МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

КРИПТОАНАЛИЗ ДВУХКАСКАДНОГО КОНЕЧНО-АВТОМАТНОГО ГЕНЕРАТОРА С ФУНКЦИОНАЛЬНЫМ КЛЮЧОМ[1]

И. В. Боровкова, И. А. Панкратова, Е. В. Семенова

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Рассматривается криптографический генератор $G = A_1 \cdot A_2$, представляющий собой последовательное соединение двух абстрактных конечных автоматов A_1 и A_2 над полем $\mathbb{F}_2$. Ключом генератора является функция f_1 выходов автомата A_1 и, возможно, начальные состояния автоматов. Задача криптоанализа генератора G состоит в определении его ключа по заданному отрезку $\gamma = z(1)z(2)\dots z(l)$ его выходной последовательности. Описаны алгоритмы анализа автомата A_2 в общем случае и для конечно-автоматного генератора (δ, τ)-шагов, позволяющие найти поступающий на вход автомата A_2 прообраз $u(1)\dots u(l)$ последовательности γ. Значения $u(t)$ суть значения функции f_1 на наборах $x(t)$, $t = 1, 2, \dots, l$, где $x(t)$ — состояние автомата A_1 в момент времени t. Если начальное состояние $x(1)$ и класс функций C_1, которому принадлежит f_1, известны, то задача поиска функции f_1 сводится к доопределению частичной булевой функции до функции в классе C_1.

Ключевые слова: *конечный автомат, криптографический генератор, генератор (δ, τ)-шагов, криптоанализ, метод DSS.*

DOI 10.17223/20710410/42/3

CRYPTANALYSIS OF 2-CASCADE FINITE AUTOMATA GENERATOR WITH FUNCTIONAL KEY

I. V. Borovkova, I. A. Pankratova, E. V. Semenova

National Research Tomsk State University, Tomsk, Russia

E-mail: iborovkova95@gmail.com, pank@mail.tsu.ru, katrinevs@mail.ru

A cryptographic generator under consideration is a serial connection $G = A_1 \cdot A_2$ of two finite state machines (finite automata) $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ (it is autonomous) and $A_2 = (\mathbb{F}_2, \mathbb{F}_2^n, \mathbb{F}_2, g_2, f_2)$. The key of the generator is the function f_1 and possibly the initial states $x(1), y(1)$ of the automata A_1, A_2. The cryptanalysis problem for G is the following: given an output sequence $\gamma = z(1)z(2)\dots z(l)$, find the generator's key. Two algorithms for analysis of A_2 are presented, they allow to find a preimage $u(1)\dots u(l)$ of γ in general case and in the case when A_2 is the Moore automaton with the transition function $g_2(u, y) = \neg u g^{\delta}(y) + u g^{\tau}(y)$ for some $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and $\delta, \tau \in \mathbb{N}$. This preimage is an input to A_2 and an output from A_1. The values $u(t)$ equal the values $f_1(x(t))$ where $x(t)$ is the state of A_1 at a time t, $t = 1, 2, \dots, l$. If the

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

initial state $x(1)$ and a function class C_1 containing f_1 are known, then f_1 can be determined by its specifying in the class C_1.

Keywords: *finite automaton, cryptographic generator, (δ,τ)-step generator, cryptanalysis, DSS method.*

1. Определение генератора

Рассматривается двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$, схема которого показана на рис. 1. Генератор представляет собой последовательное соединение автономного автомата $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ (с функцией переходов $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$ и функцией выходов $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$) и автомата $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ (с функцией переходов $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$ и функцией выходов $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2$).

