УДК 681.391.1

«ИГРЫ В ПРЯТКИ» С ПЕРЕХВАТЧИКОМ СООБЩЕНИЙ

И. Л. Ерош,
доктор техн. наук, профессор
В. В. Скуратов,
ст. преподаватель
Санкт-Петербургский государственный университет аэрокосмического приборостроения

В статье решается задача введения в заблуждение незаконного перехватчика при передаче сообщений. Рассматриваются различные варианты усиленной защиты некоторых наиболее важных фрагментов сообщения как с использованием булевых преобразований, так и с использованием матричных преобразований передаваемого текста в поле Галуа GF(2).

The problem of misleading of illegal message interceptor while transferring messages is being solved in the article. Different variants of intensified defence of some most important fragments of messages with usage of both Boolean and matrix transformations in the GF(2) Galois field of transferred text are considered.

При передаче сообщений секретным, как правило, является не все сообщение, а только некоторые его фрагменты, например, даты, суммы, адреса. Расшифровка остальных фрагментов сообщения, хотя и доставит некоторую радость незаконному перехватчику, вряд ли сможет в большинстве случаев нанести существенный ущерб передающей и принимающей стороне. Поэтому подлинно секретные фрагменты необходимо защищать особенно тщательно, например, так, чтобы ввести перехватчика в заблуждение относительно их истинного смысла.

В работах [1, 2] был предложен метод булевых преобразований двоичных последовательностей для решения различных задач защиты информации. Основная идея метода состоит в том, что, имея две двоичные последовательности A и B длины n, можно найти булеву функцию F, которая преобразует последовательность A в последовательность B. На последовательность A накладывается очень слабое ограничение – она должна быть ненулевой. В более общем случае, если имеется s двоичных последовательностей A_i (i = 1, 2, 3, ..., s) и столько же последовательностей B_i (i = 1, 2, 3, ..., s), то при незначительных ограничениях на последовательности A_i (они не должны быть связаны сдвигом) можно найти булеву функцию F, которая из любой последовательности A_i будет строить соответствующую ей последовательность B_i.

Рассмотрим различные примеры введения перехватчика в заблуждение относительно смысла передаваемого сообщения.

Пример 1. Обозначим множество последовательностей $\{A_i\}$ через A и разобьем это множество на непересекающиеся подмножества последовательностей: A_1, A_2, ..., A_k, т. е.

$A_1 \cup A_2 \cup ... \cup A_k = A$;

$A_1 \cap A_2 \cap ... \cap A_k = \varnothing$, где $\varnothing$ – пустое множество.

Каждому подмножеству A_j поставим в соответствие одну последовательность B_j (j = 1, 2, ..., k) и найдем булеву функцию F, которая из каждой последовательности подмножества A_j будет строить соответствующую ему последовательность B_j. В этом случае возможно ввести незаконного перехватчика в заблуждение, если k абонентов будут подписывать свои сообщения разными паролями (двоичными последовательностями из соответствующего подмножества), а центр будет идентифицировать их с помощью булевой функции F.

Например, пусть множество последовательностей A состоит из подмножеств:

A_1: {101101, 011011, 111010}; A_2: {001111, 100011}; A_3: {101011, 110110}.

Пусть последовательности B_j (j = 1, 2, 3) будут соответственно равны:

B_1 = 100111, B_2 = 011110, B_3 = 010101.

По методике, изложенной в работе [1], найдем:

$$F = -1^*0^*1^*\rceil 3 \vee 0^*1^*2 \vee \rceil -1^*\rceil 0^*1 \vee \rceil 0^*\rceil 2 \vee -3^*1 + -1^*\rceil 1^*3 \vee$$
$$\rceil -4^* -3^*\rceil -2 \vee \rceil -2^*\rceil 0^*\rceil 4 \vee \cdot \rceil -2^*0^*1^*\rceil 3 \vee -2^*\rceil 0^*\rceil 1 \vee$$
$$-2^*\rceil -1^*\rceil 1^*\rceil 2 \vee -3^*2 \vee \rceil -2^*0^*\rceil 1^*3.$$

Полученная функция F из любой последовательности множества A_1 будет строить одну по-

следовательность B_1, из любой последовательности множества A_2 будет строить одну последовательность B_2 и т. д.

