Algebraic Cryptanalysis of Block Ciphers

Rustem Biyashev1, Dilmuhanbet Dyusenbayev1, Kunbolat Algazy1 and Nursulu Kapalova1,∗

1Information Security laboratory Institute of Information and Computational Technologies CS MES RK Almaty, Kazakhstan
∗Corresponding author

Abstract— Algebraic methods of cryptanalysis are applicable to present-day ciphers. These methods are based on generation of an equation system, where elements of a ciphertext and a key are chosen as variables of the system. When implementing a linearization method to solve the equation system, we consider a possibility to find partial elements of a key. Generally, cryptosystems use S-blocks, which are the only element contributing to nonlinearity of a ciphering transformation and the level of its strength against cryptanalytic attacks. In this paper we present, by the example of encryption algorithm Kuznechik, an algebraic analysis applicable to some block ciphers.

Keywords—algebraic cryptanalysis; linearization; S-block; residual variables; equation systems; truth tables; Boolean function; Zhegalkin polynomial

I. INTRODUCTION

While developing cryptosystems, engineers in many applications use S-blocks, which are the only element contributing to nonlinearity of a ciphering transformation and the level of its strength against cryptanalytic attacks.

Algebraic methods are based on formation of a nonlinear equation system representing transformations in S-blocks, and then solving the nonlinear system. In the equations composed for a linear transformation, the number of variables is not increasing when passing to next rounds, i.e. the number of variables after the first round is equal to the number of variables after the n-th round. In contrast, in nonlinear transformations the number of variables increases after each round owing to multiplications of variables from a previous round. An algorithm complexity can be evaluated on the basis of the systems of linear equations resulting from linearization.

The size of variables chosen for a study depends on the type of the algorithm under consideration, as well as preferences of a cryptanalyst. For example, variables can have a size of a bit or byte.

When implementing a linearization method to solve a set of equations, it is possible to find elements of a key in parts. Our study was conducted with the developed software enabling generation of an equation system for linear and nonlinear transformations, namely for substitution S-blocks. The essence of the method is generation of equations representing nonlinear substitution transformations of S-blocks with a subsequent attempt to solve the generated equation systems and obtain parts of a cipher key. When performing the attack, a cryptanalyst can see in a stepwise manner changes in outputs after each round.

The attack was conducted against encryption algorithm Kuznechik. Our study has shown that all processes of the algorithm, apart from S-block, can be presented in a linear form. If a ciphertext has been known then the respective plaintext and key are considered as variables [1,2]. Each output of S-block can be converted with an input and represented by nonlinear expressions. We then obtain a set of nonlinear equations and through linearization pass to a linear equation system. Here, the number of variables is increasing exponentially with each round. So, in our studies we used an abridged linearization with residual variables. By residual variables we mean a grouped sum of terms, which are products of two or more variables.

II. BLOCK CIPHER «KUZNECHIK»

This is a symmetric block encryption algorithm with a block size of 128 bits and a 256-bit key, generated with a Feistel network. The cipher is built on the basis of an SP-network, which is a transformation involving several identical rounds, where each round comprises of linear and nonlinear conversions followed by applying a key. An SP-network transforms the whole input block as against a half-block in a Feistel network. The length of input block of the algorithm is 128 bits, and the length of master key is 256 bits [3, 4].

Kuznechik peculiarities:

– Round keys are generated from a master key based on a Feistel network, where a round transformation of the original algorithm serves as a function;
– A linear transformation can be performed with a shift register.

When implementing encryption and decryption algorithms, the following transformations are used:

\[
X[k]: V_{128} \rightarrow V_{128}, X[k](a) = k \oplus a, \text{ where } k, a \in V_{128};
\]

\[
S: V_{128} \rightarrow V_{128}, S(a) = S(a_{15}||…||a_0) = \pi(a_{15})||…||\pi(a_0),
\]

where \(a = a_{15}||…||a_0 \in V_{128}, a_i \in V_8, i = 0, 1, \ldots, 15;\)

\[
L: V_{128} \rightarrow V_{128} - L(a) = R^{16}(a), \text{ where } a \in V_{128};
\]

The linear transformation is given by mapping \(\ell: V_8 \rightarrow V_8\), which is defined as follows:

\[
\ell(a_{15}, \ldots, a_0) = \Psi(148 \cdot \Delta(a_{15}) + 32 \cdot \Delta(a_{14}) + 133 \cdot \Delta(a_{13}) + 16 \cdot \Delta(a_{12}) + 194 \cdot \Delta(a_{11}) + 192 \cdot \Delta(a_{10}) + 1 \cdot \Delta(a_9) + 251 \cdot \Delta(a_8) + 1 \cdot \Delta(a_7) + 192 \cdot \Delta(a_6) + 194 \cdot \Delta(a_5) + 16 \cdot \Delta(a_4) + 133 \cdot \Delta(a_3) + 32 \cdot \Delta(a_2) + 148 \cdot \Delta(a_1) + 1 \cdot \Delta(a_0))
\]
III. ALGEBRAIC CRYPTANALYSIS OF KUZNECHIC ALGORITHM

Let \((x^0_1, x^0_2, \ldots, x^0_n)\) and \((x^1_1, x^1_2, \ldots, x^1_k)\) be input and output bit blocks of the linear transformation at the initial step. Output bits are expressible by:

\[
x^1_i = q_{i,1}x^0_1 \oplus q_{i,2}x^0_2 \oplus \cdots \oplus q_{i,n}x^0_n,
\]

where coefficients \(q_{i,j} \in \{0,1\}, \ i,j = \overline{1,n} (n – block length)\) reflect the contribution of bit \(x^0_j\) to the equation. In Kuznechik algorithm, \(n = 384, m = 128\).

