bhv

Михаил Масленников

Практическая криптография

+ CD-ROM

- Некоторые идеи мэтра криптографии Клода Шеннона
- Основные криптологические процедуры
- Построение качественного генератора гаммы
- Хэш-функция и электронная подпись

МАСТЕР РЕШЕНИЙ

УДК 681.3.06
ББК 32.81
М31

Масленников М. Е.

М31 Практическая криптография. — СПб.: БХВ-Петербург, 2003. — 464 с.: ил.

ISBN 5-94157-201-8

Книга российского криптографа посвящена прикладным проблемам современной криптографии. Наряду с основными теоретическими положениями рассматривается: создание криптографического ядра, встраивание криптографических алгоритмов в Microsoft Outlook и Lotus Notes, создание автоматизированной системы документооборота, технология отпечатков пальцев. Все программное обеспечение, описываемое в книге, создано в Borland C++ Builder. На прилагаемом к книге компакт-диске находятся демонстрационные версии некоторых программ и документация.

Для широкого круга IT-специалистов и специалистов, отвечающих за безопасность систем

УДК 681.3.06
ББК 32.81

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Анатолий Адаменко*
Зав. редакцией	*Анна Кузьмина*
Редактор	*Петр Науменко*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Оформление серии	*Via Design*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.11.02.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 37,41.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-201-8

Содержание

Глава 2

Стойкость шифра

Интуитивно ясно, что стойкость шифра — это его способность противостоять попыткам взлома. Что такое попытка взлома? Это попытка потенциального злоумышленника получить некоторую дополнительную информацию об открытом тексте, зашифрованном с помощью этого шифра. А что такое "информация об открытом тексте"? Как дать этому, в общем-то довольно туманному, понятию строгое математическое определение? Ответ на эти вопросы был получен американским математиком Клодом Шенноном в конце 1940-х годов. К точным математическим формулам мы перейдем чуть позже, а пока, для неискушенного в вопросах математики читателя, я позволю себе привести цитату, посвященную Клоду Шеннону из книги Дэвида Кана.

"Клод Шеннон родился в городе Петоски в штате Мичиган 30 апреля 1916 года. Поступив в Мичиганский университет, Шеннон занялся серьезным изучением электротехники и математики. Именно там у него впервые проявился интерес к теории связи и криптографии.

В Массачусетском технологическом институте Шеннон написал диссертацию, в которой содержалось множество новаторских идей, связанных с разработкой телефонных систем. Получив степень доктора математических наук, Шеннон поступил на службу в лабораторию компании "Белл", которая была заинтересована в реализации этих идей на практике.

"Во время второй мировой войны, — рассказывал Шеннон, — компания "Белл" работала над засекречиванием информации. Я тогда занимался системами связи и был назначен в несколько комиссий, изучавших криптоаналитические методы. Начиная примерно с 1941 г. исследования в области математической теории связи и теории шифров велись мной одновременно. Я трудился в обеих областях сразу, и кое-какие идеи в одной из них возникали у меня, когда я работал в другой. Я не хочу сказать, что одна из этих областей доминирует над другой. Просто они настолько тесно связаны, что их невозможно разделить". Хотя разработка обеих теорий была в основном завершена примерно к 1944 г., Шеннон продолжал уточнять полученные

результаты до 1948—1949 гг., когда они были опубликованы в виде двух отдельных статей в солидном теоретическом журнале "Bell system technical journal".

В обеих статьях Шеннона — "Математическая теория связи" и "Теория связи в секретных системах" — идеи излагаются в краткой, математической форме. Точный и выразительный стиль изложения Шеннона вдохнул в них жизнь. В результате его первая статья породила теорию информации, а вторая — теорию шифров.

Главной в работах Шеннона является концепция избыточной информации. Избыточность, по Шеннону, означает, что в сообщении содержится больше символов, чем в действительности требуется для передачи информации. Избыточность связана с излишком правил, обременяющих все языки. Второй источник языковой избыточности происходит из человеческой лености, которая заставляет людей выбирать легко выговариваемые и узнаваемые звуки. Процесс корректорской правки текста сродни криптоанализу, ибо при вскрытии шифров криптоаналитики также используют свое знание правил фонетики, грамматики, идиом, слов-функций и фонетических склонностей, которые в совокупности и придают языку избыточность.

Тут я позволю себе немного дополнить признанного мэтра криптографии. Появление компьютеров и специфической компьютерной информации (текста, набранного в текстовом редакторе, файла, содержащего рисунок, exe-файла и т. п.) еще больше обострило проблему избыточности. К упоминаемым Шенноном правилам фонетики и грамматики добавились стандартные служебные символы, добавляемые компьютерной программой, пробелы, возврат каретки и перевод строки, различные заголовки, информация о версии, Copyright, информация о пользователе и многое, многое другое. Возьмите текстовой файл, набранный в Microsoft Word, и посмотрите, сколько места в нем занимает сам текст, а сколько — служебные символы. В совокупности избыточность компьютерной информации оказалась во много раз выше избыточности простой разговорной речи.

