Использование однократного гаммирования

С точки зрения теории криптоанализа метод шифрования однократной случайной равновероятной гаммой той же длины, что и открытый текст, является невскрываемым (далее для краткости авторы будут употреблять термин «однократное гаммирование», держа в уме все вышесказанное). Обоснование, которое привел Шеннон, основываясь на введенном им же понятии информации, не дает

возможности усомниться в этом — из-за равных априорных вероятностей криптоаналитик не может сказать о дешифровке, верна она или нет. Кроме того, даже раскрыв часть сообщения, дешифровщик не сможет хоть сколько-нибудь поправить положение — информация о вскрытом участке гаммы не дает информации об остальных ее частях.

Логично было бы предположить, что для организации канала конфиденциальной связи в открытых сетях следовало бы воспользоваться именно схемой шифрования однократного гаммирования. Ее преимущества вроде бы очевидны. Есть, правда, один весомый недостаток, который сразу бросается в глаза, — это необходимость иметь огромные объемы данных, которые можно было бы использовать в качестве гаммы. Для этих целей обычно пользуются датчиками настоящих случайных чисел (в западной литературе аналогичный термин носит название True Random Number Generator или TRNG). Это уже аппаратные устройства, которые по запросу выдают набор случайных чисел, генерируя их с помощью очень большого количества физических параметров окружающей среды. Статистические характеристики таких наборов весьма близки к характеристикам «белого шума», что означает равновероятное появление каждого следующего числа в наборе. А это, в свою очередь, означает для нас действительно равновероятную гамму[38].

К сожалению, для того чтобы организовать конфиденциальный канал передачи данных, потребуется записать довольно большое количество этих данных и обменяться ими по секретному каналу[39]. Уже одно это условие делает однократное гаммирование во многих случаях неприемлемым. В самом деле, зачем передавать что-то по открытому незащищенному каналу, когда есть возможность передать все это по секретному защищенному? И хотя на простой вопрос, является ли метод использования однократной случайной равновероятной гаммы стойким к взлому, существует положительный ответ, его использование может оказаться попросту невозможным.

Да и к тому же метод однократного гаммирования криптостоек только в определенных, можно даже сказать, тепличных условиях. Что же касается общего случая, то все не так просто.

Показать слабости шифра однократного гаммирования можно, говоря наукообразно, с помощью примера или, что называется, «на пальцах». Представим следующую ситуацию.

Допустим, в тайной деловой переписке используется метод однократного наложения гаммы на открытый текст. Напомним, что «наложение» гаммы не что иное, как сложение ее элементов с элементами открытого текста по некоторому

---

[38] Многие известные фирмы занимаются производством таких устройств. В их число входит даже IBM, которая имеет свою собственную лабораторию, исследующую статистические свойства подобных генераторов случайных чисел. В конце 2000 года на рынок компьютерных комплектующих поступили экспериментальные серии материнских плат от IBM, которые дополнительно содержали запатентованный IBM генератор случайных чисел. Его свойства хорошо задокументированы, а с выборками больших объемов были проведены масштабные статистические тесты.

[39] Например записав данные на DVD-диск и послав его специальным курьером с охраной. Хотя в этом случае нет гарантии того, что этого курьера не перехватят и не подкупят.

фиксированному модулю. Значение модуля представляет собой известную часть алгоритма шифрования.

Чтобы дальнейшие рассуждения были как можно более понятны, рассмотрим следующее свойство шифротекста. Предположим, что мы знаем часть гаммы, которая была использована для зашифрования текста «Приветствую, мой ненаглядный сосед!» в формате ASCII, кодировка WIN-1251 (см. листинг 7.1).

Листинг 7.1

```
П р и в е т с т в у ю ,  _ м о й
CF F0 E8 E2 E5 F2 F1 F2 E2 F3 FE 2C 20 EC EE E9

_ н е н а г л я д н ы й _ с о с
20 ED E5 ED E0 E3 EB FF E4 ED FB E9 20 F1 EE F1

е д !
E5 E4 21
```

Верхние строки в этих трех парах строк соответствуют символам открытого текста, а нижние строки представляют собой соответствующие шестнадцатеричные их обозначения.

