменных, из которых каждая является двоичной цифрой:

$$\chi(a,b) = \begin{cases} 0, \text{ если } a = b \\ 1, \text{ если } a \neq b \end{cases}.$$

В булевой алгебре эта функция называется «исключающим или». Для наглядности вот таблица её значений:

$\chi(a,b)$	0	1
0	0	1
1	1	0

Вместо записи $\chi(a,b)$ будем также употреблять запись $a \circ b$. Тогда непосредственной проверкой легко убедиться, что

$$\chi(\chi(a,b),b) = a \circ b \circ b = a\,.$$

Это означает, что если b есть некоторый ключ, то схема шифрования $f(x) = \chi(x,b)$ обратима сама с собой: $f\big(f(x)\big) = x$. С другой стороны, справедливо также соотношение

$$\chi(\chi(\chi(a,b),c),b) = a \circ b \circ c \circ b = a \circ c = \chi(a,c).$$

Это, в свою очередь, означает, что если взять ключ c и построить схему шифрования (также самообратимую!) $g(x) = \chi(x,c)$, то будет иметь место

$$f^{-1}\left(g\big(f(x)\big)\right) = f\left(g\big(f(x)\big)\right) = g(x),$$

что́ нам и требуется!

Итак, секретная передача сообщения без предварительного согласования ключа вполне возможна. Сначала Алиса придумывает свой ключ, являющийся последовательностью битов, и побитово применяет к сообщению и ключу функцию χ (в качестве a выступает очередной бит сообщения, в качестве b — очередной бит ключа). Зашифрованное сообщение передаётся Бобу, который, не зная ключа Алисы, придумывает свою ключевую последовательность битов… и далее всё по уже описанной схеме.

Рассмотрим простой пример. Пусть необходимо передать сообщение 01100010 (десятичное 98). Алиса выбрала ключ 10110011 (десятичное 179). Записывая одно число под другим, она получает зашифрованное сообщение:

01100010
10110011
⇓
11010001

В десятичной системе это число 209. Боб, получив его, выбирает свой ключ 10101010 (десятичное 170) и проделывает такую же операцию:

11010001
10101010
⇓
01111011

В десятичной системе это 123. Алиса снова применяет к нему свой ключ 179:

01111011
10110011
⇓
11001000

Это десятичное 190. Бобу осталось повторно применить к нему свой ключ 170, и он в результате получает исходное сообщение 98:

11001000
10101010
⇓
01100010

В полном согласии с теорией, схема работает!

Нам осталось поговорить только об одной, но очень важной проблеме. А насколько устойчиво к взлому χ-шифрование? Вообще говоря, если длина ключа близка к длине сообщения (а ещё лучше — совпадает с ней, тогда ключ не приходится повторять с начала), и ключ сгенерирован случайным образом (например, на основе оцифровки белого шума), то взлом практически невозможен. Правда, это при условии, что ключ используется лишь один раз, а в описанной схеме Алисе приходится применять его к сообщению два раза, и оба раза пересылать результат применения по открытому каналу… Но если сообщение короткое и не содержит лингвистической информации (для чего опять же могут применяться методы устранения избыточности), то схема весьма и весьма надёжна. В приведённом выше примере из трёх последовательно переданных чисел 209, 123, 190 вообще нельзя извлечь никакой информации в принципе.

Слабые места её кроются в другом. Во-первых, необходима *трёхкратная* передача сообщения туда-сюда (два раза от Алисы к Бобу и один раз от Боба к Алисе), и нет гарантии, что канал связи будет функционировать достаточное для этого время. Во-вторых, передача сообщения требует *совместных* усилий Алисы и Боба, что может быть проблематичным, например, при большой разнице во времени.

СОЗДАНИЕ КЛЮЧА БЕЗ ЕГО ПЕРЕДАЧИ

Второй подход, который мы рассмотрим, позволяет Алисе и Бобу совместными усилиями создать секретный ключ на основе переговоров только по открытому достоверному каналу. В итоге они получат одно и то же число, которое может быть использовано для шифрования, например, по описанной ранее χ-схеме, причём никто другой, подслушивающий их переговоры, это число получить не сможет.