Suppose \((x^0_1, x^0_2, \ldots, x^0_n)\) is an output vector of round \(r\). The equations governing bits of the output vector of round \(r\) are given by formula:

\[
x^1_i = q_{i,1}x^0_1 \oplus q_{i,2}x^0_2 \oplus \cdots \oplus q_{i,n}x^0_n
\]

The regularity above cannot be attributed merely to S-block. If we linearize the S-block equation represented by output bits, then the equation can be transformed by input bits following:

\[
y_i = q_i \bigoplus \bigoplus_{1 \leq j \leq 8} (q_{ij}x_j) \bigoplus \bigoplus_{1 \leq k \leq 8} (q_{ik}x_k)
\]

The first term of linear transformation (4) \((148 \cdot X)\) can be expressed as follows:

\[
(Z + z^2 + z^3) \cdot (x^7 \cdot z^2 + x^6 \cdot z^3 + x^5 \cdot z^4 + x^4 \cdot z^5 + x^3 \cdot z^6 + x^2 \cdot z^7 + z^8)
\]

We repeat the steps for the other terms.

Now we perform a modulo operation, which result in the finite augmented field \(\text{GF}(2)[z]/(z^8 + z^7 + z^6 + z^5 + z^4 + z^3 + z^2 + z + 1)\).

Providing that the block length is 128, we will get not more than 4096 output variables for 16 input bytes. Here, inputs of S-blocks involve 8 variables as against 16 ones. If the number of variables is \(k\), then the number of combinations will not exceed \(K = \sum_{i=1}^{k} C_i^k\). The general case, the S-block output variables for \(k\) input variables can be expressed as follows:

\[
y_i = a_{i0} \bigoplus \bigoplus_{j=1}^{k} (a_{ij}x_j) \bigoplus \bigoplus_{1 \leq j \leq 5k} (a_{ij,j,2}x_jx_k) \bigoplus \cdots \bigoplus a_{i,1, \ldots, k} x_1 \times \cdots \times x_k
\]

where \(i = 1, 2\).
where residues we denote as residual variables.

involving in a Zhegalkin polynomial \([5,6]\), i.e. you can describe under consideration uses 256 variables for key generation, and

bytes as bits, then after \(r\)-th cycle the output bits will follow the formula:

\[
a_{16} = (x'_{127}, \ldots, x'_{120}),
\]

where

\[
x'_{127} = x'_{127} \oplus x'_{126} \oplus x'_{125} \oplus x'_{124} \oplus x'_{123} \oplus x'_{122} \oplus x'_{121} \oplus x'_{120}
\]

\[
= q_{127}^{r+1} x_{127}^0 \oplus q_{126}^{r+1} x_{126}^0 \oplus \cdots \oplus q_{120}^{r+1} x_{120}^0
\]

A byte shift represented in bits is as follows:

\[
x_i^{r+1} = x_i^r - 8 = q_{i,1}^{r+1} x_{i}^0 \oplus q_{i,2}^{r+1} x_{i}^0 \oplus \cdots \oplus q_{i,n}^{r+1} x_{i}^0
\]

Formula (10) is true for all \(i, r\), i.e. an output bit of every cycle can be expressed by elements of previous cycles. For any \(i, r\) equation \(q_i = q_i^r\) holds true.

Formula (10) involves \(n\) variables. The equation system under consideration uses 256 variables for key generation, and 384 variables for encryption.

B. Analysis of S-blocks

We look at S-block in its bit representation:

\[
(x_1, x_2, \ldots, x_8) \rightarrow (y_1, y_2, \ldots, y_8),
\]

where \(x_i, y_i \in \{0,1\}, i = \overline{1,8}\)

The bit representations are set as eight truth tables. For each truth table we construct a Boolean function of 8 variables involving in a Zhegalkin polynomial \([5,6]\), i.e. you can describe each output bit of the S-box. In the equations above our concern was only with monomials involving one variable. The sum of the rest terms we denote as residual variables.