Для зашифрования была использована гамма из листинга 7.2 (значения чисел даны тоже в шестнадцатеричном формате).

Листинг 7.2

```
BD 8E 1E 72 9C 26 43 AD E7 8B 89 7C 91 06 DE E2
2B DC F9 C9 E1 6D BB 91 43 68 F3 6D 85 2F 0A 74
0F AA 7A 8D A1 8C F8 4A 5D 52 7B BB 7C 01 25 93
```

Пусть криптоаналитику стали доступны ее первые десять байтов и он также знает, что было использовано наложение гаммы как сложение по модулю два[40]. В языке Си существует общеизвестный оператор «^», который ее выполняет. Алгоритм шифрования на языке Си описывается короткой и простой функцией (см. листинг 7.3).

Листинг 7.3

```
char text[1024] = "Приветствую, мой ненаглядный сосед!";
char gamma[1024] = {
      0xBD, 0x8E, 0x1E, 0x72, 0x9C, 0x26, 0x43, 0xAD, 0xE7, 0x8B, 0x89,
      0x7C, 0x91, 0x06, 0xDE, 0xE2, 0x2B, 0xDC, 0xF9, 0xC9, 0xE1, 0x6D,
      0xBB, 0x91, 0x43, 0x68, 0xF3, 0x6D, 0x85, 0x2F, 0x0A, 0x74, 0x0F,
      0xAA, 0x7A, 0x8D, 0xA1, 0x8C, 0xF8, 0x4A, 0x5D, 0x52, 0x7B, 0xBB,
      0x7C, 0x01, 0x25, 0x93
};

for (int i = 0; i < strlen(text); i++)
      text[i] ^= gamma[i];
```

[40] Сложение по модулю 2 представляет собой сложение в поле характеристики два, в простонародье именуемое операцией XOR или «Исключающее ИЛИ».

А ее результат выполнения, то есть зашифрованный текст, выглядит, как показано в листинге 7.4.

Листинг 7.4

```
72 7E F6 90 79 D4 B2 5F 05 78 77 50 B1 EA 30 0B
0B 31 1C 24 01 8E 50 6E A7 85 08 84 A5 DE E4 85
EA 4A 5B
```

Поскольку криптоаналитику известны первые десять байтов гаммы, он может дешифровать первые десять байтов шифротекста с помощью все той же программы из листинга 7.3. Но кроме этого, он может использовать известные ему байты гаммы, чтобы подделать начало сообщения, заменив десять байтов оригинального шифротекста на свои собственные. Для этого он может выбрать поддельное сообщение, например «До свидания». Затем зашифровать его гаммированием и поместить в начало шифротекста вместо зашифрованной части исходного сообщения — слова «Приветствую». В этом случае гамма будет выглядеть так, как в листинге 7.5 (жирным выделен замещенный участок).

Листинг 7.5

```
79 60 3E 83 7E CE A7 4D 0A 63 76 50 B1 EA 30 0B
0B 31 1C 24 01 8E 50 6E A7 85 08 84 A5 DE E4 85
EA 4A 5B
```

Получатель, расшифровав сообщение, увидит фразу «До свидания, мой ненаглядный сосед!», а не оригинальную «Приветствую, мой ненаглядный сосед!». При этом у него не будет никакой возможности проверить, действительно ли это написал отправитель или кто-то другой. Так что, скорее всего, он поверит присланному сообщению.

Вернемся теперь немного назад и попробуем проанализировать зашифрованный однократной гаммой документ, содержащий что-либо более существенное. К примеру, коммерческую тайну. Поставим себя на место криптоаналитика, задачей которого стало вскрытие перехваченного зашифрованного документа.

Этот документ вполне может быть контрактом на оказание услуг в сфере защиты информации конкурирующей фирмы «Рога и Копыта». Как и большинство деловых бумаг, подобный контракт имеет фиксированный бумажный формуляр (примитивный вариант изображен в листинге 7.6), в который попросту вписываются нужные значения полей — фамилии, суммы и ставятся подписи. При переводе в электронный вид этот документ принимает вполне определенную фиксированную электронную форму в формате какого-нибудь текстового процессора.

