МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

КРИПТОАВТОМАТЫ С ФУНКЦИОНАЛЬНЫМИ КЛЮЧАМИ[1]

Г. П. Агибалов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Определяется понятие криптографического автомата (называемого также криптоавтоматом) как некоторого класса C автоматных сетей с фиксированной структурой N, построенных с помощью операций последовательного, параллельного и с обратной связью соединений инициальных конечных автоматов с функциями переходов и выходов, принадлежащими произвольным функциональным классам. Ключ криптоавтомата может содержать в себе начальные состояния и функции переходов и выходов некоторых компонент в N так, что задание любого конкретного ключа k влечёт за собой выбор вполне определённой автоматной сети N_k в C в качестве нового криптографического алгоритма. В случае обратимого автомата сети этот алгоритм может выступать в роли алгоритма шифрования. Функционирование сети N_k в дискретном времени описывается канонической системой уравнений конечного автомата, сопоставляемого этой сети. Её структура описывается системой уравнений, объединяющей в себе системы канонических уравнений компонент сети. Криптоанализ криптоавтомата осуществляется путём решения функциональной или структурной системы уравнений сети N_k и доопределения возникающих при этом частичных функций её компонент в заданных классах. В роли одного из инструментов решения автоматных уравнений используется метод DSS, который в применении к некоторой системе уравнений E является итерацией следующей тройки действий: 1) E разделяется (Divided) на две подсистемы E' и E'', где E' легко решаема; 2) E' решается (Solved); 3) решение E' подставляется (Substituted) в E''. В криптографических системах криптоавтоматы находят применение в качестве их компонент — криптографических генераторов, блоков замены, фильтров, комбайнеров, ключевых хеш-функций, а также систем шифрования, симметричных и с открытым ключом, и схем цифровой подписи. Определение и криптоанализ иллюстрируются на примере автономного криптоавтомата, обобщающего известную схему криптографического генератора с альтернативным управлением, или с перемежающимся шагом, построенного на регистрах сдвига с линейной обратной связью. Представлен ряд атак на этот криптоавтомат с ключами различных типов, сочетающих в себе комбинаторные или функциональные свойства, выраженные начальными состояниями или функциями выходов компонент в схеме криптоавтомата.

Ключевые слова: *конечный автомат, автоматная сеть, криптоавтомат, криптоавтомат с альтернативным управлением, криптоанализ, метод DSS, доопределение частичных функций.*

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

DOI 10.17223/20710410/36/5

CRYPTAUTOMATA WITH FUNCTIONAL KEYS

G. P. Agibalov

National Research Tomsk State University, Tomsk, Russia

E-mail: agibalov@isc.tsu.ru

In this paper, we describe the cryptautomata and some cryptanalysis techniques for them. In cryptographic systems, the cryptautomata are widely used as its primitives including key-stream generators, s-boxes, cryptofilters, cryptocombiners, key hash functions as well as symmetric and public-key ciphers, digital signature schemes. Here, 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. The choosing a certain key k produces a certain network N_k from C to be a 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 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 cryptautomaton with the alternative control. 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, cryptautomaton generator with alternative control, cryptanalysis, linearization attack, "devide-and-solve-and-substitute", partially defined function completion.*

1. Автоматные сети

В современной криптографии важное место занимают криптосистемы и их компоненты, представляющие собой сети, построенные с помощью операций последовательного, параллельного и с обратной связью соединений инициальных конечных автоматов. Первые две операции бинарные. Для любых двух автоматов $A_i = (X_i, S_i, Y_i, g_i, f_i, s_i(1))$ с конечными входными алфавитами X_i, выходными алфавитами Y_i, множествами состояний S_i, начальными состояниями $s_i(1)$ и с функциями переходов и выходов соответственно $g_i : X_i \times S_i \to S_i$ и $f_i : X_i \times S_i \to Y_i$, $i \in \{1, 2\}$, их результатами являются автоматы $A = (X, S, Y, g, f, s(1))$, в которых $S = S_1 \times S_2$, $s(1) = s_1(1)s_2(1)$ и для любых $x \in X$ и $s = s_1 s_2 \in S$:

— для последовательного соединения $A_1 \cdot A_2$ верно $X = X_1$, $Y = Y_2$, $g(x, s) = (g_1(x, s_1), g_2(f_1(x, s_1), s_2))$ и $f(x, s) = f_2(f_1(x, s_1), s_2)$;

— для параллельного соединения $(A_1 \parallel A_2)h$ с отображением связи $h : Y_1 \times Y_2 \to Y$ верно $X = X_1 = X_2$, $g(x,s) = (g_1(x,s_1), g_2(x,s_2))$ и $f(x,s) = h(f_1(x,s_1), f_2(x,s_2))$.

Третья операция унарная: в автомате A сети hA_1, являющейся результатом её применения к автомату A_1 с отображением связи $h : X \times Y_1 \to X_1$, верно $S = S_1$, $Y = Y_1$, $g(x,s) = g_1(h(x, f_1(s)), s)$, $f(x,s) = f_1(s)$ для произвольного X и любых $x \in X$ и $s \in S$.

В определении последнего соединения предполагается, что A_1 есть автомат Мура, в нём функция выходов зависит только от состояния.

Формально автоматная сеть и её компоненты определяются индуктивно следующим образом:

1. Всякий конечный автомат является автоматной сетью и одновременно её единственной компонентой.

2. Последовательное, параллельное и с обратной связью соединения автоматных сетей есть автоматная сеть. Её компонентами являются компоненты этих сетей.

3. Других автоматных сетей нет.

По определению, всякая автоматная сеть однозначно определяется своими компонентами и отображениями связи между ними.

2. Канонические уравнения

Функционально любую автоматную сеть можно описать системой уравнений, называемой её канонической системой уравнений (КСУ) и определяемой по индукции построения сети следующим образом.

1. Каноническая система уравнений автомата $A = (X, S, Y, g, f, s(1))$ есть система уравнений от переменных $x(t), s(t), y(t)$, $t = 1, 2, \ldots$, со значениями в X, S, Y соответственно, записываемая как

$$\begin{aligned} &y(t) = f(x(t), s(t)),\\ &s(t+1) = g(x(t), s(t)),\ t \geqslant 1,\\ &s(1) \text{ — начальное условие.} \end{aligned}$$

Далее эта система обозначается КСУ$(A; x, s, y, s(1))$.