Надо отметить, что очень долгое время считалось, будто создать секретный ключ на основе полностью открытых переговоров невозможно в принципе. Лишь в 1976 году американцы Уитфилд Диффи (Wietfield Diffie) и Мартин Хеллман (Martin Hellman) опубликовали алгоритм, опровергающий это мнение; позднее они постоянно упоминали, что он был разработан под сильным влиянием идей Ральфа Ме́ркля (Ralph Merkle), которого следует по справедливости считать одним из соавторов.

Значительно позднее, в 1997 году, были опубликованы материалы, из которых следовало, что незадолго до Диффи и Хеллмана этот же алгоритм был найден англичанином Малькольмом Уильямсоном (Malcolm Williamson), но тот работал по контракту на государственную спецслужбу (Штаб-квартиру правительственной связи Великобритании), которая и засекретила разработку на долгое время.

Алгоритм Диффи–Хеллмана сам по себе прост, но его изложение требует введения одного предварительного обозначения.

Пусть a и b — это два целых неотрицательных (натуральных) числа. Остаток от их целочисленного деления a/b в математике обозначают $a \bmod b$ — так называемая *модулярная операция*.

Процесс совместной генерации ключа начинается с того, что Алиса и Боб договариваются о двух числах P и Q. На основе этих двух чисел они будут пользоваться функцией

$$f(x) = Q^x \bmod P\,.$$

Числа P и Q, вообще говоря, должны выбираться так, чтобы числа $f(x)$ распределялись возможно более случайным образом. Такому выбору посвящено множество работ, выработано много хороших рекомендаций, так что можно просто воспользоваться одним из готовых рецептов. В качестве Q на практике чаще всего берут одно из первых простых чисел (например, 2 или 5), а в качестве P — большое число порядка 10^{300}.

Далее Алиса выбирает некоторое число a и находит $A = f(a)$. Аналогично, Боб выбирает число b и находит $B = f(b)$. В качестве a и b обычно используются числа порядка 10^{100}.

Алиса и Боб открыто обмениваются числами A и B, не сообщая друг другу a и b. Теперь Алисе осталось найти $B^a \bmod P$, а Бобу $A^b \bmod P$. Чудесным образом получившиеся у них числа совпадут — это следует из длинного, но незамысловатого равенства

$$B^a \bmod P = \left(Q^b \bmod P\right)^a \bmod P = Q^{ab} \bmod P = (Q^a \bmod P)^b \bmod P = A^b \bmod P\,.$$

Это общее число Алиса и Боб могут использовать в качестве ключа.

Функция $f(x)$, использующаяся в схеме Диффи–Хеллмана является чрезвычайно труднообратимой (это так называемая проблема *дискретного логарифмирования*). На данный момент не существует практически пригодных методов (то есть эффективных существенно более, нежели полный перебор), которые позволяли бы найти ключ $Q^{ab} \bmod P$ без знания секретных чисел a и b. Неэффективность же перебора гарантируется порядками используемых чисел.

Для примера рассмотрим алгоритм Диффи–Хеллмана с не очень большими числами. Пусть $P = 1234567$ и $Q = 2$, так что

$$f(x) = 2^x \bmod 1234567.$$

Допустим, что Алиса выбрала $a = 239$, а Боб $b = 311$. Тогда

$$A = f(239) = 305945,$$

$$B = f(311) = 828429.$$

Происходит обмен этими числами, после чего Алиса находит

$$828429^{239} \bmod 1234567 = 623197.$$

Боб, со своей стороны, находит

$$305945^{311} \bmod 1234567 = 623197.$$

Это одно и то же число-ключ. Кстати, даже для таких сравнительно небольших чисел не хватит никакого калькулятора — например, число 305945^{311} содержит 1707 десятичных разрядов! Для работы с подобными величинами, конечно, нужны специальные программы, использующие при расчётах специальные методы из теории чисел.

Алгоритм Диффи–Хеллмана также требует взаимодействия сообщающихся сторон, однако с точки зрения передачи информации гораздо эффективнее предыдущей схемы — хотя бы уже потому, что не требует многократной передачи сообщения туда-сюда.

ИДЕЯ НЕСИММЕТРИЧНОГО КЛЮЧА

Вместе с алгоритмом, рассмотренным в предыдущем параграфе, Диффи, Хеллман и Меркль придумали чрезвычайно интересную концепцию, которую назвали «шифрованием с несимметричным ключом» или «схемой с открытыми ключами». Как можно понять из этих названий, для шифровки и расшифровки сообщений здесь используются *различные* ключи. Тут основным автором концепции являлся уже Ральф Меркль.

