УДК 004.056

В.М. Рудницький, Д.А. Жиляєв

Черкаський державний технологічний університет, Черкаси

ПОБУДОВА ОБЕРНЕНИХ ФУНКЦІЙ ДЛЯ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

В роботі дано означення логічної функції кодування при роботі з байтами повідомлення, розглянутий алгоритм кодування повідомлення з використанням логічної функції. Показана можливість еквівалентного подання логічної функції у вигляді певною мірою модифікованої перестановки, наведена таблиця функцій кодування-декодування. Показано правило одержання оберненої функції з функції кодування на основі використання теорії перестановок що базується на знаходження перестановки, оберненої до даної з деякими особливостями.

Ключові слова: *логічна функція, кодування, перестановка.*

Вступ

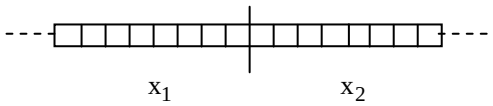
Постановка проблеми. Сучасні системи захисту інформації в основному базуються на використанні криптографічного додавання до повідомлення яке захищається деякої випадкової послідовності, що в загальному випадку можна подати у вигляді формули $X_i^* = X_i \oplus Y_i$, де X_i – і-й розряд повідомлення, яке захищається; Y_i – і-й розряд випадкової послідовності тієї ж довжини; $\oplus$ – операція додавання по модулю 2. Вибір в загальному випадку послідовності Y зазвичай значно ускладнює алгоритм шифрування, який, як правило, повинен реалізовуватися максимально швидко. Одним із напрямків дослідження можливості вибору допоміжної послідовності Y є виділення цієї послідовності безпосередньо з вихідного повідомлення, тобто використання для кодування власне частини даного повідомлення. Вдалий вибір правила формування допоміжної послідовності дозволить значно спростити алгоритм шифрування повідомлення.

Аналіз останніх досліджень і публікацій. Серед останніх досліджень і публікацій варто насамперед виділити [2], де був запропонований загальний алгоритм кодування повідомлення з допомогою логічних функцій. в [2] Був презентований алгоритм визначення спеціальних логічних функцій, а в [1] наведені результати синтезу таких функцій, доведено коректність процедур кодування і декодування. Проте, в розглянутих роботах перетворення здійснювалися над бітом повідомлення, не розглядалася можливість визначення функції декодування по вигляду функції кодування.

Мета статті. Вказати критерії визначення функції декодування при роботі з байтами повідомлення.

Означення логічної функції для кодування повідомлення

Відомо, що обчислювальні пристрої найбільш ефективно працюють з байтами повідомлення [5]. Розглянемо два сусідні байти повідомлення X:



Означення 1. Вектор-функцією $\bar{F}$ будемо називати деяке перетворення двох сусідніх байтів і позначимо $\bar{F}: \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix}$ – деяке відображення множини (повідомлення) самої на себе.

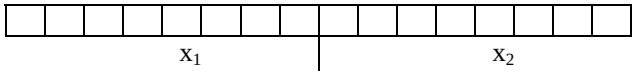
Перетворення можна записати у вигляді системи:

$$\begin{cases} x_1^* = a_{11}x_1 \oplus a_{12}x_2 \oplus b_1, \\ x_2^* = a_{21}x_1 \oplus a_{22}x_2 \oplus b_2. \end{cases}$$

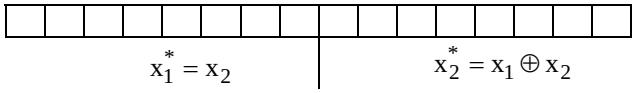
Щоб забезпечити невиродженість перетворення накладаються обмеження: $a_{ij} = \{0; 1\}$, $b_i = \{0; 1\}$, $i = \overline{1,2}$, $j = \overline{1,2}$, $a_{11} \cdot a_{22} - a_{12} \cdot a_{21} \neq 0$.

Приклад дії перетворення:

Нехай $\bar{F}: \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_2 \\ x_1 \oplus x_2 \end{pmatrix}$, тоді фрагмент деякого повідомлення X: …



перетвориться у X^*: …



Разом з перетворенням $\bar{F}$ можна розглядати наступне рівносильне йому перетворення [1]:

Нехай $\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$ – таблиця можливих значень двох логічних функцій. Можна покласти

$$f_1 = \begin{pmatrix}0\\0\\1\\1\end{pmatrix} \text{ і } f_2 = \begin{pmatrix}0\\1\\0\\1\end{pmatrix}.$$

Тоді, якщо вважати, що вихідна таблиця $\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}$ відповідає тотожному перетворенню $\overline{F}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}$ то будь-яку функцію $\overline{F}$ можна подати у вигляді перестановки вихідної таблиці.