Продолжим рассказ Дэвида Кана о Шенноне. "С чего начинается криптоанализ? При исправлении ошибки все избыточные элементы, используемые для правки, лежат в готовом виде на поверхности. В криптограмме все наоборот — они незаметны. Криптоаналитик производит подсчет частот букв криптограммы и соотносит полученные результаты с известными частотами букв предполагаемого языка, на котором записан открытый текст. Откуда у криптоаналитика уверенность в том, что частоты букв открытого текста данной криптограммы примерно совпадают с частотами эталонного открытого текста? Разве не может это соответствие нарушиться из-за различий в словарном запасе корреспондентов и в темах их переписки? Нет, не может, ибо избыточные элементы языка превалируют над остальными. Сила ума Шеннона, его огромный вклад в теорию шифровального дела выразились в открытии избыточности как основы криптоанализа. Шеннон первым сумел объяснить постоянство частот встречаемости букв, а тем самым и такое зависящее от него явление, как криптоанализ, дав возможность глубоко понять процесс аналитического вскрытия шифров".

Применение идей Шеннона на практике мы уже видели в прошлой главе, на примере Русской армии 1914 года.

А теперь — немного математики, взятой из учебника "Лекции по теории информации", МГУ, раздел "Количество информации в дискретном сообщении. Энтропия".

"Предположим, что источник сообщений может в каждый момент времени случайным образом принять одно из конечного множества возможных состояний. Такой источник называют дискретным источником сообщений. При этом принято говорить, что различные состояния реализуются вследствие выбора их источника. Каждому состоянию источника U ставится в соответствие условное обозначение в виде знака. Совокупность знаков u_1, u_2, ..., u_i, ..., u_N, соответствующих всем N возможным состояниям источника, называют его алфавитом, а количество состояний N — объемом алфавита. Формирование таким источником сообщений сводится к выбору им некоторого состояния u_i и выдачи соответствующего знака. Таким образом, под элементарным дискретным сообщением будем понимать символ u_i, выдаваемый источником, при этом в течение некоторого времени T источник может выдать дискретное сообщение в виде последовательности элементарных дискретных сообщений, представляющей собой набор символов u_i (например, u_5, u_1, u_3), каждый из которых имеет длительность t_i секунд. В общем случае необязательно одинаковую для различных i. Такая модель источника сообщений соответствует реальной ситуации, имеющей место в телеграфии ($t_i \neq \text{const}$) и передаче данных ($t_i = \text{const}$). Отдельные состояния источника могут выбираться им чаще, другие реже. Поэтому в общем случае он хранится дискретным ансамблем U, т. е. полной совокупностью состояний с вероятностями их появления, составляющими в сумме 1.

$$U = \begin{pmatrix} u_1 & u_2 & \dots & u_i & \dots & u_N \\ P(u_1) & P(u_2) & \dots & P(u_i) & \dots & P(u_N) \end{pmatrix}, \qquad \sum_{i=1}^{N} P(u_i) = 1, \tag{1.1}$$

где $P(u_i)$ — это вероятность выбора состояния u_i источником сообщений. При выдаче источником сообщений в виде последовательности элементарных дискретных сообщений, полным вероятностным описанием является вероятность совместного появления набора различных символов u_i в момент t_1, t_2, ..., t_n, где n — длина последовательности

$$P(u_i^{t_1}, u_j^{t_2}, \dots, u_k^{t_i}, \dots, u_l^{t_n}).$$[1]

[1] В нашем случае эта модель описывает множество всевозможных открытых текстов, которые могут быть затем подвергнуты зашифровыванию.

Располагая такими сведениями об источнике, можно вычислить вероятность любого отрезка сообщения длиной меньше n.[1]