2. Каноническая система уравнений сети $A_1 \cdot A_2$ записывается в виде

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; y_1, s_2, y, s_2(1))$$

и обозначается КСУ$(A_1 \cdot A_2; x, s_1, s_2, y, s_1(1)s_2(1)$.

3. Каноническая система уравнений сети $(A_1 \parallel A_2)h$ записывается в виде

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; x, s_2, y_2, s_2(1)),\ y(t) = h(y_1(t), y_2(t)),\ t \geqslant 1,$$

и обозначается КСУ$((A_1 \parallel A_2)h; x, s_1, s_2, y, s_1(1)s_2(1))$.

4. Каноническая система уравнений сети hA_1 записывается в виде

$$\text{КСУ}(A_1; x_1, s_1, y_1, s_1(1)),\ x_1(t) = h(x(t), y_1(t)),\ y(t) = y_1(t),\ t \geqslant 1,$$

и обозначается КСУ$(hA_1; x, s_1, y, s_1(1))$.

По определению, переменные в КСУ любой автоматной сети N подразделяются на входную — x, выходную — y, внутренние — s_i (они же переменные состояний компонент в N) и вспомогательные — x_j, y_k (посредством их осуществляется связь между компонентами в N). Внутренние и вспомогательные переменные называются иногда промежуточными. Переменная t трактуется как дискретное время, а $u(t)$ — как значение любой другой переменной u в момент времени t. Значение переменной состояния некоторой компоненты сети в момент $t = 1$ называется начальным состоянием этой компоненты.

3. Определение криптоавтомата

Понятие криптоавтомата восходит к работе [1], где под ним подразумевается конечный автомат с ключом и приводятся примеры описания такими криптоавтоматами генераторов ключевого потока MUGI и KNOT — в поточных шифрах, блоков замены L, M, R, S и управляющего ими блока stepping — в японской шифрмашине Purple и симметричного конечно-автоматного шифра Закревского. Здесь мы расширяем это понятие, подразумевая под криптоавтоматом класс автоматных сетей с ключом, который может включать в себя и начальные состояния компонент сети, и их функции переходов и выходов так, что любое фиксирование значения ключа выделяет в классе некоторую конкретную сеть для выполнения соответствующего криптографического преобразования.

Условимся далее множество всех функций, имеющих одинаковые области соответственно определения и значений и обладающих некоторыми фиксированными свойствами, называть (функциональным) классом. Так, можно говорить, например, о классах булевых функций, зависящих от одних и тех же множеств переменных и обладающих ограниченной сложностью задания или вычисления и одинаковыми криптографическими свойствами — нелинейностью, корреляционной иммунностью и т. п. Аналогично, можно говорить и об автоматных классах, или классах автоматов с функциями переходов и выходов из некоторых фиксированных функциональных классов. Наконец, можно говорить и о классах автоматных сетей, в которых между компонентами разных сетей существует взаимно однозначное соответствие, такое, что соответствующие компоненты принадлежат одному и тому же автоматному классу. Таким образом, автоматные сети из одного и того же класса имеют общую структуру (схему соединений компонент) и могут различаться лишь начальными состояниями компонент и их функциями переходов и выходов в своих классах. Принимая за ключ конкретные значения этих величин (начальных состояний и (или) функций переходов и выходов) в каких-либо компонентах автоматных сетей класса, мы тем самым выделяем в этом классе конкретную автоматную сеть, подобно тому как, фиксируя значение ключа в криптосистеме, мы превращаем её в конкретный криптоалгоритм. Эти рассуждения приводят нас к следующим определениям.

Определим *класс автоматной сети* N как множество $C(N)$ всех автоматных сетей, которые получаются из N с помощью операций замены в N начальных состояний некоторых компонент, и (или) функций переходов некоторых компонент, и (или) функций выходов некоторых компонент с сохранением их функциональных классов. Ясно, что этими операциями можно любую сеть в $C(N)$ получить из любой другой сети в $C(N)$, т. е. для любой сети N' в $C(N)$ имеет место равенство $C(N') = C(N)$, в связи с чем можно говорить о *классе автоматных сетей* как о любом таком их множестве C, в котором существует сеть N со свойством $C(N) = C$, или, что то же самое, для всех N в C верно это свойство. В дальнейшем класс автоматных сетей понимается именно в этом смысле.

Тройка $\Sigma = (C, I, K)$ называется *криптографическим автоматом*, или сокращённо *криптоавтоматом*, если C есть некоторый класс автоматных сетей, $C = C(N)$ для некоторой сети N, $I = \{I_\mathrm{s}, I_\mathrm{t}, I_\mathrm{o}\}$, где I_s, I_t и I_o суть множества, некоторые из которых непустые и состоят из номеров компонент в N, чьи начальные состояния (s — state), функции переходов (t — transition) и функции выходов (o — output) соответственно составляют ключ криптоавтомата, и K — множество всех возможных значений этого ключа. В нём C называется *сетевым классом*, I — *носителем ключа* и K — *ключевым пространством* криптоавтомата. Так, пусть N состоит из компонент $A_1, A_2, \ldots, A_r$, где

A_i для каждого $i \in \{1, 2, \ldots, r\}$ есть автомат с множеством состояний S_i и с функциями переходов и выходов из некоторых функциональных классов G_i и F_i соответственно. Тогда $I_\mathrm{s}, I_\mathrm{t}, I_\mathrm{o}$ суть подмножества в $\{1, 2, \ldots, r\}$, такие, что для любого $i \in \{1, 2, \ldots, r\}$ если $i \in I_\mathrm{s}$, $i \in I_\mathrm{t}$ или $i \in I_\mathrm{o}$, то соответственно для каждого $s \in S_i$, $g \in G_i$ или $f \in F_i$ найдётся сеть в C, в i-й компоненте которой s является начальным состоянием, g — функцией переходов или f — функцией выходов соответственно, а если $i \notin I_\mathrm{s}$, $i \notin I_\mathrm{t}$ или $i \notin I_\mathrm{o}$, то соответственно начальные состояния, функции переходов или функции выходов i-й компоненты во всех сетях в C одинаковы. Кроме того, ключевое пространство криптоавтомата Σ равно $K = K_\mathrm{s} \times K_\mathrm{t} \times K_\mathrm{o}$, где $K_\mathrm{s} = \prod\limits_{i \in I_\mathrm{s}} S_i$, $K_\mathrm{t} = \prod\limits_{i \in I_\mathrm{t}} G_i$, $K_\mathrm{o} = \prod\limits_{i \in I_\mathrm{o}} F_i$ и $\prod$ — символ декартова произведения.

