МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

О ДВУХКАСКАДНЫХ КОНЕЧНО-АВТОМАТНЫХ КРИПТОГРАФИЧЕСКИХ ГЕНЕРАТОРАХ И МЕТОДАХ ИХ КРИПТОАНАЛИЗА[1]

Г. П. Агибалов, И. А. Панкратова

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 $\mathbb{F}_2$ с множествами состояний $\mathbb{F}_2^n$, $n > 1$, и $\mathbb{F}_2^m$, $m > 1$, соответственно, с выходным алфавитом $\mathbb{F}_2$ и с функциями выходов $f_1(x)$ и $f_2(u, y)$ из некоторых классов булевых функций от n и $m + 1$ переменных соответственно. Автомат A_1 автономный с произвольной функцией переходов $g_1(x)$, автомат A_2 неавтономный с входным алфавитом $\mathbb{F}_2$ и функцией переходов $g_2(u, y)$, в которой $g_2(0, y) = g^\delta(y)$ и $g_2(1, y) = g^\tau(y)$ для некоторых различных нетрицательных целых δ и τ и отображения $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$. В каждый момент времени $t = 1, 2, \ldots$ автомат A_1 из состояния $x(t)$ переходит в состояние $x(t + 1) = g_1(x(t))$ и вырабатывает выходной символ $u(t) = f_1(x(t))$, автомат A_2 из состояния $y(t)$ переходит в состояние $y(t + 1) = g_2(u(t), y(t))$ и вырабатывает выходной символ $z(t) = f_2(u(t), y(t))$, который и является выходным символом генератора G. Ключом генератора может быть любой непустой набор элементов из ряда $x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau$. Задача криптоанализа генератора G состоит в определении его ключа по заданному конечному отрезку $\gamma = z(1)z(2)\ldots z(l)$ его выходной последовательности. Показано, что в генераторе G с линейным автоматом A_2 ключ $y(1)$ вскрывается с полиномиальной сложностью решением системы линейных уравнений, а ключ $(x(1), y(1))$ — линеаризационной атакой сложности не более 2^n. Предложен метод, позволяющий в произвольном генераторе G с известными функциями g_2 и f_2 вычислить по γ отрезок управляющей последовательности $\beta = u(1)u(2)\ldots u(l)$ на выходе A_1 и тем самым открыть две возможности для криптоанализа такого G: 1) найти его ключ $(x(1), y(1))$ атакой «встреча посредине» со сложностью 2^m и 2) свести задачу криптоанализа G к криптоанализу автомата A_1 — найти ключ последнего по β. Сложность метода полиномиальная, если $y(1)$ не входит в ключ, и не превосходит 2^m в противном случае. Если ключом в A_1 служит функция f_1, то его вскрытие, в свою очередь, сводится к доопределению частичной булевой функции со значениями $u(t)$ на состояниях $x(t)$ для $t = 1, 2, \ldots, l$ до функции в классе функции f_1. Аналогично, к доопределению частичной булевой функции со значениями $z(t)$ на парах $(u(t), y(t))$ для $t = 1, 2, \ldots, l$ до функции в классе функции f_2 сводится вскрытие ключа произвольного генератора G с ключом f_2. Сообщается об известных авторам алгоритмах доопределения частичной булевой функции от сколь угодно большого набора переменных до функции из класса полностью определённых булевых функций, существенно зависящих от малого или ограниченного количества переменных из этого набора.

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

Ключевые слова: *конечный автомат, криптографический генератор, генератор (δ, τ)-шагов, криптоанализ, линеаризационная атака, атака «разделяй и решай», атака «разделяй — решай — подставляй», атака «встреча посредине».*