Листинг 7.6

```
Д О Г О В О Р N____

Я, _____, именуемый в дальнейшем...
...заключаю договор на сумму _____ ...
Подпись ЗАКАЗЧИКА                Подпись ИСПОЛНИТЕЛЯ
```

Криптоаналитик наверняка легко может получить копию этого электронного документа. Он может сделать это, например, под предлогом заключения договора о тестировании компьютеров на предмет возможной утечки информации. Исследуя эту копию, он узнает тип и электронный формат документа, который у него уже есть в зашифрованном виде. Таким образом, он становится обладателем так называемой *вторичной информации* о шифре.

С высокой долей вероятности может оказаться, что к нему в руки попал документ, например, в формате Microsoft Word. Этот формат содержит очень много избыточной информации, которая сама по себе может быть полезна для криптоанализа (подробнее об этом см. раздел «Защита текстовых документов Microsoft Word»).

Самое главное в любом формате — перечень значений различных полей. Многие поля формата текстовых файлов Microsoft Word (MSWord) просто заполняются фиксированными значениями. Тогда, по сути, криптоаналитик сразу же получает довольно много пар «открытый текст — зашифрованный текст», поскольку многие участки того файла, который находится в его руках, либо вовсе фиксированы, либо легко угадываются.

Это наверняка дало бы криптоаналитику множество преимуществ, если бы документ был зашифрован каким-нибудь блочным шифром, но, к сожалению, совершенно бесполезно в случае однократного использования равновероятной гаммы.

Анализируя шифротекст, мы можем сложить известные нам значения полей формата MSWord с зашифрованным текстом и получить участки гаммы. Но, к сожалению, эта информация ничем не поможет нам в поиске возможных значений для остальных участков гаммы, поскольку они вырабатывались независимо друг от друга и знание одной части не дает ни капли информации о других частях.

С другой стороны, получив текстовый формуляр в электронном виде и сравнивая его с зашифрованным текстом, мы можем с некоторой уверенностью указать на местоположение тех или иных элементов текста в зашифрованном документе. То есть, имея стандартный формуляр, мы можем опираться не только на легко угадываемые значения полей формата MSWord, но и уже на текстовые сообщения, которые хранятся в файле в этом формате.

Посмотрим на добытый криптоаналитиком формуляр типового документа. Для этого можно использовать HEX-редактор или даже быстренько набросать программку вывода содержимого файла с помощью функции print_hex() (см. главу «Отступление для программистов»).

По смещению 500h (в десятичном эквиваленте 500h = 1280) байтов в содержимом файла можно найти отлично видимый невооруженным взглядом текст уже знакомого документа, приведенный выше[41] (см. листинг 7.7).

[41] В последних версиях формата документов Microsoft Word текст записывается в формате Unicode, то есть по два байта на символ; к счастью, и эта трудность достаточно просто преодолевается использованием соответствующего средства для просмотра и редактирования документов в формате Unicode, например, встроенного в командную оболочку менеджера файлов Far.

Листинг 7.7

```
00000500:  C4 20 CE 20 C3 20 CE 20 ¦ C2 20 CE 20 D0 20 20 20   Д О Г О В О Р  .
00000510:  4E 5F 5F 5F 5F 0D 0D DF ¦ 2C 20 5F 5F 5F 5F 5F 5F   N____  Я, ______
00000520:  5F 5F 5F 5F 5F 5F 5F 5F ¦ 5F 5F 5F 5F 5F 2C 20 E8   _____________, и
00000530:  EC E5 ED F3 E5 EC FB E9 ¦ 20 E2 20 E4 E0 EB FC ED   меуемый в дальн
00000540:  E5 E9 F8 E5 EC 85 0D 85 ¦ E7 E0 EA EB FE F7 E0 FE   ейшем:  заключаю
00000550:  20 E4 EE E3 EE E2 EE F0 ¦ 20 ED E0 20 F1 F3 EC EC    договор на сумм
00000560:  F3 20 5F 5F 5F 5F 5F 5F ¦ 5F 0D 0D 0D CF EE E4 EF   у _______ Подп .
00000570:  E8 F1 FC 20 C7 C0 CA C0 ¦ C7 D7 C8 CA C0 09 09 CF   ись ЗАКАЗЧИКА  П
00000580:  EE E4 EF E8 F1 FC 20 C8 ¦ D1 CF CE CB CD C8 D2 C5   одпись ИСПОЛНИТЕ
00000590:  CB DF 0D 0D 21 00 91 C4 ¦ 02 A1 01 00 9C C4 02 9D   ЛЯ..............
```