Каждый ключ $k = k_\mathrm{s} k_\mathrm{t} k_\mathrm{o}$ в ключевом пространстве K криптоавтомата Σ состоит из трёх частей, не все из которых пустые: k_s — набор значений начальных состояний компонент криптоавтомата с номерами в I_s, k_t — набор функций переходов компонент криптоавтомата с номерами в I_t и k_o — набор функций выходов компонент криптоавтомата с номерами в I_o. Его задание выделяет в C однозначно некоторую конкретную сеть N_k, в которой начальные состояния компонент A_i для $i \in I_\mathrm{s}$ принадлежат набору k_s, функции переходов компонент A_i, $i \in I_\mathrm{t}$, — набору k_t и функции выходов компонент A_i, $i \in I_\mathrm{o}$, — набору k_o.

Мы допускаем к рассмотрению и автономные автоматы (в них функции переходов и выходов зависят только от состояния), и комбинационные автоматы (в них функция выходов не зависит от состояний и нет нужды ни в множестве последних, ни в функции переходов), и обратимые автоматы (в них входная последовательность однозначно определяется по выходной и, возможно, по начальному состоянию). По определению, каждая автоматная сеть является конечным автоматом, поэтому все понятия, введённые для автоматов, переносятся на автоматные сети и на криптоавтоматы, построенные на основе их классов. Так, в случае комбинационности, автономности или обратимости автомата сети сама сеть также называется комбинационной, автономной или обратимой соответственно. Можно говорить, следовательно, и о криптоавтоматах такого рода.

4. Примеры криптоавтоматов

Кроме приведённых выше примеров однокомпонентных криптоавтоматов, под определение криптоавтомата на основе класса автоматных сетей подпадают и роторные машины, включая Энигму, и уже упоминавшаяся пурпурная шифрсистема Purple, и конечно-автоматные криптосистемы с открытым ключом для шифрования и цифровой подписи семейства FAPKC [1, 2].

Комбинационные криптоавтоматы обычно состоят из комбинационных компонент и в таком составе часто могут применяться в роли блоков замены в тех блочных шифрах с аддитивными раундовыми ключами, которые являются листьями «дерева», выросшего из корня DES. Более того, сами шифры на этом дереве допускают описание комбинационными криптоавтоматами. Чтобы это понять, достаточно заметить, что сложение бит раундового ключа со значениями на входе блока замены превращает функции последнего в функции, зависящие ещё и от ключа шифра.

По определению, каждая автономная сеть является последовательным соединением автоматных сетей, в котором на месте первой компоненты (автомата A_1) выступает автономный автомат. Автономные сети порождают последовательности выходных символов, и основанные на них криптоавтоматы мы называем конечно-автоматными

криптографическими генераторами. Огромное множество примеров таких криптоавтоматов доставляют конечно-автоматные обобщения криптографических генераторов, построенных в виде схем из регистров сдвига с линейными обратными связями (LFSRs). Одно такое обобщение рассмотрено в [3, 4]. Далее, в п. 8, мы рассмотрим криптоавтоматное обобщение ещё одного генератора на LFSRs — криптографического генератора с альтернативным управлением [5], называемого в [6] the alternating step generator и в [7] — генератором с перемежающимся шагом.

5. Задачи криптоанализа

Предполагается, что криптоаналитику, исследующему криптоавтомат (C, I, K), известны все составляющие его множества C, I и K. Для автономного криптоавтомата $\Sigma = (C, I, K)$ есть только одна задача криптоанализа: по кратчайшему отрезку $\gamma = y(1)y(2)\dots y(l)$, $l \geqslant 1$, последовательности на выходе сети N_k в C узнать его ключ $k \in K$. Задача фактически распадается на три других: 1) сформулировать необходимые и достаточные условия единственности автоматной сети в C, порождающей γ; 2) разработать метод, позволяющий по любому такому отрезку γ построить каждую из сетей в C, которая может сгенерировать эту γ; 3) по любой сети N_k в C, порождающей γ, вычислить ключ k. Умея это делать, можно сформулированную задачу криптоанализа автономного криптоавтомата $\Sigma = (C, I, K)$ решать следующим образом: последовательность на выходе неизвестной сети N_k в C наблюдается до тех пор, пока не будет получен отрезок γ, для которого выполнены условия единственности в задаче 1, после чего методом в задаче 2 строится искомая сеть, а потом методом задачи 3 по этой сети вычисляется искомый ключ.

Для неавтономного криптоавтомата $\Sigma = (C, I, K)$ можно указать по меньшей мере две задачи криптоанализа: полное раскрытие — по заданным отрезкам $\alpha = x(1)x(2)\dots x(l)$ и $\gamma = y(1)y(2)\dots y(l)$, $l \geqslant 1$, соответствующих входной и выходной последовательностей неизвестной сети N_k в C вычислить ключ k, и раскрытие сообщения — по отрезку γ на выходе сети N_k вычислить отрезок α на её входе, который она преобразует в γ. Оптимизационный вариант каждой из этих задач формулируется аналогично задаче криптоанализа автономного криптоавтомата.

Раскрытие ключа в любом случае возможно тривиальным методом — так называемой атакой грубой силы (от англ. Brute-Force Attack (BFA)), состоящей в опробовании каждого ключа в K на соответствие определяемой им сети в C исходным данным: выходной последовательности — в автономном случае или выходной и входной последовательностей — в неавтономном случае. Вычислительная сложность этого метода, естественно, оценивается мощностью ключевого пространства криптоавтомата.