Зокрема, в наведеному вище прикладі те саме перетворення можна записати у вигляді $\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\0&1\\1&0\end{pmatrix}$. Враховуючи тотожність вказаних перетворень, можна легко обчислити кількість можливих функцій $\overline{F}$. Так як кожна функція являє собою перестановку чотирьох пар елементів, то кількість таких перестановок дорівнює 4! = 24.

Використавши викладене в [1] і [2] можна подати функції кодування-декодування у вигляді таблиці (табл. 1)

Таблиця 1

Функції кодування-декодування

Кодування		Декодування	
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}$	$\overline{F}_1:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}$	$\overline{F}_1:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_2:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_2:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&1\\1&0\end{pmatrix}$	$\overline{F}_3:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&1\\1&0\end{pmatrix}$	$\overline{F}_3:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\1&1\\0&1\end{pmatrix}$	$\overline{F}_4:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_5:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_5:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\1&1\\0&1\end{pmatrix}$	$\overline{F}_4:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&0\\0&1\end{pmatrix}$	$\overline{F}_6:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&0\\0&1\end{pmatrix}$	$\overline{F}_6:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_7:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_7:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&1\\0&0\end{pmatrix}$	$\overline{F}_8:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&1\\0&0\end{pmatrix}$	$\overline{F}_8:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\0&0\\1&1\end{pmatrix}$	$\overline{F}_9:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_{18}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\end{pmatrix}$

Кодування		Декодування	
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\0&0\\1&1\end{pmatrix}$	$\overline{F}_{10}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\0&0\\1&1\end{pmatrix}$	$\overline{F}_{10}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{11}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{15}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{12}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_{23}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{13}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\1&1\\0&1\end{pmatrix}$	$\overline{F}_{20}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{14}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{14}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{15}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{11}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{16}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_{21}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&0\\1&1\end{pmatrix}$	$\overline{F}_{17}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&0\\1&1\end{pmatrix}$	$\overline{F}_{17}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_{18}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\0&0\\1&1\end{pmatrix}$	$\overline{F}_9:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&1\\1&0\end{pmatrix}$	$\overline{F}_{19}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&1\\1&0\end{pmatrix}$	$\overline{F}_{19}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\1&1\\0&1\end{pmatrix}$	$\overline{F}_{20}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{13}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_{21}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{16}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&1\\0&0\end{pmatrix}$	$\overline{F}_{22}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&0\\0&1\end{pmatrix}$	$\overline{F}_{24}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_{23}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{12}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&0\\0&1\end{pmatrix}$	$\overline{F}_{24}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&1\\0&0\end{pmatrix}$	$\overline{F}_{22}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\end{pmatrix}$

Очевидно, що множина логічних функцій кодування-декодування замкнена, тобто для кожної з 24 логічних функцій кодування обов'язково серед цих самих функцій є і функція декодування, до того ж тільки одна. Постає питання про можливість безпосереднього вибору функції декодування для кожної функції кодування по її вигляду і взагалі про критерії, яким повинна задовольняти функція, щоб бути функцією декодування для деякої функції кодування.

При розробці критеріїв для визначення функції декодування можна реалізовувати два підходи: 1) виходячи з означення вектор-функції як деякого відображення множини самої на себе; 2) розглядаючи функцію $\overline{F}$ як перестановку вихідної таблиці.

При реалізації першого підходу ми працюємо власне з деяким оператором А, який не є лінійним, і тому не маючи можливості використовувати відомі властивості лінійних операторів маємо суттєву проблему з формулюванням будь-яких критеріїв. Тому варто реалізувати другий підхід.