Пример 2. В работе [2] рассмотрен случай многократно вложенного скрытого смысла в сообщениях.

Пусть A, B, C, ... – некоторые равномощные множества осмысленных сообщений. Если каждое сообщение представляет собой двоичную последовательность длины n, то можно найти булеву функцию F, которая из каждой последовательности A строит соответствующую ей последовательность B, из каждой последовательности B – соответствующую ей последовательность C и т. д. В этом случае незаконный перехватчик, даже вскрывая передаваемые сообщения, не может точно знать, какое из сообщений является верным.

Например, пусть множество A состоит из последовательностей: {10111011, 01111010, 10101111}, множество B пусть содержит следующие последовательности: {11101001, 10100111, 00110111}, множество C состоит из следующих последовательностей: {11011001, 00111101, 11101110}. Найдем функцию F, которая из каждой последовательности A строит соответствующую ей последовательность B, из каждой последовательности B – соответствующую ей последовательность C. По методике работы [1] получаем:

$$F = -3^*-1^*0^*1^*]3\vee 0^*1^*2^*]3\vee]0^*2^*3\vee$$
$$]-1^*]2^*3\vee]-1^*]0^*2\vee]-1^*]0^*]1\vee]-4^*]1^*]2\vee$$
$$]0^*]1^*3\vee]-1^*]1^*]2\vee-2^*-1^*]0^*]2\vee]-2^*]2^*3^*]4\vee$$
$$-3^*]1^*]2\vee-2^*-1^*0^*1^*]3\vee-1^*0^*1^*2\vee]-1^*]1^*3\vee$$
$$]0^*]1^*]2\vee]-2^*]1^*]2\vee-2^*]1^*3.$$

Пример 3. Основную часть сообщения можно шифровать слабым шифром, а наиболее секретные фрагменты – дополнительно, шифром с повышенной криптостойкостью.

В работе [2] для дополнительного шифрования сообщений предлагалось использовать преобразование с помощью матриц размера (n, n) в поле Галуа GF(2). Преобразование сообщений с помощью умножения матрицы на двоичные блоки сообщений в поле GF(2) можно рассматривать как булевы линейные преобразования. Изменение размеров блоков и матриц шифрования позволяет создать дополнительные трудности перехватчику, причем это можно делать в наиболее ответственных местах передаваемого текста. Так, основной текст сообщения можно шифровать с помощью одной матрицы размера (n, n), а наиболее ответственные фрагменты – двумя или тремя матрицами разных размеров с блоками различных размеров. Кроме того, в наиболее ответственные места шифрованного сообщения можно замешивать гаммы случайных последовательностей.

Например, зашифруем фразу: Tom stays in **Rome 2004-03-26 at 7** o'clock. В этой фразе наиболее секретный фрагмент выделен жирным шрифтом. Запишем эту фразу в кодах ASCII в шестнадцатеричном представлении и разобьем на фрагменты по 17 символов в каждом. Последний неполный фрагмент дополним нулем: 546A6B20737461797 320696E20526F6D65 20323030342D30332 D3236206174203720 6F27636C6F636B2E0.

Сначала зашифруем сообщение путем умножения слева на матрицу M размера (5, 5). Матрицу M возьмем в виде

$$M = \begin{pmatrix} 11011 \\ 01101 \\ 11110 \\ 01110 \\ 11010 \end{pmatrix}.$$

Обратная матрица имеет вид:

$$M^{-1} = \begin{pmatrix} 00110 \\ 11100 \\ 00101 \\ 11011 \\ 10001 \end{pmatrix}.$$

$$\begin{pmatrix} 11011 \\ 01101 \\ 11110 \\ 01110 \\ 11010 \end{pmatrix} \times \begin{pmatrix} 546A6B20737461797 \\ 320696E20526F6D65 \\ 20323030342D30332 \\ D3236206174203720 \\ 6F27636C6F636B2E0 \end{pmatrix} =$$

$$= \begin{pmatrix} DA68FCA80E73FFF32 \\ 7D13C5BE5E68ADCB7 \\ 957DAFF4553DA4EE0 \\ C117C4D42649C5977 \\ B54F9FC4611094DD22 \end{pmatrix}.$$

Шифрованное сообщение выглядит следующим образом: DA68FCA80E73FFF32 7D13C5BE5E68ADCB7 957DAFF4553DA4EE0 C117C4D42649C5977 B54F9FC4611094DD2.

