УДК 681.391.1

ПЕРЕДАЧА СО СКРЫТЫМ СМЫСЛОМ

И. Л. Ерош,
д-р техн. наук, профессор
Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП)

В статье решается задача, в которой сообщение, принятое различными группами пользователей, может быть прочитано не одинаково, так как в сообщении заложен многократный скрытый смысл. При этом передача сообщения может выполняться как в открытом виде, так и в зашифрованном (во втором случае его необходимо предварительно расшифровать). Рассмотрены два варианта решения этой задачи. В первом при передаче сообщения в осмысленный текст вставляется некоторый фрагмент, который разными группами пользователей должен читаться по-разному. Во втором каждая группа пользователей читает сообщение, адресованное только ей, остальные пользователи получают бессмысленный набор символов.

The article is dedicated to resolution of a problem of interpretation of messages being received by different groups of users that may comprehend given message in different ways,because of multiple hidden implications associated with them. Message itself can be transmitted as well in clear as encrypted format. Prior decryption is required in the second case. Two possible resolutions of this problem are considered. The first resolution implies the insertion of a certain fragment into meaningful text of a message. Different users should interpret this fragment in different way. The second resolution assumes that each group of the users receives meaningful message addressed especially to it. The rest of the users are getting meaningless set of characters.

Введение

В ряде случаев при передаче сообщений требуется обеспечить дополнительную секретность, смысл которой состоит в том, чтобы некоторые пользователи, имеющие доступ к расшифрованному сообщению, не получали полной информации о точных датах, суммах и других данных, предназначенных для особого круга лиц из числа пользователей.

Реализация передачи со скрытым смыслом

Рассмотрим задачу передачи сообщения, в которое заложен скрытый смысл, при этом для каждой группы пользователей (получателей сообщения) этот смысл может быть различным.

Для выявления скрытого смысла сообщения будем использовать булевы преобразования двоичных последовательностей. Напомним основные теоремы из [1], с помощью которых будет обеспечиваться скрытый смысл полученного сообщения.

Пусть $A(a_1, a_2, \ldots, a_n)$ и $B(b_1, b_2, \ldots, b_n)$ — две двоичных последовательности длины n. Каждый элемент b_i является результатом преобразования булевой функцией F элемента a_i и некоторых элементов из «окружения» a_i. Так, если последовательность A имеет вид 0100100101001110 и задана функция $F = 0^* \urcorner 3 \vee -2$, это означает, что каждый элемент

$$b_i = a_i^* \urcorner (a_{i-3}) \vee a_{i-2},$$

где * — конъюнкция, $\urcorner$ — инверсия, $\vee$ — дизъюнкция.

Последовательность B в этом случае имеет вид 0101001001010111. В шестнадцатеричном коде такое преобразование будет записываться как 494E → → 5257, а в кодах ASCII этому преобразованию соответствует in → rw.

Теорема 1. Для того чтобы существовала булева функция F, такая, что $F(A) = B$, *необходимо и достаточно*, чтобы A была ненулевой последовательностью.

Теорема 2. Пусть $A_1, A_2, \ldots, A_s$ и $B_1, B_2, \ldots, B_s$ — два набора двоичных последовательностей длины n; s — целое, равное числу пар последовательностей. Для того чтобы существовала функция F, преобразующая любую последовательность A_i в последовательность B_i, *достаточно*, чтобы в A_i не было двух последовательностей, связанных сдвигом.

Для теоремы 2 можно сформулировать *необходимое и достаточное* условие: в таблице истинности функции F, построенной по методике работы [1], не должно быть двух одинаковых строк (набо-

ров), на которых функция принимала бы различные значения. К сожалению, это *необходимое и достаточное* условие проверяется труднее, чем просто *достаточное*.

Например, пусть требуется обеспечить преобразование A → A, 5 → D. Последовательности A и 5 связаны сдвигом, т. е. не удовлетворяют *достаточному* условию, но из таблицы истинности видно, что в ней нет одинаковых строк, на которых функция принимала бы разные значения (*необходимое и достаточное* условие). Поэтому находится общая функция преобразования: $F = \neg 1$. Если же взять A → 2, 5 → D, то такая функция не может быть построена, так как на одинаковых наборах в таблице истинности функция должна принимать различные значения.

Рассмотрим два варианта решения задачи, которые реализуются с помощью различных классов булевых функций.

Вариант 1. В передачу произвольной длины вставляется некоторый фрагмент, при обработке которого функцией F_i каждый пользователь (например, солдат, офицер и генерал) читает текст, предназначенный только ему, т. е. фрагмент содержит разный смысл для разных пользователей (рис. 1).

