Каждый из восьми s-боксов размера 6×4 алгоритма DES можно представить как 32 s-бокса размера 4×4 с помощью фиксаций битов a_1 и a_6, управляющих выбором одной из четырёх подстановок степени 16 s-бокса, где $a_1, a_2, \ldots, a_6$ — биты входного набора s-бокса. При фиксации других битов биективность s-бокса не обеспечена. Установлено, что 6 s-боксов не обладают свойством 2 (наличие неподвижных точек). Ряд s-боксов не обладают свойством 4: у 16 s-боксов $p_s = 8/16$, у 14 — $p_s = 6/16$, и имеется по одному s-боксу, у которых $p_s = 4/16$ и $10/16$.

В алгоритме ГОСТ 28147-89 используется 8 s-боксов размера 4×4, имеются рекомендации по их выбору [3]. Установлено, что из восьми s-боксов три не обладают свойством 2 (наличие неподвижных точек); у всех s-боксов $p_s = 4/16$.

Выводы

1. Построенное множество s-боксов размера 4×4, обладающее рядом позитивных свойств, может быть использовано при решении задач синтеза перспективных криптографических алгоритмов.

2. Созданное программное обеспечение может быть использовано для исследования s-боксов размера 4×4, используемых в различных действующих и перспективных криптографических системах.

ЛИТЕРАТУРА

1. *Menyachikhin A.* Spectral-Linear and Spectral-Difference Methods for Generating Cryptographically Strong S-Boxes. CTCrypt Preproc., 2016.
2. *Фомичев В. М.* Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. 424 с.
3. Рекомендации по стандартизации ТК 26 «Задание узлов замены блока подстановки алгоритма шифрования ГОСТ 28147-89». 2013.

УДК 519.7 DOI 10.17223/2226308X/10/43

CRYPTAUTOMATA: DEFINITION, CRYPTANALYSIS, EXAMPLE[1]

G. P. Agibalov

This conference paper is an extended abstract of a recent article in Prikladnaya Diskretnaya Matematika (2017, No. 36), where we presented the definition of the cryptautomata and described some cryptanalysis techniques for them. In cryptosystems, the cryptautomata are widely used as its primitives including cryptographic generators, s-boxes, filters, combiners, key hash functions as well as symmetric and public-key ciphers, and digital signature schemes. A cryptautomaton is defined as a class C of automata networks of a fixed structure N constructed by means of the series, parallel, and feedback connection operations over initial finite automata (finite state machines) with transition and output functions taken from some predetermined functional classes. A cryptautomaton key can include initial states, transition and output functions of some components in N. Choosing a certain key k produces a certain network N_k from C to be a new cryptographic algorithm. In case of invertibility of N_k, this algorithm can be used for encryption. The operation (functioning) of any network N_k in the discrete time is described by the canonical system of equations of its automaton. The structure of N_k is described by the union of canonical systems of equations of its components. The cryptanalysis problems for a cryptautomaton are considered as the problems of solving the operational or structural system of equations

[1]Работа поддержана грантом РФФИ, проект №17-01-00354.

of N_k with the corresponding unknowns that are key k variables and (or) plaintexts (input sequences). For solving such a system E, the method DSS is used. It is the iteration of the following three actions: 1) E is Divided into subsystems E' and E'', where E' is easy solvable; 2) E' is Solved; 3) the solutions of E' are Substituted into E'' by turns. The definition and cryptanalysis of a cryptautomaton are illustrated by giving the example of the autonomous alternating control cryptautomaton. It is a generalization of the LFSR-based cryptographic alternating step generator. We present a number of attacks on this cryptautomaton with the states or output functions of its components as a key.

Keywords: *finite automaton, automata network, cryptautomaton, alternating control cryptautomaton, cryptanalysis, "divide-and-solve-and-substitute", partially defined function completion.*

1. Definition

In this paper, we will present an extended abstract of the recent article [1] devoted to the definition of the cryptautomata and to description of some cryptanalysis techniques for them. Here is a formal mathematical definition: a *cryptautomaton* is a three-tuple (C, I, K), where C, the network class, is a finite set of possible automaton networks; I, the *keyplace*, is a finite set of possible key variables, and K, the *keyspace*, is a finite set of possible keys. The automaton networks under consideration are constructed of some initial finite automata (finite-state machines) by using the operations of series, parallel, and feedback connections, and they themselves uniquely define some initial finite automata.