Расшифрование сообщения выполняется путем умножения обратной матрицы M^{-1} на фрагменты шифрованного сообщения, т. е.

$$\begin{pmatrix} 00110 \\ 11100 \\ 00101 \\ 11011 \\ 10001 \end{pmatrix} \times \begin{pmatrix} DA68FCA80E73FFF32 \\ 7D13C5BE5E68ADCB7 \\ 957DAFF4553DA4EE0 \\ C117C4D42649C5977 \\ B54F9FC4611094DD22 \end{pmatrix} =$$

$$= \begin{pmatrix} 546A6B20737461797 \\ 320696E20526F6D65 \\ 20323030342D30332 \\ D3236206174203720 \\ 6F27636C6F636B2E0 \end{pmatrix}.$$

Зашифруем дополнительно выделенный жирным шрифтом фрагмент сообщения **Rome 2004-03-26 at 7** путем умножения на матрицу вида

$$M_1 = \begin{pmatrix} 0100010011 \\ 1100001010 \\ 0010111000 \\ 0101101001 \\ 1000001110 \\ 1001101100 \\ 1100111000 \\ 0010011011 \\ 1011001101 \\ 1010101101 \end{pmatrix}.$$

Обратная матрица имеет вид

$$M^{-1} = \begin{pmatrix} 1110010010 \\ 1110111111 \\ 0010100101 \\ 0111000100 \\ 0111000111 \\ 0011111100 \\ 0100011110 \\ 1010011111 \\ 0000110011 \\ 0101110000 \end{pmatrix}.$$

Сам фрагмент после предварительного шифрования матрицей M выглядит так: E68ADCB 7957DAFF4553DA4EE0C117C4D42649C59.

Фрагмент состоит из 40 символов, поэтому его можно разбить для шифрования этой матрицей на десять блоков по четыре символа в каждом. После шифрования получим фрагмент в следующем виде: A664 7448 68BF B636 D4B2 6C1F C7FF E3BF 3006 CACF.

Легко проверяется, что умножением полученного фрагмента на матрицу M_1^{-1} восстанавливается фрагмент, зашифрованный матрицей M.

$$\begin{pmatrix} 0100010011 \\ 1100001010 \\ 0010111000 \\ 0101101001 \\ 1000001110 \\ 1001101100 \\ 1100111000 \\ 0010011011 \\ 1011001101 \\ 1010101101 \end{pmatrix} \times \begin{pmatrix} E68A \\ DCB 7 \\ 957D \\ AFF4 \\ 553D \\ A4EE \\ 0C11 \\ 7C4D \\ 4264 \\ 9C59 \end{pmatrix} = \begin{pmatrix} A664 \\ 7448 \\ 68BF \\ B636 \\ D4B2 \\ 6C1F \\ C7FF \\ E3BF \\ 3006 \\ CACF \end{pmatrix}.$$

Если незаконный перехватчик знает матрицу M и попытается восстановить сообщение, умножив слева его на матрицу M^{-1}, он получит полностью искаженное сообщение, т. е. искажению подвергнется не только выделенный фрагмент, но и все остальные участки сообщения. То же будет, если перехватчик знает матрицу M_1 и попытается вскрыть все сообщение, умножив его слева на обратную матрицу.