Идея этой схемы заключается в следующем. Предположим, что Алиса изготовила множество одинаковых замко́в, ключ к которым есть только у неё. Конструктивная особенность замка такова, что *защёлкнуть* его может кто угодно — для этого ключ не нужен, — однако после этого *открыть* защёлкнутый замок может лишь сама Алиса своим ключом. Эти «алисины замки́» при желании можно взять в любом почтовом отделении, по которым она распространила их — а сделать это можно простой рассылкой. Совершенно аналогично поступает Боб (у него, конечно, другие замки, и открываются они другим ключом).

Теперь, если Алиса хочет отправить секретное сообщение Бобу, она идёт с этим сообщением на почту, берёт там «замо́к Боба», кладёт своё сообщение в шкатулку и защёлкивает на ней взятый замок. Всё. Теперь, кроме Боба, никто не сможет открыть шкатулку и прочесть сообщение. Шкатулка посылается Бобу по почте.

Совершенно аналогично действует Боб, отправляя ответное сообщение Алисе. Он кладёт свой ответ в шкатулку, защёлкивает на ней «алисин замок» и отправляет шкатулку по почте. Открыть шкатулку и прочесть ответ может лишь Алиса.

Если теперь перейти от аналогий к реальному шифрованию, то защёлкивание замка соответствует шифрованию сообщения неким ключом, который Алиса предоставляет всем желающим. Этот ключ называется *открытым*. Снятие же замка соответствует дешифровке сообщения *другим* ключом, который доступен только Алисе. Этот ключ называется *закрытым*.

Главная проблема заключается в том, чтобы гарантировать принадлежность алисиного открытого ключа именно Алисе. Если исходить из нашего предположения о достоверности открытого канала, то этой проблемы просто не возникнет: то, что Алиса разослала от своего имени, дойдёт до получателей в целости и сохранности. В реальности всё далеко не так просто, и приходится принимать специальные меры для гарантии достоверности.

Также очень неплохо было бы, если бы пара ключей работала и «в обратную сторону», то есть сообщение, зашифрованное алисиным закрытым ключом, можно было расшифровать с помощью её открытого ключа. Разумеется, таким образом ничего секретного не передашь (ведь открытый ключ есть у всех желающих), зато читатели сообщения будут иметь стопроцентную гарантию того, что сообщение было написано именно Алисой. Это так называемая *задача цифровой подписи*.

Концепция несимметричного шифрования была опубликована в 1975 году и воспринята криптографическим сообществом с большим энтузиазмом. На уровне концепции всё выглядело просто великолепно, и казалось, что более изящную схему придумать вообще нельзя. Но когда начались поиски математических функций, — а таких функций нужно было *две* — которые обеспечивали бы требуемое «парное» взаимодействие шифрования и дешифрования, исследователи приуныли. Эта задача оказалась гораздо более сложной, нежели выглядела с первого взгляда.

Разрешить её удалось лишь два года спустя другим трём математикам на другом конце Соединённых Штатов Америки.

Этими тремя учёными были Рональд Ривест (Ronald Rivest), Ади Шамир (Adi Shamir) и Леонард Эйдлеман (Leonard Adleman). В 1977 году они опубликовали схему, которую назвали аббревиатурой из первых букв своих фамилий — RSA. В основе их идеи лежали простые числа.

Как известно, *простым* называется натуральное число, не имеющее других делителей, кроме самого́ себя и единицы (такие делители ещё называют *нетривиальными*). Натуральные числа, не являющиеся простыми, называются *составными*.