В таком случае данные зашифрованного файла по этому же смещению могут выглядеть так же, как и в листинге 7.8 (авторы берут на себя смелость немного свободно обращаться с шифротекстом, однако, как будет показано, это не меняет ситуации в целом):

Листинг 7.8

```
00000500:  F2 A9 BC 9B 37 FD 49 C0 ¦ 1E 89 20 EC 3A B9 F1 2E   Єй+Ы7¤I+-Й ь:¦ё.
00000510:  C8 96 C8 76 6E BC 3B A9 ¦ 8B BA 91 1D 82 70 E1 A6   +Ц+vn+;йЛ¦С_Врсж
00000520:  78 1B 22 09 F2 70 72 CB ¦ 85 C8 BC DC 62 DC F4 D9   x_».Єpr-E++_b_Ї+
00000530:  B6 E0 0E CB 6E AD 1B B9 ¦ 4C ED 44 A2 A2 0C CA 2D   ¦p-nн_¦LэDвв--..
00000540:  51 B4 27 6E E6 17 72 10 ¦ 7C 79 29 EF E4 A5 08 A0   Q¦'nц_r_|y)яфе_a
00000550:  5C 5C B7 FA 73 DF CC 9F ¦ CB AA A8 8C E0 82 25 84   \\+·s_¦Я-киМрВ%Д
00000560:  7E DA 16 35 8F 86 6C 52 ¦ 2D 8F F2 FB 2A C7 E9 85   ~+_5ПЖlR-ПЄ_*¦щЕ
00000570:  B8 9E F3 45 D9 1A 57 5E ¦ E1 A2 67 37 E4 D7 71 A2   +ЮєЕ+_W^свg7ф+qв
00000580:  B6 1B B9 C3 79 5D CA 2F ¦ AF 44 A2 20 65 F2 24 9B   ¦_¦+y]-/пDв еЄ$Ы
00000590:  BC 83 D4 34 05 7E FA AA ¦ 83 F4 AF EA 72 93 BB 57   +Г+4_~·кГЇпъrУ+W
```

Если теперь данные из зашифрованного файла сложим по модулю два (ведь по правилу Керкхоффа нам известен способ сложения гаммы с открытым текстом) с данными типового договора, то получим не что иное, как несколько байтов гаммы (см. листинг 7.9). Точно так же мы поступали с текстом «Приветствую, мой ненаглядный сосед!», рассматривая способ подделывания подобных сообщений.

Листинг 7.9

```
Байты из типового договора по смещению 500h:

C4 20 CE 20 C3 20 CE 20 C2 20 CE 20 D0 20 20 20   Д О Г О В О Р  .

Складываем с байтами шифротекста договора по тому же смещению:

F2 A9 BC 9B 37 FD 49 C0 1E 89 20 EC 3A B9 F1 2E   Єй+Ы7¤I+-Й ь:¦ё.

И получаем байты гаммы:

36 89 72 BB F4 DD 87 E0 DC A9 EE CC EA 99 D1 0E   Єй+Ы7¤I+-Й ь:¦ё.
```

Теперь допустим, мы каким-либо образом узнали, что сделка была оформлена на общую сумму $15000. Следовательно, эта сумма была проставлена в документе, зашифрованный вариант которого мы имеем.

Посмотрим на документ — поле для ввода суммы сделки находится по смещению 562h (1378) в файле. Можно предположить, и это почти наверняка окажется так, что и в зашифрованном файле по тому же смещению находится строка «$15000».