The set C is completely defined by any automaton network $N \in C$ and consists of all the automaton networks that can only differ from N in some parameters of some components. For every such a parameter, the set of these components is presented in the keyplace I, and the parameter itself — in the keyspace K as a part of a key. Here, by the parameters of an automaton $A_i = (X_i, S_i, Y_i, g_i, f_i, s_i(1))$ in N, we mean its initial state $s_i(1)$, its transition function $g_i : X_i \times S_i \to S_i$, and its output function $f_i : X_i \times S_i \to Y_i$. It is supposed, that the parameters $s_i(1)$, g_i, and f_i are elements of, respectively, the set S_i of states in A_i, a class G_i of some functions $g : X_i \times S_i \to S_i$, and a class F_i of some functions $f : X_i \times S_i \to Y_i$.

So, if N consists of r components A_i, $i \in \{1, 2, \ldots, r\}$, then the keyplace I is the three-tuple of sets I_s, I_t, and I_o that are subsets of $\{1, 2, \ldots, r\}$, and the keyspace K of the cryptautomaton is the Cartesian product $\prod$ of sets K_s, K_t, and K_o, where $K_\mathrm{s} = \prod\limits_{i \in I_\mathrm{s}} S_i$, $K_\mathrm{t} = \prod\limits_{i \in I_\mathrm{t}} G_i$, and $K_\mathrm{o} = \prod\limits_{i \in I_\mathrm{o}} F_i$. Thus, a key in K is a three-tuple $k_\mathrm{s}k_\mathrm{t}k_\mathrm{o}$, where $k_\mathrm{s} \in K_\mathrm{s}$, $k_\mathrm{t} \in K_\mathrm{t}$, and $k_\mathrm{o} \in K_\mathrm{o}$, that is, a cryptautomaton key can be composed of initial states of some components in N, of transition functions in G_i for some $i \in \{1, 2, \ldots, r\}$, and of output functions in F_j for some $j \in \{1, 2, \ldots, r\}$.

Each key k in K defines a certain automaton network N_k in C and $C = \{N_k : k \in K\}$. The operation (functioning) of this network N_k in discrete time is described by the canonical system of equations of its automaton as well as by the union of canonical systems of equations describing the operations of components in N_k. The second system of equations also describes the structure (circuit) of N_k. In case that, for any k, the automaton of N_k is invertible, the cryptautomaton (C, I, K) is a cipher.

2. Cryptanalysis

There are many different cryptanalysis problems for a given cryptautomaton (C, I, K). Some of them are put as follows: given a finite output sequence γ of a network N_k in C and, possibly, an input sequence α which N_k transforms into γ, determine the key k and (or) the

sequence α. For solving these problems, we offer to solve the following two mathematical problems:

1) finding solutions of the systems of equations describing the operation or structure of the automaton networks in the network class C;

2) completing partially defined functions in a functional class, that is, for given a partially defined function ϕ and a class Φ of completely defined functions, it is required to find a function in Φ which coincides with ϕ on its domain.

In fact, the second problem is connected with the first one and appears after partial determining unknown output or transition functions of some components in the network N_k by solving its system of equations.

The system E of equations of any automaton network is recursively easy solvable (r.e.s.), that is, it has a nonempty subsystem $E_1 \subseteq E$ with a small effective subset U of unknowns such that assigning any possible values to them makes E_1 to be easily solvable and the subsystem $E_2 = E \setminus E_1$ becomes r.e.s. after substitution of any solution of E_1 into it. Thus, every solution of E can be computed by the method DSS [1, 2], consisting of three repeated actions: Divide E into E_1 and E_2, Solve E_1, and Substitute solutions of E_1 into E_2.