Ещё Евклид (ок. 365 — ок. 300 до н.э.) доказал, что простых чисел бесконечно много. Чуть позже Эратосфе́н (276–192 до н.э.) построил алгоритм, позволяющий «отсеивать» простые числа из натуральных. Алгоритм этот (он так и называется — «решето Эратосфена») на са́мом деле неэффективен, поскольку фактически сводится к простому перебору…

Древние греки знали также и *основную теорему арифметики*: любое натуральное число $n \in \mathbb{N}$ можно единственным образом представить в виде произведения простых чисел p_i (возможно, возведённых в некоторые степени):

$$n = p_1^{m_1} \cdot p_2^{m_2} \cdot \ldots \cdot p_k^{m_k}.$$

Из теории множеств следует, что должна существовать однозначная функция, которая каждому натуральному n ставит в соответствие n-ое простое число. Поисками такой функции математики занимаются уже почти две с половиной тысячи лет, однако по существу никаких успехов достигнуто не было. Единственным заметным успехом было лишь то, что в XIX и XX веках были разработаны сравнительно неплохие алгоритмы проверки чисел на простоту… но чтобы найти n-ое по порядку простое число, всё равно приходится последовательно просматривать натуральные числа. Короче говоря, сколько-нибудь удовлетворительного решения задача о простых числах так и не получила.

Именно этим воспользовались авторы RSA для генерации ключей. Рассмотрим их алгоритм подробнее.

Для того, чтобы создать свои ключи — открытый и закрытый — Алиса выбирает пару *очень больших* простых чисел P и Q. Этот шаг выполняется лишь единожды, а ключи создаются на долгое время, поэтому времени на генерацию P и Q не жалко. Крайне желательно, чтобы эти два числа не были слишком близки друг к другу.

Далее Алиса находит числа $N = P \cdot Q$ и $\varphi = (P-1)(Q-1)$. После этого числа P и Q можно уже уничтожить, однако раскрывать их *ни в коем случае нельзя*!

Обратим внимание: P и Q являются *простыми*, так что разложение $N = P \cdot Q$ согласно основной теореме арифметики будет *единственным*.

Теперь Алисе нужно найти ещё два числа δ и ε. Из них ε (где $1 < \varepsilon < \varphi$) должно быть таким, чтобы оно не имело общих делителей с φ, кроме единицы — этому условию удовлетворить несложно. Второе число δ должно быть таким, чтобы выполнялось условие

$$\delta\varepsilon \bmod \varphi = 1\,.$$

Иными словами, $\delta\varepsilon = 1 + k\varphi$, где $k \in \mathbb{Z}$. Из теории чисел следует, что такое δ всегда существует, причём для его нахождения можно применить слегка модифицированный алгоритм поиска наибольшего общего делителя (он носит имя уже упоминавшегося Евклида), и эта операция тоже не потребует особо много времени.

На практике в качестве ε обычно берут не очень большое простое число, в двоичной записи которого присутствуют две единицы, например $\varepsilon = 17$ (двоичный эквивалент 10001) или же $\varepsilon = 65537$ (двоичное 10000000000000001). Такой выбор упрощает умножение и возведение в степень.

После этого можно уничтожить и число φ, хотя раскрывать его также нельзя.

Всё. Ключи готовы. Открытым ключом является пара чисел (N, ε), а закрытым — число δ.

Предположим, что Боб хочет отправить Алисе сообщение, выражаемое числом $m < N$. Функция шифрования имеет вид

$$f(m) = m^{\varepsilon} \bmod N\,.$$

Найдя $c = f(m)$, Боб пересылает это число. Алиса, чтобы расшифровать сообщение, должна воспользоваться своим закрытым ключом δ и функцией дешифровки

$$f^{-1}(c) = c^{\delta} \bmod N\,.$$

Алгоритм RSA предусматривает и возможность цифровой подписи. Если Алиса хочет подписать своё сообщение m, то она должна воспользоваться той же функцией f^{-1}, вычислить $s = f^{-1}(m)$ и отправить вместе с сообщением m подпись s. Всякий получатель сообщения, имеющий в своём распоряжении открытый ключ Алисы (N, ε), может удостовериться в авторстве, проверив равенство $m = f(s)$.

Рассмотрим пример. Пусть для своей пары ключей Алиса выбрала простые числа 11 и 23. Тогда

$$N = 11 \cdot 23 = 253,$$

$$\varphi = 10 \cdot 22 = 220.$$

Она может положить $\varepsilon = 3$, так как 220 на 3 не делится. Осталось лишь найти такое δ, чтобы выполнялось $3\delta = 220k + 1$, $k \in \mathbb{Z}$. Для небольших чисел его нетрудно найти и перебором. В нашем случае $\delta = 147$, так как $3 \cdot 147 = 441$, а $441 = 220 \cdot 2 + 1$.