В каждом элементарном сообщении содержится для его получателя определенная информация: совокупность сведений о состоянии дискретного источника сообщения. Определяя количественную меру этой информации, мы совершенно не будем учитывать ее смыслового содержания, также ее значения для конкретного получателя. Очевидно, что при отсутствии сведений о состоянии источника имеется неопределенность относительно того, какое сообщение u_i из числа возможных им выбрано, а при наличии этих сведений данная неопределенность полностью исчезает. Естественно количество информации, содержащейся в дискретном сообщении, измерять величиной исчезнувшей неопределенности. Введем меру этой неопределенности, которую можно рассматривать и как меру количественной информации. Мера должна удовлетворять ряду естественных условий, одним из них является необходимость ее монотонного возрастания с увеличением возможности выбора, т. е. объема алфавита источника N. Кроме того, желательно, чтобы вводимая мера обладала *свойством аддитивности*, заключающемся в следующем: если 2 независимых источника с объемами алфавита N и M рассматривать как один источник, одновременно реализующий пары состояний n_i и m_i, то в соответствии с принципом аддитивности полагают, что неопределенность объединенного источника равна сумме неопределенностей исходных источников. Поскольку объем алфавита объединенного источника равен NM, то искомая функция при равной вероятности состояний источников должна удовлетворять условию $f(NM) = f(N) + f(M)$. Можно математически строго показать, что единственной функцией, при перемножении аргументов которой значение функций складываются, является логарифмическая функция. Поэтому перечисленные требования выполняются, если в качестве меры неопределенности источника с равновероятными состояниями и характеризующего его ансамбля U принять логарифм объема алфавита источника

$$H(u) = \log N.$$

Легко видеть, что:

- с ростом N величина $H(U)$ монотонно возрастает;
- в случае если объем алфавита источника N равен 1, т. е. когда неопределенность отсутствует,

$$H(u) = \log 1 = 0;$$

- величина $H(U)$ обладает свойством аддитивности, поскольку

$$\log(NM) = \log(N) + \log(M).$$

[1] То есть вероятность любого открытого текста длины, не превосходящей n.

Впервые данная мера была предложена Хартли в 1928 г. Основание логарифма не имеет принципиального значения и определяет только масштаб или единицу количества информации. Чаще всего в качестве основания используют число 2, при этом единица количества информации называется двоичной единицей или битом и представляет собой информацию, содержащуюся в одном дискретном сообщении источника равновероятных сообщений с объемом алфавита равным двум. При выборе основания логарифма равным 10 получаем десятичную единицу, называемую дитом. Иногда используют натуральную единицу количества информации, называемую натом, при этом основание логарифма равно $e \approx 2{,}7$. Рассматриваемая мера количества информации может иметь лишь ограниченное применение, поскольку предполагает равную вероятность выбора источником любого из возможных его состояний.[1]

В более общем случае, когда вероятности различных состояний источника не одинаковы, степень неопределенности конкретного состояния зависит не только от объема алфавита источника, но и от вероятности этого состояния. В такой ситуации количество информации, содержащееся в одном дискретном сообщении u_k, целесообразно определить как функцию вероятности появления этого сообщения $P(u_k)$ и характеризовать величиной

$$i(u_k) = -\log P(u_k) = \log \frac{1}{P(u_k)} \qquad (1.2)$$

Теперь количество информации, содержащееся в дискретном сообщении, зависит от степени неожиданности этого сообщения, характеризуемой вероятностью его появления. Количество информации в сообщении тем больше, чем оно более неожиданно. Если источник выдает последовательность зависимых между собой элементарных сообщений, то наличие предшествующих сообщений может изменить вероятность последующего а, следовательно, и количество информации в нем.[2]

Оно должно определяться по условной вероятности $P(u_k/u_{k-1}, u_{k-2}, ...)$ выдачи сообщений u_k при известных предшествующих сообщениях u_{k-1}, u_{k-2}, ..., тогда количество информации

$$i(u_k/u_{k-1}, u_{k-2}, ...) = -\log P(u_k/u_{k-1}, u_{k-2}, ...) \qquad (1.3)$$

Определения (1.2) и (1.3) количества информации являются случайной величиной, поскольку сами сообщения являются случайными. Его распределение вероятностей, определяемое распределением вероятностей сообщений

---

[1] В нашем случае открытые тексты выбираются, как правило, неравновероятно.

[2] Именно так и обстоит дело, к примеру, в русском, да и в любом другом разговорном языке. Например, за буквой 'ы' не может следовать 'ь'. Наличие таких запретов помогает криптоаналитику вскрыть шифр.

в данном ансамбле для цифровой характеристики всего ансамбля или источника сообщения, используется для математического ожидания количества информации в отдельных сообщениях, называемого *энтропией*.

$$H(U) = M\left\{\log\frac{1}{P(u_i)}\right\} = \sum_{i=1}^{N} P(u_i)\cdot\log\left(\frac{1}{P(u_i)}\right) \qquad (1.4)$$

Чем больше энтропия источника, тем больше степень неожиданности выдаваемых им сообщений в среднем, т. е. тем более неопределенным является ожидание сообщений. Впервые мера (1.4) была преложена Клодом Шенноном в его фундаментальной работе "Математические основы теории связи", опубликованной в 1948 г., в которой были заложены основы современной теории информации".