Пример 4. В сообщении передается информация о конечных пунктах прибытия самолета. Это сообщение может быть вскрыто перехватчиком, который прочтет название конечного пункта прибытия. Однако законный получатель, обрабатывая полученные данные, узнает истинное место назначения. Например, A_i –ложные пункты, B_i – истинные:

Сообщения A_i (расшифрованные перехватчиком): Landing in	Сообщения B_i (расшифрованные законным получателем): Landing in
Rome 524F4D45202020202020	New York 4E455720594F524B2020
Bombay 424F4D42415920202020	Paris 50415249532020202020
Berlin 4245524C494E20202020	Bagdad 424147444144202020
Washington 57415348494E47544F4E	Moscow 4D4F534B4F5720202020
Stockholm 53544F434B484F4C4D20	Venice 56454E49434520202020
Zurich 5A555249434820202020	Calcutta 43414 C43555454412020

Для нахождения функции F переведем названия городов в коды ASCII в шестнадцатеричном представлении и по методике [1] найдем:

$$F = 4*2*0*\rceil-1*-6\vee\rceil4*\rceil3*0*-2*\rceil-3*\rceil-5\vee$$
$$\rceil5*\rceil1*0*\rceil-2*-3*\rceil-4*\rceil-5\vee2*1*\rceil0*\rceil-1*\rceil-2*\rceil-7\vee$$
$$7*\rceil3*\rceil2*\rceil0*-2*\rceil-3*-4\vee5*\rceil2*\rceil1*\rceil0*\rceil-3*-6\vee$$
$$1*\rceil0*\rceil-1*\rceil-2*\rceil-3*-5\vee\rceil2*\rceil1*-1*\rceil-4*\rceil-5*-7\vee$$
$$\rceil0*\rceil-1*-3*\rceil-6*\rceil-7*-9\vee$$
$$\rceil16*\rceil3*\rceil\ 2*0*\rceil-1*\rceil-2*\rceil-3\vee4*3*\rceil2*0*-2*\rceil-3\vee$$
$$\rceil3*2*1*\rceil0*-4*\rceil-6\vee\rceil3*\rceil2*\rceil1*0*\rceil-1*-2*\rceil-8\vee$$
$$2*\rceil1*\rceil0*\rceil-1*-2*\rceil-3*\rceil-10\vee\rceil11*3*0*\rceil-2*\rceil-3*\rceil-5\vee$$
$$\rceil8*3*2*1*0\vee\rceil7*2*1*-1\vee\rceil6*1*0*-1*-2\vee$$
$$2*\rceil1*\rceil0*\rceil-1*\rceil-2*-3*-6\vee5*0*\rceil-1*-2*\rceil-4\vee$$
$$8*\rceil2*\rceil1*\rceil0*-1*\rceil-3\vee\rceil5*3*\rceil1*\rceil0*\ -3*-4\vee$$
$$\rceil4*3*\rceil1*-1*\rceil-2*\rceil-3*\rceil-4\vee$$
$$\rceil6*\rceil5*\rceil3*\rceil1*0*\rceil-2*-3*-6\vee\ 2*1*0*-2\vee$$
$$\rceil1*\rceil0*\rceil-1*\rceil-2*-4*-6\vee\rceil5*\rceil2*1*0*\rceil-1*-3*-5\vee$$
$$\rceil4*\rceil1*0*-1*\rceil-2*-6\vee\rceil2*0*-2*-3*\rceil-4*-8\vee$$
$$5*\rceil3*\rceil1*0*\rceil-1*\rceil-8\vee\rceil7*3*\rceil2*\rceil-1*-2*\rceil-4*\ -5\vee$$
$$3*\rceil1*0*\rceil-1*\rceil-2*-3*\rceil-6*\rceil-7\vee$$
$$4*\rceil3*2*\rceil1*\rceil0*-1*\rceil-3*\rceil-\ 7*\rceil-8\vee$$
$$3*\rceil2*1*\rceil0*\rceil-1*-2*\rceil-4*\rceil-9\vee6*3*2*-4\vee$$
$$4*\rceil2*1*0*-1\vee3*\rceil1*0*-1*-2\vee2*\rceil0*-1*-2*-3\vee$$
$$1*-1*-4*-5\vee10*3*\rceil1*-1*\rceil-4\vee$$
$$\rceil6*3*0*\rceil-1*\rceil-2*\rceil-3*\rceil-7\vee7*0*\rceil\ -1*\rceil-2*-3\vee$$
$$\rceil7*6*\rceil1*\rceil0*-1*-4\vee4*3*0*\rceil-2*-3\vee5*\rceil3*1*0*-1\vee$$
$$\rceil3*\rceil2*0*\rceil-1*\rceil-2*\rceil-3*-4\vee\ \rceil3*\rceil2*0*\rceil-2*\rceil-6*-12\vee$$

