Математика и её приложения: ЖИМО. 2011. Вып. 1 (8). С. 75 – 78.

УДК 517.19

Е. А. Михайлова[1]

О реализации схемы В. Пана защиты информации в канале со стираниями

Ключевые слова: кодек, матрица Коши, q-ичный канал, помехи, стирания, поле Галуа.

Разработаны и реализованы алгоритмы кодирования и декодирования схемы В. Пана защиты информации при передаче данных по каналу со стираниями. Программная реализация проведена на языке C++ при помощи библиотеки работы с конечными полями WinNTL.

Keywords: codec, Cauchy matrix, q-ary channel, interference, erasure, Galois field.

We develop and realize the coding and decoding algorithms of V. Pan's scheme for information security, when data transmitted by means of erasure channel. Program implemenetation realized in C++ language with the WinNTL-library for operation with finite fields.

1. Введение и постановка задачи

Рассмотрим задачу передачи данных по цифровому каналу с помехами, порождающими ошибки типа стирания. Будем предполагать, что поле Галуа $\mathbb{F}_{p^r}$ является алфавитом информационных сообщений и входным канальным алфавитом, а $\mathbb{F}_{p^r} \cup \{*\}$ — выходным канальным алфавитом [1]. Под стиранием будем понимать событие, состоящее в замене при передаче по каналу некоторого символа из $\mathbb{F}_{p^r}$ на символ $*$. На вход канала поступают информационные сообщения над полем $\mathbb{F}_{p^r}$, а на выходе будет получено сообщение над $\mathbb{F}_{p^r} \cup \{*\}$.

В. Паном в [2] предложен метод защиты информации от стираний при передаче по каналу. Метод основан на использовании составной матрицы, одной частью которой является единичная, а другой — матрица Коши.

Целью данной работы является разработка и реализация алгоритмов кодирования и декодирования схемы В. Пана. Программная реализация проведена на языке $C++$, при помощи библиотеки WinNTL для работы с конечными полями [3].

2. Модель защиты

Рассмотрим предложенную в [2] схему защиты от стираний в канале передачи данных, у которого алфавит информационных сообщений и входной канальный алфавит — поле Галуа $\mathbb{F}_{p^r}$, а выходной канальный алфавит — $\mathbb{F}_{p^r} \cup \{*\}$.

[1]Южный Федеральный Университет; E-mail: mikhailovakaterina@yandex.ru.

© Михайлова Е. А., 2011

Множество всех $(k \times m)$-матриц над полем $\mathbb{F}$ будем обозначать $\mathrm{Mat}_{k,m}(\mathbb{F})$, или, если поле не требует уточнения, $\mathrm{Mat}_{k,m}$. Матрицей Коши будем называть прямоугольную матрицу вида

$$C(c,d) \;=\; \begin{pmatrix} \frac{1}{c_1-d_1} & \frac{1}{c_2-d_1} & \cdots & \frac{1}{c_l-d_1} \\ \frac{1}{c_1-d_2} & \frac{1}{c_2-d_2} & \cdots & \frac{1}{c_l-d_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{c_1-d_k} & \frac{1}{c_2-d_k} & \cdots & \frac{1}{c_l-d_k} \end{pmatrix} \in \mathrm{Mat}_{k,l}(\mathbb{F}_{p^r}) \qquad (1)$$

определенную для различных $k+l$ элементов поля $\mathbb{F}_{p^r}$

$$c = (c_1, c_2, \ldots, c_l), \quad d = (d_1, d_2, \ldots, d_k).$$

Как показано в [4], в этом случае ранг матрицы (1) равен $\min\{l; k\}$.

В процессе кодирования информационный поток над полем $\mathbb{F}_{p^r}$ разбивается на блоки длины k. Кодирование блока производится умножением информационного сообщения m на кодирующую блочную матрицу:

$$G \;=\; (\; I_k \mid C(c,d)\;)(\in Mat_{k\times n}), \qquad (2)$$

где I_k — единичная $(k \times k)$-матрица. Кодирование имеет вид:

$$s \;=\; mG = \;(\; m \mid mC(c,d)\;). \qquad (3)$$

Заметим, что отправляемое сообщение состоит из двух частей — скопированное исходное сообщение и исходное сообщение, умноженное на матрицу Коши. Закодированный вектор s передается по каналу. В канале могут происходить помехи, порождающие стирания, при этом стертые символы заменяются на $*$.

