УДК 004.056

В.М. Рудницький, В.Г. Бабенко, Д.А. Жиляєв

Черкаський державний технологічний університет, Черкаси

АЛГЕБРАЇЧНА СТРУКТУРА МНОЖИНИ ЛОГІЧНИХ ОПЕРАЦІЙ КОДУВАННЯ

В роботі дано означення логічної операції кодування при роботі з байтами повідомлення, розглянутий алгоритм кодування повідомлення з використанням логічної операції. Показана можливість еквівалентного подання логічної операції у вигляді перестановки і доведено, що множина дворозрядних логічних операцій з введеною операцією композиція є групою, ізоморфною групі S^4.

Ключові слова: *логічна операція, кодування, перестановка, група.*

Вступ

Постановка проблеми. Проблема підвищення швидкодії систем захисту інформації є актуальною, тому що безпосередньо випливає із тенденції збільшення обсягів інформації, що передається та обробляється в інформаційних системах. Одним із варіантів рішення даної проблеми є застосування багаторозрядних операцій криптографічного кодування під управлінням криптосистем.

Аналіз останніх досліджень і публікацій. Серед останніх досліджень і публікацій варто насамперед виділити [1], де був запропонований загальний алгоритм кодування повідомлення з допомогою логічних функцій. В [3, 4] був презентований алгоритм визначення спеціальних логічних функцій, а в [5, 6] наведені результати синтезу таких функцій, доведено коректність процедур кодування і декодування.

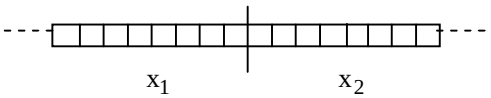
Проте, в розглянутих роботах перетворення здійснювалися над бітом повідомлення, не розглядалася можливість визначення функції декодування по вигляду функції кодування.

Слід відмітити, що дані про існування робіт, де наводиться математичне обґрунтування припущення, що синтезовані функції, які застосовуються в якості операцій на етапі криптографічного кодування, утворюють алгебраїчну групу, відсутні.

Основною **метою даної статті** є доведення того факту, що множина синтезованих логічних операцій криптографічного кодування та задана множина операцій утворюють алгебраїчну систему, а також математичне обґрунтування того припущення, що множина дворозрядних логічних операцій з введеною операцією композиція належить до такого класу алгебраїчних систем як група.

Означення логічної операції кодування повідомлення

Відомо, що обчислювальні пристрої найбільш ефективно працюють з байтами повідомлення. 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: …

0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	0
x_1								x_2							

перетвориться у X^*: …

1	0	1	0	0	1	1	0	1	1	0	1	1	1	1	1
$x_1^* = x_2$								$x_2^* = x_1 \oplus x_2$							

Разом з перетворенням $\bar{F}$ можна розглядати наступне рівносильне йому перетворення [3]:

Якщо таблиця можливих значень двох логічних операцій $T = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$, то тоді $f_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$ і $f_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$.

Тоді, якщо вважати, що вихідна таблиця T відповідає тотожному перетворенню $\bar{F}:\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

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 [3] можна поставити у відповідність кожній логічній операції перестановку елементів повідомлення таким чином:

Перестановка елементів повідомлення	Логічна операція
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}$	$\overline{F}_1:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_2:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&1\\1&0\end{pmatrix}$	$\overline{F}_3:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\1&1\\0&1\end{pmatrix}$	$\overline{F}_4:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_5:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&0\\0&1\end{pmatrix}$	$\overline{F}_6:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_7:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&1\\0&0\end{pmatrix}$	$\overline{F}_8:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\0&0\\1&1\end{pmatrix}$	$\overline{F}_9:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\0&0\\1&1\end{pmatrix}$	$\overline{F}_{10}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&0\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{11}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&1\\1&1\\0&0\end{pmatrix}$	$\overline{F}_{12}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{13}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&1\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{14}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\0&1\\1&0\end{pmatrix}$	$\overline{F}_{15}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&0\\1&0\\0&1\end{pmatrix}$	$\overline{F}_{16}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_2\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\0&0\\1&0\\1&1\end{pmatrix}$	$\overline{F}_{17}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\0&0\\0&1\\1&1\end{pmatrix}$	$\overline{F}_{18}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\0&1\\1&1\\1&0\end{pmatrix}$	$\overline{F}_{19}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&0\\1&0\\1&1\\0&1\end{pmatrix}$	$\overline{F}_{20}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}0&1\\1&1\\1&0\\0&0\end{pmatrix}$	$\overline{F}_{21}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&0\\1&1\\0&1\\0&0\end{pmatrix}$	$\overline{F}_{22}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\0&1\\0&0\\1&0\end{pmatrix}$	$\overline{F}_{23}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus x_2\oplus1\\x_1\oplus1\end{pmatrix}$
$\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\to\begin{pmatrix}1&1\\1&0\\0&0\\0&1\end{pmatrix}$	$\overline{F}_{24}:\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\oplus1\\x_1\oplus x_2\oplus1\end{pmatrix}$