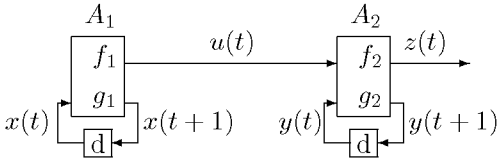

Рис. 1. Схема генератора G

Генератор функционирует в дискретном времени $t = 1, 2, \ldots$, в каждый момент t которого автомат A_1, находясь в состоянии $x(t) \in \mathbb{F}_2^n$, выдаёт выходной символ $u(t) = f_1(x(t))$ и переходит в следующее состояние $x(t+1) = g_1(x(t))$, а автомат A_2, находясь в состоянии $y(t) \in \mathbb{F}_2^m$, принимает от A_1 символ $u(t)$, выдаёт на выход генератора выходной символ $z(t) = f_2(u(t), y(t))$ и переходит в следующее состояние $y(t+1) = g_2(u(t), y(t))$. Последовательность $u(1)\ldots u(l)$, $l \in \mathbb{N}$, выходных символов автомата A_1 называется управляющей последовательностью автомата A_2, а последовательность $z(1)\ldots z(l)$ выходных символов автомата A_2 — выходной последовательностью генератора G. Ключом генератора может быть любое непустое подмножество множества $\{x(1), y(1), f_1, g_1, f_2, g_2\}$.

2. Криптоанализ генератора G

2.1. Основная задача

Задача криптоанализа состоит в определении ключа генератора по его выходной последовательности. Рассмотрим сначала случай, когда ключом служит только функция f_1, все остальные параметры известны. Как правило, известен ещё и класс функций, которому принадлежит f_1, потому что выходные функции автоматов в генераторе должны обладать определёнными свойствами: иметь ограниченную сложность задания, полиномиальную вычислимость, достаточную криптографическую стойкость и т. д. Целью данной работы является решение ряда вспомогательных задач для следующей основной задачи.

Задача 1

Дано: $\gamma = z(1)\ldots z(l)$ — выходная последовательность генератора; $x(1), y(1)$ — начальные состояния автоматов A_1, A_2; g_1 — функция переходов автомата A_1; C_1 — класс функций, которому принадлежит функция выходов автомата A_1; g_2 и f_2 — функции соответственно переходов и выходов автомата A_2.

Найти: функцию выходов $f_1 \in C_1$, такую, что $z(t) = f_2(f_1(x(t)), y(t))$ при $x(t+1) = g_1(x(t))$ и $y(t+1) = g_2(f_1(x(t)), y(t))$ для $t = 1, \ldots, l$.

Поскольку функция f_1 является ключом генератора, криптоаналитику неизвестна управляющая последовательность $u(1)u(2)\ldots$ Знание этой последовательности упрощает решение задачи 1, давая информацию о некоторых значениях функции f_1, а именно

$$u(t) = f_1(g_1^{t-1}(x(1))), \tag{1}$$

где $g_1^0(x) = x$; $g_1^t(x) = g_1(g_1^{t-1}(x))$, $t = 1, \ldots, l$. В связи с этим основная задача 1 распадается на две вспомогательные задачи:

1) анализ автомата A_2 — по выходной последовательности генератора G найти управляющие последовательности автомата A_2;
2) анализ автомата A_1 — по управляющей последовательности на выходе автомата A_1 найти его функцию выходов f_1.

2.2. Анализ автомата A_2

Обозначим $U(\gamma, y(1))$ множество всех управляющих последовательностей $u(1)\ldots u(l)$, отображаемых автоматом $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ в начальном состоянии $y(1)$ в выходную последовательность $\gamma = z(1)\ldots z(l)$, т. е. таких, что

$$f_2(u(t), y(t)) = z(t),\ y(t+1) = g_2(u(t), y(t)), \quad t = 1, \ldots, l. \tag{2}$$

Задача анализа автомата A_2 ставится следующим образом.

Дано: γ — выходная последовательность автомата A_2; $y(1)$ — его начальное состояние; g_2, f_2 — функции переходов и выходов.

Найти: множество $U(\gamma, y(1))$.

Для решения этой задачи построим граф, вершины которого расположены по ярусам с номерами $t \in \{1, 2, \ldots, l, l+1\}$ и помечены состояниями автомата A_2, дуги помечены значениями 0 и 1. На первом ярусе — вершина с меткой $y(1)$. Для каждой вершины v с меткой q построенного яруса t, $t = 1, \ldots, l$, составляем уравнение $z(t) = f_2(u, q)$ относительно $u \in \{0, 1\}$ и добавляем к вершине v столько потомков на ярусе $t+1$, сколько решений имеет это уравнение. Для каждого пути в графе от вершины первого яруса к вершине $(l+1)$-го яруса выписываем последовательность меток $u(1)\ldots u(l)$ дуг этого пути. По сути, это есть реализация метода DSS (Devide, Solve and Substitute) [1–3]. Более подробно действия описаны в алгоритме 1.