6. Доопределение частичной функции в заданном классе

Для произвольного класса F функций $f : V^n \to V$ над конечным множеством (алфавитом) V и для произвольной частичной функции $f' : D_{f'} \subset V^n \to V$ мы говорим, что f' *доопределима в классе* F, если существует функция $f \in F$, такая, что $f'(v) = f(v)$ для всех $v \in D_{f'}$; в этом случае говорят, что f' *доопределима до* f.

В криптоанализе криптографических систем с функциональными ключами, в том числе и криптоавтоматов, возникает проблема доопределения некоторых частичных функций до функций в заданном классе F, состоящая в выяснении существования и единственности такого доопределения и в построении всевозможных функций в классе, до которых доопределима заданная частичная функция. Решение этой проблемы для любого класса F возможно тривиальным методом — так называемым исчерпывающим

поиском (от англ. Exhaustive Search Method (ESM)), который заключается в проверке для каждой функции в F, доопределима ли до неё заданная частичная функция. Вычислительная сложность этого метода, естественно, оценивается мощностью класса F. Других общих методов решения данной проблемы (для всех классов F) неизвестно. В [8, 9] можно найти её нетривиальное решение для класса F булевых функций от n переменных, в котором булевы функции зависят существенно от ограниченного числа $k < n$ последних.

Здесь мы продемонстрируем эту проблему в криптоанализе уже упоминавшегося криптоавтоматного генератора с альтернативным управлением. Основным инструментом в этом криптоанализе служит метод DSS решения систем автоматных уравнений.

7. Метод DSS

7.1. Определение

Подмножество L переменных в некоторой системе уравнений E над конечным алфавитом называется *эффективным множеством*, если фиксирование любых возможных значений этих переменных превращает E в легко решаемую (например, за полиномиальное или меньшее время) систему (ЛРС). В частности, таковым является подмножество переменных, при любом фиксировании которых все уравнения в системе над конечным полем превращаются в линейные. Оно называется линеаризационным множеством переменных системы [5, 10].

Система уравнений E над k-элементным алфавитом с q-элементным эффективным множеством переменных L решается со сложностью k^q методом DS (Devide and Solve), или по-русски «разделяй и решай», состоящим в подстановке в систему E поочерёдно различных наборов значений переменных в L и в решении получаемой каждый раз ЛРС. В случае совместности последней её решение, взятое вместе с подставленным в E набором значений переменных в L, является решением системы E. Метод DS на основе линеаризационного множества переменных со значениями в конечном поле называется линеаризационной атакой [5, 10]. В нём решается частный случай ЛРС — система линейных уравнений.

Система уравнений E называется *рекурсивно легко решаемой системой* (РЛРС), если в ней существует непустая подсистема уравнений E' с небольшим эффективным подмножеством (ныне это не более 3–4 десятков) переменных, такая, что подсистема уравнений $E \setminus E'$ подстановкой в неё любого решения подсистемы E' преобразуется в РЛРС. По определению, такая система решается кратным применением метода DS к её подсистеме и подстановки полученных решений подсистемы в её дополнение. В [3] данный метод решения РЛРС назван DSS — Devide, Solve and Substitute («разделяй, решай и подставляй»).

Можно сказать, что метод DS является частным случаем метода DSS, а именно методом DSS с однократным применением метода DS. Таким образом, и линеаризационная атака на систему уравнений над конечным полем — это тоже частный случай метода DSS.

Далее мы продемонстрируем этот метод применительно к каноническим системам уравнений конечных автоматов и автоматных сетей над полем $\mathbb{F}_2$ из двух элементов. Переложение излагаемого материала на автоматные системы уравнений над произвольным алфавитом V возможно простой заменой $\mathbb{F}_2$ на V, которая не меняет существа метода и лишь увеличивает его вычислительную сложность.

7.2. Решение канонической системы уравнений автомата

Покажем, как методом DSS может быть решена каноническая система уравнений E произвольного конечного автомата $A = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g, f, s(1))$ над $\mathbb{F}_2$ при известных функциях g и f, заданных начальном состоянии $s(1) \in \mathbb{F}_2^m$ и выходной последовательности $y(1)y(2)\dots y(l) \in \mathbb{F}_2^l$ и неизвестной входной последовательности $x(1)x(2)\dots x(l) \in \mathbb{F}_2^l$:

$$y(t) = f(x(t), s(t)),\ s(t+1) = g(x(t), s(t)),\ 1 \leqslant t \leqslant l.$$

В самом деле, при $t = 1$ имеем уравнение $y(1) = f(x(1), s(1))$, которое для $x(1)$ имеет либо два решения — 0 и 1, если $y(1) = f(0, s(1)) = f(1, s(1))$, либо одно решение — некоторое $b(1) \in \mathbb{F}_2$, если $y(1) = f(b(1), s(1)) \neq f(\neg b(1), s(1))$, либо ни одного решения, если $y(1) \neq f(0, s(1)) = f(1, s(1))$. В последнем случае система E несовместна. При $2 \leqslant t \leqslant l$ для каждого из найденных решений для $x(1)x(2)\dots x(t-1)$ вычисляем $s(t) = g(x(t-1), s(t-1))$ и получаем уравнение $y(t) = f(x(t), s(t))$, которое тоже может иметь одно ($b(t)$), два (0 и 1) или ни одного решения для $x(t)$. В последнем случае рассматриваемый вариант частичного решения $x(1)x(2)\dots x(t-1)$ отвергается, а в остальных случаях из него получаются соответственно один или два варианта расширенного частичного решения $x(1)x(2)\dots x(t-1)x(t)$ — соответственно с $x(t) = b(t)$ или с $x(t) = 0$ и с $x(t) = 1$. Если на некотором шаге с номером $t \leqslant l$ множество полученных частичных решений оказывается пустым, система E объявляется несовместной; в противном случае все её решения для $x(1)x(2)\dots x(l)$ находятся на l-м шаге. Алгоритмически метод представляет собой обход дерева решений высоты не более l со степенями исхода вершин не более 2. В нём на путях от корня к листьям, имеющих длину l, лежат все решения КСУ автомата — все его входные последовательности $x(1)x(2)\dots x(l)$, преобразуемые им в выходную последовательность $y(1)y(2)\dots y(l)$.