When linearizing the equation system we obtain as follows

\[
y_1(x_1, x_2, \ldots, x_8) = \delta_1,
\]

\[
y_2(x_1, x_2, \ldots, x_8) = \delta_2, y_3(x_1, x_2, \ldots, x_8) = x_8 \oplus x_3 \oplus x_2 \oplus \delta_3
\]

\[
y_4(x_1, x_2, \ldots, x_8) = x_7 \oplus x_5 \oplus x_3 \oplus x_1 \oplus \delta_4,
\]

\[
y_5(x_1, x_2, \ldots, x_8) = \delta_5,
\]

\[
y_6(x_1, x_2, \ldots, x_8) = x_7 \oplus x_5 \oplus x_3 \oplus x_2 \oplus \delta_6,
\]

\[
y_7(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_4 \oplus x_3 \oplus x_1 \oplus \delta_7,
\]

\[
y_8(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_5 \oplus x_3 \oplus x_2 \oplus \delta_8
\]

A similar abridgement can be performed using residual variables, which are products of three different variables. As an example:

\[
y_1(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_6 \oplus x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus \delta_1.
\]

\[
y_2(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_6 \oplus x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus \delta_1.
\]

\[
y_3(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_6 \oplus x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus \delta_1.
\]

\[
y_4(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_6 \oplus x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus \delta_1.
\]

\[
y_5(x_1, x_2, \ldots, x_8) = x_8 \oplus x_7 \oplus x_6 \oplus x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus \delta_1.
\]

C. Linearization Abridgement

Let \(\sum_{i=1,j=1,n}(q_{1,i,j} x_i y_j) \oplus \cdots \oplus q_{n,n} x_1 \ldots x_n\) be a part of expression (8). We denote the remaining part as a new variable \(\delta_i \in \{0,1\}\), then formula (8) becomes as follows:

\[
x_i^r = q_i^r \oplus q_{i,1}^r x_1^0 \oplus q_{i,2}^r x_2^0 \oplus \cdots \oplus q_{i,n}^r x_n^0 \oplus \delta_i
\]

where \(x_i^r, i = \overline{1,128}\) are bits of a ciphertext.

We divide linear equation system (14) into two parts:

\[
F(K, X) = q_{i,0}^r \oplus q_{i,1}^r x_1^0 \oplus q_{i,2}^r x_2^0 \oplus \cdots \oplus q_{i,n}^r x_n^0
\]

and residual variables

\[
Q = (\delta_1, \delta_2, \ldots, \delta_{128}), F(K, X) + Q = Y
\]

Where \(K = (x_1, x_2, \ldots, x_{256})\) are key bits, \(X = (x_{257}, x_{258}, \ldots, x_{384})\) are plaintext bit, and \(Y = (y_1, y_2, \ldots, y_{128})\) are ciphertext bits.

Let there be given two blocks of the cipher \(F_1(K, X_1) \oplus Q_1 = Y_1\) and \(F_2(K, X_2) \oplus Q_2 = Y_2\).

If we add the plaintext variables to residuals, we will obtain:
\[ F'(K) \oplus Q_1^* = Y_1^*, F'(K) \oplus Q_2^* = Y_2^* \]  
(16)

When adding together equation systems (16), we get as follows:

\[ Q_1^* \oplus Q_2^* = Y_1^* \oplus Y_2^* \]  
(17)

The resulting equation system has 256 variables. In a general way, it needs an exhaustive search of \(2^{256}\) operations to retrieve the key. By using equation system (17) the complexity of exhaustive search will be \(2^{128}\).

IV. SUMMARY

Based on the research findings it can be proposed as follows:

- If equation system (15) provides for a possibility to calculate variables of the system in parts, then the complexity of the exhaustive search will be less than \(2^{128}\);
- If equation system (15) does not provide for a possibility to calculate variables of the system in parts, then the complexity of the exhaustive search will be more than \(2^{128}\);

With the developed software implementing the approach described above, it is possible to trace involvement of each key bit in the process of transformation. Our study of Kuznechik algorithm has shown that the transformation results lead to the second proposition, i.e. the complexity of the exhaustive search will be more than \(2^{128}\). The algebraic method discussed in the paper can also be implemented to other block encryption algorithms.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Education and Science of the Republic of Kazakhstan.

REFERENCES

[1] Voronin R. I., Algebraic cryptoanalysis of one-round S-AES, PDM, 2011, 4, 29–31.
[2] Kapalova N., Dyusenbayev D., Security analysis of an encryption scheme based on nonpositional polynomial notations // Journal Open Engineering. - 2016. – 6: p. 250-258.
[3] National Standard of the Russian Federation, GOST R 34.12-2015, Information technology, Cryptographic data security, Block ciphers. Moscow, Moscow, Standartinform, 2015, p. 21.
[4] E.A. Ischukova, R.A. Koshutskiy, L.K. Babenko. Development and implementation of high speed data encryption algorithm Kuznechik, Electronic scientific journal of the Kursk State University. 2015. No. 4 (08).
[5] S.D. Shaporev. Mathematical logic. Course of lectures and tutorial. – SPb.: BHV-Peterburg, 2007. – p. 416: il.
[6] A.G. Rostovtsev, E.B. Makhovenko. Theoretical cryptography. NPO Professional, Sankt-Petersburg, 2004.