УДК 519.7
DOI: 10.25559/SITITO.14.201803.586-593

МЕТОД ПОСТРОЕНИЯ ПОЛИНОМА ОДНОЙ ПЕРЕМЕННОЙ НАД КОНЕЧНЫМ ПОЛЕМ РАУНДОВОЙ ФУНКЦИИ БЛОЧНЫХ ШИФРОВ

С.А. Белов
Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия

A METHOD OF CONSTRUCTING A BLOCK CIPHERS ROUND FUNCTION'S POLYNOMIAL OVER A FINITE FIELD

Sergey A. Belov
Lomonosov Moscow State University, Moscow, Russia

© Белов С.А., 2018

Ключевые слова

Блочный шифр; шифр PRESENT; интерполяционный криптоанализ; матрица Вандермонда; многочлен над конечным полем.

Аннотация

В работе предлагается метод построения раундовой функции в виде полинома одной переменной над конечным полем. Предложенный метод основан на вычислении исходного криптографического преобразования в специальных точках конечного поля и последующем обращении матрицы Вандермонда. Для этого класса матриц существуют алгоритмы вычисления обратной матрицы, которые значительно эффективнее стандартного алгоритма обращения с помощью метода Гаусса. В работе был использован алгоритм Трауба, вычислительная сложность которого пропорциональна квадрату размера заданной матрицы. Метод применим для блочных итеративных шифров специального вида (SP-сеть). Для этого класса шифров приведены математические оценки алгебраических параметров полиномов раундовых функций над конечным полем. Количественные значения оценок посчитаны для актуального российского стандарта шифрования «Кузнечик». Представлены оценки вычислительной сложности предлагаемого метода. Проведены практические вычисления полиномов одной переменной для преобразования над конечными полями с различными характеристиками. Приведены практические результаты измерений времени работы при построении полиномов в конечных полях различной размерности. С помощью представленного метода в явном виде вычислен многочлен одной переменной над конечным полем раундовой функции блочного шифра PRESENT

Keywords

Block cipher; PRESENT cipher; interpolation cryptoanalysis; Vandermonde matrix; finite field polynomial.

Abstract

The work outlines the method of construction of round function as a polynomial of one variable over the finite field. The proposed method is based on the calculation of the initial cryptographic transformation at special points of the finite field and the subsequent inversion of Vandermonde matrix. For this class of matrices, there are algorithms for calculating the inverse matrix, which are much more efficient than the standard algorithm of inversion using the Gauss method. In the proposed work, the Traub algorithm is used. The computational complexity of Traub algorithm is proportional to the square of the size of a given matrix. The method is applicable to block iterative ciphers of special type (SP-network). For this type of ciphers, mathematical evaluations of algebraic parameters of polinomials of round functions over the finite fields are provided. Quantative values of estimations are calculated for Russian encryption standard "Kuznechik". The estimates of computational complexity of the proposed method are provided. The article contains practical results of estimations of work time for polynomials notation for finite fields of varying dimensions. The proposed method is used for explicit calculation of the polynomial of one variable over the finite field of round function of block cipher PRESENT.

Об авторе:

Белов Сергей Алексеевич, аспирант, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, ГСП-1, г. Москва, Ленинские горы, д. 1); ORCID: https://orcid.org/0000-0002-7923-0129, serbel.sci@gmail.com

Введение

Исследование стойкости шифров к различным методам криптоанализа является одним из важнейших направлений симметричной криптографии. Основными методами криптоанализа блочных шифров являются дифференциальный [1] и линейный [2] криптоанализ. В то же время существуют методы криптоанализа, эффективные против криптографических преобразований, устойчивых к дифференциальному и линейному криптоанализу. Примером является интерполяционный криптоанализ, предложенный Кнудсеном в работе [3]. Основной идеей метода является представление криптографического преобразования многочленом над конечным полем соответствующей размерности и последующем применении процедуры интерполяции к этому многочлену. Стойкость криптографического преобразования к интерполяционному криптоанализу существенно зависит от алгебраических параметров полинома над конечным полем. Поскольку размер блока современных блочных шифров достаточно велик, построить представление шифра в виде полинома над конечным полем в явном виде не представляется возможным. В таком случае алгебраические свойства блочного итеративного шифра оценивают в зависимости от соответствующих алгебраических свойств раундового преобразования. В работе представлены оценки алгебраических параметров раундовых функций блочных шифров и метод построения раундовых преобразований блочных шифров в виде полинома над конечным полем.

Работа построена следующим образом. В разделе 2 приведены основные обозначения, используемые в дальнейшем. В разделе 3 представлены основные сведения об интерполяционном криптоанализе. В разделе 4 дан обзор алгоритмов обращения матрицы Вандермонда. В 5 разделе более детально рассмотрен алгоритм Трауба для обращения матрицы Вандермонда. В разделе 6 приводятся необходимые теоремы и доказательства о полиномах в конечных полях. В разделе 8 рассматриваются раундовые функции SP-сетей. В разделе 8 приводится алгоритм построения полинома одной переменной раундовой функции произвольного блочного шифра. В разделе 9 описано устройство шифра PRESENT. В разделах 10 и 11 приводятся результаты и численные характеристики предложенного алгоритма для различных параметров.

1. Соглашения и обозначения

V_n – множество всех векторов из 0 и 1 длины n.

$GF(q)$ – конечное поле из q элементов.

Известно, что любая функция $F: GF(q) \to GF(q)$ может быть представлена в виде многочлена одной переменной над полем $GF(q)$ степени не более $q-1$.

