**Транспортное кодирование**

Поскольку системы шифрования данных часто используются для кодирования текстовой информации : переписки, счетов, платежей электронной коммерции, и при этом криптосистема должна быть абсолютно прозрачной для пользователя, то над выходным потоком криптосистемы часто производится транспортное кодирование, то есть дополнительное кодирование (не шифрование !) информации исключительно для обеспечения совместимости с протоколами передачи данных.

Все дело в том, что на выходе криптосистемы байт может принимать все 256 возможных значений, независимо от того был ли входной поток текстовой информацией или нет. А при передаче почтовых сообщений многие системы ориентированы на то, что допустимые значения байтов текста лежат в более узком диапазоне : все цифры, знаки препинания, алфавит латиницы плюс, возможно, национального языка. Первые 32 символа набора ASCII служат для специальных целей. Для того, чтобы они и некоторые другие служебные символы никогда не появились в выходном потоке используется транспортное кодирование.

Наиболее простой метод состоит в записи каждого байта двумя шестнадцатиричными цифрами-символами. Так байт 252 будет записан двумя символами 'FC'; байт с кодом 26, попадающий на специальный символ CTRL-Z, будет записан двумя допустимыми символами '1A'. Но эта схема очень избыточна : в одном байте передается только 4 бита информации.

На самом деле практически в любой системе коммуникации без проблем можно передавать около 68 символов (латинский алфавит строчный и прописной, цифры и знаки препинания). Из этого следует, что вполне реально создать систему с передачей 6 бит в одном байте (26<68), то есть кодировать 3 байта произвольного содержания 4-мя байтами из исключительно разрешенных (так называемых печатных) символов. Подобная система была разработана и стандартизирована на уровне протоколов сети Интернет – это система Base64 (стандарт RFC1251).

Процесс кодирования преобразует 4 входных символа в виде 24-битной группы, обрабатывая их слева направо. Эти группы затем рассматриваются как 4 соединенные 6-битные группы, каждая из которых транслируется в одиночную цифру алфавита base64. При кодировании base64 входной поток байтов должен быть упорядочен старшими битами вперед.

Каждая 6-битная группа используется как индекс для массива 64-х печатных символов. Символ, на который указывает значение индекса, помещается в выходную строку. Эти символы выбраны так, чтобы быть универсально представимыми и исключают символы, имеющие специальное значение (".", CR, LF).

Алфавит Base64

Значение Код Значение Код Значение Код Значение Код

0 A 17 R 34 i 51 z

1 B 18 S 35 j 52 0

2 C 19 T 36 k 53 1

3 D 20 U 37 l 54 2

4 E 21 V 38 m 55 3

5 F 22 W 39 n 56 4

6 G 23 X 40 o 57 5

7 H 24 Y 41 p 58 6

8 I 25 Z 42 q 59 7

9 J 26 a 43 r 60 8

10 K 27 b 44 s 61 9

11 L 28 c 45 t 62 +

12 M 29 d 46 u 63 /

13 N 30 e 47 v заполнитель =

14 O 31 f 48 w

15 P 32 g 49 x

16 Q 33 h 50 y

Выходной поток (закодированные байты) должен иметь длину строк не более 76 символов. Все признаки перевода строки и другие символы, отсутствующие в таблице 1, должны быть проигнорированы декодером base64. Среди данных в Base64 символы, не перечисленные в табл. 1, переводы строки и т.п. должны говорить об ошибке передачи данных, и, соответственно, программа-декодер должна оповестить пользователя о ней.

Если в хвосте потока кодируемых данных осталось меньше, чем 24 бита, справа добавляются нулевые биты до образования целого числа 6-битных групп. А до конца 24-битной группы может оставаться только от 0 до 3-х недостающих 6-битных групп, вместо каждой из которых ставится символ-заполнитель "=". Поскольку весь входной поток представляет собой целое число 8-битных групп (т.е., просто байтных значений), то возможны лишь следующие случаи:

1. Входной поток оканчивается ровно 24-битной группой (длина файла кратна 3). В таком случае выходной поток будет оканчиваться четырьмя символами Base64 без каких либо дополнительных символов.
2. "Хвост" входного потока имеет длину 8 бит. Тогда в конце выходного кода будут два символа Base64, с добавлением двух символов "=".
3. "Хвост" входного потока имеет длину 16 бит. Тогда в конце выходного будут стоять три символа Base64 и один символ "=".

Так как символ "=" является хвостовым заполнителем, его появление в теле письма может означать только то, что конец данных достигнут. Но опираться на поиск символа "=" для обнаружения конца файла неверно, так как, если число переданных битов кратно 24, то в выходном файле не появится ни одного символа "=".