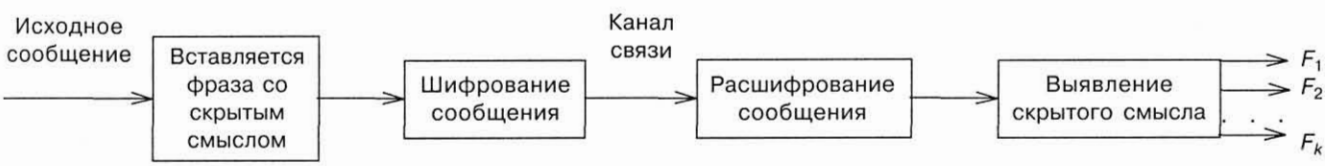
Пусть, например, выбраны следующие вставляемые в осмысленный текст фразы (таблица, столбец 1).

В первом столбце таблицы приведены фразы на русском языке, соответствующие им фразы на английском языке и английский вариант в кодах ASCII в шестнадцатеричном представлении. Во втором, третьем и четвертом столбцах приведены фразы, которые должны читаться соответственно солдатом (с помощью функции F_1), офицером (с помощью функции F_2) и генералом (с помощью функции F_3).

Каждую функцию F_i будем находить для преобразования 16-байтового блока сообщения. Если блоки сообщения оказываются короче 16 байт, то в конце фразы добавляются пробелы. Можно находить преобразования любой длины; блоки по 16 байт выбраны для упрощения примера.

Солдат, офицер и генерал этот текст читают совершенно по-разному с помощью собственных функций F_i. Для данного примера, используя процедуру, описанную в [1], вычисляем:

$$F_1 = \neg 5^* \neg 4^* 0^* \neg - 1^* - 2^* \neg - 3^* \neg - 4 \vee \neg 3^* \neg 1^* 0^* \neg - 2^* - 3^* \neg - 5 \vee$$
$$4^* \neg 3^* \neg 2^* \neg 1^* \neg 0^* - 1^* \neg - 3 \vee 4^* 0^* \neg - 1^* \neg - 2^* \neg - 4 \vee$$
$$8^* 2^* \neg 1^* \neg 0^* \neg - 1^* \neg - 3^* \neg - 6 \vee 7^* 2^* 0^* - 1^* - 7 \vee$$
$$2^* \neg 1^* 0^* - 1^* - 4 \vee 4^* 0^* \neg - 1^* - 2^* - 3 \vee$$
$$\neg 6^* 3^* \neg 0^* \neg - 1^* \neg - 5^* \neg - 6^* - 7 \vee 6^* 2^* 0^* \neg - 2^* \neg - 6^* \neg - 7 \vee$$
$$5^* \neg 3^* \neg 2^* 1^* - 1^* \neg - 2^* - 5 \vee 7^* \neg 3^* \neg 2^* \neg 1^* 0^* \neg - 1^* - 2 \vee$$
$$3^* \neg 1^* - 1^* - 3^* \neg - 4^* - 7 \vee \neg 5^* \neg 4^* 2^* \neg 1^* \neg 0^* - 8 \vee$$
$$6^* \neg 2^* \neg 1^* 0^* \neg - 2^* \neg - 3 \vee 7^* 2^* \neg - 1^* \neg - 2 \vee$$



■ ***Рис. 1.*** *Выявление скрытого смысла по варианту 1*

■ ***Таблица.*** *Выявление смысла исходных сообщений разными группами пользователей (с функциями F_i)*

Фраза, включенная в передаваемое сообщение	Солдат с F_1 читает	Офицер с F_2 читает	Генерал с F_3 читает
Сейчас хорошая погода Good weather now 474F4F44574 54154484552 4E4F572020	Оставаться на месте To remain on a place 544F52454D41 494E4F4E4150 4C414345	Готовь оружие Prepare the weapon 50524550415245 54484557454150 4F4E	Позвонить командующему Call to commander 43414C4C54484543 4F4D4D414E444552
Пора обедать Time tohave dinner 54494D4554 4F4841564544 494E4E4552	Взять противогазы To take gas masks 544F54414B4547 41534D41534B53 2020	Проверить охрану To check up guard 544F434845434B 55504755415244 2020	Завтра наступление Tomorrow approach 544F4D4F52524F57 415050524F414348
Полная луна The complete moon 544845434F4D504C 4554454D4F4F4E20	Отбой через час Tosleep in one hour 544F534C454550 494E4F4E45484F 5552	Танцы до утра Dance till morning 44414E43455449 4C4C4D4F524E 494E47	Полезная встреча Useful meeting 55534546554C4D45 4554494E47202020