DOI 10.17223/20710410/35/4

ABOUT 2-CASCADE FINITE AUTOMATA CRYPTOGRAPHIC GENERATORS AND THEIR CRYPTANALYSIS

G. P. Agibalov, I. A. Pankratova

National Research Tomsk State University, Tomsk, Russia

E-mail: agibalov@isc.tsu.ru, pank@isc.tsu.ru

A cryptographic generator under consideration is a serial connectiom $G = A_1 \cdot A_2$ of two finite state machines (finite automata) A_1 and A_2 both defined over the two-element field $\mathbb{F}_2$. The first one is an autonomous automaton $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ with the state set $\mathbb{F}_2^n$, $n > 1$, the output alphabet $\mathbb{F}_2$, a transition function $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$, and an output function $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$. The second one is a non-autonomous automaton $A_2 = (\mathbb{F}_2, \mathbb{F}_2^n, \mathbb{F}_2, g_2, f_2)$ with the state set $\mathbb{F}_2^m$, $m > 1$, the input and output alphabets $\mathbb{F}_2$, an output function $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^n \to \mathbb{F}_2$, and a transition function $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$, for which there exist a map $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and some different integers δ and τ such that, for all $u \in \mathbb{F}_2$ and $y \in \mathbb{F}_2^m$, $g_2(u, y) = \neg u g^\delta(y) + u g^\tau(y)$, where $g^0(y) = y$ and $g^r(y) = g(g^{r-1}(y))$ for every integer $r \geqslant 1$. Thus, the generator G is a finite autonomous automaton $G = A_1 \cdot A_2 = (\mathbb{F}_2^n \times \mathbb{F}_2^m, \mathbb{F}_2, h, f)$, where $h : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n \times \mathbb{F}_2^m$ and $h(x, y) = (g_1(x), g_2(f_1(x), y))$, $f : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2$ and $f(x, y) = f_2(f_1(x), y)$, $x \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2^m$. As we see, all the functions in the definition of G are Boolean ones, and the functions g_1 and g_2, g are vector functions of dimensions n and m respectively. Further, it is assumed that each of them belongs to a predetermined class of Boolean functions and the classes of functions f_1 and f_2 are denoted by C_1 and C_2 respectively. At any time moment $t = 1, 2, \ldots$, the automaton A_1 goes from its state $x(t)$ to a state $x(t + 1) = g_1(x(t))$ and produces an output symbol $u(t) = f_1(x(t))$, the automaton A_2 receives $u(t)$ and goes from its state $y(t)$ to a state $y(t + 1) = g_2(u(t), y(t))$ and produces an output symbol $z(t) = f_2(u(t), y(t))$ which is the output symbol generated by G at this moment. In general, a key of the generator can be defined as any non-empty subset of the set $\{x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau\}$. The cryptanalysis problem for G is the following: given a finite beginning $\gamma = z(1)z(2)\ldots z(l)$ of a sequence generated by G, find the generator key value. For solving the problem, the generator is described by a system of logical equations, connecting bits in γ with the initial states and values of transition and output functions in automata A_1 and A_2. It is shown that in G with the linear A_2, the key $y(1)$ is determined with the polynomial complexity by solving a system of linear equations and the key $(x(1), y(1))$ — by the linearization attack (trying different initial states of A_1) with the complexity 2^n which is much less than 2^{n+m} — the complexity of the bruitforce attack. For an arbitrary G with the known g_2 and f_2, a method is proposed allowing to compute from γ a string of the output sequence $\beta = u(1)u(2)\ldots u(l)$ of A_1 and so to reveal two possibilities for cryptanalysis of this G: 1) to determine its key $(x(1), y(1))$ by the meet-in-the-middle attack with the complexity 2^m or 2) to reduce the cryptanalysis problem for G to the cryptanalysis problem for A_1, that is, to determine the key of A_1 by β. The complexity of the method is polynomial if $y(1)$ is not included in the key and is not more than 2^m

otherwise. In case when the key of an arbitrary G is f_1, this key can be determined by identifying f_1 with a function $f \in C_1$ satisfying the equalities $f(x(t)) = u(t)$, $t = 1, 2, \ldots, l$. Similarly, if the key of G is f_2, the determination of f_2 is reduced to the construction of a function $f \in C_2$ satisfying the equalities $f(u(t), y(t)) = z(t)$, $t = 1, 2, \ldots, l$. In connection with the last, it is told about some algorithms, known to the authors, for extending a partially defined Boolean function in a large set of variables to a function from a class of completely defined Boolean functions essentially depending on a little or bounded number of variables in this set.