9*⌉1*⌉0*–2*⌉–5∨8*5*0*–1∨6*0*⌉–1*–2*–3∨
4*⌉3*2*1*–5∨⌉5*3*1*⌉0*–1*–2∨
⌉4*⌉3*⌉2*0*⌉–1*–2∨⌉6*⌉5*2*⌉1*–1*⌉–3*⌉–4∨
6*4*⌉2*⌉1*⌉0*⌉–4∨0*⌉–2*⌉–3*⌉–6*–11*⌉–13∨
5*⌉3*2*⌉0*–1*–3*⌉–4*⌉–5∨1*0* –1*–7*–10∨
8*2*⌉1*0*–1*⌉–2∨ 5*3*⌉2*⌉1*–1*–3∨
⌉2*1*⌉0*⌉–1*–2*⌉–3*–4*⌉–5∨
⌉5*⌉2*⌉1*–1*–4*⌉–7∨1*⌉–1*⌉–2*⌉–4*–10*–12∨
⌉4*⌉3*⌉0*⌉–1*–2*–13*–15∨
⌉1*⌉0*–3*⌉–6*–14*–16∨3*⌉0*⌉–1*⌉–2*–16*–
18∨17*⌉6*1*⌉–1*⌉–2*⌉–4*⌉–10*⌉–11∨
15*⌉2*–1*⌉–12*⌉–15*–20∨
12*⌉1*⌉0*⌉–1*⌉–2*⌉–10*⌉–18∨
14*⌉3*⌉1*⌉–5*⌉–6*⌉–21∨
⌉19*11*⌉2*⌉1*⌉0*⌉–1*⌉–2*⌉–16*⌉–17.

Перехватчик читает названия городов, приведенные в левом столбце, а законный получатель, используя функцию F, читает названия городов, приведенные в правом столбце. Функция F найдена из полного списка замен таблицы. Если перехватчику удалось каким-либо образом узнать несколько замен и построить функцию, то, применяя эту функцию к остальным городам, он будет получать бессмысленный набор символов.

Пример 5. В работе [2] рассматривалась задача распределения ресурсов вычислительной системы с использованием булевых преобразований запросов. Рассмотрим похожую задачу, в которой незаконный перехватчик может быть введен в заблуждение из-за неполного знания всех возможных запросов и соответствующих правильных ответов.

Рассмотрим такую ситуацию. Красная Шапочка договорилась с бабушкой о том, что, приходя к ней, она будет сначала стучать в дверь, а затем на запрос бабушки давать соответствующий ответ. Если ответ будет правильным, бабушка откроет Красной Шапочке дверь. Поскольку Красная Шапочка приходит к бабушке часто и каждый раз пары запрос–ответ должны быть разными, список запросов и ответов оказался очень длинным, например, содержащим 200 пар. При этом и запрос и ответ представляют собой для увеличения криптостойкости бессмысленный набор из десяти знаков (букв и цифр). Красной Шапочке папа помог составить булеву функцию F', которая по запросу бабушки формировала правильный ответ. Серый Волк неделю сидел в кустах около дома бабушки и тщательно записывал все запросы и ответы. Из полученных данных он сформировал собственную функцию F', которую и решил использовать, когда внучка почему-то не пришла в обычное время к бабушке. На запрос бабушки Серый Волк ответил с помощью своей функции F', сформированной из неполного списка и, конечно, ответ оказался неверным. Бабушка вызвала полицию и та арестовала Волка. Интересно, какова вероятность того, что функция F', построенная Волком, случайно «угадает» ответ на запрос?