Итак, Шеннон впервые вывел точную формулу для меры неопределенности открытого текста. И с точки зрения криптографии сделал фундаментальный вывод: обосновал абсолютно стойкий шифр, т. е. такой шифр, который никто и никогда не сможет вскрыть. А именно:

Определение 2.1

Шифр является абсолютно стойким, если энтропия открытого текста при условии известного шифртекста равна безусловной энтропии открытого текста.

Попросту говоря, если наличие шифртекста не дает криптоаналитику никакой новой информации об открытом тексте. Конечно же, этому условию не удовлетворяет простая замена: в ней статистика шифртекста — это просто переставленная статистика открытого текста. По шифртексту криптоаналитик в этом случае сразу же определяет места появления наиболее часто встречающихся символов в открытом тексте, т. е. сразу же получает огромную дополнительную информацию об открытом тексте.

Теперь осталось сделать последний шаг — построить пример абсолютно стойкого шифра. И это, имея формулы Шеннона, описывающие энтропию, оказалось возможно.

Определение 2.2

Если шифр получается путем наложения на открытый текст случайной и равновероятной гаммы, то такой шифр является абсолютно стойким.

Из теории вероятностей известно, что при сложении двух случайных величин, если одна из них является случайной и равновероятной, то сумма, независимо от распределения другой величины, также будет случайной и равновероятной. Если при гаммировании произвольный открытый текст (неравновероятная случайная величина) складывается со случайной и равновероятной гаммой, то шифртекст будет случайным и равновероятным.

Подставляя в формулу (1.4) значение $P(u_i) = 1/N$ для всех i, получаем максимальное значение $H(U) = \log(N)$.

Помните слова Дэвида Кана из прошлой главы о том, что "одноразовые шифрблокноты обеспечивают надежную защиту для сообщений российских разведчиков, военных, дипломатов и работников тайной политической полиции"? Это и есть пример абсолютно стойких шифров. В них пишется случайная и равновероятная гамма, и вскрыть их, не имея второго экземпляра блокнота, теоретически невозможно.

Однако в большинстве практических случаев одноразовые шифрблокноты, т. е. таблички со случайной гаммой, сразу после наложения которой они уничтожаются, неудобны. Запас гаммы ограничен, для предотвращения компрометации нужно соблюдать очень строгие правила при хранении неиспользованных блокнотов. При защите компьютерной информации применение шифрблокнотов возможно только в уж очень экзотических ситуациях, например, при смене секретных ключей для рассылки новых ключей по электронной почте. В повседневной, реальной жизни используются программные генераторы гаммы. А их уже необходимо анализировать и не только на предмет того, насколько случайную и равновероятную гамму они вырабатывают, но и с точки зрения возможности определения начальных параметров генератора по какому-то отрезку гаммы или, крайний случай, по всей гамме. Такие начальные параметры и являются секретным ключом системы шифрования.

Насколько правомерна такая задача? В каких случаях потенциальному злоумышленнику может быть известен отрезок гаммы? Давайте не забывать, что в открытом тексте могут быть достаточно длинные стандарты. Криптоаналитиков часто выручают выражения вроде: "В ответ на Вашу телеграмму от", "На Ваш исходящий № от ", "Уважаемый господин ", "С уважением" и т. п. Они позволяют определить некоторые отрезки гаммы, и если по этой информации оказывается возможным вычислить начальные параметры генератора гаммы, то сразу же вскрывается и весь остальной шифрованный текст. Например, если в качестве генератора гаммы используется линейная рекуррентная последовательность вида

$$x_{i+n} = x_i + x_{i+1} + \ldots + x_{i+n-1},$$

а секретным ключом являются первые n значений, то если хотя бы в одном месте криптоаналитику будет известно n знаков гаммы, решив систему линейных уравнений он без труда вычислит секретный ключ и весь открытый текст.

Более подробно о требованиях, предъявляемых к генераторам гаммы, мы поговорим в следующей главе, а сейчас, раз мы говорим о стойкости шифра и уже определили понятие абсолютной стойкости, подумаем о том, что следует понимать под гарантированной стойкостью.

Любой программный генератор гаммы в принципе может быть вскрыт. Число начальных параметров ограничено, поэтому всегда можно предполагать, что потенциальный злоумышленник попробует их все перебрать. Но представим себе, что он попробует сделать это, как ранее в приведенном примере, когда каждый элемент ключа — байт, а $n = 17$. Количество всевозможных вариантов перебора составит

$$(2^8)^{17} = 2^{136} = 10^{40}.$$

(Здесь и дальше мы будем пользоваться распространенным у криптографов округлением $2^{10} \approx 10^3$.)