Keywords: *finite automaton, cryptographic generator, (δ, τ)-step generator, cryptanalysis, linearization attack, "devide and solve" attack, "devide — solve — substitute" attack, "meet-in-the middle" attack.*

1. Определение генератора

Рассматриваемый здесь криптографический генератор (будем обозначать его G) представляет собой последовательное соединение двух абстрактных конечных автоматов над полем $\mathbb{F}_2$ — управляющего и управляемого. Первый является некоторым автономным автоматом $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ с множеством состояний $\mathbb{F}_2^n$, $n > 1$, и выходным алфавитом $\mathbb{F}_2$, с функцией переходов $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$ и функцией выходов $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$, а второй — некоторым неавтономным автоматом $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ с входным и выходным алфавитами $\mathbb{F}_2$, с множеством состояний $\mathbb{F}_2^m$, $m > 1$, с функцией выходов $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2$ и функцией переходов $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$, для которой существуют отображение $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ и различные целые неотрицательные числа δ и τ, такие, что для всех $u \in \mathbb{F}_2$ и $y \in \mathbb{F}_2^m$ если $u = 0$, то $g_2(u, y) = g^\delta(y)$, и если $u = 1$, то $g_2(u, y) = g^\tau(y)$, т. е. $g_2(u, y) = \neg u g^\delta(y) + u g^\tau(y)$, где, как обычно, $g^0(y) = y$ и $g^r(y) = g(g^{r-1}(y))$ для любого целого $r \geqslant 1$. Таким образом, генератор G — это конечный автономный автомат $G = A_1 \cdot A_2 = (\mathbb{F}_2^n \times \mathbb{F}_2^m, \mathbb{F}_2, h, f)$, где $h : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n \times \mathbb{F}_2^m$ и $h(x, y) = (g_1(x), g_2(f_1(x), y))$, $f : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2$ и $f(x, y) = f_2(f_1(x), y)$, $x \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2^m$. Все функции в определении G, как видим, булевы, причём функции g_1 и g_2, g векторные размерности n и m соответственно.

Генератор G функционирует в дискретном времени $t = 1, 2, \ldots$, в каждый момент t которого его автомат A_1, находясь в состоянии $x(t) = x_1(t)x_2(t)\ldots x_n(t) \in \mathbb{F}_2^n$, выдаёт выходной управляющий символ $u(t) = f_1(x(t))$ и переходит в следующее состояние $x(t + 1) = g_1(x(t))$, а автомат A_2 в этот момент, находясь в состоянии $y(t) = y_1(t)y_2(t)\ldots y_n(t) \in \mathbb{F}_2^m$, принимает от A_1 символ $u(t)$, выдаёт свой выходной символ $v(t) = f_2(u(t), y(t))$ и переходит в следующее состояние $y(t+1) = g_2(u(t), y(t))$. Значением $z(t)$ на выходе генератора G в момент t является значение $v(t)$ на выходе автомата A_2 в этот момент.

Предполагается, что значение любой функции в генераторе G на любом наборе значений её аргументов вычисляется за полиномиальное время от числа последних.

Можно показать, что в частном случае, когда функции g_1 и g_2 являются функциями переходов некоторых регистров сдвига с линейной обратной связью и длиной n и m соответственно и $f_2(u, y_1, y_2, \ldots, y_m) = y_1$, генератор G функционально эквивалентен генератору (δ, τ)-шагов [1], в котором функция переходов второго регистра есть g.

Функционирование генератора G во времени описывается формально следующей системой E векторных булевых уравнений с переменными $x(t), y(t), u(t)$, $t = 1, 2, \ldots$:

$$\begin{cases} u(t) = f_1(x(t)), \\ x(t+1) = g_1(x(t)), \\ x(1) = x_1(1)x_2(1)\dots x_n(1); \\ z(t) = f_2(u(t), y(t)), \\ y(t+1) = g_2(u(t), y(t)) = \neg u(t) g^{\delta}(y(t)) + u(t) g^{\tau}(y(t)), \\ y(1) = y_1(1)y_2(1)\dots y_m(1); \\ t \geqslant 1. \end{cases}$$

В ней подсистема E_1 из первого, второго и третьего уравнений описывает работу управляющего автомата A_1, подсистема E_2 из четвёртого, пятого и шестого уравнений — работу управляемого автомата A_2.