Если злоумышленник[42] задастся целью подделать сумму сделки, он может в этом преуспеть, сделав всего несколько простых вычислительных действий при помощи средств модулярной арифметики.

Итак, следуя данному немного выше примеру и сделав предположение, что сумма равна ни больше ни меньше $15000, сложим по модулю два байты строки в зашифрованном тексте, начиная со смещения 562h (именно там находится сумма договора), с байтами строки «$15000» — предполагаемой нами суммой. Результат отражен в листинге 7.10.

Листинг 7.10

```
7E DA 16 35 8F 86 6C 52 2D 8F F2 FB 2A C7 E9 85   ~+_5ПЖlR-П╞_*|щE

складываем с байтами строки "$15000":

      24 31 35 30 30 30                           $15000

И получаем байты гаммы:

      32 04 BA B6 5C 62                           2_||\b
```

Получив таким простым образом уже не просто несколько байтов гаммы, а участок гаммы, где находится интересующая нас информация, мы можем зашифровать все, что нам заблагорассудится. В нашем случае это любая сумма договора. А затем мы можем поместить результат по выбранному смещению обратно в зашифрованный файл и выслать его действительному получателю.

Уменьшим слегка сумму контракта, сложив байты шифротекста с байтами строки «$00000» (см. листинг 7.12):

Листинг 7.11

```
складываем байты гаммы:

      32 04 BA B6 5C 62                           2_||\b

с байтами строки "$00000":

      24 30 30 30 30 30                           $00000

и снова получаем байты шифротекста:

      16 34 8A 86 6C 52                           .4КЖlR
```

В результате этих действий получаем почти тот же зашифрованный текст, но теперь в контракте стоит совершенно другая сумма (см. листинг 7.12, рамочкой обведены изменившиеся байты).

[42] Авторы ни в коем случае не хотели бы ставить себя и тем более читателя на место злоумышленника, но — «увы и ах!» — криптоаналитику всегда приходится это делать.

Листинг 7.12

```
00000500:  F2 A9 BC 9B 37 FD 49 C0 ¦ 1E 89 20 EC 3A B9 F1 2E   Єй+Ы7¤I+-Й ь:¦ё.
00000510:  C8 96 C8 76 6E BC 3B A9 ¦ 8B BA 91 1D 82 70 E1 A6   +Ц+vn+;йЛ¦С_Врсж
00000520:  78 1B 22 09 F2 70 72 CB ¦ 85 C8 BC DC 62 DC F4 D9   x_».Єpr-Е++_b_Ї+
00000530:  B6 E0 0E CB 6E AD 1B B9 ¦ 4C ED 44 A2 A2 0C CA 2D   ¦р-пн_¦LэDвв--..
00000540:  51 B4 27 6E E6 17 72 10 ¦ 7C 79 29 EF E4 A5 08 A0   Q¦'nц_r_|y)яфе_а
00000550:  5C 5C B7 FA 73 DF CC 9F ¦ CB AA A8 8C E0 82 25 84   \\+·s_¦Я-киМрВ%Д
00000560:  7E DA 16 34 8A 86 6C 52 ¦ 2D 8F F2 FB 2A C7 E9 85   ~+.4КЖlR-ПЄ_*¦щЕ
00000570:  B8 9E F3 45 D9 1A 57 5E ¦ E1 A2 67 37 E4 D7 71 A2   +ЮєЕ+_W^свg7ф+qв
00000580:  B6 1B B9 C3 79 5D CA 2F ¦ AF 44 A2 20 65 F2 24 9B   ¦_¦+у]-/пDв еЄ$Ы
00000590:  BC 83 D4 34 05 7E FA AA ¦ 83 F4 AF EA 72 93 BB 57   +Г+4_~·кГЇпъrУ+W
```

Таким образом, для выбранного специалистами фирмы «Рога и Копыта» способа шифрования (то есть однократного шифрования равновероятной гаммой) существуют вполне приемлемые условия, при которых может быть использован по крайней мере один способ подделки сообщений. А это уже весьма ощутимый результат, который может оказаться «последней ошибкой».