In [1], we illustrated the method DSS by solving canonical systems of equations of finite automata, series, parallel, and feedback automaton networks over the field $\mathbb{F}_2$ of two elements. The solution problem was the following one: given an output sequence of an automaton network N, find the input sequences of N. Besides, we defined an autonomous cryptautomaton with alternating control over $\mathbb{F}_2$ (that is a generalization of the cryptographic alternating step generator on LFSRs [3]) and illustrated the method DSS and the problem of completing partially defined functions by several attacks on this cryptautomaton with some different keyplaces I and corresponding keyspaces K.

3. Autonomous alternating control cryptautomaton

Let Σ be an autonomous cryptautomaton (C, I, K). It is called an *alternating control cryptautomaton* if each automaton network N in C is a *network with alternating control*, that is, N is a series-parallel connection of three automata: an autonomous automaton A_1, $A_1 = (\mathbb{F}_2^{m_1}, \mathbb{F}_2, g_1, f_1, s_1(1))$, and two unautonomous automata A_2 and A_3, $A_i = (\mathbb{F}_2, \mathbb{F}_2^{m_i}, \mathbb{F}_2, g_i, f_i, s_i(1))$, $i \in \{2, 3\}$, both controlled by A_1 in such a way that, for any their input symbol y_1 (produced on the output of A_1) and states s_2 и s_3 respectively, the *alternation condition* $g_2(y_1, s_2) = s_2 \Leftrightarrow g_3(y_1, s_3) \neq s_3$ is true, and both producing output symbols y_2 and y_3 respectively with the sum $y_2 \oplus y_3 \bmod 2$ on the output of N. For each $i \in \{1, 2, 3\}$, it is supposed that $S_i = \mathbb{F}_2^{m_i}$, $g_i \in G_i$, and $f_i \in F_i$, where G_i and F_i are some functional classes. The following is the canonical system of equations of the network N with alternating control:

$$
\begin{aligned}
&y_1(t) = f_1(s_1(t)),\\
&s_1(t+1) = g_1(s_1(t)),\\
&y_2(t) = f_2(y_1(t), s_2(t)),\\
&s_2(t+1) = g_2(y_1(t), s_2(t)),\\
&y_3(t) = f_3(y_1(t), s_3(t)),\\
&s_3(t+1) = g_3(y_1(t), s_3(t)),\\
&y(t) = y_2(t) \oplus y_3(t), \quad t \geqslant 1,\\
&s_1(1)s_2(1)s_3(1) \text{ — initial state,}
\end{aligned}
$$

where the first two equations describe the automaton A_1, the next five equations — the parallel subnetwork N' of the automata A_2 and A_3.

Here, for cryptanalysis of an alternating control cryptautomaton Σ, we describe some attacks on it with a known output sequence $\gamma = y(1)y(2)\dots y(l)$, $l \geqslant 1$, in order to determine its key k by using the method DSS in solving the canonical system of equations of a network N_k in C and by completing partially defined output functions of its components in their classes. The attacks depend on the type of keyplace I in Σ.

1. $I_{\mathrm{s}} = \{1\}$, $I_{\mathrm{t}} = I_{\mathrm{o}} = \varnothing$; $K_{\mathrm{s}} = S_1 = \mathbb{F}_2^{m_1}$, $K_{\mathrm{t}} = K_{\mathrm{o}} = \varnothing$; $K = K_{\mathrm{s}} = \mathbb{F}_2^{m_1}$; $k = s_1(1) \in K$.

A t t a c k 1: 1) given γ on the output of Σ, use the method DSS and compute the input sequences of parallel subnetwork N' that are, simultaneously, the output sequences of the automaton A_1; 2) for each of these sequences, find an initial state $s_1(1)$ of the automaton A_1 by an exhaustive key search.

Computational complexity of the attack equals 2^{m_1}.

2. $I_{\mathrm{s}} = \{1, 2\}$, $I_{\mathrm{t}} = I_{\mathrm{o}} = \varnothing$; $K_{\mathrm{s}} = S_1 \times S_2 = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2}$, $K_{\mathrm{t}} = K_{\mathrm{o}} = \varnothing$; $K = K_{\mathrm{s}} = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2}$; $k = s_1(1)s_2(1) \in K$.