Корректность алгоритма. Пусть $c_1, \ldots, c_l$ — последовательность меток дуг некоторого пути от первого к $(l+1)$-му ярусу, а $q_1, \ldots, q_{l+1}$ — последовательность меток вершин этого пути. По построению $q_{t+1} = g_2(c_t, q_t)$, $f_2(c_t, q_t) = z(t)$, т. е. выполнены условия (2) при $u(t) = c_t$, $y(t) = q_t$. Следовательно, $c_1 \ldots c_l \in U(\gamma, y(1))$.

Полнота алгоритма. Пусть $u(1)\ldots u(l) \in U(\gamma, y(1))$, т. е. выполнены условия (2). Тогда $f_2(u(1), y(1)) = z(1)$ и по построению на ярусе 2 есть вершина v с меткой $q = g_2(u(1), y(1))$, соединённая с вершиной первого яруса дугой с меткой $u(1)$. Положим $y(2) = q$; ввиду того, что $f_2(u(2), y(2)) = z(2)$, на ярусе 3 есть вершина, соединённая с v дугой, помеченной $u(2)$, и т. д. до яруса $(l+1)$. Значит, $u(1)\ldots u(l)$ есть последовательность дуг некоторого пути от первого до $(l+1)$-го яруса.

Алгоритм 1 реализован на языке ЛЯПАС-Т [4, 5]. Граф в программе представляется логическим комплексом L, элементы которого соответствуют вершинам и хранят их метки; для элемента $L[i]$, $i = 0, 1, \ldots$, потомками являются элемент $L[2i+1]$ (дуга к нему от $L[i]$ помечена знаком 0) и элемент $L[2i+2]$ (дуга помечена знаком 1); если вершина отсутствует или удаляется, то элементу присваивается специальное значение (-1).

Алгоритм 1. Анализ автомата A_2

Вход: $\gamma = z(1)\dots z(l)$ — выходная последовательность автомата A_2; $y(1)$ — его начальное состояние; g_2, f_2 — функции переходов и выходов.

Выход: множество $U(\gamma, y(1))$.

1: На ярусе 1 — одна вершина с меткой $y(1)$.
2: **Для** $t = 1, 2, \dots, l$ строим ярус $t+1$ по следующим правилам.
3: **Если** вершин на ярусе t нет, **то**
4: выход из алгоритма с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
5: Рассматриваем каждую вершину v на ярусе t; пусть q — метка вершины v.
6: **Если** $z(t) = f_2(0, q)$, **то**
7: к вершине v добавляем потомка с меткой $g_2(0, q)$, соединяем v с потомком дугой с меткой 0.
8: **Если** $z(t) = f_2(1, q)$, **то**
9: к вершине v добавляем потомка с меткой $g_2(1, q)$; соединяем v с потомком дугой с меткой 1.
10: **Если** $z(t) \neq f_2(0, q) = f_2(1, q)$, **то**
11: потомков у вершины v нет; удаляем вершину v и дуги, ведущие в неё; поднимаемся по ярусам вверх, удаляя по пути все вершины, не имеющие потомков, и дуги, ведущие в них. Если граф стал пустым, то выход с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
12: Вершины яруса $t+1$, имеющие одинаковые метки, отождествляем.
13: Выполняем обход построенного графа в глубину. Последовательность меток дуг каждого пути, идущего из вершины 1-го яруса к вершинам $(l+1)$-го яруса, задаёт возможную управляющую последовательность; включаем её в множество $U(\gamma, y(1))$.