В случае, если A есть автомат Мура (в нём функция выходов зависит от входного символа фиктивно), то уравнение $y(t) = f(x(t), s(t))$ в его канонической системе имеет для $x(t)$ два решения (0 и 1) или ни одного, вследствие чего в её дереве решений из каждой неконцевой вершины исходят ровно две дуги.

Заметим также, что если A является автоматом некоторой автоматной сети, то метод DSS по выходной последовательности этой сети находит её возможные входные последовательности. Кроме того, если A есть некоторая компонента автоматной сети над $\mathbb{F}_2$, то метод DSS вычисляет входные последовательности этой компоненты по её выходной последовательности в сети. Далее мы покажем, как эти задачи решаются методом DSS непосредственно по канонической системе уравнений автоматной сети без построения её автомата. Ввиду индуктивности определения автоматных сетей достаточно показать это для базовых операций их построения — последовательного, параллельного и с обратной связью соединений автоматов над $\mathbb{F}_2$.

7.3. Решение канонической системы уравнений последовательной автоматной сети

Начнём с рассмотрения канонической системы уравнений последовательного соединения автоматов $A_1 \cdot A_2$ над F_2: КСУ$(A_1; x, s_1, y_1, s_1(1))$, КСУ$(A_2; y_1, s_2, y, s_2(1))$. В ней подсистема

$$y(t) = f_2(y_1(t), s_2(t)),\ s_2(t+1) = g_2(x_2(t), s_2(t)),\ 1 \leqslant t \leqslant l,$$

с начальным условием $s_2(1)$ является канонической системой уравнений автомата A_2 и может быть решена методом DSS, как только что описано, при известных его функ-

циях g_2 и f_2 и заданных его начальном состоянии $s_2(1) \in \mathbb{F}_2^{m_2}$ и выходной последовательности $y(1)y(2)\dots y(l) \in \mathbb{F}_2^l$. В результате будут найдены в $\mathbb{F}_2^l$ возможные входные последовательности $y_1(1)y_1(2)\dots y_1(l)$ автомата A_2, среди которых присутствуют и все те выходные последовательности автомата A_1 с неизвестными атрибутами $s_1(1), g_1, f_1$, которые он может выработать в ответ на входные последовательности сети, вызывающие на её выходе последовательность $y(1)y(2)\dots y(l)$.

При известных $s_1(1), g_1, f_1$ аналогичным образом по выходной последовательности $y_1(1)y_1(2)\dots y_1(l)$ автомата A_1 вычисляются его входные последовательности $x_1(1)x_1(2)\dots x_1(l)$, являющиеся одновременно входными последовательностями сети.

7.4. Решение канонической системы уравнений параллельной автоматой сети

Рассмотрим теперь каноническую систему уравнений $E = \text{КСУ}((A_1 \parallel A_2)h; x, s_1, s_2, y, s_1(1)s_2(1))$ параллельного соединения автоматов $(A_1 \parallel A_2)h$ над $\mathbb{F}_2$ с отображением $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; x, s_2, y_2, s_2(1)),\ y(t) = h(y_1(t), y_2(t)),\ 1 \leqslant t \leqslant l.$$

Пусть далее $M_h(t) = \{(y_1^1(t), y_2^1(t)), (y_1^2(t), y_2^2(t)), \dots, (y_1^{n_t}(t), y_2^{n_t}(t))\} = h^{-1}(y(t)) = \{(y_1(t), y_2(t)) \in \mathbb{F}_2^2 : h(y_1(t), y_2(t)) = y(t)\}$, $t \geqslant 1$. Здесь $n_t = |M_h(t)| \leqslant 4$. После подстановки в систему E вместо пар переменных $(y_1(t), y_2(t))$ их возможных значений $(y_1^j(t), y_2^j(t))$ для $j \in \{1, 2, \dots, n_t\}$ из $M_h(t)$ эта система принимает вид E^j:

$$\begin{aligned}
&y(t) = h(y_1^j(t), y_2^j(t)),\\
&y_1^j(t) = f_1(x(t), s_1(t)),\\
&y_2^j(t) = f_2(x(t), s_2(t)),\\
&s_1(t+1) = g_1(x(t), s_1(t)),\\
&s_2(t+1) = g_2(x(t), s_2(t)),\ 1 \leqslant t \leqslant l,\\
&s_1(1), s_2(1) \text{ — начальные условия.}
\end{aligned}$$

Последовательно для t со значениями 1, 2, …, l из второго и третьего уравнений находим для неизвестного $x(t)$ либо два решения — 0 и 1 (говорим, что имеет место случай 2), если

$$y_1^j(t) = f_1(0, s_1(t)) = f_1(1, s_1(t)) \text{ и } y_2^j(t) = f_2(0, s_2(t)) = f_2(1, s_2(t)),$$

либо одно решение — 0 или 1 (случай 1), если

$$(y_1^j(t) = f_1(0, s_1(t))) \wedge (y_2^j(t) = f_2(0, s_2(t))) \wedge [(y_1^j(t) \neq f_1(1, s_1(t))) \vee (y_2^j(t) \neq f_2(1, s_2(t)))] \vee$$
$$\vee (y_1^j(t) = f_1(1, s_1(t))) \wedge (y_2^j(t) = f_2(1, s_2(t))) \wedge [(y_1^j(t) \neq f_1(0, s_1(t))) \vee (y_2^j(t) \neq f_2(0, s_2(t)))],$$

либо ни одного решения (случай 0), если

$$[(y_1^j(t) \neq f_1(0, s_1(t))) \vee (y_2^j(t) \neq f_2(0, s_2(t)))] \wedge [(y_1^j(t) \neq f_1(1, s_1(t))) \vee (y_2^j(t) \neq f_2(1, s_2(t)))].$$

Найденное так множество решений для $x(t)$ в E^j обозначим $B_j(t)$ и положим $B(t) = B_1(t) \cup \dots \cup B_{n_t}(t)$. По построению $B(t) \subseteq \mathbb{F}_2$. В случае $B(t) = \varnothing$ система E несовместна. В противном случае для каждого из найденных решений $x(t)$ в $B(t)$ вычисляем $s_1(t+1) = g_1(x(t), s_1(t))$ и $s_2(t+1) = g_2(x(t), s_1(t))$ и аналогичным образом находим множество $B(t+1)$ решений в $\mathbb{F}_2$ для $x(t+1)$. После l таких шагов, т. е.