Ключом генератора G теоретически может быть любое непустое подмножество множества $\{x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau\}$. Требование стойкости криптографического генератора накладывает определённые ограничения на применяемые в нём булевы функции. Кроме того, не любую булеву функцию можно задать практически. В этой связи предполагается, что каждая функция в генераторе принадлежит некоторому классу функций, ограниченных по сложности и обладающих некоторыми криптографическими свойствами, и этот класс, в том числе для функции в составе ключа, общеизвестен. При известном ключе и заданных других параметрах генератора уравнения данной системы E позволяют однозначно вычислить порождаемую генератором выходную последовательность $z(1)z(2)\dots$

Здесь мы ограничимся случаями, в которых ключ генератора есть $x(1)$ — начальное состояние A_1, $y(1)$ — начальное состояние A_2, $(x(1), y(1))$ — начальное состояние G, f_1 — функция выходов A_1 или f_2 — функция выходов A_2. Далее классы функций f_1 и f_2 обозначаются C_1 и C_2 соответственно.

2. Постановка задачи

Задача криптоанализа произвольного генератора G заключается в определении его ключа в известном классе по заданному конечному отрезку $\gamma = z(1)z(2)\dots z(l)$ последовательности, порождаемой им на этом ключе. В такой постановке задача возникает в криптоанализе поточного шифра, использующего данный генератор, атакой на шифр с известным или выбираемым открытым текстом. Её решение может быть получено как решение конечной подсистемы $E(l)$ указанной системы уравнений E с $t = 1, 2, \dots, l$. В случае неединственности решения системы $E(l)$ обычно рекомендуется задаться более длинным отрезком γ.

Решение системы $E(l)$ может быть найдено атакой грубой силы, или исчерпывающим поиском, т. е. перебором возможных ключей с вычислением при каждом выбранном ключе k начального отрезка γ' длины l порождаемой последовательности и сравнением его с отрезком γ. В случае $\gamma' = \gamma$ ключ k принимается за ответ задачи. Сложность этой атаки определяется размером ключевого пространства. Так, если ключом генератора служит его начальное состояние, то сложность атаки равна 2^{n+m}. Если же ключом является какая-либо из функций генератора, то сложность атаки равна мощности класса этой функции. Ниже приведены примеры атак на любой генератор G со сложностью меньше, чем у атаки грубой силы. О некоторых из этих атак докладывалось на 15 Сибирской школе-семинаре «Компьютерная безопасность и криптография» [2].

3. Атака на состояние генератора

Подмножество L (скалярных) переменных в некоторой системе уравнений S над некоторым кольцом называется *эффективным множеством*, если фиксирование любых возможных значений этих переменных превращает S в легко решаемую (например, за полиномиальное или меньшее время) систему уравнений (ЛРС). В частности, таковым является подмножество переменных, при любом фиксировании которых все уравнения в системе превращаются в линейные. Оно называется линеаризационным множеством переменных системы [3, 4]. Очевидно, что для любой ЛРС пустое множество переменных эффективное, а в системе булевых уравнений (СБУ) с r переменными любые $r-1$ переменных образуют линеаризационное множество.

Всякая СБУ S с q-элементным эффективным множеством L решается со сложностью 2^q методом DS (Devide and Solve), или по-русски «разделяй и решай», состоящим в подстановке в систему S 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 S набором значений переменных в L, является решением системы S. Метод DS на основе линеаризационного множества переменных называется линеаризационной атакой [3, 4]. В нём решается частный случай ЛРС — система линейных уравнений (СЛУ).

Автомат A_2 называется линейным, если все координатные функции в g_2 и функция f_2 линейные. Нетрудно заметить, что в системе $E(l)$ уравнений генератора G с линейным управляемым автоматом A_2 и ключом $(x(1), y(1))$ переменные $x_1(1)$, $x_2(1)$, …, $x_n(1)$ в $x(1)$ образуют линеаризационное множество системы уравнений $E(l)$, ибо фиксирование криптоаналитиком определённых значений этих переменных позволяет ему вычислить и тем самым зафиксировать в $E(l)$ определённые значения остальных переменных, кроме $y_j(t)$ для $j = 1, 2, \ldots, m$ и $t = 1, 2, \ldots, l$, и, благодаря линейности функций g_2 и f_2, превратить $E(l)$ в некоторую СЛУ с переменными из последнего списка, включающего и компоненты в $y(1)$. Решение этой системы — заключительный шаг в линеаризационной атаке. Её сложность, как видим, равна 2^n, что много меньше 2^{n+m} — сложности атаки грубой силы. Кроме того, показанная возможность вычисления $y(1)$ по $x(1)$ означает, что при наличии в ключе вектора $x(1)$ добавление к ключу вектора $y(1)$ не повышает стойкости любого генератора G. Согласно [1, с. 331], для генераторов (δ, τ)-шагов этот факт давно известен. Здесь он установлен для более широкого класса криптографических генераторов.