4*3*]2*0]5*0*]–1*–2*–3v8*2*0*–5*–7v
8*]5*1*0*–2*–8v4*]3*1*0*–1v
]3*]2*]0*–1*]–3*]–4*–5v2*]0*]–1*]–2*]–3*–8v
]2*0*]–1*]–5*–9v9*]3*2*]0*–1*]–3*–4v
3*2*1*–4*]–6v0*]–1*–2*]–3*]–4*–8v]6*2*]1*0*–1v
]4*3*2*]1*0v7*1*]0*]–2*]–5*–6v]6*2*]1*0*]–6*–7v
]4*2*]1*]0*]–1*–2*]–4v]2*1*]0*]–1*–2*]–3*]–5*]–8v
5*]3*1*0*–5v3*]1*0*–1*–7v
]1*]0*–1*–2*–8v]4*3*]–1*–2*]–4v]6*3*]1*0*–1v
5*]1*]0*–1*–2*]–5v3*]2*–1*]–3*–4*–6v
7*4*]3*]0*–1*]–2*–3v5*]1*0*–9v]9*3*]0*]–1*–4v
6*]5*1*0*–1*–2v7*2*]0*]–1v
2*1*]0*–1*]–5v5*]1*]0*]–1*–5v]3*]2*]1*0*–2*]–8v
7*]6*3*1*]–1*]–5*]–6v
]4*1*]0*–1*]–3*–5*]–7*]–8v8*2*0*–2*–4v
0*–2*–3*–4v3*]1*]–1*]–2*]–3v
]5*1*]0*]–1*]–2*]–3*]–4v]3*]0*]–1*–3*]–7v
1*]0*]–1*]–2*]–3*]–4*]–9;

F_2 = 12*]3*]1*]0*]–1*]–2*–3v4*0*]–1*]–2*]–4v
7*3*]2*]0*]–1*–3*]–6v]5*1*]0*–1*]–3*–5v
2*1*0*–1*]–8v]7*2*0*–1*–2v]3*]2*]1*]0*–3*–7v
1*]0*]–1*]–2*–4v]3*]2*1*0*]–1*]–2v]6*3*1*]0*
]–1*–3*]–6*–7v]8*5*]2*1*–1*]–2*–3*–5v
4*]2*]1*0*]–3*]–7v]3*2*1*]0*–2*–6v
2*1*–1*–6*]–7v8*3*]–1*–8v]8*2*]1*0*–1v
5*1*0*]–1*–3v]0*]–1*]–2*–4*–5v
]2*]1*]0*]–1*]–2*]–3*–9v12*]1*0*]–6v
3*]1*0*]–1*–9v]9*]3*2*]0*]–2*]–3*–4v
]8*3*]2*1*]–1*–2v]5*]4*0*–2*–3v
6*]4*]1*]0*]–1*]–3v]5*1*–1*–2*]–7v5*]1*0*–5v
3*]1*–2*–7v]6*]3*]1*]0*]–1*]–2*–3v0*–3*–4*]–5v
6*]4*0*]–1*]–3*]–4*–6v
]4*2*]1*]0*–2*]–3*]–4v]4*3*]0*]–1*–2*]–5v
3*]2*0*–1*]–3*]–6v3*]2*]0*–1*]–3*–4*–6v
7*4*]3*]0*–1*]–2*–3v3*2*]1*]0*–2v
2*1*–2*–4*–6*]–7v9*8*]5*1*–2v]7*1*0*–1*–6v
1*]0*]–1*–4*–5v5*2*]1*0*]–2v]5*]1*0*–2*–4*]–6v
3*]1*–3*]–5*–13v]6*4*]1*–2*]–9v6*3*]2*0*]–1v
5*]1*]0*–1*–2*]–8v4*0*–1*–2v]5*1*]–1*–2*–3v
]5*1*]–1*]–2*]–3*]–4v]3*]2*]1*–2*]–6;