$Tr(x) = \sum_{i=0}^{n-1} x^{2^i}$ – функция "след" в поле $GF(2^n)$. Определённая таким образом функция является отображением из $GF(2^n)$ в $GF(2)$.

Два базиса $\alpha_0, \dots, \alpha_{n-1}$ и $\beta_0, \dots, \beta_{n-1}$ конечного поля будем называть дуальными, если $Tr(\alpha_i \beta_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$

Пусть F – функция над $GF(q)$. Через $deg(F)$ будем обозначать степень полинома этой функции над конечным полем.

Булева функция от n переменных $f(x_1, \dots, x_n)$ может быть единственным образом представлена в виде многочлена от переменных $x_1, \dots, x_n$. Такое представление называется полиномом Жегалкина и имеет вид

$$c_0 \bigoplus_{1 \le i_1 \le \dots \le i_k \le n, k \in \{1,2,\dots,n\}} c_{i_1,i_2,\dots,i_k} x_{i_1} x_{i_2} \dots x_{i_k}$$

Пусть f – булева функция от n переменных. Через $def(f)$ будем обозначать степень булевой функции, т.е. число переменных в самом длинном слагаемом полинома Жегалкина функции f.

При изложении будем рассматривать функцию $F: \{0,1\}^n \to \{0,1\}^n$, с одной стороны, как функцию над конечным полем $F: GF(2^n) \to GF(2^n)$, с другой стороны, как вектор булевых функций $F(x) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$, где $f_i(x_1, \dots, x_n)$ – булева функция от n переменных. В таком случае будем говорить, что булевы функции $f_1, \dots, f_n$ составляют функцию F.

2. Интерполяционный криптоанализ

В работе [3] Якобсен и Кнудсен предложили новый тип атак на блочные шифры, названный ими интерполяционным криптоанализом. Обозначим размер блока в битах n, открытый текст x, а шифротекст y, x и y могут быть представлены как вектора длины p, в котором каждый элемент состоит из m бит (при этом выполняется равенство $p * m = n$):
$x = (x_1, \dots, x_p) \in GF(2^n), x_i \in V_m$
$y = (y_1, \dots, y_p) \in GF(2^n), y_i \in V_m$

В работе рассмотрены два варианта атаки: первый из них, названный авторами global deduction, состоит в следующем. Фиксируется некоторый ключ k. В этом случае результат работы шифра после j раундов может быть представлен как полином над конечным полем

$$y_j = g_{j,k}(x) = f_{k_j}(f_{k_{j-1}}(\dots f_{k_1}(x)..))$$

Имея достаточное количество пар открытый – зашифрованный текст, атакующий может восстановить полином g при помощи интерполяционной формулы Лагранжа. Восстановив многочлен $g_{j,k}$, атакующий может зашифровывать сообщения без знания секретного ключа. Поменяв местами x и y атакующий также может расшифровывать сообщения без знания секретного ключа. Второй вариант атаки, названный авторами instance deduction, аналогичен первому, только дополнительно атакующий имеет возможность фиксировать части

открытого текста $(x_1,\dots,x_p)$ для того, чтобы уменьшить сложность раундовой функции. В работе также предложен метод восстановления ключа. Практическая возможность осуществления атаки была продемонстрирована Якобсеном и Кнудсеном на примере шифра PURE [4] и семействе шифров SHARK [5]. Авторы работы также продемонстрировали возможность применения интерполяционного криптоанализа к семейству шифров SNAKE.

3. Алгоритмы обращения матрицы Вандермонда

Рассмотрим многочлен над конечным полем степени $n-1$. Задача интерполяции многочлена сводится к решению систем линейных уравнений с матрицей Вандермонда, которая имеет вид:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{pmatrix}$$

Известно, что если $x_1,..,x_n$ различны, то матрица Вандермонда обратима. Помимо классического алгоритма Гаусса существует множество алгоритмов обращения матрицы Вандермонда, учитывающих её особую структуру. В работе [7] приведены явные выражения для элементов матрицы, обратной матрице Вандермонда специального вида над конечным полем $GF(q)$. В работе [8] приведён итеративный алгоритм вычисления элементов обратной матрицы со сложностью $O(n^3)$. Известны алгоритмы с лучшей оценкой сложности: алгоритм Трауба [9], алгоритм Паркера [10], алгоритм Бьёрка-Перейа [11], обобщённый алгоритм Паркера-Трауба [12], алгоритм Ян и Янга [13], имеющие вычислительную сложность $O(n^2)$.

4. Алгоритм Трауба

Перечисленные выше алгоритмы обращения матрицы Вандермонда имеют одинаковую вычислительную сложность $O(n^2)$ и различаются, главным образом, в вопросе накопления ошибок при вычислениях, что не существенно для случая матрицы над конечным полем. В представленной работе для обращения матрицы был использован алгоритм Трауба [9, 12]. В этом разделе приводится более подробное описание этого алгоритма. Пусть V – матрица Вандермонда размерности n. Обозначим:

$$P(x)=\prod_{k=1}^{n}(x-x_k)=x^n+\sum_{k=0}^{n-1}a_k*x^k$$

Алгоритм (Трауб)

1. Вычислить коэффициенты $P(x)$ с помощью следующей процедуры:

$$\begin{bmatrix} a_0^1 \\ a_1^1 \end{bmatrix}=\begin{bmatrix} -x_1 \\ 1 \end{bmatrix}\begin{bmatrix} a_0^k \\ a_1^k \\ \vdots \\ a_k^k \end{bmatrix}=\begin{bmatrix} 0 \\ a_0^{k-1} \\ \vdots \\ a_{k-1}^{k-1} \end{bmatrix}-x_k*\begin{bmatrix} a_0^{k-1} \\ \vdots \\ a_{k-1}^{k-1} \\ 0 \end{bmatrix}$$