при $t = l$, получаем множество $\{x(1)x(2)\dots x(l) : x(t) \in B(t), t \in \{1, 2, \dots, l\}\}$ всех решений системы уравнений E параллельного соединения автоматов. Его элементы суть всевозможные входные последовательности сети $(A_1 \parallel A_2)h$, которые она с фиксированными компонентами и их начальными состояниями преобразует в выходную последовательность $y(1)y(2)\dots y(l)$.

7.5. Решение канонической системы уравнений автоматной сети с обратной связью

Рассмотрим наконец каноническую систему уравнений $E = \text{КСУ}(hA_1; x, s_1, y, s_1(1))$ автоматного соединения hA_1 над $\mathbb{F}_2$ с обратной связью $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$

$$\text{КСУ}(A_1; x_1, s_1, y_1),\ x_1(t) = h(x(t), y_1(t)),\ y(t) = y_1(t),\ 1 \leqslant t \leqslant l.$$

В ней подсистема уравнений компоненты A_1 с $1 \leqslant t \leqslant l$ решается методом DSS, как каноническая система уравнений автомата Мура (см. п. 7.2) при заданных $s_1(1), g_1, f_1$ и выходной последовательности сети $y(1)y(2)\dots y(l)$. Результатом решения являются входные последовательности $x_1(1)x_1(2)\dots x_1(l)$ автомата A_1. По каждой из них и выходной последовательности сети методом DSS решается система уравнений

$$x_1(t) = h(x(t), y(t)),\ 1 \leqslant t \leqslant l,$$

и находятся входные последовательности сети $x(1)x(2)\dots x(l)$, а именно: для каждого целого j, $1 \leqslant j < l$, и для каждого решения $x(1)x(2)\dots x(j-1)$ её подсистемы с $1 \leqslant t \leqslant j-1$ её подсистема с $1 \leqslant t \leqslant j$ имеет либо два решения $x(1)x(2)\dots x(j-1)x(j)$ с $x(j) \in \mathbb{F}_2$, если $x_1(j) = h(0, y(j)) = h(1, y(j))$, либо одно решение с $x(j) = b$, $b \in \mathbb{F}_2$, если $x_1(j) = h(b, y(j)) \neq h(\neg b, y(j))$, либо ни одного решения, если $x_1(j) \neq h(0, y(j)) = h(1, y(j))$.

8. Криптогенератор с альтернативным управлением

8.1. Определение

Рассмотрим *автоматные сети с альтернативным управлением*, представляющие собой последовательно-параллельные соединения трёх автоматов над полем $\mathbb{F}_2$. В каждой такой сети $N = A_1 \cdot (A_2 \parallel A_3)h$ один из автоматов, A_1, управляющий, он является автономным: $A_1 = (\mathbb{F}_2^{m_1}, \mathbb{F}_2, g_1, f_1, s_1(1))$, два других, A_2 и A_3, управляемые, они неавтономные: $A_i = (\mathbb{F}_2, \mathbb{F}_2^{m_i}, \mathbb{F}_2, g_i, f_i, s_i(1))$, $i \in \{2, 3\}$. В ней выход автомата A_1 соединён со входами автоматов A_2 и A_3 непосредственно, а выходы A_2 и A_3 соединены с выходом сети через отображение связи $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$, такое, что для любых y_2, y_3 в $\mathbb{F}_2$ имеет место $h(y_2, y_3) = y_2 \oplus y_3$, где $\oplus$ — операция сложения по mod 2. Альтернативность управления автоматами A_2 и A_3 со стороны автомата A_1 в N выражается в следующих ограничениях на их функции переходов, справедливых для любого их входного символа x и любых их состояний s_2 и s_3 соответственно и называемых далее *условиями альтернативности*: $g_2(x, s_2) = s_2 \Leftrightarrow g_3(x, s_3) \neq s_3$.

Автомат сети $N = A_1 \cdot (A_2 \parallel A_3)h$ с альтернативным управлением есть $A = (\mathbb{F}_2^{m_1} \times \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}, \mathbb{F}_2, g, f, s_1(1)s_2(1)s_3(1))$, и в нём для любых $s_i \in \mathbb{F}_2^{m_i}$, $i \in \{1, 2, 3\}$, верно: $g(s_1s_2s_3) = (g_1(s_1), g_2(f_1(s_1), s_2), g_3(f_1(s_1), s_3))$ и $f(s_1s_2s_3) = f_2(f_1(s_1), s_2) \oplus f_3(f_1(s_1), s_3)$.

Каноническая система уравнений E этой сети записывается как

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

а каноническая система уравнений её автомата A — как

$$\begin{aligned}
&y(t) = f_2(f_1(s_1(t)), s_2(t)) \oplus f_3(f_1(s_1(t)), s_3(t)),\\
&s_1(t+1)s_2(t+1)s_3(t+1) = (g_1(s_1(t)), g_2(f_1(s_1(t)), s_2(t)), g_3(f_1(s_1(t)), s_3(t))),\ t \geqslant 1,\\
&s_1(1)s_2(1)s_3(1) \text{ — начальное условие.}
\end{aligned}$$

В системе E подсистемы E_1 из первого и второго уравнений, E_2 из третьего и четвёртого уравнений и E_3 из пятого и шестого уравнений описывают работу автоматов A_1, A_2 и A_3 соответственно, седьмое уравнение описывет выход сети. Подсистема E' из уравнений с номерами от третьего до седьмого включительно и с начальными условиями $s_2(1)$ и $s_3(1)$ описывает работу параллельной подсети $(A_2 \parallel A_3)h$ сети N. Входные последовательности этой подсети можно вычислить по выходной последовательности сети применением к E' метода DSS, как это описано в п. 7.4 со следующими уточнениями, связанными со специфичностью отображения связи h и свойством альтернативного функционирования автоматов A_2 и A_3 в ней:

1) множество $M_h(t)$, равное $h^{-1}(y(t))$, состоит из двух пар в $\mathbb{F}_2^2$ — 00 и 11, если $y(t) = 0$, либо 01 и 10, если $y(t) = 1$;
2) система уравнений E несовместна не только когда $B(t) = \varnothing$, но и когда

$$(s_2(t+1) \neq s_2(t)) \wedge (s_3(t+1) \neq s_3(t)) \vee (s_2(t+1) = s_2(t)) \wedge (s_3(t+1) = s_3(t))$$

для некоторого t, $1 \leqslant t \leqslant l$.