F_3 = 0*]–1*–2*]–4*]–6v]8*]4*2*]1*]0*–2*]–3v
]4*1*]–1*–2*]–3*]–4*]–12v
]13*3*]2*]–1*–2*]–4v]3*2*]0*]–1*]–2*–3v
]7*]1*0*]–2*]–7v]6*]3*2*0*–2*]–3v5*0*]–1*]–3v
4*3*1*]–1*]–6v2*1*0*–1*]–8v3*]2*1*0*]–1v
]1*0*–1*]–2*–4v]3*]2*]1*–1*–3*–7v
]4*2*]–1*]–2*–4v]1*]0*]–1*–2*–3*]–4v
]3*2*0*]–1*–4*–6v]7*4*]1*0*]–1*–2*]–3v
6*]2*0*]–2*]–3v]7*3*1*]0*]–1*]–2*]–6*–7v
6*3*]0*–1*–2*]–3v1*0*–1*–6*]–7v
1*0*–1*–2*]–8v8*]4*0*–1*–3v10*]0*–1*–4*–6v
]5*3*2*–3*–4v3*]1*0*]–5*–9v
]9*]3*2*]0*–1*]–3*–4v3*1*]0*]–1*]–2*–5v
1*–1*]–2*]–3*–5*–7v6*]1*–1*–3*–7v
3*1*–2*]–4*]–5v]4*]2*0*–2*–3v3*]1*]0*–2*–7v
4*]2*]1*]–1*]–4v2*]0*]–1*]–2*]–3*]–6v
2*]1*]–1*]–4*–5*]–7v]2*1*]0*]–1*–2*]–3*]–5*]–8v
5*]3*0*–5v6*]2*]1*0*–1v2*]0*–1*–2*–8v
]5*2*]0*–1*–2*]–5v5*1*]0*–1*]–2*–8v]5*]3*]1*
–1*]–2*–3*]–4v2*–1*–2*]–3*]–4v
]2*0*–2*–3*]–5v6*]1*0*]–1*–11v]8*3*1*0*]–1*–5v
]7*2*1*0*–1v]3*]2*0*–2*–3v
]3*1*]0*]–1*]–2*–6v5*]3*]2*1*–5*]–7v
4*]2*]1*0*]–8v]6*]1*]0*–2*–4*–8v
]3*]1*0*]–2*]–5*–13v4*]2*]1*–1*]–3*–14v
7*1*]–1*–3*]–7v]3*2*1*0*–3*]–7v
7*]2*0*–1*]–2*]–3v]5*1*]–1*–2*–3v
]3*]2*–1*–4*–5v4*]2*]1*]0*]–1*]–2.

Булевы преобразования выполняются очень быстро и практически не влияют на скорость передачи сообщений.

Многократно вложенный смысл проиллюстрируем примером. Возьмем несколько сообщений, различных по смыслу:

A. Вперед в атаку!
B. Держим оборону.
C. Какие будут распоряжения?
D. Срочно назад!

В английском варианте и кодах ASCII это выглядит так:

A. Forwards in attack!
ASCII: 464F525741524453494E41545441434B.
B. We keep a defense.
ASCII: 57454B4545504144454645E53452020.
C. What orders will be?
ASCII: 574841544F524445525257494C4C4245.
D. Urgently we back!
ASCII: 555247454E544C5957454241434B2020.