4. Метод DSS

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

Теперь можно заметить, что подсистема $E'(l)$, состоящая из трёх уравнений в $E(l)$ — четвёртого, пятого и шестого, т. е. подсистема уравнений управляемого автомата A_2, при любом фиксировании значений переменных в $y(1)$ становится РЛРС с переменными $u(t)$ для $t = 1, 2, \ldots, l$ и $y(t)$ для $t = 2, 3, \ldots, l$ и может быть решена методом DSS. В самом деле, при $t = 1$ и при фиксированном значении $y(1)$ имеем

уравнение

$$z(1) = f_2(u(1), y(1))$$

относительно $u(1)$, которое имеет либо два решения (0 и 1), если $z(1) = f_2(0, y(1)) = f_2(1, y(1))$, либо одно решение c, если $z(1) = f_2(c, y(1)) \neq f_2(\neg c, y(1))$, и не имеет решений в оставшемся случае $z(1) \neq f_2(0, y(1)) = f_2(1, y(1))$.

При $t = 2$, используя найденное на предыдущем шаге значение $u(1)$, вычисляем $y(2) = \neg u(1) g^{\delta}(y(1)) + u(1) g^{\tau}(y(1))$ и получаем уравнение

$$z(2) = f_2(u(2), y(2))$$

относительно $u(2)$, которое также может иметь два, одно или ни одного решения. Продолжая процесс для $t = 3, \ldots, l$, получим последовательности $u(1), \ldots, u(l)$ и $y(1), \ldots, y(l)$, которые являются кандидатами соответственно на выходную последовательность управляющего автомата и последовательность состояний управляемого автомата при известных $y(1), g_2, f_2$ и выходной последовательности $z(1), z(2), \ldots, z(l)$ генератора G.

Если начальное состояние $y(1)$ автомата A_2 не входит в ключ генератора, то криптоаналитику известно его значение и он может решить $E'(l)$ методом DSS, как описано выше, и найти возможные значения для $u(1), u(2), \ldots, u(l)$ на входе управляемого автомата A_2, которые являются возможными значениями функции f_1 на наборах $x(t)$ значений её переменных $x_1(t)x_2(t)\ldots x_n(t)$ для некоторых $t \in \{1, 2, \ldots, l\}$. Сложность этого вычисления полиномиальная, подробно процедура описана далее в п. 5 (алгоритм 1).

Если же состояние $y(1)$ автомата A_2 принадлежит ключу, то криптоаналитик может применить к системе $E'(l)$ метод DS, а именно: фиксируя в $E'(l)$ поочерёдно возможные значения состояния $y(1)$ и получая каждый раз некоторую РЛРС, найти такое значение $y(1)$, при котором полученная РЛРС совместна, решить её методом DSS и в результате получить и ключевое значение $y(1)$, и некоторые значения для $u(1), u(2), \ldots, u(l)$ функции f_1 на соответствующих наборах $x(t)$ значений её переменных. Сложность этой атаки равна 2^m. Её реализация сводится к последовательному применению алгоритма 1 для всех значений $y(1) \in \mathbb{F}_2^m$.

Таким образом, со сложностью не более 2^m криптоаналитик независимо от того, входит или не входит $y(1)$ в ключ, но зная f_2 и g_2 (а вместе со вторым и δ, τ, g), может получить доступ к некоторым значениям в отрезке $\beta = u(1)u(2)\ldots u(l)$ выходной последовательности автомата A_1 и тем самым открыть две возможности для решения задачи криптоанализа произвольного генератора G: 1) вычислить его ключ $(x(1), y(1))$ атакой «встреча посредине» и 2) свести её к криптоанализу управляющего автомата A_1, т. е. к определению ключа последнего по его выходноу отрезку β. Далее эти возможности рассматриваются в п. 6 и 7 соответственно.