Розглянемо вже наведене нами перетворення $\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\0&1\\1&0\end{pmatrix}$, яке відповідає, зокрема, вектор-функції $\overline{F}_{13}$. Позначимо елементи таблиці через a_i. тобто $\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\Leftrightarrow\begin{pmatrix}a_1\\a_2\\a_3\\a_4\end{pmatrix}$. Таким чином розглянуте перетворення запишеться у вигляді $\begin{pmatrix}a_1\\a_2\\a_3\\a_4\end{pmatrix}\to\begin{pmatrix}a_1\\a_4\\a_2\\a_3\end{pmatrix}$, або однією перестановкою $\begin{pmatrix}a_1&a_1\\a_2&a_4\\a_3&a_2\\a_4&a_3\end{pmatrix}$, транспонувавши яку одержимо перестановку в традиційному вигляді [3]: $\begin{pmatrix}a_1&a_2&a_3&a_4\\a_1&a_4&a_2&a_3\end{pmatrix}$. Таким чином, кодування повідомлення задається з допомогою шифру перестановки. При використанні шифрів перестановки [3] процедура декодування повідомлення полягає в повторному кодуванні вже одержаного зашифрованого повідомлення деякою новою перестановкою, яка називаються оберненою.

Використавши правило знаходження оберненої перестановки [3] одержимо $\begin{pmatrix}a_1&a_2&a_3&a_4\\a_1&a_4&a_2&a_3\end{pmatrix}^{-1}=\begin{pmatrix}a_1&a_2&a_3&a_4\\a_1&a_3&a_4&a_2\end{pmatrix}$, таким чином перестановка $\begin{pmatrix}a_1&a_2&a_3&a_4\\a_1&a_3&a_4&a_2\end{pmatrix}$ і буде тією, яка відповідатиме функції декодування. Виконавши попередні дії в зворотному порядку будемо мати:

$$\begin{pmatrix}a_1&a_2&a_3&a_4\\a_1&a_3&a_4&a_2\end{pmatrix}\Rightarrow\begin{pmatrix}a_1&a_1\\a_2&a_3\\a_3&a_4\\a_4&a_2\end{pmatrix}\Rightarrow\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\1&1\\0&1\end{pmatrix},$$

тобто функцією декодування для $\overline{F}_{13}$ буде функція $\overline{F}_{20}$ що підтверджується практичною перевіркою. Повністю аналогічно функції декодування визначаються для всіх $\overline{F}_i$, $i=\overline{1,24}$.

Висновкия

В статті подано у вигляді таблиці в узагальненому вигляді перелік дворозрядних логічних функцій кодування байтів повідомлення разом з їх представленням у вигляді перестановок. Сформульовано критерій знаходження функції декодування для кожної логічної функції кодування повідомлення. Серед подальших напрямків дослідження є, зокрема, узагальнення критерію вибору функції декодування для більших розрядів.

Списокялітература

1. Бабенко В.Г. Алгоритми вибору логічних функцій для криптографії / В.Г. Бабенко, Т.В. Дахно, В.М. Рудницький // 2-а міжнародна наукова конференція «Сучасні інформаційні системи. Проблеми та тенденції розвитку». – Х.: ХНУРЕ, 2007. – С. 423-424.

2. Рудницький В.М. Синтез математичних моделей пристроїв декодування / В.М. Рудницький, В.Г. Бабенко // Системи обробки інформації: зб. наук. пр. – Х.: Х УПС, 2009. – Вип. 2 (76). – С. 124-128.

3. Жиляєв Д.А. Особливості захисту технологічної інформації на основі перестановок / Д.А. Жиляєв // Вісник інженерної академії України. – К., 2007. – Вип. 3-4. –. С. 37-41.

Надійшла до редколегії 15.06.2009

Рецензент: д-р техн. наук, проф. І.В. Чумаченко, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.

ПОСТРОЕНИЕяОБРАТНЫХяФУНКЦИЙяДЛяСИСТЕМяЗАЩИТЫяИНФОРМАЦИИя

В.Н. Рудницкий, Д.А. Жиляев

В работе дано определение логической функции кодирования при работе с байтами сообщения, рассмотрен алгоритм кодирования, показана возможность эквивалентного представления логической функции в виде перестановки, приведена таблица функций кодирования-декодирования. Показано правило получения обратной функции.

Ключевые слова: *логическая функция, кодирование, перестановка.*

CONSTRUCTIONяOFяREVERSEяFUNCTIONSяFORяTHEяSYSTEMSяOFяPROTECTIONяTOяINFORMATIONя

V.N. Rudnitsky, D.A. Zhylyaev

Determination of Boolean function of encoding is in-process given during work with the bytes of report, the algorithm of encoding is considered, possibility of equivalent presentation of Boolean function is rotined as transposition, the table of functions of encoding-decoding is resulted. The rule of receipt of reverse function is rotined.

Keywords: *Boolean function, encoding, transposition.*