In this case, the key of Σ is computed by a meet-in-the-middle attack. In advance, before the attack, for each possible value a of unknown $s_1(1)$, compute $s_1(t+1) = g_1(s_1(t))$ and $y_1(t) = f_1(s_1(t))$ for $t \in \{1, 2, \dots, l\}$ and $s_1(1) = a$ and store a in memory by address $H(y_1(1)y_1(2)\dots y_1(l))$, where $H : \mathrm{F}_2^l \to \mathrm{F}_2^{m_1}$ is a hash function.

A t t a c k 2: given γ on the output of Σ, use the method DSS and compute the input sequences of subnetwork N' for different values of $s_2(1)$ chosen unless, for some its value b, a sequence β will be obtained on the input of N' such that there is a value a of $s_1(1)$ in memory by address $H(\beta)$; in this case the pair (a, b) is taken for the result — the key k.

Computational complexity of the attack equals 2^{m_2}.

Remark: the attack remains valid after exchanging roles of A_2 and A_3 in it.

3. $I_{\mathrm{s}} = \{1, 2, 3\}$, $I_{\mathrm{t}} = I_{\mathrm{o}} = \varnothing$; $K_{\mathrm{s}} = S_1 \times S_2 \times S_3 = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathrm{F}_2^{m_3}$, $K_{\mathrm{t}} = K_{\mathrm{o}} = \varnothing$; $K = K_{\mathrm{s}} = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}$; $k = s_1(1)s_2(1)s_3(1) \in K$, and the set of variables $y_1(1), y_1(2), \dots, y_1(l)$ is a linearization set in the system of equations E' of the subnetwork N' of the network N.

A t t a c k 3: for each $s_1(1)$ in S_1, 1) compute $s_1(t+1) = g_1(s_1(t))$ and $y_1(t) = f_1(s_1(t))$ for $t \in \{1, 2, \dots, l\}$; 2) execute the linearization attack on E', namely: substitute the values $y_1(1), y_1(2), \dots, y_1(l)$ into E', solve the obtained system E'' of linear equations by Gauss method and find the values of unknowns $s_2(t)$ и $s_3(t)$, $t \in \{1, 2, \dots, l\}$; 3) from each solution of E'' satisfying the alternation condition for all t, $1 \leqslant t \leqslant l$, take the values of $s_2(1)$ and $s_3(1)$ and fix the three-tuple $(s_1(1)s_2(1)s_3(1))$ as one of the values of the key k.

Computational complexity of the attack equals 2^{m_1}.

Remark. So we have proved that in this case, the real key of the alternating control cryptautomaton is the initial state of the controlling automaton and its estending by means of initiall states of controlled automata doesn't increase the cryptographic security of the cryptautomaton. For the LFSR-based cryptographic alternating step generators, this fact was shown earlier in [4].

4. $I_{\mathrm{s}} = I_{\mathrm{t}} = \varnothing$, $I_{\mathrm{o}} = \{1\}$; $K_{\mathrm{s}} = K_{\mathrm{t}} = \varnothing$, $K_{\mathrm{o}} = F_1$; $K = K_{\mathrm{o}} = F_1$; $k = f_1 \in K$.

A t t a c k 4: 1) compute $s_1(t+1) = g_1(s_1(t))$, $t \in \{1, 2, \dots, l-1\}$; 2) as in Attack 1, step 1, compute the input sequences of subnetwork N' of the network N by method DSS; 3) by any of them $y_1(1)y_1(2)\dots y_1(l)$ and the internal sequence $s_1(1)s_1(2)\dots s_1(l)$ of the automaton A_1, construct a partially defined function f_1' as $f_1'(s_1(t)) = y_1(t)$ for $t \in \{1, 2, \dots, l\}$; 4) in the class F_1, find a function f_1 which is an extension of f_1' and, in case of success of this operation, give f_1 as one of the values of the key k.

Remark: to obtain all the values of the key k under which the cryptautomaton produces γ, the construction in the step 3 is executed for every sequence computed in the step 2.

5. $I_{\mathrm{s}} = I_{\mathrm{t}} = \varnothing$, $I_{\mathrm{o}} = \{2\}$; $K_{\mathrm{s}} = K_{\mathrm{t}} = \varnothing$, $K_{\mathrm{o}} = F_2$; $K = K_{\mathrm{o}} = F_2$; $k = f_2 \in K$.