Например, часть списка запросов и ответов пусть выглядит так:

Запрос в виде символов ASCII кодов и его шестнадцатеричное представление	Ответ в виде символов ASCII кодов и его шестнадцатеричное представление
PqWzt: 5071577A74	QcfSU: 5163665355
W+Vr,: 772B76522C	GFL/: 47466F4C2F
Yt-!J: 79542D214A	HWQHu: 6857514875
?Ic,f: 3F49632C66	X=Er!: 583d457221
)nTWk: 296E54574B	&:ZrV: 263A5A7256

Для упрощения примера длина запросов и ответов сокращена до пяти символов и число пар взято равным пяти. Даже в этом случае Красной Шапочке трудно запомнить все пары запрос–ответ. Функция F, построенная Красной Шапочкой, имеет вид

F = ⌉2*1*⌉–1*–2*⌉–3*–4∨3*⌉1*0*⌉–2*⌉–5∨
⌉5*2*1*⌉0*–2*⌉–3∨3*⌉2*⌉1*0*–1∨
⌉5*⌉1*⌉–1*⌉–2*–4*⌉–6∨⌉3*2*⌉1*0*⌉–1*–2∨
⌉3*1*⌉0*–1*⌉–2*–3*⌉–5∨3*⌉1*⌉0*–1*⌉–2*–3∨
⌉1*⌉–1*⌉–2*⌉–3*–4∨4*3*⌉0*–1*⌉–2*⌉–4∨
4*⌉1*0*⌉–1*–2*⌉–4∨3*1*0*⌉–2* –3∨
⌉1*0*⌉–1*–3*⌉–5∨⌉4*2*1*–1*⌉–2*–6∨
5*4*1*⌉0*⌉–1∨2*1*0*–1*⌉–3∨⌉4*2*⌉1*⌉0*–1*–5∨
3*⌉2*0*⌉–1*!–2∨⌉3*⌉2*1*⌉–1*–2*⌉–6∨
⌉6*2*1*⌉–1*⌉–2*–3∨4*1*⌉–1*–2*–3∨
2*1*⌉0*–1*–4*–5∨⌉2*⌉1*⌉0*–2*–8∨
⌉3*⌉2*⌉1*⌉0*–1*–2∨⌉4*2*1*0∨
4*⌉2*⌉0*⌉–1*⌉–2*⌉–3∨4*2*⌉0*–1*⌉–2*⌉–4∨
0*⌉–1*–2*–3*⌉–4∨ ⌉6*⌉1*0*⌉–1*⌉–5∨
⌉4*⌉3*⌉2*1*⌉0*–4∨⌉5*⌉1*⌉0*–1*⌉–2*⌉–4∨
⌉2*1*0*–1*–5∨6*1*⌉–1*–2*–4∨⌉3*2*1*⌉0*–1*–2∨
2*⌉1*⌉0*–1*⌉–3*–4∨⌉2*⌉1*0*–1*–3*⌉–6∨
⌉3*0*⌉–1*⌉–4*⌉–5∨⌉3*1*0*–1*⌉–2*⌉–3∨
⌉2*⌉0*⌉–1*⌉–2*–3*–4∨4*0*⌉–1*–2*–3∨
⌉3*2*0*–1*⌉–3*⌉–4∨⌉4*⌉2*⌉1*⌉0*⌉–1.

Пусть Серый Волк подсмотрел три первых пары запрос–ответ. Он построит примерно такую функцию:

F' = ⌉2*0*–1*–3∨⌉3*2*0*⌉–1*–2∨4*⌉0*–1*⌉–2*–3∨
4*⌉2*1*⌉–1*⌉–3∨4*2*⌉0*–1*⌉–2∨
⌉2*⌉1*0*⌉–1*⌉–5∨⌉4*⌉3*1*–4∨
⌉5*⌉1*⌉0*–1*⌉–2*⌉–4∨3*⌉2*⌉1*0*–1∨
1*⌉0*⌉–1*–2*⌉–5∨⌉3*2*1*–2 ∨⌉1*⌉0*–1*–2*⌉–3∨
⌉4*0*⌉–1*–4∨0*⌉–1*⌉–2*⌉–3*–4∨⌉3*1*⌉–1∨
⌉2*⌉1*⌉–1*–2*–3∨0*⌉–1*⌉–2*⌉–5∨
⌉3*⌉2*0*–1*⌉–2∨⌉0*⌉–1*⌉–2*–3*–4∨
3*1*0*–1*–2∨3*2*⌉1*⌉0∨⌉3*2*0*–1*⌉–3∨
⌉4*⌉1*⌉0*⌉–1.

При ответе на четвертый и пятый запросы Серый Волк ответит неверно. Красная же Шапочка на все запросы ответит верно.

Пример 6. Рассмотрим теперь случай, когда из практически одинаковых сообщений, передаваемых по открытому каналу, законный полу-

чатель извлекает полезную информацию, в то время как незаконному перехватчику все передаваемые сообщения будут казаться совершенно одинаковыми.

Выберем какое-нибудь сообщение, например «Спокойное море»: The sea quiet. Будем добавлять пробелы и знаки препинания и даже введем грамматические неточности, но так, чтобы не исказить смысл сообщения. Каждому варианту будем сопоставлять новое выражение. Результаты преобразований будут выглядить так:

Спокойное море: The sea quiet 546865202073656120717569657 4	Сильный холод: Strong cold 5374726F6E6720636F6C64202020
Спокойное море: The sea quiet 5468652073656120207175696574	Дождь и град: Rain and hails 5261696E20616E64206861696C73
Спокойное море: A sea quiet 4120207365612020717569657420	Скоро ураган: Soon hurricane 536F6F6E2068757272696361 6E65
Спокойное море: The sea quiet 5468652073656120717569657420	Жара наступает: The heat comes 5468652068656174206 36F6D6573

В левом столбце все фразы кажутся совершенно одинаковыми, так их читает перехватчик. Однако законный получатель с помощью функции F читает их совершенно по-разному. Функция F в этом случае имеет вид