Криптоавтомат $\Sigma = (C, I, K)$ называется *криптогенератором с альтернативным управлением*, если в нём $C = C(N)$ для некоторой автоматной сети N с альтернативным управлением. По определению, всякая сеть в C является автоматной сетью $A_1 \cdot (A_2 \parallel A_3)h$ с альтернативным управлением, в ней для каждого $i \in \{1, 2, 3\}$ состояния в i-й компоненте принадлежат множеству $S_i = \mathbb{F}_2^{m_i}$, а функции переходов и выходов — некоторым функциональным классам G_i и F_i соответственно. Элементы I_s, I_t и I_o носителя I ключа в Σ являются подмножествами в $\{1, 2, 3\}$, а декартовы множители K_s, K_t и K_o ключевого пространства K — декартовыми произведениями множеств S_i для $i \in I_\mathrm{s}$, G_i для $i \in I_\mathrm{t}$ и F_i для $i \in I_\mathrm{o}$ соответственно. Наконец, в ключе $k = k_\mathrm{s}k_\mathrm{t}k_\mathrm{o} \in K$ имеем: $k_\mathrm{s} \in K_\mathrm{s}$, $k_\mathrm{t} \in K_\mathrm{t}$ и $k_\mathrm{o} \in K_\mathrm{o}$.

8.2. Криптоанализ

Пусть далее $\Sigma = (C, I, K)$ есть произвольный криптогенератор с альтернативным управлением. Предполагается, что целью криптоанализа генератора является нахождение ключа по заданной его выходной последовательности $\gamma = y(1)y(2)\ldots y(l), l \geqslant 1$.

Допуская, разумеется, существование иных методов решения этой задачи, мы остановимся здесь, главным образом, на применениях метода DSS и задачи доопределения частичных булевых функций в атаках на генератор Σ с тем или иным носителем $I = \{I_{\rm s}, I_{\rm t}, I_{\rm o}\}$ ключевого пространства $K = K_{\rm s} \times K_{\rm t} \times K_{\rm o}$. Начнём с простейшего случая.

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

А т а к а 1: 1) по заданной γ на выходе генератора методом DSS вычисляем входные последовательности параллельной подсети $N' = (A_2 \parallel A_3)h$ сети N, являющиеся одновременно выходными последовательностями автомата A_1; 2) для любой из таких последовательностей атакой грубой силы (BFA) находим начальное состояние $s_1(1)$ автомата A_1.

Вычислительная сложность атаки равна 2^{m_1}.

Примечание: если вычисление на шаге 2 выполнить для каждой последовательности, вычисленной на шаге 1, то будут получены все значения ключа k, при которых генератор Σ вырабатывает данную γ.

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

В этом случае атака относится к разряду meet-in-the-middle — встреча посредине. Предварительно, до атаки, для каждого значения a переменной $s_1(1)$ в S_1 вычисляются $s_1(t+1) = g_1(s_1(t))$ и $y_1(t) = f_1(s_1(t))$ для $t \in \{1, 2, \ldots, l\}$ и $s_1(1) = a$, и это a сохраняется по адресу $H(y_1(1)y_1(2)\ldots y_1(l))$, где $H : \mathbb{F}_2^l \to \mathbb{F}_2^{m_1}$ есть некоторая хеш-функция.

А т а к а 2: имея на выходе генератора последовательность γ, методом DSS вычисляем входные последовательности подсети N' при разных значениях переменной $s_2(1)$, выбираемых в S_2 до тех пор, пока при некотором её значении b на входе N' не будет получена некоторая последовательность β с некоторым непустым содержимым a по адресу $H(\beta)$, и тогда пару (a, b) принимаем за результат атаки — значение ключа k.

Вычислительная сложность атаки равна 2^{m_2}.

Примечание: атака остаётся в силе, если в ней автоматы A_2 и A_3 переставить местами.

3. $I_s = \{1, 2, 3\}$, $I_{\rm t} = I_{\rm o} = \varnothing$; $K_{\rm s} = S_1 \times S_2 \times S_3 = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}$, $K_{\rm t} = K_{\rm o} = \varnothing$; $K = K_{\rm s} = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}$; $k = s_1(1)s_2(1)s_3(1) \in K$, и переменные $y_1(1), y_1(2), \ldots, y_1(l)$ образуют линеаризационное множество системы уравнений E' подсети N' сети N.

А т а к а 3: для каждого $s_1(1)$ в S_1 1) вычисляем $s_1(t+1) = g_1(s_1(t))$ и $y_1(t) = f_1(s_1(t))$ для $t \in \{1, 2, \ldots, l\}$; 2) осуществляем линеаризационную атаку на E', а именно: найденные на шаге 1 значения $y_1(1), y_1(2), \ldots, y_1(l)$ подставляем в E' и полученную систему E'' линейных уравнений решаем (методом Гаусса, например) относительно булевых неизвестных в векторах $s_2(t)$ и $s_3(t)$, $t \in \{1, 2, \ldots, l\}$; 3) из каждого из тех решений системы E'', которые удовлетворяют условиям альтернативности для всех t, $1 \leqslant t \leqslant l$, выписываем значения $s_2(1)$ и $s_3(1)$ и тройку $s_1(1)s_2(1)s_3(1)$ принимаем за одно из значений ключа k.

Вычислительная сложность атаки равна 2^{m_1}.

Примечание: эффективным ключом автоматного криптогенератора с альтернативным управлением в этом случае является начальное состояние одного автомата — управляющего, его удлинение посредством начальных состояний управляемых автоматов не добавляет стойкости генератору. Для генератора с альтернативным управлением на регистрах сдвига с линейной обратной связью это показано в [5].