В качестве коэффициентов взять результат на -ом шаге, т.е. $a_j=a_j^n$;

2. Для $j=1\dots n$ выполнить следующее:

   2.1. Вычислить $q_k(j)$ по формуле:
   $$q_0(x)=1, q_k(x)=x*q_{k-1}(x)+a_{n-k}, k=1\dots n$$
   2.2. Вычислить $P'(j)=q'_n(j)$ по формуле:
   $$q'_1(x)=1, q'_k(x)=q_{k-1}(x)+x*q'_{k-1}(x), k=2\dots n$$
   2.3. Вычислить -ый столбец матрицы V^{-1}
   $$q_k(x_j)*(P'(x_j))^{-1}, k=0\dots n-1$$
3. Конец алгоритма.

Алгоритм Трауба позволяет вычислить все элементы обратной матрицы за время, пропорциональное n^2, что для больших матриц существенно быстрее, чем вычисление обратной матрицы классическим алгоритмом Гаусса.

5. Характеристики многочлена над конечным полем

Пусть $GF(2)$ – поле из двух элементов, а $GF(2^n)$ – его расширение степени n. Тогда булеву функцию от n переменных $f(x_0,\dots,x_{n-1})$ можно представить в виде функции $F: GF(2^n)\to GF(2)$. Или, так как $GF(2)$ является подполем $GF(2^n)$, $F: GF(2^n)\to GF(2^n)$. Обозначим через $\alpha_0,\dots,\alpha_{n-1}$ базис конечного поля, а через $\beta_0,\dots,\beta_{n-1}$ – дуальный к нему базис. Тогда для любого $x\in GF(2^n)$ однозначно определено разложение по базису $x=\sum_{i=0}^{n-1}\alpha_i x_i$, а в силу двойственности базисов верно $x_i=Tr(\beta_i x)$. Таким образом, для любой булевой функции $f(x_0,\dots,x_{n-1})$ имеет место равенство $f(x_0,\dots,x_{n-1})=f(Tr(\beta_0 x),Tr(\beta_1 x),\dots,Tr(\beta_{n-1}x))$. Такое представление булевой функции называется трейс-представлением. Трейс-представления используется для анализа свойств булевых функций, в том числе криптографических [14, 15, 16].

Теорема 1. [17] Пусть f – булева функция от n переменных, $deg(f)=d$. $F(x): GF(2^n)\to GF(2)$ – представление f над полем $GF(2^n)$. Тогда $deg(F)\le 2^n-2^{n-d}$, количество мономов в F не превосходит $\sum_{i=0}^{d}C_n^i$.

Лемма 1. Пусть два набора чисел $A=(2^{k_1},2^{k_2}\dots,2^{k_t}), 0\le k_i\le n, i\in\{1,2,\dots,t\}$ и $B=(2^{p_1},2^{p_2}\dots,2^{p_q}), 0\le p_i\le n,\ i\in\{1,2,\dots,q\}$ имеют одинаковую сумму и $A\cap B=\emptyset$. Тогда в одном из наборов присутствуют повторяющиеся числа из множества $\{1,2,4,\dots,2^{n-1}\}$.

Доказательство. Если ни в наборе A, ни в наборе B нет повторяющихся элементов, то $\sum_{i=0}^{t}2^{k_i}$ и $\sum_{j=0}^{q}2^{p_j}$ являются представлениями одного и того же числа в двоичной системе счисления. В силу единственности

представления числа в двоичной системе счисления, приходим к противоречию с условием $A \cap B = \emptyset$. Если в одном из наборов повторяется число 2^n, то сумма набора не меньше чем $2^n + 2^n = 2^{n+1}$. Предположим, что во втором наборе нет повторяющихся элементов. Так как $A \cap B = \emptyset$, то во втором наборе не может присутствовать число 2^n. Тогда сумма второго набора не превосходит $1 + 2 + 4 + \ldots + 2^{n-1} = 2^n - 1 < 2^n$. Приходим к противоречию с условием равенства сумм наборов.

Лемма 2. Пусть два набора чисел $A = (2^{k_1}, 2^{k_2} \ldots, 2^{k_t}), 0 \le k_i \le n, i \in \{1,2,\ldots,t\}$ и $B = (2^{p_1}, 2^{p_2} \ldots, 2^{p_q}), 0 \le p_i \le n, i \in \{1,2,\ldots,q\}$ имеют одинаковую сумму по модулю $2^{n+1} - 1$ и $A \cap B = \emptyset$. Тогда в одном из наборов присутствуют повторяющиеся элементы.

Доказательство. Предположим, что ни в наборе A, ни в наборе B нет повторяющихся элементов. Так как $A \cap B = \emptyset$, и элементы наборов ограничены сверху числом 2^n, выполнено $\sum_{i=0}^{t} 2^{k_i} + \sum_{j=0}^{q} 2^{p_j} \le 1 + 2 + \ldots + 2^n = 2^{n+1} - 1$. Так как сумма набора больше либо равна 1, сумма каждого из наборов строго меньше $2^{n+1} - 1$. По предположению, в наборах нет повторяющихся элементов, следовательно, $\sum_{i=0}^{t} 2^{k_i}$ и $\sum_{j=0}^{q} 2^{p_j}$ являются представлениями одного и того же числа в двоичной системе счисления. В силу единственности представления числа в двоичной системе счисления, приходим к противоречию с условием $A \cap B = \emptyset$.