Если ключ (17 начальных значений) выбирался случайно и равновероятно, то по теории вероятности в среднем половину из этого (10^{40}) числа вариантов надо будет опробовать, пока не попадется истинный вариант. Говорить о половине при таких значениях несерьезно, это 5×10^{39}, т. е. почти те же порядки, при округлении мы огрубили эту величину гораздо больше. Как можно наглядно представить себе эту величину?

Производительность современного компьютера примем за 10^{10} простейших операций в секунду. Тогда за 1 час = 3600 сек компьютер выполнит $3{,}6 \times 10^{13}$ простейших операций, за сутки — $24 \times 3{,}6 \times 10^{13} \approx 10^{15}$, за год — $3{,}6 \times 10^{17}$, округляем до 10^{18}. Для перебора 10^{40} значений такому компьютеру потребуется 10^{22} лет. Ясно, что это нереальная задача. Предположим, что для перебора используется не один, а несколько (например, 10^6 — миллион) таких компьютеров. Им всем, работая в параллельном режиме, для перебора потребуется 10^{16} лет. Даже с учетом постоянно растущей производительности компьютеров, ясно, что задача перебора всевозможных значений имеет разумные пределы. Отодвинем ее до неразумных (с учетом постоянно возникающих сообщений о возможности появления сверхпроводниковых, квантовых, биологических и прочих новейших технологий) и примем величину в 10^{100} как гарантированную оценку стойкости.

Сложнее с другой стороной дела. В приведенном выше примере мы видели, что, несмотря на высокую оценку тотального перебора, шифр элементарно вскрывался за счет плохо выбранного генератора гаммы. А искусство выбора и всестороннего изучения генератора гаммы и есть искусство математика-криптографа. Об этом — в следующей главе.

Глава 3

Генератор гаммы

Поскольку качественный генератор гаммы является неотъемлемой частью шифрсистемы, то давайте поподробнее рассмотрим, каким требованиям он должен удовлетворять.

Шеннон писал: "С криптографической точки зрения секретная система почти тождественна системе связи при наличии шума". В теории связи термин "шум" имеет особое значение. Под шумом подразумевается любая помеха, создающая ошибки при передаче по каналу связи. Шеннон исходит из того, что шум схож с наложением гаммы. "Основное различие между ними заключается, во-первых, в том, что преобразование при помощи шифра имеет обычно более сложный характер, чем возникающее за счет шума в канале; во-вторых, в том, что ключ в секретной системе выбирается из конечного множества, тогда как шум обычно вносится в канал постоянно и выбирается из бесконечного множества".

Понятие "конечное — бесконечное множество" весьма относительно. Конечное множество, состоящее из 10^{100} элементов, с точки зрения практических задач организации его перебора, вполне можно считать бесконечным. И задача построения качественного генератора гаммы сводится как раз к тому, чтобы накладываемая на открытый текст гамма по своим характеристикам напоминала бы "белый шум", т. е. шум, который полностью заглушает все осмысленные параметры открытого текста. А принципиальное отличие от традиционного шума в том, что генератор гаммы должен уметь повторять с точностью до единого знака этот шум, с тем чтобы была возможность расшифровать зашифрованный ранее текст. А это, как правило, невозможно осуществить без использования такого традиционного математического аппарата, как рекуррентные последовательности, т. е. такие, в которых каждый последующий знак вычисляется как некоторая постоянная функция от n предыдущих. В примере из прошлой главы в роли такой функции выступала сумма n предыдущих знаков. На этом примере мы смогли убедиться, что не любая рекуррентная последовательность пригодна для

использования в качестве генератора гаммы. Давайте рассмотрим этот и некоторые другие примеры несколько подробнее.

Поскольку в этой книге речь, в основном, идет о защите компьютерной информации, то в качестве алфавита открытого и шифрованного текста у нас часто (но не всегда) будет выступать вся таблица ASCII-символов, т. е. множество всевозможных целых неотрицательных чисел от 0 до 255 с определенными на нем операциями сложения и вычитания по модулю 256. Такую конструкцию в математике еще называют кольцом вычетов по модулю 256 и обозначают как $Z/256$.[1]

$$x_{i+n} = x_i + x_{i+1} + \ldots + x_{i+n-1}$$

Предположим, что в качестве начальных значений этого рекуррентного соотношения выбраны все нули. Тогда совершенно очевидно, что вся вырабатываемая гамма также будет состоять из одних нулей, т. е. получается, что у такого генератора гаммы есть потенциально опасный ключ. Такую точку в теории рекуррентных последовательностей называют изолированной, а соответствующий ключ можно назвать критическим.