В соответствии с методом защиты информации от стираний, предложенным в работе [2], построим алгоритмы кодирования и декодирования. Вследствие того, что информация передается блоками, достаточно рассмотреть кодирование и декодирование одного блока.

Алгоритм 1.

Вход: кодирующая матрица (2); информационный вектор $m \in \mathrm{Mat}_{1,k}(\mathbb{F}_{p^r})$.

Выход: закодированный вектор $s \in \mathrm{Mat}_{1,n}(\mathbb{F}_{p^r})$.

1) Вычислить $s_1 \;=\; mC(c,d) \in \mathrm{Mat}_{1,n-k}(\mathbb{F}_{p^r})$.

2) Вернуть сгенерированный вектор $s \;=\; (\; m \mid s_1\;) \in \mathrm{Mat}_{1,n}(\mathbb{F}_{p^r})$.

Перейдем к декодированию. Далее для произвольной матрицы A через $A_{(i)}$ будем обозначать i-ю строку, а через $A^{(j)}$ — j-й столбец; для вектора a через $a_{(i)}$ будем обозначать i-й элемент. Предположим, что по каналу пришел вектор $r \in \mathrm{Mat}_{1,n}(\mathbb{F}_{p^r} \cup \{*\})$. Алгоритм декодирования основан на отборе столбцов кодирующей матрицы, соответствующих номерам элементов r, полученным без стираний. При этом учитывается, что кодирующая матрица содержит единичную, и, поэтому, некоторые элементы вектора r являются частью информационного сообщения.

Алгоритм 2.

Вход: полученный по каналу вектор $r \in \mathrm{Mat}_{1,n}(\mathbb{F}_{p^r} \cup \{*\})$.

Выход: информационный вектор m, если количество полученных без стирания символов не меньше k, и ошибка декодирования в противном случае.

1) Если количество полученных без стираний символов меньше k, вернуть сообщение об ошибке декодирования.

2) Если первые k символов сообщения r получены без стираний, сгенерировать и вернуть вектор $m = (r_1, \ldots, r_k)$. Выйти из алгоритма.

3) Построить упорядоченное по возрастанию множество

$$I = \{i_1; \ldots; i_k\}$$

номеров первых k полученных без стирания элементов вектора r. Построить упорядоченные по возрастанию множества

$J_1 = \{j_{1,1}; \ldots; j_{1,b}\} = \{i_s \in I : i_s \leqslant k\}$,

$J_2 = \{j_{2,1}; \ldots; j_{2,(k-b)}\} = \{i_s \in I : i_s > k\}$,

$J_3 = \{j_{3,1}; \ldots; j_{3,(k-b)}\} = \{1; 2; \ldots; k\} - J_1$.

4) Сгенерировать два вектора:

$h = (r_{(j_{1,1})}, r_{(j_{1,2})}, \ldots, r_{(j_{1,b})})(\in \mathrm{Mat}_{1,b})$,

$t = (r_{(j_{2,1})}, r_{(j_{2,2})}, \ldots, r_{(j_{2,(k-b)})})(\in \mathrm{Mat}_{1,k-b})$

(векторы h и t составлены из полученных без стираний элементов левой и правой части сообщения r соответственно (см. (3))).

5) Сгенерировать две матрицы:

$H = (G^{(j_{1,1})}\ G^{(j_{1,2})}\ \cdots\ G^{(j_{1,b})})(\in \mathrm{Mat}_{k,b})$,

$T = (G^{(j_{2,1})}\ G^{(j_{2,2})}\ \cdots\ G^{(j_{2,(k-b)})})(\in \mathrm{Mat}_{k,k-b})$

(матрицы H и T составлены из столбцов кодирующей матрицы, номера которых совпадают с номерами полученных без стираний элементов левой и правой части вектора r соответственно (см. (2))).

6) Сгенерировать матрицу:

$H_1 = (G^{(j_{3,1})}\ G^{(j_{3,2})}\ \cdots\ G^{(j_{3,k-b})})(\in \mathrm{Mat}_{k,k-b})$

(матрица H_1 составлена из столбцов кодирующей матрицы, номера которых совпадают с номерами стертых элементов левой части вектора r).