Теорема 2. Пусть f – булева функция от n переменных, $deg(f) = d$. $F(x): GF(2^n) \to GF(2)$ – представление f над полем $GF(2^n)$. $F(x) = \sum_{i=0}^{2^n-1} a_i x^i$. Тогда $a_i = 0$ для тех i, которые не являются суммой по модулю $2^n - 1$ строго возрастающего набора, состоящих из чисел $\{1,2,\ldots,2^{n-1}\}$, длины не более d.

Доказательство. Рассмотрим булеву функцию $f(x_1,\ldots,x_n)$ степени m как функцию $GF(2^n) \to GF(2)$. Полином Жегалкина этой булевой функции имеет вид

$$c_0 \bigoplus_{1 \le i_1 \le \ldots \le i_k \le n, k \in \{1,2,\ldots,m\}} c_{i_1,i_2,\ldots,i_k} x_{i_1} x_{i_2} \ldots x_{i_k}$$

Выберем базис конечного поля $\alpha_1,\ldots,\alpha_n$. Подставляя вместо булевых переменных $x_1,\ldots,x_n$ соответствующие им выражения $Tr(\beta_i x)$ (где β_i – элементы дуального базиса), получаем выражение

$$c_0 \bigoplus_{1 \le i_1 \le \ldots \le i_k \le n, k \in \{1,2,\ldots,m\}} c_{i_1,i_2,\ldots,i_k} Tr(\beta_{i_1} x) Tr(\beta_{i_2} x) \ldots Tr(\beta_{i_k} x)$$

Полином функции $Tr(x)$ содержит только степени двойки $\{1,2,4,\ldots,2^{n-1}\}$, следовательно, в полиномиальном представлении булевой функции f над конечным полем $F = \sum_{i=0}^{2^n-1} a_i x^i, x \in GF(2^n)$ для тех i, которые не представимы в виде суммы по модулю $2^n - 1$ степеней двойки, с числом слагаемых не более d, $a_i = 0$. Перестановка элементов множества не меняет его сумму, поэтому достаточно рассмотреть только упорядоченные неубывающие наборы. Согласно лемме 1, если в наборе присутствуют повторяющиеся числа из множества $\{1,2,\ldots,2^{n-1}\}$, то существует набор с такой же суммой, в котором нет повторений. В то же время, у двух строго возрастающих наборов не могут быть одинаковые суммы, так как иначе возникает противоречие с леммой 2.

Теорема 3. Пусть $F(x): GF(2^n) \to GF(2^n)$, $F(x) = \sum_{i=0}^{2^n-1} a_i x^i$. $F(x) = \big(f_0(x), \ldots, f_{n-1}(x)\big)$, где $f_0, \ldots, f_{n-1}$ – булевы функции. d – максимальная степень среди функций $f_0, \ldots, f_{n-1}$

Тогда:

1. Количество мономов $F(x)$ не превосходит $\sum_{i=0}^{d} C_n^i$;
2. $deg(F(x)) \le 2^n - 2^{n-d}$;
3. $a_i = 0$ для тех i, которые не являются суммой строго возрастающего набора, состоящих из чисел $\{1,2,\ldots,2^{n-1}\}$ длины не более d.

Доказательство. Обозначим базис конечного поля $GF(2^n)$ $\alpha_0,\ldots,\alpha_{n-1}$. Пусть f_i имеет представление над конечным полем

$$F_i(x) = \sum_{k=0}^{2^n-1} a_k^i x^k, F_i(x): GF(2^n) \to GF(2)$$

Тогда для функции $F(x)$ верно представление $F(x) = \sum_{j=0}^{n-1} \alpha_j F_j(x)$. Из этого представления и теорем 1 и 2 следуют утверждения теоремы.

6. SP-сети

Будем говорить, что раундовый блочный шифр является SP-сетью [18], если его раунды имеют следующий вид. Каждый раунд состоит из трёх стадий. На первой стадии осуществляется побитовый XOR блока с раундовым ключом. На второй стадии блок делится на m подблоков длины p бит каждый. К каждому подблоку применяются нелинейные преобразования, называемые S-блоками, каждый из которых осуществляет перестановку $\{0,1\}^p \to \{0,1\}^p$. На третьей стадии применяется обратимое линейное преобразование $P: V_n \to V_n$. Таким образом, раундовая функция может быть представлена в виде $R(x) = P(S(x \oplus k))$, где k — раундовый ключ.

Теорема 4. Пусть $S(x): GF(2^n) \to GF(2^n), S(x) = (s_0(x),\ldots,s_{n-1}(x))$, где $s_0,\ldots,s_{n-1}$ – булевы функции. d – наибольшая степень среди функций $s_0, \ldots, s_{n-1}$. $k \in GF(2^n)$. $P(x): GF(2^n) \to GF(2^n)$ – обратимое линейное преобразование. $R(x) = P(S(x+k)): GF(2^n) \to GF(2^n), R(x) = (r_0(x),\ldots,r_{n-1}(x))$, где $r_0,\ldots,r_{n-1}$ – булевы функции. r – наибольшая степень среди функций $r_0, \ldots, r_{n-1}$ Тогда

1. $r \le s$;
2. Количество мономов в $R(x)$ не превосходит $\sum_{i=0}^{d} C_n^i$.