2.3. Анализ автомата A_1

Перейдём ко второй вспомогательной задаче: по множеству $U(\gamma, y(1))$ найти функцию выходов f_1 автомата A_1. Пусть $\beta = u(1)\dots u(l)$ — произвольная последовательность из $U(\gamma, y(1))$. Положив $h_\beta(g_1^{t-1}(x(1))) = u(t)$, $t = 1, \dots, l$, получим частично определённую булеву функцию $h_\beta(x)$. В соответствии с формулами (1) функция f_1 является доопределением функции h_β для некоторой $\beta \in U(\gamma, y(1))$. И наоборот: из описания работы генератора G следует, что если f_1' — любое доопределение функции h_β, то автомат $A_1' = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1')$ в состоянии $x(1)$ за l тактов работы выдаст управляющую последовательность β, а генератор $G' = A_1' \cdot A_2$ — выходную последовательность γ.

Обозначим $F(\gamma, x(1), y(1))$ множество всех булевых функций f, таких, что автомат $A = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f)$ в состоянии $x(1)$ за l тактов работы выдаёт управляющую последовательность из множества $U(\gamma, y(1))$, а именно:

$$f(x) \in F(\gamma, x(1), y(1)) \;\Leftrightarrow\; u(1)\dots u(l) \in U(\gamma, y(1)), \text{ где } u(t) = f(g_1^{t-1}(x(1))),\ t = 1, \dots, l.$$

Тогда любая функция из множества $F(\gamma, x(1), y(1)) \cap C_1$ является решением основной задачи 1. Получаем следующую постановку задачи.

Дано: множество $U(\gamma, y(1))$; $x(1)$ — начальное состояние автомата A_1; g_1 — его функция переходов; C_1 — класс функций, которому принадлежит f_1.

Найти: множество $F(\gamma, x(1), y(1)) \cap C_1$.

Необходимые действия описаны в алгоритме 2.

Алгоритм 2. Анализ автомата A_1

Вход: множество $U(\gamma, y(1))$; $x(1)$ — начальное состояние автомата A_1; g_1 — его функция переходов; C_1 — класс функций, которому принадлежит f_1.

Выход: множество $M = F(\gamma, x(1), y(1)) \cap C_1$ возможных функций выходов автомата A_1.

1: Положим $M := \varnothing$.

2: **Для** каждой последовательности $u(1) \dots u(l) \in U(\gamma, y(1))$

3: находим частично определённую функцию h, полагая $h(g_1^{t-1}(x(1))) = u(t)$, $t = 1, \dots, l$;

4: находим все доопределения функции h в классе C_1, добавляем их в множество M.

Способ выполнения шага 4 алгоритма 2 зависит от конкретного класса C_1. В связи с этим актуальны следующие задачи: поиск условий существования (несуществования) доопределения заданной частично определённой булевой функции в данном классе, условий единственности такого доопределения, метода построения всех её доопределений и др. Примеры их решения в случае, когда классом C_1 является множество функций с заданным или ограниченным числом существенных переменных, можно найти в [6, 7].

Количество повторений шагов 3, 4 алгоритма 2 зависит от мощности множества $U(\gamma, y(1))$. Компьютерные эксперименты показывают, что эта мощность сильно меняется даже при незначительном изменении параметров генератора G, например при изменении только начальных состояний автоматов A_1 и A_2.

В частном случае, когда функция $f_2(u, y)$ зависит от u линейно, всегда получим $|U(\gamma, y(1))| = 1$: при $f_2(u, y) = u \oplus \varphi(y)$ уравнение $z(t) = f_2(u, y)$ имеет единственное решение $u = z(t) \oplus \varphi(y)$, следовательно, в алгоритме 1 каждая вершина графа имеет одного потомка и путь от первого до последнего яруса единственный.

Для решения задачи 1 достаточно последовательно применить алгоритмы 1 и 2.

2.4. Некоторые обобщения основной задачи

Рассмотим случаи, когда начальные состояния автоматов A_1 и/или A_2 входят в ключ генератора вместе с функцией f_1.

Задача 2

Пусть ключом генератора является пара $(y(1), f_1)$. Задача криптоанализа ставится так же, как задача 1, за исключением того, что $y(1)$ неизвестно, его надо найти вместе с функцией f_1. Решениями будут все пары (y, f), такие, что $U(\gamma, y) \neq \varnothing$ и $f \in F(\gamma, x(1), y) \cap C_1$.