5. Алгоритм DSS

Задача — методом DSS определить некоторые значения ключевой функции f_1 в случае, когда остальные параметры генератора известны. Алгоритм состоит в построении графа, вершины которого расположены по ярусам с номерами $t \in \{1, 2, \ldots, l\}$, вершины помечены натуральными числами, дуги — значениями 0 и 1.

Алгоритм 1. DSS

Вход: $z(1), \ldots, z(l)$; $x(1), y(1)$; $g, \sigma, \tau, g_1, f_2$
Выход: значения $f_1(x^{(i)})$, $i = 1, \ldots, k$, для некоторых $x^{(1)}, \ldots, x^{(k)} \in \mathbb{F}_2^n$

1: На ярусе 1 — одна вершина u (особая — её метка не имеет значения); $t := 2$.
2: **Если** $z(1) \neq f_2(0, y(1)) = f_2(1, y(1))$, **то**
3: выход из алгоритма с ответом «$y(1)$ не может быть начальным состоянием автомата A_2»,
4: **иначе**
5: на ярус 2 помещаем вершину v с меткой 1; соединяем вершину u с вершиной v двумя дугами (с метками 0 и 1), если $z(1) = f_2(0, y(1)) = f_2(1, y(1))$; одной дугой с меткой 0 (0-дугой), если $z(1) = f_2(0, y(1)) \neq f_2(1, y(1))$, и одной дугой с меткой 1 (1-дугой), если $z(1) = f_2(1, y(1)) \neq f_2(0, y(1))$.
6: **Если** вершин на ярусе t нет, **то**
7: выход из алгоритма с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
8: Рассматриваем каждую вершину v на ярусе t; пусть j — метка вершины v.
9: **Если** $z(t+1) = f_2(0, g^{\sigma+j}(y(1))) = f_2(1, g^{\tau+j}(y(1)))$, **то**
10: к вершине v добавляем два потомка на ярусе $t+1$: 0-дуга соединяет вершину v с вершиной, помеченной $\sigma + j$; 1-дуга — вершину v с вершиной, помеченной $\tau + j$.
11: **Если** $z(t+1) \neq f_2(0, g^{\sigma+j}(y(1))) = f_2(1, g^{\tau+j}(y(1)))$, **то**
12: 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 поднялись до яруса 1, то выход с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
13: **Если** $z(t+1) = f_2(0, g^{\sigma+j}(y(1))) \neq f_2(1, g^{\tau+j}(y(1)))$, **то**
14: к вершине v добавляем одного потомка с меткой $\sigma + j$; соединяем v с потомком 0-дугой.
15: **Если** $z(t+1) = f_2(1, g^{\tau+j}(y(1))) \neq f_2(0, g^{\sigma+j}(y(1)))$, **то**
16: к вершине v добавляем одного потомка с меткой $\tau + j$; соединяем v с потомком 1-дугой.
17: Вершины яруса t, имеющие одинаковые метки, отождествляем.
18: **Если** $t < l - 1$, **то**
19: увеличиваем t на 1, переходим к шагу 6.
20: Анализируем построенный граф.
21: **Для** $t = 1, \ldots, l-1$
22: **Если** все дуги, соединяющие вершины t-го и $(t+1)$-го ярусов, имеют одинаковую метку $c \in \{0, 1\}$, **то**
полагаем $f_1(g_1^{t-1}(x(1))) = c$.

Если начальное состояние $y(1)$ автомата A_2 входит в ключ генератора, то, применяя алгоритм 1 для всех значений $y(1) \in \mathbb{F}_2^m$, можно ожидать, что количество «кандидатов» на роль начального состояния автомата A_2 будет много меньше, чем 2^m; так, в среднем в 1/4 случаев алгоритм закончит работу на шаге 3; ещё примерно 1/4 значений отсеется на шаге 7 при $t = 2$ и т. д. Получение более точных оценок и их зависимости от параметров генератора составляет предмет дальнейших исследований.

6. Атака «встреча посредине»

Пусть ключом генератора G является пара $(x(1), y(1)) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ и $l > n$.