Функция F, преобразующая $F(A) = B$; $F(B) = C$; $F(C) = D$, имеет вид

F = 1*0*⌉–1*–2*⌉–7∨⌉8*0*–1*–2*⌉–3∨0*–3*–4∨
⌉4*⌉2*⌉0*⌉–1*–2*⌉–3*–4∨3*⌉1*⌉0*–2*–5∨
⌉3*⌉1*0*⌉–3*⌉–4*–5∨4*⌉1*⌉0*⌉–1*⌉–2∨
5*⌉–1*⌉–2*⌉–3*–5*⌉–8∨3*2*1*0*⌉–6∨1*0*–1*–2*⌉–∨
⌉8*3*0*–4∨⌉6*⌉0*⌉–1*–2*⌉–3*–4*–7∨
⌉7*⌉3*⌉2*1*⌉–1*–3*–4*⌉–5∨⌉6*4*⌉2*⌉1*⌉–1*⌉–2∨
4*⌉2*⌉1*⌉–1*⌉–2*⌉–3*⌉–7∨4*2*⌉1*⌉–1*⌉–4*⌉–5∨
7*5*3*1*⌉0*–3∨5*⌉–1*–2*–4∨
6*⌉3*1*–2*⌉–5*⌉–7∨5*⌉2*0*⌉–1*⌉–6*–7∨
4*⌉3*⌉1*⌉0*–1*⌉–3*⌉–7∨⌉3*2*⌉1*0*⌉–1*–3∨
⌉2*1*⌉–1*–2*⌉–3*–4*⌉–5*⌉–6∨6*–1*–3*–8∨
4*3*⌉2*0*⌉–6∨⌉1*⌉0*⌉–1*–2*–3∨5*3*⌉1*⌉–3*⌉–5∨
7*1*0*–1∨⌉5*0*–1*–2*⌉–3∨1*–5*–6*–7∨
6*4*2*⌉0*⌉–6∨⌉7*⌉1*0*⌉–1*⌉–2*–4*–6∨
6*⌉3*⌉2*0*–2*–4∨5*1*–1*⌉–2*–5*–7∨
⌉6*2*⌉1*0*!–1*–2*–4∨⌉5*⌉2*⌉1*⌉0*⌉–1*–3∨
⌉2*⌉1*⌉–1*⌉–2*⌉–3*–6∨4*⌉3*0*⌉–1*–2*⌉–4∨
2*⌉1*⌉0*⌉–1*–2*⌉–4*⌉–5*–8∨
⌉5*3*1*–1*⌉–3*⌉–7∨3*2*⌉1*⌉0*–2∨
3*⌉1*0*–1*⌉–2∨4*2*⌉1*⌉–1*⌉–4*–6∨
4*2*⌉1*0*⌉–6*–7∨6*5*2∨⌉3*1*0*⌉–2*–7∨
4*⌉1*⌉0*–1*–2∨5*3*2*⌉1*–6∨4*⌉3*2*1*⌉0*–2∨
⌉5*3*⌉0*⌉–1*–3*–5∨
⌉5*1*⌉–1*⌉–2*⌉–3*⌉–4*⌉–5∨⌉3*⌉2*⌉1*⌉0*⌉–1*⌉–2*
⌉–3*⌉–4∨⌉3*1*0*⌉–2*⌉–3*⌉–6∨0*–2*–3*–9∨
7*⌉1*–1*–3*–5*–6∨⌉7*2*1*0*–2∨
⌉3*2*⌉1*0*⌉–4*⌉–6∨⌉5*0*⌉–1*–2*–3∨
⌉3*2*⌉1*⌉0*–1*⌉–3*–4*⌉–8∨⌉6*1*0*–1*⌉–2*⌉–3∨
7*5*1*–1*⌉–2*–3*–5∨5*3*⌉2*⌉1*⌉0*–1*⌉–4*–7∨
⌉1*0*⌉–1*–4*–6*⌉–7∨⌉8*⌉7*⌉3*2*0*–2*⌉–4∨
⌉6*⌉5*⌉4*⌉2*⌉1*⌉0*–3*–5∨⌉4*⌉1*⌉0*–2*–4*–6∨
⌉6*2*0*⌉–1*⌉–2*⌉–3*⌉–4∨4*⌉3*⌉2*⌉1*⌉0*–3∨
4*⌉1*–3*–4.

Следует заметить, что никто из пользователей, имея в руках преобразователь с функцией F, не получает информации о том, когда и какая фраза, переданная центром, в какую фразу будет преобразована с помощью этой функции. Этой информацией владеет только центр.

Вариант 2. В рассмотренном примере (вариант 1) можно использовать произвольный, но ограниченный набор фраз, которые каждый абонент воспринимает по-своему. В варианте 2 система построена таким образом, чтобы передаваемый текст был совершенно произвольным без каких-либо ограничений. Это можно сделать с использованием так называемых обратимых булевых функций, которые лежат в классе линейных [1], т. е. таких, что уравнение $F(X) = B$ всегда разрешимо при любых B относительно X. Тогда, снабдив каждую группу абонентов своей функцией F, можно передавать по каналу некоторый текст (X), а каждый абонент, используя свою функцию, будет читать только то, что предназначено центром непосредственно ему. Остальные абоненты будут извлекать из передаваемого текста бессмысленный набор букв и символов (рис. 2)

Если центр хочет передать конкретному пользователю некоторое сообщение B, он берет функцию F этого пользователя и вычисляет двоичную последовательность $F^{-1}(B) = A$, затем передает A в зашифрованном виде по каналу связи. После расшифрования только тот получатель, кто владеет преобразователем с этой конкретной функцией, может восстановить сообщение: $F(A) = B$. Остальные пользователи, пытаясь восстановить сообщение с помощью своих функций, получают бессмысленную последовательность символов. Не представляет трудности нахождение таких функций для побуквенного преобразования сообщения, однако в этом случае все сводится к обычному шифру замены, который легко вскрывается. Лучше выполнять преобразования сразу нескольких символов (например, 10–20 символов текста), а еще лучше — блоками битов, не кратных длине символов (т. е. не кратных четырем битам). Однако в приводимых примерах для большей наглядности длины блоков выбираются кратными четырем битам. Операции прямого и обратного преобразования выполняются по mod 2.

Пусть требуется передать конкретному получателю фразу: We act in a campaign tomorrow. В кодах ASCII эта фраза с учетом пробелов между словами будет выглядеть так: 57452041435420494E 204120434F4D504149474E20544F4D4F52524F5720.