В самом деле, ввиду $f \in F(\gamma, x(1), y)$, автомат $A = (\mathbb{F}_2^m, \mathbb{F}_2, g_1, f)$ в состоянии $x(1)$ за l тактов работы выдаёт управляющую последовательность из множества $U(\gamma, y)$, которая отображается автоматом $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ в начальном состоянии y в выходную последовательность γ.

Для поиска решений можно применить такой метод: поочерёдно для всех $y \in \mathbb{F}_2^m$ полагаем $y(1) = y$, выполняем алгоритм 1 и если $U(\gamma, y) \neq \varnothing$, то алгоритм 2.

Аналогично рассуждая, получаем следующие задачи и их решения.

Задача 3

Если ключом генератора является пара $(x(1), f_1)$, то решениями задачи криптоанализа будут все пары (x, f), такие, что $F(\gamma, x, y(1)) \cap C_1 = M \neq \varnothing$ и $f \in M$. Для их

нахождения применяем алгоритм 1 (заметим, что при «правильном» $y(1)$ множество $U(\gamma, y(1))$ всегда непусто). Затем для каждого $x(1) = x \in \mathbb{F}_2^n$ выполняем алгоритм 2.

Задача 4

Наконец, если ключ — тройка $(x(1), y(1), f_1)$, то решение задачи криптоанализа имеет следующий вид: $\{(x, y, f) : U(\gamma, y) \neq \varnothing \;\&\; F(\gamma, x, y) \cap C_1 = M \neq \varnothing \;\&\; f \in M\}$. Для его нахождения перебираем все $x(1) = x \in \mathbb{F}_2^n$ и решаем для них задачу 2, или перебираем все $y(1) = y \in \mathbb{F}_2^m$ и решаем для них задачу 3.

Предложенные решения задач 2–4, скорее всего, не являются лучшими и даже приемлемыми; поиск более эффективных методов составляет предмет дальнейших исследований.

Компьютерные эксперименты с задачей 2 в случае, когда никаких ограничений на функцию f_1 не накладывается (класс C_1 содержит все булевы функции от n переменных), дали следующие результаты. Пусть $Y = \{y \in \mathbb{F}_2^m : U(\gamma, y) \neq \varnothing\}$ — множество начальных состояний автомата A_2, в которых он отображает хотя бы одну управляющую последовательность в выходную последовательность γ. Будем оценивать среднее значение $|Y|$ при случайном выборе параметров генератора. Как и ожидалось, оно уменьшается с ростом длины l выходной последовательности γ и стабилизируется в некотором значении $|Y|_{\text{ср}}$ при некотором l (разном для разных n и m). В частности, $|Y|_{\text{ср}} \approx 2$, если функция $f_2(u, y)$ не зависит от u, $|Y|_{\text{ср}} \approx 2^{m-1}$, если она имеет вид $u \vee \varphi(y)$ или $u \wedge \varphi(y)$. Исключение составляет случай, когда функция f_2 зависит от u линейно — $f_2(u, y) = u \oplus \varphi(y)$; в этом случае всегда $|Y| = 2^m$ (т. е. $Y = \mathbb{F}_2^m$), потому что автомат A_2 в любом начальном состоянии $y(1)$ отображает последовательность $u(1) \dots u(l)$ в последовательность $\gamma = z(1) \dots z(l)$, если взять $u(t) = z(t) \oplus \varphi(y(t))$, $t = 1, \dots, l$.

3. Криптоанализ конечно-автоматного генератора (δ, τ)-шагов

Рассмотрим важный частный случай [2, 3] конечно-автоматного генератора $G = A_1 \cdot A_2$, в котором автомат A_2 является автоматом Мура, т. е. его функция выходов не зависит от u, а именно: $f_2(u, y) = f(y)$ для некоторой функции $f : \mathbb{F}_2^m \to \mathbb{F}_2$; функция переходов автомата A_2 имеет вид $g_2(u, y) = \neg u g^{\delta}(y) + u g^{\tau}(y)$ для некоторых $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ и $\delta, \tau \in \mathbb{N}$. По аналогии с известным генератором (δ, τ)-шагов на регистрах сдвига с линейной обратной связью [8] будем называть его конечно-автоматным генератором (δ, τ)-шагов.