Для каждого значения a переменной $x(1)$ в $\mathbb{F}_2^n$, пользуясь системой уравнений E_1 для $t = 1, 2, \ldots, l$, вычислим $\alpha = a_1 a_2 \ldots a_l$ как её решение для набора переменных $u(1), u(2), \ldots, u(l)$ и сохраним a по адресу $H(\alpha)$, где $H : \mathbb{F}_2^l \to \mathbb{F}_2^n$ есть некоторая хеш-функция. Эти действия выполняются предварительно, т. е. до проведения атаки, и их сложность не входит в её сложность. Во время атаки, имея выходную последовательность генератора $\gamma = z(1)z(2)\ldots z(l)$, методом DSS решается система E_2 уравнений автомата A_2 при разных значениях b переменной $y(1)$, выбираемых в $\mathbb{F}_2^m$ до тех пор, пока при некотором b в системе для набора переменных $u(1), u(2), \ldots, u(l)$ не будет получено решение $\beta = b_1 b_2 \ldots b_l$ с непустым содержимым a по адресу $H(\beta)$, и тогда пара (a, b) принимается за результат криптоанализа — искомое значение ключа. Сложность этой атаки не превосходит 2^m.

В альтернативном варианте атаки «встреча посредине» эти два её шага меняются местами, а именно: сначала по известной γ для каждого значения b переменной $y(1)$ методом DSS вычисляется β и значение b сохраняется по адресу $H(\beta)$ для некоторой хеш-функции $H : \mathbb{F}_2^l \to \mathbb{F}_2^m$, а затем последовательно для различных значений a переменной $x(1)$ вычисляются соответствующие значения α до тех пор, пока по адресу $H(\alpha)$ не будет найдено непустое b, и тогда пара (a, b) принимается за результат криптоанализа. Сложность этого варианта атаки не превосходит $2^m + 2^n$.

Разумеется, в каждом из вариантов атаки результат криптоанализа может быть ошибочным, т. е. не эквивалентным истинному ключу, но вероятность этого с ростом длины l отрезка γ может только падать.

7. Криптоанализ управляющего автомата

Итак, требуется по заданной выходной последовательности $u(1)u(2)\ldots u(l)$ автомата A_1 вычислить его ключ, которым в произвольном случае может быть любое подмножество в $\{x(1), g_1, f_1\}$. Если ключ есть $\{x(1)\}$, то $x(1)$ может быть вычислено как решение подсистемы E_1 уравнений A_1. Это может быть сделано любым подходящим методом [4], либо методом DS при наличии в E_1 малого эффективного множества переменных, в частности — линеаризационной атакой при наличии в E_1 малого линеаризационного множества переменных, либо методом грубой силы при малом n. Ясно, что сложность любой такой атаки не превосходит 2^n.

В случае, когда ключом генератора A_1 является его функция выходов f_1, принадлежащая некоторому классу C_1 булевых функций от n переменных, его криптоанализ, в свою очередь, сводится к доопределению в классе C_1 функции f', частично определённой соотношениями $u(t) = f'(x_1(t), x_2(t), \ldots, x_n(t))$, $t = 1, 2, \ldots, l$. Нетривиальные методы решения этой последней задачи в значительной степени определяются заданным классом C_1, который, опять же в свою очередь, может определяться ограничениями на сложностные характеристики функций в классе — на количество существенных аргументов, мощность эффективного или линеаризационного множества аргументов (любое фиксирование которых превращает функцию в вычислимую с полиномиальной сложностью или линейную соответственно), длину АНФ или ДНФ, степени мономов или ранги элементарных конъюнкций в них и т. п., или криптографическими свойствами этих функций [5, 6] — сбалансированностью, корреляционной иммунностью, устойчивостью, алгебраической иммунностью и др. Подобных классов булевых функций, в том числе полезных для практических применений в генераторах A_1 и интересных в теории криптоанализа таких генераторов, можно указать великое множество.