Доказательство. Рассмотрим функцию $R(x) = P(S(x+k))$ как вектор-функцию, состоящую из булевых функций $r_0(x),\ldots,r_{n-1}(x)$. Сложение с константой в конечном поле эквивалентно прибавлению по модулю два 0 или 1 к каждой из булевых функций $s_0(x),\ldots,s_{n-1}(x)$ и

потому не изменяет степени булевых функций. Применение линейного преобразования P состоит из операций сложения по модулю два булевых функций друг с другом, в результате чего степень булевых функций может только уменьшиться, из чего следует утверждение 1 теоремы. Утверждение 2 следует из утверждения 1 и теоремы 3.

В качестве примера рассмотри раундовую функция шифра «Кузнечик»[1]. Раундовая функция шифра «Кузнечик» с раундовым ключом k задаётся преобразованием $F(x) = LSX[k](x)$, в котором
$X[k](x): V_{128} \to V_{128} = x \oplus k$
$S(x): V_{128} \to V_{128} = S(x_{15}||\dots||x_0) = \pi_{15}||\dots||\pi_0$
$\pi(x): V_8 \to V_8$ – нелинейное преобразование
$L(x): V_{128} \to V_{128} = R^{16}(x)$
$R(x): V_{128} \to V_{128} = R(x_{15}||\dots||x_0) = l(x_{15}||\dots||x_0)||x_{15}||\dots||x_1$
$l(x): V_8^{16} \to V_{128}$ – линейное преобразование
Непосредственно проверяется, что максимальная степень булевой функции, составляющей преобразование $\pi(x)$ равна 7. Следовательно, степень полинома функции F над $GF(2^{128})$ не превосходит $2^{128} - 2^{121} = 337623910929368631717566993311207522304$, а количество мономов не превосходит $\sum_{i=0}^{7} C_{128}^{i} = 100224990433$.

Покажем, как при помощи доказанных утверждений построить полином одной переменной раундовой функции блочного шифра над конечным полем в явном виде.

7. Алгоритм построения полинома раундовой функции

Пусть R(x) = $c_1 x^{\alpha_1} + c_2 x^{\alpha_2} + \dots + c_n x^{\alpha_n}$ – полином над конечным полем. Задача состоит в том, чтобы вычислить коэффициенты $c_1, c_2, \dots, c_n$ имея возможность вычислять значения многочлена в любых наперёд заданных точках. Покажем, как вычислить значения коэффициентов $c_1, c_2, \dots, c_n$, вычислив многочлен в n точках. Выбрав различные точки $x_1, \dots, x_n$, получаем систему линейных уравнений $Ac = y$, где $c = (c_1, \dots, c_n)^T$ - вектор коэффициентов, y –вектор значений многочлена в выбранных точках, $y = (y_1, \dots, y_n)^T, y_i = R(x_i)$. Матрица A представляет собой матрицу вида

$$\begin{pmatrix} x_1^{\alpha_1} & x_1^{\alpha_2} & \dots & x_1^{\alpha_n} \\ x_2^{\alpha_1} & x_2^{\alpha_2} & \dots & x_2^{\alpha_n} \\ \dots & \dots & \dots & \dots \\ x_n^{\alpha_1} & x_n^{\alpha_2} & \dots & x_n^{\alpha_n} \end{pmatrix}$$

Чтобы гарантировать существование и единственность решения будем выбирать точки следующим образом. Пусть a – элемент конечного поля порядка больше чем n. Такой элемент заведомо существует, так как известно, что мультипликативная группа конечного поля является циклической. Поэтому в качестве элемента a всегда можно выбрать образующий элемент этой группы. Выберем точки следующим образом:
$x_1 = 1$
$x_2 = a^1$
$\dots$
$x_n = a^{n-1}$
При таком выборе точек матрица A принимает вид

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ a^{\alpha_1} & a^{\alpha_2} & \dots & a^{\alpha_n} \\ \dots & \dots & \dots & \dots \\ (a^{n-1})^{\alpha_1} & (a^{n-1})^{\alpha_2} & \dots & (a^{n-1})^{\alpha_n} \end{pmatrix}$$

Произведём замену $t_1 = a^{\alpha_1}, \dots, t_n = a^{\alpha_n}$. Матрица A будет иметь вид

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_n \\ \dots & \dots & \dots & \dots \\ t_1^{n-1} & t_2^{n-1} & \dots & t_n^{n-1} \end{pmatrix}$$

То есть матрица A – это транспонированная матрица Вандермонда. Так как, в силу выбора элемента a, элементы $t_1, \dots, t_n$ являются различными, матрица A невырождена, следовательно, приведённая выше система линейных уравнений имеет единственное решение. Алгоритм построения полинома раундовой функции блочного шифра следующий:

Алгоритм

1. Вычислить n – максимально возможное количество мономов, входящих в многочлен по теореме 3;
2. Вычислить $\alpha_1, \dots, \alpha_n$ – степени, которые могут входить в многочлен по теореме 3;
3. Найти элемент a такой что $ord(a) > n$ (в качестве a можно взять образующий элемент поля);
4. Вычислить точки $x_1, \dots, x_n$ по формуле $x_i = a^{i-1}, i = 1 \dots n$;
5. Вычислить $y = (y_1, \dots, y_n)^T$ – значения многочлена $f(x)$ в точках $x_1, \dots, x_n, y_i = f(x_i)$;
6. Вычислить матрицу A, где $A_{i,j} = a^{i\alpha_j}$;
7. Вычислить матрицу A^{-1} по алгоритму Трауба;
8. Вычислить $c = A^{-1}y$, где $c = (x_1, \dots, c_n)^T$ — искомые коэффициенты;
9. Вернуть c;
10. Конец алгоритма.