Обозначим выходную последовательность автомата A_2 через $\gamma = z(1)z(2) \dots z(l)$, где $z(t) = f(y(t))$; $y(t+1) = g_2(u(t), y(t))$, $t = 1, 2, \dots, l$; введём в рассмотрение ещё одну последовательность $S = s_1 s_2 \dots$, где $s_i = f(q(i))$; $q(1) = y(1)$; $q(i+1) = g(q(i))$, $i = 1, 2, \dots$ Если ключом генератора является только функция f_1 выходов первого автомата, т. е. во всей описанной схеме, кроме f_1, криптоаналитику неизвестна только управляющая последовательность $u(1)u(2) \dots u(l)$, то последовательность S можно вычислить заранее, до атаки. Задача анализа автомата A_2 сводится в этом случае к ещё одной вспомогательной задаче — классической задаче поиска в последовательности S такой подпоследовательности $s_{i_1} s_{i_2} \dots s_{i_l}$, что $s_{i_t} = z(t)$, $t = 1, \dots, l$, но со следующим ограничением: $i_{t+1} - i_t \in \{\delta, \tau\}$ для всех $t = 1, \dots, l-1$. Будем называть последовательность индексов $i_1 \dots i_l$, для которой выполнены указанные условия, допустимой; допустимых последовательностей для одних и тех же γ, S, δ, τ может быть несколько.

Для каждой допустимой последовательности индексов $i_1 \dots i_l$ полагаем $u(t) = 0$, если $i_{t+1} - i_t = \delta$, и $u(t) = 1$, если $i_{t+1} - i_t = \tau$. Заметим, что ввиду $z(1) = f(y(1)) = f(q(1)) = s_1$ всегда $i_1 = 1$ и что по выходной последовательности длины l можно

найти только $(l-1)$ символов управляющей последовательности — от $u(l)$ значения $z(1),\dots,z(l)$ не зависят. Задача анализа автомата A_2 для конечно-автоматного генератора (δ,τ)-шагов ставится следующим образом.

Дано: $\gamma = z(1)\dots z(l)$ — выходная последовательность автомата A_2; $\delta,\tau \in \mathbb{N}$; последовательность $S = s_1 s_2 \dots$

Найти: множество $U(\gamma, y(1))$.

Необходимые действия описаны в алгоритме 3. В процессе работы алгоритм строит таблицу T — двумерную таблицу с l строками переменной длины, t-я строка которой содержит возможные значения i_t в допустимых последовательностях индексов. Этот шаг аналогичен построению очередного яруса в алгоритме 1. Этап просеивания соответствует удалению из графа вершин, не имеющих потомков. На этапе 3 рекурсивно строятся все допустимые последовательности индексов; множество $M[t]$, $t = 1,\dots,l-1$, содержит допустимые последовательности для префикса $z(1)\dots z(t)$, каждая из которых может быть продолжена одним или двумя способами; эти продолжения записываются в $M[t+1]$. На этапе 4 для каждой допустимой последовательности индексов в $M[l]$ строится соответствующая ей управляющая последовательность. Алгоритм 3 и метод решения задачи 2 на его основе реализованы на языке C++; компьютерные эксперименты дали результаты, аналогичные полученным для алгоритма 1.

Алгоритм 3. Анализ автомата A_2 для конечно-автоматного генератора (δ,τ)-шагов

Вход: $\gamma = z(1)\dots z(l)$ — выходная последовательность автомата A_2; $\delta,\tau \in \mathbb{N}$; последовательность $S = s_1 s_2 \dots$

Выход: множество $U(\gamma, y(1))$.

Э т а п 1. *Построение таблицы*

$T[1] := \{1\}$.

Для $t = 2,\dots,l$

  $T[t] := \{k+\delta : k \in T[t-1]\ \&\ s_{k+\delta} = z(t)\} \cup \{k+\tau : k \in T[t-1]\ \&\ s_{k+\tau} = z(t)\}$.

Э т а п 2. *Просеивание*

Для $t = l,\dots,2$

  из $T[t-1]$ удаляем все элементы j, такие, что $\forall k \in T[t]\ (j \neq k-\delta\ \&\ j \neq k-\tau)$.