Итак, открытым ключом является пара (253,3), а закрытым — число 147.

Если Боб хочет отправить Алисе сообщение 177, то он вычисляет $177^3 \bmod 253 = 232$ и отправляет ей число 232. Алиса, получив его, вычисляет $232^{147} \bmod 253 = 177$ и получает исходное сообщение.

(Заметим, что на этот раз для примера были взяты совсем маленькие числа, и тем не менее 232^{147} является 348-значным числом!)

Если же Алиса хочет отправить сообщение 177, подписанное своим закрытым ключом, то она вычисляет $177^{147} \bmod 253 = 78$ и отправляет адресатам [177,78]. Адресат проверяет равенство $177 = 78^3 \bmod 253$ и, поскольку оно выполняется, может быть уверен в подлинности сообщения.

Взломать шифр RSA можно лишь одним способом — разложить на простые множители число N из открытого ключа. Тогда можно будет найти и число φ, а затем и закрытый ключ δ. Чтобы отыскать разложение, придётся последовательно искать все простые числа от 2 до $\sqrt{N}$ (уже большие вычислительные затраты) и для каждого очередного проверять, делится ли на него N. Поскольку числа огромны, а деление из всех арифметических операций является наиболее затратной, сложность работы возрастает неимоверно. До тех пор, пока не будут найдены сколько-нибудь эффективные алгоритмы разложения, схема RSA будет оставаться неуязвимой.

Для иллюстрации можно привести следующий факт. В августе 1977 года, когда алгоритм RSA был впервые опубликован, известный популяризатор математики Мартин Гарднер (Martin Gardner) рассказал о нём на страницах журнала “Scientific American” и предложил читателям попытаться расшифровать сообщение, к которому прилагался использованный открытый ключ. В качестве N выступало число порядка 10^{129} — смехотворно малое по меркам нынешней криптографии. За решение задачи взялись несколько специализированных криптографических и множество общематематических лабораторий, располагавших прекрасными специалистами и мощнейшими для своего времени компьютерами. Уже значительно позднее, в начале 1990-х годов, к проекту мог подключиться любой желающий доброволец, имевший интернет-доступ (для чего следовало скачать и установить специальное программное обеспечение, возлагавшее на компьютер малую часть задачи).

И вот такими-то объединёнными колоссальными усилиями задача сравнительно небольшой размерности была решена лишь в конце апреля 1994 года — почти через 17 лет! Что же говорить о нынешних временах, когда реально надёжной длиной ключа N считается 2^{1024} бит, то есть в качестве N рекомендуется брать числа порядка 10^{308} (оценка 2006 года)?!

Следует упомянуть и ещё один факт. По иронии судьбы, идентичный RSA алгоритм также был разработан до Ривеста, Шамира и Эйдлемана, причём, так же как и в случае с алгоритмом Диффи–Хеллмана, это произошло тоже в Великобритании, и создал его тоже сотрудник британской Штаб-квартиры правительственной связи! Этим сотрудником был Клиффорд Кокс (Clifford Cocks), описавший алгоритм во внутренних документах своей организации ещё в 1973 году. Как видно, математика не терпит секретности — тем более, что в истории криптографии были и другие подобные случаи…

ЗАКЛЮЧЕНИЕ

Как уже упоминалось в начале, история криптографии насчитывает много веков, и безнадёжной затеей было бы пытаться рассказать всю эту многовековую историю в одной небольшой популярной статье. Я лишь попытался в доступной форме поведать о нескольких методах, предложенных в XX веке, каждый из которых без преувеличения можно назвать революционным. Самую большую революцию произвела схема шифрования с открытым ключом и вдохновлённый ей алгоритм RSA.

Алгоритм этот активнейшим образом применяется и по сей день — так, на основе RSA построена популярнейшая криптографическая система PGP (Pretty Good Privacy).

Всем, кого заинтересовала тема этой статьи, я рекомендую прочесть книгу Саймона Сингха под названием «Книга шифров: тайная история шифров и их расшифровки». Это великолепное сочетание истории криптографии с элементарным введением в саму́ науку; книга также содержит прекрасную библиографию, которая позволит желающим познакомиться с криптографией ещё ближе.