Аналогичным и еще более простым с точки зрения криптоанализа примером является шифрование методом однократного гаммирования электронных почтовых сообщений.

Сообщения электронной почты в сети Интернет передаются (как и многие другие данные простых сетевых протоколов прикладного уровня) в текстовом виде, приемлемом для анализа невооруженным взглядом. В листинге 7.13 представлен пример почтового сообщения в том виде, в котором оно путешествует в сети от компьютера к компьютеру. А листинг 7.14 содержит то же самое сообщение, но так, как его видит рядовой пользователь.

Листинг 7.13

```
From POPmail Tue Nov 17 18:45:15 1998
 (with Netcom Interactive pop3d (v1.21.1 1998/05/07) Wed Nov 18 02:39:15 1998)
X-From_: seg@crow.sec.gov  Tue Nov 17 20:36:28 1998
Received: from crow.sec.gov (crow.sec.gov [204.192.28.11]) by
multi33.netcomi.com (8.8.5/8.7.4) with ESMTP id UAA32707 for <fc@all.net>;
Tue, 17 Nov 1998 20:36:28 -0600
Received: (from seg@localhost) by crow.sec.gov  id VAA01882 for fc@all.net;
Tue, 17 Nov 1998 21:37:15 -0500
From: Joe Segreti <seg@crow.sec.gov>
Message-Id: <199811180237.VAA01882@crow.sec.gov>
Subject: Ответ на предложение
To: fc@all.net
Date: Tue, 17 Nov 1998 21:37:14 -0500 (EST)
X-Mailer: ELM [version 2.4 PL25 PGP7]
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit

Доброго времени суток!

Уважаемые коллеги, предлагаем вам следующую форму сотрудничества.
...
```

Листинг 7.14

```
Date: Tue, 17 Nov 1998 21:37:14 -0500 (EST)
From: Joe Segreti <seg@crow.sec.gov>
To: fc@all.net
Subject: Ответ на предложение

Доброго времени суток!

Уважаемые коллеги, предлагаем вам следующую форму сотрудничества.
...
```

И в том и в другом случае в сообщении остается много служебных слов, которые используются почтовым программным обеспечением для доставки сообщения адресату. Например, ключевое слово **Date:** и следующая за ним дата отсылки письма является обязательным атрибутом любого письма. И точно так же, как подвергся анализу текстовый файл формата MSWord, мы можем подвергнуть пристальному изучению это электронное сообщение, подставив, к примеру, другую дату отсылки письма или изменив приветствие и подпись[43].

Собственно говоря, все приведенные примеры и недостатки и есть ответ на вопрос, почему почти никто не пользуется методом однократного гаммирования — все зависит от того, как именно это делается и какими ресурсами обладает пользователь шифросистемы.

Криптографическая защита исходных текстов

Время реакции на весьма динамично развивающуюся среду веб-технологий определяет эффективность работы любого Интернет-сайта. Современные технологии создания веб-контента в режиме реального времени дали профессиональному веб-мастеру мощнейшие инструменты для управления потоками информации в Интернет. В условиях постоянного роста производительности использование языков сценариев или, как их еще называют, scripting languages или scripts стало одним из опорных решений фундаментального подхода организации инфраструктуры Интернет-сайтов. Мощные, легко осваиваемые специализированные языки программирования, ориентированные на разработчиков веб-сайтов, получили широчайшее распространение. Языки сценариев изначально были ориентированы на быстрое и эффективное решение иных задач, нежели языки программирования системного уровня, поскольку они создавались как логически связующие компоненты к уже готовым программным решениям. Преимущества такого подхода перед традиционным статическим наполнением веб-порталов видны невооруженным взглядом. Это и гибкость построения гипертекстовых переходов, и возможность создания отчетов по записям в базах данных в реальном времени, и генерация комплексных веб-документов из существующих элементарных компонентов.

[43] Чтобы узнать приветствие и подпись другого человека, иногда бывает достаточно просто написать ему письмо так, как будто бы вы ошиблись адресом. Ответ от него с сообщением об этом будет наверняка содержать всю необходимую информацию.