■ ***Рис. 2.*** *Выявление скрытого смысла по варианту 2*

Разобьем сообщение на блоки по 20 бит, что будет соответствовать 2,5 буквам в каждом блоке:

57452 | 04143 | 54204 | 94E20 | 41204 | 34F4D |

50414 | 9474E | 20544 | F4D4F | 52524 | F5720.

Возьмем четырехразрядное обратимое преобразование с добавленным пятым столбцом и пятой строкой вида

$$F_1 = \begin{pmatrix} 01100 \\ 11111 \\ 01111 \\ 00111 \\ 00001 \end{pmatrix}; \quad F_1^{-1} = \begin{pmatrix} 01100 \\ 00110 \\ 10110 \\ 10101 \\ 00001 \end{pmatrix}.$$

Произведение этих матриц дает единичную матрицу: $F_1 \times F_1^{-1} = I$.

Правый столбец и нижняя строка этих матриц позволяют учитывать инверсию элементов при умножении по mod 2.

Умножим матрицу F_1^{-1} на фрагменты текста по четыре блока каждый. Последний блок состоит из всех единиц, т. е. может быть записан как FFFFF:

$$\begin{pmatrix} 01100 \\ 00110 \\ 10110 \\ 10101 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 57452 \\ 04143 \\ 54204 \\ 94E20 \\ FFFFF \end{pmatrix} = \begin{pmatrix} 50347 \\ C0C24 \\ 97876 \\ FC9A9 \\ FFFFF \end{pmatrix};$$

$$\begin{pmatrix} 01100 \\ 00110 \\ 10110 \\ 10101 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 41204 \\ 34F4D \\ 50414 \\ 9474E \\ FFFFF \end{pmatrix} = \begin{pmatrix} 64B59 \\ C435A \\ 8515E \\ EE9EF \\ FFFFF \end{pmatrix};$$

$$\begin{pmatrix} 01100 \\ 00110 \\ 10110 \\ 10101 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 20544 \\ F4D4F \\ 52524 \\ F5720 \\ FFFFF \end{pmatrix} = \begin{pmatrix} A686B \\ A7204 \\ 87740 \\ 8DF9F \\ FFFFF \end{pmatrix}.$$

В результате получим закодированный текст: 50347C0C2497876FC9A964B59C435A8515EEE9EFA 686BA7204877408DF9E.

Только некоторые сочетания символов соответствуют в этом тексте каким-то использованным буквам кода ASCII, т. е. текст полностью бессмысленный. Однако если полученное сообщение преобразовать функцией F_1, то будет восстановлен передаваемый текст:

$$\begin{pmatrix} 01100 \\ 11111 \\ 01111 \\ 00111 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 50347 \\ C0C24 \\ 97876 \\ FC9A9 \\ FFFFF \end{pmatrix} = \begin{pmatrix} 57452 \\ 04143 \\ 54204 \\ 94E20 \\ FFFFF \end{pmatrix};$$

$$\begin{pmatrix} 01100 \\ 11111 \\ 01111 \\ 00111 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 64B59 \\ C435A \\ 8515E \\ EE9EF \\ FFFFF \end{pmatrix} = \begin{pmatrix} 41204 \\ 34F4D \\ 50414 \\ 9474E \\ FFFFF \end{pmatrix};$$

$$\begin{pmatrix} 01100 \\ 11111 \\ 01111 \\ 00111 \\ 00001 \end{pmatrix} \times \begin{pmatrix} A686B \\ A7204 \\ 87740 \\ 8DF9F \\ FFFFF \end{pmatrix} = \begin{pmatrix} 20544 \\ F4D4F \\ 52524 \\ F5720 \\ FFFFF \end{pmatrix}.$$

Действительно, результат преобразования имеет вид: 574520414354204 94E204120434F4D5041 49474E20544F4D4F52524F5720, что полностью совпадает с исходным сообщением: We act in a campaign tomorrow.

Возьмем теперь в качестве длины блока 12 бит сообщения, т. е. каждый блок будет соответствовать 1,5 букве. Выполним то же обратное преобразование с первыми четырьмя блоками:

$$\begin{pmatrix} 01100 \\ 00110 \\ 10110 \\ 10101 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 574 \\ 520 \\ 414 \\ 352 \\ FFF \end{pmatrix} = \begin{pmatrix} 134 \\ 746 \\ 232 \\ E9F \\ FFF \end{pmatrix}.$$

Полученный фрагмент является отличным от предыдущего случая и также полностью бессмысленным. Однако после обработки его прямым преобразованием F получаем

$$\begin{pmatrix} 01100 \\ 11111 \\ 01111 \\ 00111 \\ 00001 \end{pmatrix} \times \begin{pmatrix} 134 \\ 746 \\ 232 \\ E9F \\ FFF \end{pmatrix} = \begin{pmatrix} 574 \\ 520 \\ 414 \\ 352 \\ FFF \end{pmatrix},$$

т. е. 574520414352. Это и есть начало передаваемой фразы: We act ... .