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

А т а к а 4: 1) вычисляем $s_1(t+1) = g_1(s_1(t)), t \in \{1, 2, \ldots, l-1\}$; 2) как в п. 1 атаки 1, методом DSS вычисляем входные последовательности подсети N' сети N; 3) по любой из них $y_1(1)y_1(2)\ldots y_1(l)$ и внутренней последовательности $s_1(1)s_1(2)\ldots s_1(l)$ автомата A_1 строим частичную функцию f_1' как $f_1'(s_1(t)) = y_1(t)$ для $t \in \{1, 2, \ldots, l\}$; 4) доопределяем f_1' до некоторой функции f_1 в классе F_1 и в случае успешности доопределения выдаём последнюю как одно из значений ключа k.

Примечание: если вычисление на шаге 3 выполнить для каждой последовательности, вычисленной на шаге 2, то будут получены все значения ключа k, при которых генератор Σ вырабатывает данную γ.

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

А т а к а 5: 1) вычисляем $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ в автомате A_1 и $s_3(t+1) = g_3(y_1(t), s_3(t))$, $y_3(t) = f_3(y_1, s_3(t))$ в автомате A_3 для $t \in \{1, 2, \ldots, l\}$; 2) строим частичную функцию f_2' как $f_2'(y_1(t), s_2(t)) = y(t) \oplus y_3(t)$ для $t \in \{1, 2, \ldots, l\}$; 3) доопределяем f_2' до некоторой функции f_2 в классе F_2 и в случае успешности доопределения выдаём последнюю как значение ключа k.

Примечание: атака остаётся в силе, если в ней автоматы A_2 и A_3 переставить местами.

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

А т а к а 6: 1) вычисляем $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ в автомате A_1 для $t \in \{1, 2, \ldots, l\}$, $s_2(t+1) = g_2(y_1(t), s_2(t))$ в автомате A_2 и $s_3(t+1) = g_3(y_1(t), s_3(t))$ в автомате A_3 для $t \in \{1, 2, \ldots, l-1\}$; 2) вычисляем 2^l пар последовательностей $y_{2j}(1)y_{2j}(2)\ldots y_{2j}(l)$, $y_{3j}(1)y_{3j}(2)\ldots y_{3j}(l)$, $j \in \{1, 2, \ldots, l\}$, таких, что $y_{2j}(t) = y_{3j}(t) = 0$ или $y_{2j}(t) = y_{3j}(t) = 1$, если $y(t) = 0$, либо $y_{2j}(t) = 0, y_{3j}(t) = 1$ или $y_{2j}(t) = 1, y_{3j}(t) = 0$, если $y(t) = 1$; 3) для каждого $j \in \{1, 2, \ldots, l\}$ строим частичные булевы функции f_{2j} и f_{3j} как $f_{2j}(y_1(t), s_2(t)) = y_{2j}(t)$ и $f_{3j}(y_1(t), s_3(t)) = y_{3j}(t)$, $t \in \{1, 2, \ldots, l\}$, доопределяем их до некоторых функций f_2 и f_3 в F_2 и F_3 соответственно и в случае успешности доопределения выдаём $f_2 f_3$ как одно из значений ключа k.

Вычислительная сложность атаки равна 2^l.

Примечание: если на шаге 3 для каждого j хотя бы одна из функций f_{2j} или f_{3j} оказывается недоопределимой в соответствующем классе F_2 или F_3, то задача криптоанализа генератора Σ в данных условиях не имеет решения.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Конечные автоматы в криптографии // Прикладная дискретная математика. Приложение. 2009. № 2. С. 43–73.
2. *Tao R.* Finite Automata and Application to Cryptography. TSINGHUA University Press, 2009. 406 p.
3. *Агибалов Г. П., Панкратова И. А.* О двухкаскадных конечно-автоматных криптографических генераторах и методах их криптоанализа // Прикладная дискретная математика. 2017. № 35. С. 38–47.
4. *Агибалов Г. П., Панкратова И. А.* К криптоанализу двухкаскадных конечно-автоматных криптографических генераторов // Прикладная дискретная математика. Приложение. 2016. № 9. С. 41–43.
5. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 31–41.
6. *Menezes A., van Oorshot P., and Vanstone S.* Handbook of Applied Cryptography. CRC Press Inc., 1997. 661 p.

7. *Фомичёв В. М.* Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003. 400 с.
8. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. 1970. Вып. 49. С. 12–19.
9. *Агибалов Г. П., Сунгурова О. Г.* Криптоанализ конечно-автоматного генератора ключевого потока с функцией выходов в качестве ключа // Вестник Томского государственного университета. Приложение. 2006. № 17. С. 104–108.
10. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского государственного университета. Приложение. 2006. № 17. С. 4–9.

REFERENCES

1. *Agibalov G. P.* Konechnye avtomaty v kriptografii [Finite automata in cryptography]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 43–73. (in Russian)
2. *Tao R.* Finite Automata and Application to Cryptography. TSINGHUA University Press, 2009. 406 p.
3. *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)
4. *Agibalov G. P. and Pankratova I. A.* K kriptoanalizu dvukhkaskadnykh konechno-avtomatnykh kriptograficheskikh generatorov [To cryptanalysis of 2-cascade finite automata cryptographic generators]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 41–43. (in Russian)
5. *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)
6. *Menezes A., van Oorshot P., and Vanstone S.* Handbook of Applied Cryptography. CRC Press Inc., 1997. 661 p.
7. *Fomichev V. M.* Diskretnaya matematika i kriptologiya [Discrete Mathematics and Cryptology]. Moscow, DIALOG-MEPhI Publ., 2003. 400 p. (in Russian)
8. *Agibalov G. P.* O nekotorykh doopredeleniyakh chastichnoy bulevoy funktsii [Some completions of partial Boolean function]. Trudy SPhTI, 1970, iss. 49, pp. 12–19. (in Russian)
9. *Agibalov G. P. and Sungurova O. G.* Kriptoanaliz konechno-avtomatnogo generatora klyuchevogo potoka s funktsiey vykhodov v kachestve klyucha [Cryptanalysis of a finite-state keystream generator with an output function as a key]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 104–108. (in Russian)
10. *Agibalov G. P.* Metody resheniya sistem polinomial'nykh uravneniy nad konechnym polem [Methods for solving systems of polynomial equations over a finite field]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 4–9. (in Russian)