Вычислительно наиболее сложной частью Алгоритма 1 является вычисление обратной матрицы (шаг 7), который, как было отмечено выше, имеет сложность порядка N^2 арифметических операций, где N – размер матрицы. Обозначим $S_n^d = \sum_{i=0}^{d} C_n^i$ Таким образом сложность Алгоритма 1 можно оценить как $O((S_n^d)^2)$ арифметических операций в конечном поле, где n – размерность поля, а d – максимальная степень булевой функции, составляющей функцию над полем. Отметим, что согласно теореме 4, Алгоритм 1 применим к любой SP-сети. В качестве демонстрации действия алгоритма вычислим в явном виде полином раундовой функции шифра PRESENT.

[1] ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры Стандартинформ Москва 2015

8. Шифр PRESENT

Процедура шифрования

PRESENT [19] является блочным шифром, имеющим структуру SP-сети с 32 раундами. Размер блока 64 бита. Размер ключа 80 бит или 128 бит. Каждый раунд состоит из трёх стадий. На первой стадии addRoundKey осуществляется побитовый XOR с раундовым ключом . Раундовый ключ имеет размер 64 бита и является функцией основного ключа.

$$x_i = x_i \oplus K_i$$

На второй стадии sBoxLayer применяются 16 параллельных S-блоков, каждый из которых осуществляет перестановку . Таблица значений S-блока (в шестнадцатеричной нотации) представлена ниже:

Таблица1. S-блок шифра PRESENT

Table 1. S-block cipher PRESENT

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

На третьей стадии pLayer применяется линейная перестановка. Бит с номером заменяется на бит с номером согласно приведённой таблице:

Таблица2. L-преобразование шифра PRESENT

Table 2. Cipher PRESENT L-transform

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P[j]	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
j	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P[j]	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
j	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P[j]	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
j	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P[j]	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

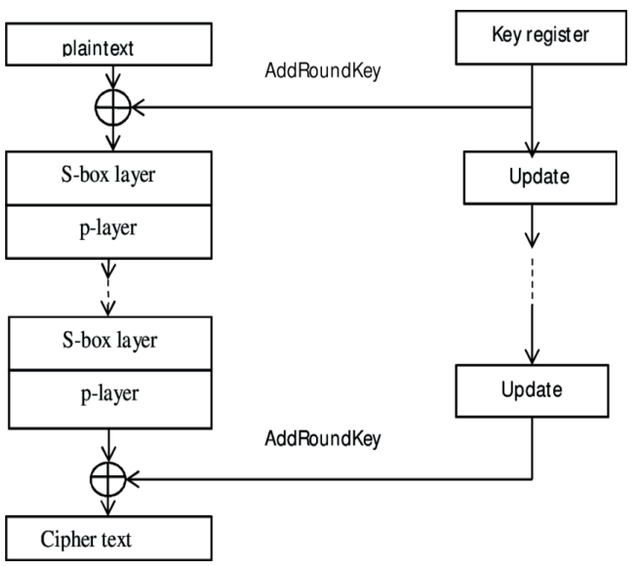

Рис. 1. Общая схема шифра PRESENT

Fig. 1. General scheme of cipher PRESENT

Выработка раундовых ключей

Как было отмечено выше, PRESENT может работать в двух режимах – с 80-битным и 128-битным ключом. Опишем процедуру генерации раундовых ключей из 80-битного ключа. Ключ сохраняется в регистре $K = k_{79}k_{78}\dots k_0$. На i-ом раунде в качестве раундового ключа K_i выбираются 64 крайних левых бита регистра K. То есть $K_i = k_{79}\dots k_{16}$. После этого регистр K изменяется по следующим правилам

1. $k_{79}\dots k_0 = k_{18}k_{17}\dots k_{20}k_{19}$ – циклический сдвиг влево на 61 позицию;
2. $k_{79}k_{78}k_{77}k_{76} = S[k_{79}k_{78}k_{77}k_{76}]$ – применение S-блока к 4 крайним правым значениям;
3. $k_{19}k_{18}k_{17}k_{16}k_{15} = k_{19}k_{18}k_{17}k_{16}k_{15} \oplus i$ – побитовое сложение по модулю два с номером раунда.

9. Построение полинома раундовой функции шифра PRESENT

Рассмотрим стадию sBoxLayer раундовой функции. На этой стадии входное слово x разбивается на части по 4 бита, затем каждая такая часть подаётся на вход S-блоку. S-блок является перестановкой $\{0,1\}^4 \to \{0,1\}^4$. S-блок может быть представлен как вектор из 4 булевых функции, каждая из которых имеет по 4 переменных.

$$S(x) = S(x_1, x_2, x_3, x_4) = (s_1(x_1, x_2, x_3, x_4), s_2(x_1, x_2, x_3, x_4),$$

$$s_3(x_1, x_2, x_3, x_4), s_4(x_1, x_2, x_3, x_4))$$

Полиномы Жегалкина булевых функций s_1, s_2, s_3, s_4:

$$s_1 = x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \oplus x_2x_3 \oplus x_4 \oplus x_3 \oplus x_1 \oplus 1$$

$$s_2 = x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_3 \oplus x_1x_4 \oplus x_3x_4 \oplus x_2 \oplus x_1 \oplus 1$$

$$s_3 = x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \oplus x_1x_2 \oplus x_1x_3 \oplus x_3 \oplus x_1$$

$$s_4 = x_2x_3 \oplus x_4 \oplus x_2 \oplus x_1$$

Максимальная степень булевых функций, входящих в S-блок равна 3. Преобразование P раундовой функции является линейным преобразованием и осуществляет перестановку битов после прохождения S-блоков. Из теоремы 3 следует, что в полиноме одной переменной над конечным полем $GF(2^{64})$ этой функции не более чем $\sum_{i=0}^{3} C_{64}^{i} = 43745$ мономов.