Заметим, что для реализации варианта 2 подходит любое обратимое преобразование

$$F: \{0,1\}^n \to \{0,1\}^n.$$

Рассмотрим множество матриц размера (n, n) с элементами и операциями из поля Галуа GF(2), т. е. будем выполнять операции сложения и умножения по mod 2. Множество таких матриц конечно и равно 2^{n^2}. Исключим из этого множества все вырожденные матрицы с определителем, равным 0 [в поле GF(2)]. Оставшееся множество матриц с определителем, равным 1, образует некоммутативную группу в поле GF(2). Для некоторой матрицы F найдем обратную матрицу F^{-1} такую, что $F \times F^{-1} = I$ — единичная матрица. Пусть сообщение, которое нужно передать, есть B. Найдем произведение $F^{-1} \times B = A$. Эту строку зашифруем и передадим по каналу связи. После расшифрования толь-

ко пользователь, который владеет матрицей преобразования F, может получить $FA = B$. Остальные будут читать бессмысленный текст. Если разных пользователей снабдить различными функциями F_i, то можно обеспечить адресную передачу сообщений.

Если имеется некоторое множество преобразований одинаковой размерности n: $F_1, F_2, \ldots, F_k$ и соответствующие им обратные преобразования F_1^{-1}, $F_2^{-1}, \ldots, F_k^{-1}$, то можно получить прямое преобразование $F = F_1, F_2, \ldots, F_k$, а обратное $F^{-1} = F_k^{-1}, F_{k-1}^{-1}, \ldots, F_1^{-1}$, так как множество матриц F образует некоммутативную группу по умножению с определителем, равным 1, над полем Галуа GF(2). Это позволяет увеличить количество различных преобразований и снабжать пользователей сети разными функциями F_i для адресной передачи сообщений.

Можно использовать матрицы с определителем, равным 1, произвольной различной размерности, меняя при этом и размеры преобразуемых блоков. Например, можно взять такие матрицы:

$$F_1 = \begin{pmatrix} 101 \\ 010 \\ 011 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 1101 \\ 0101 \\ 1110 \\ 1100 \end{pmatrix}, \quad F_3 = \begin{pmatrix} 11011 \\ 01100 \\ 10001 \\ 01110 \\ 01011 \end{pmatrix}.$$

Соответственно, обратные матрицы будут иметь вид

$$F_1^{-1} = \begin{pmatrix} 111 \\ 010 \\ 011 \end{pmatrix}, \quad F_2^{-1} = \begin{pmatrix} 1100 \\ 1101 \\ 0011 \\ 1001 \end{pmatrix}, \quad F_3^{-1} = \begin{pmatrix} 10001 \\ 11110 \\ 10110 \\ 01010 \\ 10101 \end{pmatrix}.$$

Любая фраза, состоящая, например, из 60 символов, может быть последовательно преобразована матрицей F_1 (длина блока, например, 20 символов), затем F_2 (длина блока, например, 15 символов) и в заключение F_3 (длина блоков, например, 12 символов). Восстановление исходного текста выполняется в обратном порядке, т. е. сначала текст преобразуется матрицей F_3^{-1}, затем F_2^{-1} и в заключение матрицей F_1^{-1}.

Поиск потерянного смысла

Известно, что совокупность степеней любого элемента F_i некоммутативной группы конечного порядка порождает циклическую коммутативную группу некоторого порядка s_i, т. е. $F^{s_i} = I$. В связи с этим, можно изменить передачу по варианту 2. На передающем конце отправитель будет обрабатывать сообщение $F^d \times B = A$, где d — целое, $0 < d < s_i$, а получатель сообщения, имея ту же функцию F, будет выполнять следующее преобразование: $F^{s_i - d} \times A = F^{s_i - d} \times F^d \times B = F^{s_i} \times B = I \times B = B$. При этом сообщение будет восстановлено. Если получатель сообщения не знает величины d, он может последовательно умножать полученную последовательность A слева на F до тех пор, пока не отыщет потерянный смысл — последовательность B.