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, логічній операції $\bar{F}_{13}$. Пронумеруємо елементи таблиці, тобто $\begin{pmatrix}0&0\\0&1\\1&0\\1&1\end{pmatrix}\Leftrightarrow\begin{pmatrix}1\\2\\3\\4\end{pmatrix}$.

Таким чином перетворення запишеться zr $\begin{pmatrix}1\\2\\3\\4\end{pmatrix}\to\begin{pmatrix}1\\4\\2\\3\end{pmatrix}$, або однією перестановкою $\begin{pmatrix}1&1\\2&4\\3&2\\4&3\end{pmatrix}$, транспонувавши яку одержимо перестановку в традиційному вигляді [1]:

$$\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}.$$

Таким чином, очевидно, кожній дворозрядній логічній операції можна поставити у відповідність чотирьохелементну перестановку.

Композиція логічних операцій кодування повідомлення

В множині логічних операцій введемо бінарну алгебраїчну операцію:

Означення 2. Під композицією двох логічних операцій будемо розуміти їх послідовне виконання.

Теорема 1. Множина дворозрядних логічних операцій утворює групу відносно операції композиція. Ця група ізоморфна групі S^4.

Доведення. При композиції двох дворозрядних логічних операцій відбувається послідовне переставляння елементів повідомлення, тобто виконується композиція чотирьохелементних перестановок. Відомо, що чотирьохелементні перестановки відносно операції композиція утворюють групу S^4 [2]. Таким чином, так як мы встановили пряму відповідність між елементами множини логічних операцій і елементами групи S^4, показали еквівалентність операції композиція для обох множин, то множина дворозрядних логічних операцій відносно операції композиція ізоморфна групі , а отже, і сама є групою. Теорему доведено.

Висновки

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.

В роботі показано, що множина дворозрядних логічних операцій з введеною операцією композиція є групою.

Список літератури

1. *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.*
2. *Курош А.Г. Теория групп / А.Г. Курош. – М.: Наука, 1967. – 648 с.*
3. *Рудницький В.М. Побудова обернених логічних функцій для систем захисту інформації / В.М. Рудницький, Д.А. Жиляєв // Системи обробки інформації: зб. наук. пр. – Х.: ХУПС, 2009. – Вип. 4 (78). – С. 114-116.*
4. *Бабенко В.Г. Моделювання логічних функцій для систем захисту інформації / В.Г. Бабенко, С.В. Рудницький // Методи та засоби кодування, захисту й ущільнення інформації: тези доп. Третьої Міжнародної науково-практичної конференції. – Вінниця: ВНТУ, 20-22 квітня 2011 року. – С. 82-83.*
5. *Миронець І.В. Вдосконалення методики синтезу функцій декодування на основі спеціалізованих логічних функцій / І.В. Миронець, В.Г. Бабенко // Інтегровані інтелектуальні робототехнічні комплекси: матеріали другої міжнар. наук.-практ. конф. (Київ, 2009 р.); Нац. авіац. ун-т, фак-т інформ. технологій. – К., 2009. – С. 45.*
6. *Миронець І.В. Методики синтезу функцій декодування на основі спеціалізованих логічних функцій / І.В. Миронець, В.Г. Бабенко // Проблеми інформатизації: зб. тез доп. Міжнар. наук.-техн. семінар. – Черкаси, 2009. – Вип. 1 (3). – С. 47-50.*

Надійшла до редколегії 3.10.2011

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

ПОСТРОЕНИЕ ОБРАТНЫХ ФУНКЦИЙ ДЛЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

В.Н. Рудницкий, В.Г. Бабенко, 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, группа.*

CONSTRUCTION OF REVERSE FUNCTIONS FOR THE SYSTEMS OF PROTECTION TO INFORMATION

V.N. Rudnitsky, V.G. Babenko, 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, group.*