F = 60*⌉3*⌉1*0*−2∨53*0*−2*−3−44*⌉4*0*⌉−1∨
10*⌉2*⌉1*0*⌉−1∨29*2*⌉1*−13∨1*0*−3*18∨
20*2*0*⌉−2*−22∨⌉4*⌉1*0*⌉−4*−24∨
6*⌉1*⌉0*−1*⌉−2*−36∨⌉2*⌉1*⌉0*⌉−1*−6*⌉−8∨
28*−1*−3*−23∨27*0*−2*−4∨25*0*−2*⌉−3∨
22*⌉2*0*⌉−1*−29∨21*⌉1*−1*⌉−2*−3*−30∨
20*3*1*⌉0*−2∨19*0*−3∨17*⌉3*0*⌉−1*−2*⌉−3∨
14*1*⌉0*⌉−1*−2*−37∨13*0*⌉−2*−3*−38∨11*4*0*−2∨
9*2*⌉−1*−2∨5*⌉2*0*−3*−4∨3*⌉1*⌉0*−2*⌉−3*−48∨
⌉2*⌉1*⌉0*−1*⌉−3*−52∨⌉2*⌉1*⌉−1*⌉−2*−55∨
⌉2*0*⌉−2*⌉−5∨⌉3*⌉2*⌉1*⌉0*−2*⌉−10∨
⌉3*⌉2*⌉1*⌉−1*⌉−2*⌉−3*⌉−12∨ 10*⌉2*⌉1*⌉0*−2*⌉−5∨
9*⌉1*⌉0*⌉−1*−3*⌉−6∨8*⌉2*⌉−1*⌉−7∨
7*⌉1*⌉−1*⌉−2*⌉−8*⌉−18∨2*1*0*⌉−10∨
2*−1*−2*−3*⌉−13*⌉−23∨2*1*⌉0*−2*⌉−14∨
⌉13*⌉1*⌉0*−2*−3∨⌉6*−1*⌉−2*⌉−6*−9*⌉−36∨
2*1*⌉0*⌉−4*−7*⌉−42∨39*⌉3*0*⌉−12∨
36*1*⌉−1*⌉−2*⌉−15∨30*1*−1*−2*−3*⌉−21∨
2*0*⌉−1*⌉−3*−4*⌉−24∨6*⌉3*−1*−2*−3*⌉−45∨
⌉3*⌉1*−1*⌉−2*−5*⌉−47∨⌉4*⌉3*⌉2*⌉0*⌉−1*−3*⌉−54∨
⌉59*3*⌉1*−1*⌉−2∨⌉50*18*3*⌉1*−3∨
⌉45 *13*1*⌉−1*−2*−3∨⌉40*⌉3*⌉2*1*⌉0*⌉−1*−2∨
⌉37*⌉1*⌉0*−2*⌉−4*−5∨⌉36*4*⌉2*⌉1*⌉−1∨
⌉35*3*⌉1*⌉0*⌉−2∨⌉12*⌉0*⌉−1*⌉−2*−4*−30∨
⌉11*1*⌉0*⌉−1*−4*−31∨⌉38*⌉2*⌉1*⌉0*−1*−2∨
3*1*0*⌉−26∨⌉22*3*0*−3*⌉−29∨⌉2*1*0*⌉−3*⌉−34∨
⌉14*⌉2*1*⌉−1*−3*⌉−37∨⌉13*2*⌉1*0*⌉−1*⌉−2*⌉−38∨
2*1*0*−2*⌉−42∨⌉7*0*−1*−2*⌉−44∨
⌉3*⌉2*⌉1*⌉0*⌉−1*−2*−4*⌉−48∨⌉26*2*⌉1*⌉0*⌉−1*−2∨
⌉22*⌉2*⌉0*−1*−2*⌉−3*⌉−4∨
⌉20*⌉2*⌉1*−1*⌉−2*−3*⌉−5∨⌉14*⌉1*−1*−2*⌉−4∨
⌉11*⌉3*1*−1*⌉−2*⌉−3∨4*⌉3*⌉1*⌉0*⌉−1*−9∨
⌉2*⌉1*⌉0*⌉−1*⌉−2*⌉−3*−10∨1*⌉0*⌉−1*⌉−4*−12∨
0*−1*⌉−2*−26∨11 *1*⌉0*⌉−1*⌉−2*⌉−31∨
5*⌉1*−5*⌉−7*⌉−37∨⌉8*3*⌉−1*⌉−2*−7*⌉−39∨
⌉7*⌉0*⌉−1*⌉−2*⌉−3*−8*⌉−40∨⌉41*1*0*⌉−2∨
⌉37*2*⌉0*⌉−1*−2*−14∨⌉29*2*−2*−3*−22.

Заключение

В статье приведены некоторые варианты введения незаконного перехватчика в заблуждение относительно истинного смысла сообщения. Для этого использованы различные подходы:

усиление криптостойкости некоторых наиболее важных фрагментов сообщения за счет дополнительного шифрования;

использование различных паролей многими пользователями, которые центр преобразует в более узкое множество паролей, причем так, что каждая новая комбинация будет соответствовать либо одному пользователю, либо группе пользователей;

преобразование полученного расшифрованного сообщения булевыми функциям, которые кардинально меняют смысл сообщения.

Во всех этих случаях создаются дополнительные трудности для незаконного перехватчика, причем в ряде случаев перехватчик, даже вскрыв сообщение, в принципе не может узнать, какое же сообщение является истинным.

Литература

1. **Ерош И. Л.** Защита информационных потоков в системах распределенного контроля и управления // Информационно-управляющие системы. – 2002. – № 1. – С. 40–46.
2. **Ерош И. Л.** Разграничение доступа к ресурсам в системах коллективного пользования // Информационно-управляющие системы. – 2003. – № 2–3. – С. 63–66.
3. **Ерош И. Л.** Передача со скрытым смыслом. // Информационно-управляющие системы. – 2004. – № 5. – С. 40–46.