Предположим теперь, что все начальные значения — нечетные числа и n — нечетно. Тогда, очевидно, все значения гаммы наложения также будут нечетными. После наложения такой гаммы на открытый текст все четные знаки открытого текста станут нечетными, а нечетные — наоборот, четными. По четности знаков шифртекста мы сразу же определяем четность знаков открытого текста, и выработанная гамма заведомо не удовлетворяет условиям Шеннона. Внимательному читателю предлагаю самому представить, что будет в том случае, когда все начальные значения рекуррентного соотношения — четные (независимо от четности n).

Аналогично, в случае, если все начальные значения гаммы кратны 3, 5, 7 либо любому другому простому числу, то все знаки вырабатываемой гаммы также будут кратны этому числу.

Таким образом, среди всевозможных ключей в таком генераторе заведомо есть криптографически плохие, т. е. вырабатываемая ими гамма не удовлетворяет условиям Шеннона.

Предположим теперь, что $n = 5$ и начальные значения x_0, x_1, x_2, x_3, x_4 равны:

$x_0 = 32$;

$x_1 = 32 + 128$;

---

[1] Замечание "не всегда" связано с тем, что иногда приходится сталкиваться с проблемой отображения информации на экран, например, при работе с текстовым редактором. Там среди всевозможных ASCII-символов приходится отбирать только те, которые можно отобразить на экране и которые не вызовут каких-то конфликтов при работе программного обеспечения. Я предпочитаю в качестве таких символов использовать латинские буквы и цифры.

$x_2 = 32;$

$x_3 = 32 + 128;$

$x_4 = 32.$

Тогда

$x_5 = 32 + 128;$

$x_6 = 32;$

$x_7 = 32 + 128;$

$x_8 = 32;$

и т. д.

Здесь мы столкнулись с таким криптографически важным понятием, как повторяемость, и частным случаем повторяемости — периодичностью знаков гаммы. Этот пример сродни тому случаю, когда программист, строя собственную систему защиты, использует периодически повторяющийся пароль для наложения его на открытый текст, а это, как правило, первое, что приходит в голову человеку, неискушенному в вопросах криптографии. О повторяемости гаммы мы поговорим позже, сейчас же отметим, что при выборе генератора гаммы один из первых вопросов, который должен задать себе разработчик, — как избежать описанных ранее и подобных им казусов и обосновать какие-то свойства генератора гаммы? Такое обоснование — это весьма трудоемкий и сложный процесс, требующий знаний из алгебры, теории полей, теории подстановок, теории линейных рекуррентных последовательностей, а подчас просто интуиции и опыта криптографа. Описать все возможности даже в первом приближении в рамках этой книги не представляется возможным, да я и не ставил себе такой задачи. Но некоторые вещи можно обсудить.

1. Если посмотреть на пример, то легко увидеть, что одной из его особенностей является неправильный выбор ключевых параметров. В качестве секретного ключа мы выбрали начальное заполнение, от которого затем раскручиваются все остальные знаки. В процессе выработки очередного знака в нем принимают участие только n предыдущих. Поэтому если, в силу наличия стандарта в тексте, мы сумеем определить n подряд идущих знаков гаммы, то все следующие за ними будут вычисляться автоматически. В этом примере ключевые параметры хотя и принимают участие в выработке каждого знака гаммы, но это участие зависит от предыдущих знаков выработанной гаммы, что и привело в конечном счете к возможности вычисления последующих знаков гаммы по предыдущим. Поэтому ключевые параметры должны принимать участие в выработке каждого знака гаммы *независимо* от выработанных предыдущих знаков.
2. Криптоаналитику значительно облегчило работу то обстоятельство, что выбранная рекуррентная последовательность оказалась *линейной*. Легко

составляется и решается система уравнений, уравнения дают хорошие (с точки зрения криптоанализа) результаты при сложении или вычитании различных знаков гаммы, например, такие:

$$x_{i+n} - x_{i+n-1} = x_i + x_{i+1} + \ldots + x_{i+n-2}$$

$$x_{i+n+1} - x_{i+n} = x_{i+1} + x_{i+2} + \ldots + x_{i+n-1} =$$

$$= x_{i+n} - x_{i+n-1} - x_i + x_{i+n-1} = x_{i+n} - x_i.$$

Линейность надо убрать. Но представим, например, что, убрав линейность, мы выбрали функцию

$$x_{i+n} = x_i \times x_{i+1} \times x_{i+n-1}.$$

Это означает, что любое начальное заполнение, в котором есть хотя бы один ноль, будет давать полностью нулевую гамму. И не только это. Если, например, $n = 8$, то начальное заполнение, в котором все значения равны 2, будет давать тоже полностью нулевую гамму, так как мы выполняем все операции по модулю 256.