10. Практические результаты

Описанный метод был реализован на языке C. Вычисления проводились на системе со следующей конфигурацией: процессор 2xIntel Xeon E5-2630v4 2.2 ГГц, 10x2 core 64 ГБ DDR4 ECC, 480 ГБ SSD + 4 ТБ SATA. Результаты экспериментов для различных конечных полей представлены в таблице 3. Здесь d – максимальная степень булевой функции, составляющей функцию над конечным полем, S_n^d – количество мономов (совпадает с размерностью матрицы).

Таблица 3. Результаты экспериментов

Table 3. Experimental results

Поле	d		Время
	2	11	< 1 секунды
	2	37	< 1 секунды
	3	93	< 1 секунды
	2	137	< 1 секунды
	3	697	1 секунда
	2	529	< 1 секунды
	3	5489	1 секунда
	1	65	< 1 секунды
	2	2081	31 секунда
	3	43745	8 часов

Был вычислен полином раундовой функции шифра PRESENT над полем =). Полином имеет степень 16140901064495857664 и состоит из 43745 мономов, то есть все коэффициенты, которые могут присутствовать в многочлене, ненулевые. Полную запись полинома можно найти по ссылке https://raw.githubusercontent.com/SergeyBel/science/master/PresentPolynom/present.

Заключение

В работе представлен метод построения полинома одной переменной над конечным полем раундовых функций блочных шифров. Метод основан на обращении матрицы над конечным полем специального вида. Приведены оценки сложности представленного метода и результаты практических вычислений. Важно отметить, что метод эффективен для функций над конечным полем, у которых невелики степени составляющих их булевых функций. В этом случае возможно построение полиномов даже над полями достаточно большой размерности за приемлемое на практике время. Был построен полином для раундовой функции блочного шифра PRESENT.

Благодарности

Автор благодарит научного руководителя Э.А. Применко за внимание к проводимым исследованиям.

Список использованных источников

[1] *Biham E., Shamir A.* Differential cryptanalysis of DES-like cryptosystems // Journal of Cryptology. 1991. Vol. 4, issue 1. Pp. 3-72. DOI: 10.1007/BF00630563

[2] *Matsui M.* Linear cryptanalysis method for DES cipher // Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1993. Pp. 386-397. DOI: 10.1007/3-540-48285-7_33

[3] *Jakobsen T., Knudsen L.R.* The interpolation attack on block ciphers // International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 1997. Pp. 28-40. DOI: 10.1007/BFb0052332

[4] *Nyberg K., Knudsen L.R.* Provable security against a differential attack // Journal of Cryptology. 1995. Vol. 8, issue 1.Pp. 27-37. DOI: 10.1007/BF00204800

[5] *Rijmen V.* et al. The cipher SHARK // International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 1996. Pp. 99-111. DOI: 10.1007/3-540-60865-6_47

[6] *Moriai S., Shimoyama T., Kaneko T.* Interpolation attacks of the block cipher: SNAKE // International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 1999. Pp. 275-289. DOI: 10.1007/3-540-48519-8_20

[7] *Althaus H., Leake R.* Inverse of a finite-field Vandermonde matrix (Corresp.) // IEEE Transactions on Information Theory. 1969. Vol. 15, issue 1. Pp. 173-173. DOI: 10.1109/TIT.1969.1054253

[8] *Клейбанов С.Б., Норкин К.Б.*, Привальский В.Б. Обращение матрицы Вандермонда // Автоматика и телемеханика. 1977. № 4. С. 176-177. URL: http://mi.mathnet.ru/at7343 (дата обращения: 10.05.2018).

[9] *Traub J.F.* Associated polynomials and uniform methods for the solution of linear problems // Siam Review. 1966. Vol. 8, issue 3. Pp. 277-301. DOI: 10.1137/1008061

[10] *Parker F.D.* Inverses of Vandermonde matrices // The American Mathematical Monthly. 1964. Vol. 71, issue 4. Pp. 410-411. DOI: 10.2307/2313246

[11] *Björck Å., Pereyra V.* Solution of Vandermonde systems of equations // Mathematics of Computation. 1970. Vol. 24, issue 112. Pp. 893-903. DOI: 10.1090/S0025-5718-1970-0290541-1

[12] *Gohberg I., Olshevsky V.* The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices // Journal of Complexity. 1997. Vol. 13, issue 2. Pp. 208-234. DOI: 10.1006/jcom.1997.0442

[13] *Yan S., Yang A.* Explicit algorithm to the inverse of Vandermonde matrix // 2009 International Conference on Test and Measurement. Hong Kong, 2009. Pp. 176-179. DOI: 10.1109/ICTM.2009.5413083

[14] *Баев В.В.* Некоторые нижние оценки на алгебраическую иммунность функций, заданных своими след-формами // Проблемы передачи информации. 2008. Т. 44, №. 3. С. 81-104. DOI: 10.1134/S0032946008030071

[15] *Кузьмин А.С., Марков В.Т., Нечаев А.А., Шишков А.Б.* Приближение булевых функций мономиальными //

Дискретная математика. 2006. Т. 18, № 1. С. 9-29. DOI: 10.1515/156939206776241255

[16] *Кузьмин А.С., Ноздрунов В.И.* Взаимосвязь коэффициентов полинома над полем и веса булевой функции // Прикладная дискретная математика. 2014. № 4(26). С. 28-46. URL: https://elibrary.ru/item.asp?id=22636126 (дата обращения: 10.05.2018).