7) Сгенерировать две матрицы:

$C_1 = ((T_{(j_{1,1})})^\tau\ \cdots\ (T_{(j_{1,b})})^\tau)^\tau(\in \mathrm{Mat}_{b,k-b})$,

$C_2 = ((T_{(j_{3,1})})^\tau\ \cdots\ (T_{(j_{3,(k-b)})})^\tau)^\tau(\in \mathrm{Mat}_{k-b,k-b})$

(матрица C_1 составлена из строк матрицы T, номера которых совпадают с номерами полученных без стираний элементов левой части вектора r, а матрица C_2 составлена из оставшихся строк матрицы T).

8) Вычислить

$u_1 = t - hC_1(\in \mathrm{Mat}_{1,k-b})$, $u = C_2^{-1}u_1(\in \mathrm{Mat}_{1,k-b})$.

10) Сгенерировать и вернуть информационный вектор

$m = hH + uH_1(\in \mathrm{Mat}_{1,k})$.

Выйти из алгоритма.

На основе результатов [2] доказывается теорема о том, что при использовании алгоритмов 1 и 2 сообщение будет гарантировано восстановлено, если количество стираний не превосходит $n - k$.

3. О программной реализации

Программная реализация построенного кодека осуществлена на языке $C++$ при помощи библиотеки WinNTL для работы с конечными полями [3]. Алгоритмы кодирования и декодирования реализованы соответственно модулями coding_mod и decoding_mod. Каждый элемент $a \in \mathbb{F}_{p^r}$, записывается в векторной форме вида $[a_0\, a_1\, \ldots\, a_n]$, где $a = a_0 + a_1 x + \cdots + a_n x^n$ — полиномиальное представление элемента. В модуле coding_mod функцией el_read(ifstream fmess,vector<ZZ_pE> mess) производится потоковое чтение из заданного тестового файла записанного поэлементно строками того вектора, который требуется закодировать. После чего функцией coding(vector<ZZ_pE> mess,vector<ZZ_pE> cod_mess) производится кодирование в соответствии с алгоритмом 1. В процессе выполнения этой функции происходит генерация кодовой матрицы по считанным из заданного файла векторам c и d (см. (1)) при помощи функции matr_create(ifstream fcod,vector< vector<ZZ_pE> > matr). Полученное закодированное сообщение записывается в формате, аналогичном формату файла исходного сообщения, при помощи функции el_write(ofstream fcod_mess,vector<ZZ_pE> cod_mess).

При помощи модуля decoding_mod декодируется пришедшее по каналу сообщение, которое содержит $*$ в местах стираний. Функцией el_read_analize(ifstream fdec,vector<ZZ_pE> s_mess, vector<bool> is_eras) производится его потоковое чтение. Создается массив is_eras, значения координат которого показывают, произошло ли стирание в соответстующей координате полученного сообщения. Он содержащит значение true в координатах, соответствующих $*$ в полученном сообщении, и значение false в остальных координатах. Декодирование производится в соответствии с алгоритмом 2 функцией decoding(vector<ZZ_pE> s_mess,vector<ZZ_pE> new_mess), возвращающей 0 в случае успешного декодирования и 1 при ошибке декодирования. Обращение матрицы Коши на шаге 8 алгоритма 2 производится функцией inv_Cau(int b, vector< vector<ZZ_pE> > cau). Полученный при декодировании результат записывается в файл при помощи функции el_write.

Список литературы

1. *Деундяк В. М., Косолапов Ю. В.* Математическая модель канала с перехватом второго типа // Известия высших учебных заведений. Северо-Кавказский регион. Естественные науки. – 2008. – № 3. – С. 3–8.
2. *Pan V. Y.* Matrix structure and loss-resilient encoding/decoding // Computers and Mathematics with Applications. – 2003. – V. 46. – P. 493–499.
3. *Shoup V.* NTL: A Library for doing Number Theory // `URL:http://shoup.net/ntl/`
4. *Fink T., Heinig G., Rost K.* An inversion formula and fast algorithms for Cauchy-Vandermonde matrices // Linear Algebra Appl. – 1993. – V. 183. – P. 179–191.

Поступила в редакцию 27.11.2011