В алгебре есть понятие *подстановки*. В случае модуля 256 это просто таблица со всеми значениями от 0 до 255, но как-то перемешанными. Всего можно построить 256! всевозможных подстановок по модулю 256. Множество всевозможных таких подстановок принято называть *симметрической группой* S_{256}. Давайте посмотрим, как изменится линейное уравнение, если к нему добавить некоторую подстановку π

$$x_{i+n} = \pi(x_i + x_{i+1} + \ldots + x_{i+n-1}).$$

Нетрудно видеть, что вычитание различных знаков гаммы при этом уже не будет давать такого простого результата, как в случае, когда функция была линейной. Несомненно, что использование подстановки значительно усложнит задачу криптоаналитику. В то же время скорость реализации такого преобразования практически не уменьшится, поскольку замена значения по подстановке — это одна операция обращения к памяти.

3. А что если в приведенном выше примере сделать ключевым параметром подстановку? На предварительном этапе, один раз перед сеансом выработки гаммы, мы вычисляем случайную подстановку в зависимости от секретного ключа. На скорости реализации гаммы это практически не скажется. Зато задача анализа такого генератора, при неизвестной подстановке, намного усложнится.

4. Этот генератор работает достаточно быстро. А что если использовать не все знаки подряд, а через несколько? Это еще больше усложнит работу криптоаналитика.

Это — основные идеи. Конкретная реализация зависит от фантазии разработчика, его вдумчивости, опыта и целей. Однако могу заверить, что таким

путем можно построить криптографически качественный и высокоскоростной генератор гаммы, значительно превосходящий по скорости выработки гаммы алгоритмы типа DES и ГОСТ.

Теперь — о другой страшной опасности: перекрытиях гаммы.

Что такое наложение гаммы на открытый текст? Это, обычно, процесс познакового сложения открытого текста и гаммы. Например, если открытый текст обозначить как a_1, a_2, ..., a_n, гамму — x_1, x_2, ..., x_n, то шифртекстом будет последовательность $s_1 = a_1 + x_1$, $s_2 = a_2 + x_2$, ..., $s_n = a_n + x_n$. Предположим, что с некоторого места k гамма начала повторяться, т. е. $x_1 = x_k$, $x_2 = x_{k+1}$. Отсюда следует, что

$$s_1 - s_k = a_1 - a_k,$$

$$s_2 - s_{k+1} = a_2 - a_{k+1},$$

...

Таким образом, разность знаков шифрованного текста равна разности знаков открытого текста. Появляется возможность *бесключевого чтения*: предполагая некоторое вероятное слово или фразу в открытом тексте, определяем открытый текст через k знаков и по его читаемости принимаем решение, верно наше предположение или нет. Критериев читаемости или нечитаемости текста может быть множество, например, если речь идет о русском языке и в вычисленном тексте встретилась запретная для русского языка пара ЫЬ, то такой текст признается ложным.

Теперь нетрудно заметить, что точно такой же эффект будет и в том случае, если, использовав один раз генератор гаммы и осуществив с его помощью зашифровывание открытого текста, мы попытаемся точно такой же гаммой зашифровать другой открытый текст. Начинают срабатывать те же методы бесключевого чтения. Сам генератор гаммы может быть сколь угодно хорошим, вырабатываемая им гамма удовлетворять всем условиям Шеннона, но для определения открытого текста по шифрованному нам не потребуется знание гаммы. Предполагая наличие некоторого стандарта в одном тексте, проверяем читаемость вычисленного другого открытого текста и принимаем решение об истинности или ложности сделанного предположения.

Отсюда следует один из наиболее фундаментальных принципов построения качественного генератора гаммы.

Определение 3.1

Всякий раз гамма, вырабатываемая генератором гаммы, должна быть новой, отличной от всех предыдущих гамм, выработанных этим генератором.

Что это означает на практике? Всякий раз, перед началом выработки гаммы наложения, необходимо в параметры, влияющие на выработку гаммы, добавить некоторый дополнительный параметр, называемый в разной литературе

по-разному: разовым ключом, маркантом, ключом на телеграмму и т. п. Мне больше нравится слово *маркант*, поскольку этот параметр, как таковой, чаще всего *ключом не является*. Вы должны сообщить его тому, кто будет расшифровывать текст, чтобы он смог повторить, имея секретный ключ, точно такую же гамму. Маркант иногда просто дописывается *в открытом виде* в шифрованное сообщение и пересылается вместе с шифртекстом адресату. А алгоритм выработки гаммы надо строить с тем учетом, что маркант известен потенциальному злоумышленнику. Маркант чаще всего строится по текущему моменту времени, в который осуществляется зашифровывание, хотя возможны и другие варианты, например, когда сам пользователь, нажимая на клавиши, вырабатывает случайный маркант. Единственное требование: маркант практически никогда не должен повторяться. Повторение марканта равносильно повторению гаммы и бесключевому чтению.