[17] *Carlet C.* Boolean Functions for Cryptography and Error-Correcting Codes / Y. Crama, P. Hammer (Eds.) // Boolean Models and Methods in Mathematics, Computer Science, and Engineering (Encyclopedia of Mathematics and its Applications). Cambridge: Cambridge University Press, 2010. Part III. Pp. 257-397. DOI: 10.1017/CBO9780511780448.011

[18] *Feistel H.* Cryptography and computer privacy // Scientific American. 1973. Vol. 228, issue 5. Pp. 15-23. DOI: 10.1038/scientificamerican0573-15

[19] *Bogdanov A.* et al. PRESENT: An Ultra-Lightweight Block Cipher / P. Paillier, I. Verbauwhede (Eds.) // Cryptographic Hardware and Embedded Systems – CHES 2007. CHES 2007. Lecture Notes in Computer Science. Vol. 4727. Springer, Berlin, Heidelberg, 2007. Pp. 450-466. DOI: 10.1007/978-3-540-74735-2_31

Поступила 10.05.2018; принята в печать 15.08.2018;
опубликована онлайн 30.09.2018.

References

[1] Biham E., Shamir A. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*. 1991; 4(1):3-72. DOI: 10.1007/BF00630563

[2] Matsui M. Linear cryptanalysis method for DES cipher. *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 1993. Pp. 386-397. DOI: 10.1007/3-540-48285-7_33

[3] Jakobsen T., Knudsen L.R. The interpolation attack on block ciphers. *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 1997. Pp. 28-40. DOI: 10.1007/BFb0052332

[4] Nyberg K., Knudsen L.R. Provable security against a differential attack. *Journal of Cryptology*. 1995; 8(1):27-37. DOI: 10.1007/BF00204800

[5] Rijmen V. et al. The cipher SHARK. *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 1996. Pp. 99-111. DOI: 10.1007/3-540-60865-6_47

[6] Moriai S., Shimoyama T., Kaneko T. Interpolation attacks of the block cipher: SNAKE. *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 1999. Pp. 275-289. DOI: 10.1007/3-540-48519-8_20

[7] Althaus H., Leake R. Inverse of a finite-field Vandermonde matrix (Corresp.) *IEEE Transactions on Information Theory*. 1969; 15(1):173-173. DOI: 10.1109/TIT.1969.1054253

[8] Kleibanov S.B., Norkin K.B., Prival'skii V.B. Inversion of the Vandermond matrix. *Automation and Remote Control*. 1977; 38(4):600-601. Available at: http://mi.mathnet.ru/eng/at7343 (accessed 10.05.2018).

[9] Traub J.F. Associated polynomials and uniform methods for the solution of linear problems. *Siam Review*. 1966; 8(3):277-301. DOI: 10.1137/1008061

[10] Parker F.D. Inverses of Vandermonde matrices. *The American Mathematical Monthly*. 1964; 71(4):410-411. DOI: 10.2307/2313246

[11] Björck Å., Pereyra V. Solution of Vandermonde systems of equations. *Mathematics of Computation*. 1970; 24(112):893-903. DOI: 10.1090/S0025-5718-1970-0290541-1

[12] Gohberg I., Olshevsky V. The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices. *Journal of Complexity*. 1997; 13(2):208-234. DOI: 10.1006/jcom.1997.0442

[13] Yan S., Yang A. Explicit algorithm to the inverse of Vandermonde matrix. *2009 International Conference on Test and Measurement*. Hong Kong, 2009. Pp. 176-179. DOI: 10.1109/ICTM.2009.5413083

[14] Bayev V.V. Some lower bounds on the algebraic immunity of functions given by their trace forms. *Problems of Information Transmission*. 2008; 44(3):243-265. DOI: 10.1134/S0032946008030071

[15] Kuzmin A.S., Markov V.T., Nechaev A.A., Shishkov A.B. Approximation of Boolean functions by monomial functions. *Discrete Mathematics and Applications*. 2006; 16(1):7–28. DOII: 10.1515/156939206776241255

[16] Kuzmin A.S., Nozdrunov V.I. Relationship between the coefficients of polynomials over GF(2^n) and weights of Boolean functions represented by them. *Prikladnaya Diskretnaya Matematika*. 2014; 4(26):28-46. Available at: https://elibrary.ru/item.asp?id=22636126 (accessed 10.05.2018). (In Russian)

[17] Carlet C. Boolean Functions for Cryptography and Error-Correcting Codes. Y. Crama, P. Hammer (Eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering (Encyclopedia of Mathematics and its Applications). Cambridge: Cambridge University Press, 2010. Part III. Pp. 257-397. DOI: 10.1017/CBO9780511780448.011

[18] Feistel H. Cryptography and computer privacy. *Scientific American*. 1973; 228(5):15-23. DOI: 10.1038/scientificamerican0573-15

[19] Bogdanov A. et al. PRESENT: An Ultra-Lightweight Block Cipher. P. Paillier, I. Verbauwhede (Eds.) *Cryptographic Hardware and Embedded Systems – CHES 2007*. CHES 2007. Lecture Notes in Computer Science. Vol. 4727. Springer, Berlin, Heidelberg, 2007. Pp. 450-466. DOI: 10.1007/978-3-540-74735-2_31

Submitted 10.05.2018; revised 15.08.2018;
published online 30.09.2018.

About the author:

Sergey A. Belov, graduate student, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1, Leninskie gory, GSP-1, Moscow 119991, Russia), ORCID: https://orcid.org/0000-0002-7923-0129, serbel.sci@gmail.com

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted reuse, distribution, and reproduction in any medium provided the original work is properly cited.