Рассмотрим, например, матрицу F вида

$$F = \begin{pmatrix} 11001 \\ 01101 \\ 10110 \\ 10011 \\ 00001 \end{pmatrix}.$$

Определитель этой матрицы в поле GF(2) равен 1. Совокупность степеней этой матрицы порождает коммутативную группу порядка 15, т. е. $\{F^0 = I, F^1, F^2, \ldots, F^{14}\}$ — коммутативная группа. Возьмем 14 произвольных сообщений $B_1, B_2, \ldots, B_{14}$ и зашифруем их путем умножения $F^i \times B_i = A_i$ для всех $i = 1, 2, 3, \ldots, 14$. Обозначим $F^3 = P$, тогда последовательность степеней $\{P^0, P^1, P^2, P^3, P^4\}$ тоже образует коммутативную группу, которая является подгруппой исходной коммутативной группы, составленной из степеней матрицы F. Аналогично построим коммутативную подгруппу $F^5 = Q$ и последовательность степеней $\{Q^0, Q^1, Q^2\}$. Снабдим каждую группу пользователей соответствующей функцией: F — самого ответственного получателя, а затем остальных получателей матрицами P и Q. Если теперь зашифровать все 14 сообщений и передать их по каналу связи, то после обработки полученного сообщения соответствующими функциями самый ответственный получатель прочитает все сообщения, следующий — только кратные трем и последний прочитает только сообщения, кратные пяти. Для второго и третьего часть сообщений окажутся потерянными.

Выбирая произвольные матрицы F с определителем, равным 1, в поле GF(2) и находя коммутативные подгруппы, порождаемые некоторыми элементами, можно адресовать сообщения разным пользователям.

Заключение

Для шифрования и расшифрования сообщений можно использовать любые методы, как с открытыми, так и с секретными ключами.

Основываясь на методах обеспечения скрытого смысла по варианту 1, можно организовать передачу с многократно вложенным смыслом. В этом случае каждый пользователь «прочитывая» несколько раз один и тот же фрагмент сообщения с одной функцией или разными функциями F, каждый раз извлекает из него осмысленный текст нового содержания. Возможна комбинация методов по вариантам 1 и 2.

Чтение сообщений со скрытым смыслом можно сравнить с индивидуальными очками, которые выдаются разным пользователям сети. Из расшиф-

рованного открытого текста каждый читает только то, что предназначено лично ему. В случае многократно вложенного смысла при каждом новом прочтении одного и того же фрагмента текста пользователь видит новый текст. Число вложений зависит от выбранной функции и может быть достаточно большим.

Литература

1. **Ерош И. Л.** Булевы функции. Комбинационные схемы. Преобразование двоичных последовательностей: Учебн. пособ. — СПб.: Изд-во СПбГУАП, 2001. — 30 с.
2. **Ерош И. Л.** Разграничение доступа к ресурсам в системах коллективного пользования // Информационно-управляющие системы. — 2003. — № 2–3. — С. 63–66.

ИЗДАТЕЛЬСТВО «ПОЛИТЕХНИКА» ПРЕДСТАВЛЯЕТ

Куприянов М. С., Матюшкин Б. Д.

Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования. — 2-е изд., перераб. и доп. — СПб.: Политехника, 2002. — 592 с.: ил.

Книга содержит три части. Первая часть «Процессоры цифровой обработки сигналов» посвящена архитектуре и особенностям организации DSP. Во второй части «Алгоритмы цифровой обработки сигналов» рассматриваются основы теории дискретных систем, методы анализа эффектов квантования сигналов при реализации алгоритмов обработки на DSP, базовые алгоритмы ЦОС и их реализация на DSP. Третья часть «Инструментальные средства проектирования систем ЦОС» содержит описание программных и аппаратных средств, используемых для решения задач проектирования и входящих в стартовый комплекс разработчика систем ЦОС. В приложении приведена система команд семейств DSP5600x и DSP5630x.

Книга рассчитана на инженерно-технических работников, занимающихся проектированием систем ЦОС, а также студентов соответствующих специальностей технических университетов.

Информационно-управляющие системы для подвижных объектов. Семинары ASK Lab 2001 / Под общ. ред. М. Б. Сергеева. — СПб.: Политехника, 2002. — 234 с.: ил.

В книге представлены статьи, посвященные актуальным проблемам в области разработки информационно-управляющих систем для подвижных объктов, вопросам их надежности, алгоритмического и аппаратного обеспечения, защиты информационных каналов.

Книга ориентирована на научных и инженерно-технических работников, специалистов в области встраиваемых систем управления не только авиационных комплексов, но и наземных подвижных дистанционно управляемых объектов различного назначения.