Вот такие основные правила должен соблюдать разработчик, создавая собственный генератор гаммы. Но и это еще не все.

Предположим, что мы зашифровываем банковскую информацию. Например, сумму в 1 000 000 рублей в банковском платежном поручении. И в отличие от предыдущих рассуждений, будем предполагать, что открытый и шифрованный тексты состоят только из цифр, а сложение осуществляется по модулю 10.

Пусть, например, для зашифровывания этой суммы наш генератор гаммы выработал последовательность 5017329. Тогда шифртекст — сумма открытого текста и гаммы — будет 6017329 и этот шифртекст был послан в канал связи, причем такой, к которому потенциальный злоумышленник имеет доступ. Ключа к шифру злоумышленник не знает и при правильно построенном генераторе гаммы вычислить не может. Но это ему в некоторых случаях и не нужно. Предположим, что злоумышленник имеет возможность подменять шифртекст в канале связи (например, при пересылке платежного поручения по телеграфу, почти как по Пушкину: "и в суму его пустую суют грамоту другую"). Воспользовавшись этой возможностью, злоумышленник подменит шифртекст 6017329 на 7017329.

Что произойдет на приемном конце? Получив платежное поручение с зашифрованной суммой — 7017329 — операционистка, обладая ключом к выработке гаммы, выработает ее: 5017329, и произведя расшифрование, т. е. вычитание гаммы из шифртекста, получит сумму 2 000 000, которую и зачислит на счет получателя. Нетрудно предположить, что получатель и будет тем злоумышленником, который, подменив шифртекст, не зная ключа к шифру, тем не менее сумел получить таким образом лишний миллион рублей.

Следовательно, при использовании шифра гаммирования подмена некоторых знаков в шифртексте может быть не обнаружена при расшифровании. В криптографии свойство шифра обнаруживать подмену некоторых знаков

в шифрованном тексте называют *имитостойкостью*. И разработчик должен помнить, что шифр гаммирования не обладает имитостойкостью.

Начало сентября 1992 года. Старое здание ЦБ на Неглинке. Комната-пенал, два стола вдоль стены, никаких излишеств.

— Нам нужна Ваша помощь.

Это говорит, вставая из-за стола у окна, высокий стройный человек с седой и густой шевелюрой. Спокойный, уравновешенный, без начальственных привычек. Наверное, раньше был инженером, технарем. А в ЦБ я, действующий офицер ФАПСИ, 35-летний подполковник, без ведома большого начальства. В Конторе какие-то очень уж непонятные времена начались. Сразу после августовских событий 91-го года, как только Ельцин победил, буквально на следующий день нашего начальника Главка завалили рапортами об увольнении. В моем отделении едва ли не треть ребят подали рапорта. И все — молодые, не закостенелые, кому до смерти надоели эти постоянные разговоры ни о чем в курилках да игры на компьютере. Те, кто хочет еще как-то двигаться, а не просто ждать пенсии. Визировал их, как зам. начальника отделения, без сожаления. Сам тоже твердо решил: протяну как-нибудь до 1994-го года, там будет 20 лет выслуги, пенсия, и — сразу на свободу. Начальники тоже как-то притихли. В январе 1992 года, на ежегодном отчете нашего отделения, прямо сказали: "Учитесь зарабатывать деньги, развивайте коммерцию". Позаключали разных договоров. И вот в начале 1992 года — новая метла, новый самый большой начальник. Прозвали его "папой". В мае — очередная крутая смена курса. "Всякую коммерческую деятельность запретить, все договоры разорвать!" Ну, разорвать так разорвать, офицеру что прикажут — то и сделает. Но на душе как-то муторно. А когда будет очередной прогиб Генеральной линии? Через полгода, год, два? И в какую сторону?

− Надо защитить народные деньги.

Ну, это по моей части! Система электронной подписи к тому времени уже готова, и даже более того, готова комплексная программа "Криптоцентр", в которой наворочено все, что душе угодно: подпись, шифрование на индивидуальном секретном ключе, шифрование с использованием открытых ключей, журнал учета, помощь. С собой экзотический в те времена Notebook. Ставлю его на стол, показываю "Криптоцентр". Все в диковинку, смотрят внимательно, но что-то не особенно заинтересованно.

— Понимаете, это хорошо, но нам сейчас нужно не то. У нас около 2000 РКЦ, компьютеров почти нигде нет. Связь, в основном, по почте и телеграфу. А в Сибири есть и такие, до которых 3 дня на лодке надо плыть.

"Широка страна моя родная", что и говорить. Да, наверное они правы: "Криптоцентр" тут не всем подойдет.

— Нам бы как-нибудь ваш калькулятор приспособить и полгодика продержаться. А там что-нибудь придумаем.