Э т а п 3. *Построение допустимых последовательностей индексов*

$M[1] := \{(1)\}$.

Для $t = 2,\dots,l$

  $M[t] := \varnothing$.

  Для всех $(i_1\dots i_{t-1}) \in M[t-1]$

    Если $j = i_{t-1}+\delta \in T[t]$, **то**

      $M[t] := M[t] \cup \{(i_1\dots i_{t-1}j)\}$.

    Если $k = i_{t-1}+\tau \in T[t]$, **то**

      $M[t] := M[t] \cup \{(i_1\dots i_{t-1}k)\}$.

Э т а п 4. *Построение множества* $U(\gamma, y(1))$

Для всех $(i_1\dots i_l) \in M[l]$

  полагаем $u(t) = 0$, если $i_{t+1} - i_t = \delta$, и $u(t) = 1$, если $i_{t+1} - i_t = \tau$, $t = 1,\dots,l-1$; включаем последовательность $u(1)\dots u(l-1)$ в множество $U(\gamma, y(1))$.

Заключение

Рассмотрен двухкаскадный конечно-автоматный генератор в общем случае и с ограничениями на автомат второго каскада; в обоих случаях представлены алго-

ритмы решения задачи криптоанализа генератора с функцией выходов f_1 автомата первого каскада в роли ключа. Предложены способы криптоанализа в случае, когда в ключ вместе с функцией f_1 входят начальные состояния обоих или одного из автоматов. В дальнейшем предполагается исследование генераторов с ключами, содержащими и другие их параметры.

Авторы выражают глубокую признательность Геннадию Петровичу Агибалову за постановку задачи и помощь в работе.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Криптоавтоматы с функциональными ключами // Прикладная дискретная математика. 2017. № 36. С. 59–72.
2. *Агибалов Г. П., Панкратова И. А.* О двухкаскадных конечно-автоматных криптографических генераторах и методах их криптоанализа // Прикладная дискретная математика. 2017. № 35. С. 38–47.
3. *Агибалов Г. П., Панкратова И. А.* К криптоанализу двухкаскадных конечно-автоматных криптографических генераторов // Прикладная дискретная математика. Приложение. 2016. № 9. С. 41–43.
4. *Торопов Н. Р.* Язык программирования ЛЯПАС // Прикладная дискретная математика. 2009. № 2(4). С. 9–25.
5. *Агибалов Г. П., Липский В. Б., Панкратова И. А.* О криптографическом расширении и его реализации для русского языка программирования // Прикладная дискретная математика. 2013. № 3(21). С. 93–104.
6. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. 1970. Вып. 49. С. 12–19.
7. *Агибалов Г. П., Сунгурова О. Г.* Криптоанализ конечно-автоматного генератора ключевого потока с функцией выходов в качестве ключа // Вестник Томского государственного университета. Приложение. Август 2006. № 17. С. 104–108.
8. *Фомичёв В. М.* Методы дискретной математики в криптологии. М.: ДИАЛОГ-МИФИ, 2010. 424 с.

REFERENCES

1. *Agibalov G. P.* Kriptoavtomaty s funktsional'nymi klyuchami [Cryptautomata with functional keys]. Prikladnaya Diskretnaya Matematika, 2017, no. 36, pp. 59–72.
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.
3. *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.
4. *Toropov N. R.* Yazik programmirovaniya LYaPAS [Programming language LYaPAS]. Prikladnaya Diskretnaya Matematika, 2009, no. 2(4), pp. 9–25. (in Russian)
5. *Agibalov G. P., Lipskiy V. B., and Pankratova I. A.* O kriptograficheskom rasshirenii i ego realizatsii dlya russkogo yazyka programmirovaniya [Cryptographic extension and its implementation for Russian programming language]. Prikladnaya Diskretnaya Matematika, 2013, no. 3(21), pp. 93–104. (in Russian)
6. *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)

7. *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)
8. *Fomichev V. M.* Metody diskretnoy matematiki v kriptologii [Methods of Discrete Mathematics in Cryptology]. Moscow, DIALOG-MEPhI Publ., 2010. 424 p. (in Russian)