A t t a c k 5: 1) compute $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ in the automaton A_1 and $s_3(t+1) = g_3(y_1(t), s_3(t))$, $y_3(t) = f_3(y_1, s_3(t))$ in the automaton A_3 for $t \in \{1, 2, \ldots, l\}$; 2) construct a partially defined function f_2' as $f_2'(y_1(t), s_2(t)) = y(t) \oplus y_3(t)$ for $t \in \{1, 2, \ldots, l\}$; 3) in the class F_2, find a function f_2 which is an extension of f_2' and, in case of success of this operation, give f_2 as one of the values of the key k.

Remark: the attack remains valid after exchanging roles of A_2 and A_3 in it.

6. $I_{\mathrm{s}} = I_{\mathrm{t}} = \varnothing$, $I_{\mathrm{o}} = \{2, 3\}$; $K_{\mathrm{s}} = K_{\mathrm{t}} = \varnothing$, $K_{\mathrm{o}} = F_2 \times F_3$; $K = K_{\mathrm{o}} = F_2 \times F_3$; $k = f_2 f_3 \in K$.

A t t a c k 6: 1) compute $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ in the automaton A_1 for $t \in \{1, 2, \ldots, l\}$, $s_2(t+1) = g_2(y_1(t), s_2(t))$ in the automaton A_2, and $s_3(t+1) = g_3(y_1(t), s_3(t))$ in the automaton A_3 for $t \in \{1, 2, \ldots, l-1\}$; 2) compute 2^l pairs of sequences $y_{2j}(1)y_{2j}(2)\ldots y_{2j}(l)$, $y_{3j}(1)y_{3j}(2)\ldots y_{3j}(l)$, $j \in \{1, 2, \ldots, l\}$, such that $y_{2j}(t) = y_{3j}(t) = 0 \vee y_{2j}(t) = y_{3j}(t) = 1$ if $y(t) = 0$ or $(y_{2j}(t) = 0, y_{3j}(t) = 1) \vee (y_{2j}(t) = 1, y_{3j}(t) = 0)$ if $y(t) = 1$; 3) for each $j \in \{1, 2, \ldots, l\}$, construct partial Boolean functions f_{2j} and f_{3j} as $f_{2j}(y_1(t), s_2(t)) = y_{2j}(t)$ and $f_{3j}(y_1(t), s_3(t)) = y_{3j}(t)$, $t \in \{1, 2, \ldots, l\}$; 4) in the classes F_2 and F_3, find some functions f_2 and f_3 respectively which are the extensions of f_{2j} and f_{3j} respectively and, in case of success of this operation, give $f_2 f_3$ as one of the values of the key k.

Computational complexity of the attack equals 2^l.

Remark: if, in the step 4 for every j, at least one of the functions f_{2j} or f_{3j} is not completed in the corresponding class, F_2 or F_3, then the cryptanalysis problem for the cryptautomaton Σ hasn't solution in this case.

REFERENCES

1. *Agibalov G. P.* Kriptoavtomaty s funktsional'nymi klyuchami [Cryptautomata with functional keys]. Prikladnaya Diskretnaya Matematika, 2017, no. 36, pp. 59–72. (in Russian)
2. *Agibalov G. P. and Pankratova I. A.* O dvukhkaskadnykh konechno-avtomatnykh kriptograficheskikh generatorakh i metodakh ikh kriptoanaliza [About 2-cascade finite automata cryptographic generators and their cryptanalysis]. Prikladnaya Diskretnaya Matematika, 2017, no. 35, pp. 38–47. (in Russian)
3. *Menezes A., van Oorshot P., and Vanstone S.* Handbook of Applied Cryptography. CRC Press Inc., 1997. 661 p.
4. *Agibalov G. P.* Logicheskie uravneniya v kriptoanalize generatorov klyuchevogo potoka [Logical equations in cryptanalysis of key stream generators]. Vestnik TSU. Prilozhenie, 2003, no. 6, pp. 31–41. (in Russian)