8. Атака на функцию выходов управляемого автомата

В случае, когда ключом генератора G является функция f_2, криптоаналитик, зная, а возможно, и выбирая $x(1), y(1)$ и наблюдая $z(t)$, может вычислить $x(t+1) = g_1(x(t))$, $u(t) = f_1(x(t))$ и $f_2(u(t), y(t)) = z(t)$ для $t = 1, 2, \ldots, l$. Последние равенства определяют f_2 частично, и ввиду принадлежности f_2 к некоторому известному классу C_2 булевых функций от $m + 1$ переменных задача криптоанализа G теперь опять же сводится к доопределению имеющейся частичной булевой функции f', определённой соотношениями $f'(u(t), y(t)) = z(t)$, $t = 1, 2, \ldots, l$, до функции в классе C_2.

9. Доопределение частичной функции в классе функций с малым числом существенных переменных

Авторам известен только один класс C, функции которого уже применялись в реальных конечно-автоматных генераторах в качестве одновременно функций выходов и ключей, а сами эти генераторы исследовались на стойкость к криптоаналитическим атакам. Это было 50 лет назад с участием первого автора. Речь идёт о классе булевых функций, зависящих существенно от малого числа (по тем временам — до 10) аргументов, выбираемых случайным образом из большого числа (по тем временам — за сотню) переменных, в нашем случае представляющих собой компоненты состояния $x_1, x_2, \ldots, x_n$ в автомате A_1 или $y_1, y_2, \ldots, y_m$ в автомате A_2.

Были найдены конструктивные необходимые и достаточные условия единственности доопределения части искомой функции в этом классе и разработан метод, позволяющий построить любое из существующих таких доопределений. Условия единственности и метод построения сформулированы в предположении, что криптоаналитику известно, кроме прочего, либо количество k существенных аргументов ключевой функции, либо его верхняя граница k_0. Метод построения решает свою задачу за два шага: сначала узнаёт существенные аргументы функции, затем — её значения на наборах значений этих аргументов. Для нахождения первых используются методы построения столбцовых покрытий булевой матрицы, имеющей строками покомпонентные разности наборов значений переменных искомой функции с разными значениями на них.

В полном изложении эти результаты опубликованы в [7]. На их основе сформулированы оптимальные алгоритмы криптоанализа, в которых последовательность $z(1)z(2)\ldots z(l)$ на выходе генератора наблюдается до такого наименьшего $l = 1, 2, \ldots$, при котором выполняются условия единственности доопределения, после чего вся функция строится данным методом. Результаты экспериментальных исследований этих алгоритмов на современных компьютерах, выражающие зависимости времени вычислений алгоритмами ключа и длины l потребного для этого отрезка выходной последовательности генератора от его параметров (n, k, k_0), представлены в работе [8].

ЛИТЕРАТУРА

1. *Фомичёв В. М.* Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003. 400 с.
2. *Агибалов Г. П., Панкратова И. А.* К криптоанализу двухкаскадных конечно-автоматных криптографических генераторов // Прикладная дискретная математика. Приложение. 2016. № 9. С. 41–43.
3. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского государственного университета. Приложение. Сентябрь 2003. № 6. С. 31–41.

4. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского государственного университета. Приложение. Август 2006. № 17. С. 4–9.
5. *Панкратова И. А.* Булевы функции в криптографии: учеб. пособие. Томск: Издательский Дом Томского государственного университета, 2014. 88 с.
6. *Токарева Н. Н.* Симметричная криптография. Краткий курс: учеб. пособие. Новосибирск: Изд-во Новосиб. ун-та, 2012. 234 с.
7. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. 1970. Вып. 49. С. 12–19.
8. *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.

REFERENCES

1. *Fomichev V. M.* Diskretnaya matematika i kriptologiya [Discrete Mathematics and Cryptology]. Moscow, DIALOG-MEPhI Publ., 2003. 400 p. (in Russian)
2. *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.
3. *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)
4. *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)
5. *Pankratova I. A.* Bulevy funktsii v kriptografii: ucheb. posobie [Boolean Functions in Cryptography: a Tutorial]. Tomsk, TSU Publ., 2014. 88 p. (in Russian)
6. *Tokareva N. N.* Simmetrichnaya kriptografiya. Kratkiy kurs: ucheb. posobie [Symmetric Cryptography. Short Course: a Tutorial]. Novosibirsk, NSU Publ., 2012. 234 p. (in Russian)
7. *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)
8. *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)