Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра компьютерной безопасности

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
канд. техн. наук, доцент
_____ В.Н. Тренькаев
«_23_» _января_ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА
(ДИПЛОМНАЯ РАБОТА)

Криптоанализ двухкаскадного конечно-автоматного генератора

по специальности 10.05.01 Компьютерная безопасность,
специализация (профиль) «Анализ безопасности компьютерных систем»

Обухов Петр Кириллович

Научный руководитель ВКР
канд.физ.-мат.наук.,доцент
_____ И.А. Панкратова
«_23_» _января_ 2023 г.

Автор работы
студент группы № 931723
_____ П.К. Обухов
«_23_» _января_ 2023 г.

Томск – 2023

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ООП
канд. техн. наук, доцент

_____ В.Н. Тренькаев
подпись
«27» июня 2022 г.

ЗАДАНИЕ
по выполнению выпускной квалификационной работы специалиста обучающемуся
Обухову Петру Кирилловичу
Фамилия Имя Отчество обучающегося
по специальности 10.05.01 Компьютерная безопасность, специализация (профиль) «Анализ безопасности компьютерных систем»
1 Тема выпускной квалификационной работы

Криптоанализ двухкаскадного конечно-автоматного генератора

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис / деканат – 16 января 2023 г. б) в ГЭК – 27 января 2023 г.

3 Исходные данные к работе:
Объект исследования – Двухкаскадный конечно-автоматный генератор
Предмет исследования – Атаки на двухкаскадный конечно-автоматный генератор
Цель исследования – Анализ атак на генератор

Задачи:
Разработать атаки на двухкаскадный конечно-автоматный генератор с различными вариантами ключей. Изучить периодические свойства генератора. Использовать периодические свойства в криптоанализе. Сделать выводы относительно стойкости генератора.

Методы исследования:
Теоретический и экспериментальный.

Организация или отрасль, по тематике которой выполняется работа, –
Лаборатория компьютерной криптографии ТГУ.

4 Краткое содержание работы
Описание генератора, описание атак на генератор, результаты исследования периодических свойств генератора, использование периодических свойств в криптоанализе генератора.

Научный руководитель выпускной квалификационной работы
зав. лаб. компьютерной криптографии _____ / *И.А.Панкратова*
должность, место работы *подпись* *И.О. Фамилия*

Задание принял к исполнению
студент группы 931723 _____ / *П.К.Обухов*
должность, место работы *подпись* *И.О. Фамилия*

АННОТАЦИЯ

Дипломная работа содержит 37 страниц, 15 рисунков, 2 таблицы, 4 источника литературы.

ДВУХКАСКАДНЫЙ КОНЕЧНО-АВТОМАТНЫЙ ГЕНЕРАТОР, КРИПТОАНАЛИЗ, ПЕРИОДИЧЕСКИЕ СВОЙСТВА

Актуальность:

Популярные способы генерации ключевого потока, например, такие, как линейно-конгруэнтный метод, регистры сдвига хорошо изучены и известны атаки на них. Поэтому была поставлена цель изучить альтернативный способ генерации ключевого потока - двухкаскадный конечно-автоматный генератор.

Объекты исследования: двухкаскадный конечно-автоматный генератор, криптоанализ генератора, периодические свойства генератора.

Цель работы: исследование атак на двухкаскадный конечно-автоматный генератор и периодических свойств генератора.

Задачи:

Разработать атаки на двухкаскадный конечно-автоматный генератор с различными вариантами ключей. Изучить периодические свойства генератора. Использовать периодические свойства в криптоанализе. Сделать выводы относительно стойкости генератора.

Методы исследования: теоретический и экспериментальный.

Результаты: Рассмотрены способы криптоанализа генератора, когда его ключом является одно из множеств $\{x(1)\}$, $\{y(1)\}$, $\{x(1), y(1)\}$, $\{f_2\}$, $\{f_2, y(1)\}$, $\{f_2, y(1), x(1)\}$. На данный момент, для случаев, когда ключом является $\{f_2, y(1)\}$ или $\{f_2, y(1), x(1)\}$, не

удалось построить атак со сложностью меньше полного перебора, что говорит о криптографической стойкости генератора.

Написана программа, реализующая двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$ и находящая период автомата по выходной последовательности, а также по состояниям автомата A_2. Проведены исследования свойств периодичности генератора, выведены и доказаны вспомогательные теоремы.

Рассмотрены случаи криптоанализа для ключевых множеств $\{x(1)\}$, $\{y(1)\}$, $\{x(1), f_2\}$, $\{y(1), f_2\}$ в случае, если известно, что рассматриваемый генератор имеет максимальный период. Знание о максимальности периода позволяет не перебирать значения начальных состояний обоих автоматов, а использовав одно любое состояние, найти заданное.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ..... 6

1 Определение генератора..... 8

2 Криптоанализ генератора..... 9

2.1 Случай $K=\{x(1)\}$..... 10

2.2 Случаи $K=\{y(1)\}$ и $K=\{y(1), x(1)\}$..... 12

2.3 Случаи $K=\{f_2\}$, K=$\{y(1), f_2\}$ и $K=\{x(1), y(1), f_2\}$..... 14

3 Периодические свойства генератора..... 17

3.1 Вспомогательные утверждения..... 18

3.2 Теорема о максимальности периода генератора..... 19

3.3 Утверждения, полученные в результате экспериментов..... 21

3.4 Определение максимальности периода автомата..... 25

3.5 Композиция автоматов..... 28

4 Использование периодических свойств в криптоанализе..... 30

4.1 Случаи $K=\{y(1)\}$ и $K=\{y(1), f_2\}$..... 31

4.2 Случаи $K=\{x(1)\}$ и $K=\{x(1), f_2\}$..... 33

ЗАКЛЮЧЕНИЕ..... 35

ЛИТЕРАТУРА..... 36

ВВЕДЕНИЕ

В настоящее время конечные автоматы широко используются в разных областях криптографии [1]. Конечные автоматы могут выступать в роли самостоятельной криптосистемы, например в качестве алгоритма шифрования. Такие автоматы называются шифрующими автоматами [2]. Кроме того, конечные автоматы могут выступать в качестве одной из частей криптосистемы. Так, например, конечный автомат может выступать в роли генератора ключевого потока (ГКП) в поточных шифрах (схема представлена на рисунке 1). Генерируемый ключевой символ подается на вход функции шифрования вместе с символом открытого текста. На выходе функция шифрования возвращает символ шифротекста. Аналогично работает функция расшифрования, только на вход она принимает генерируемый ключевой символ и символ шифротекста, а на выходе - символ открытого текста.

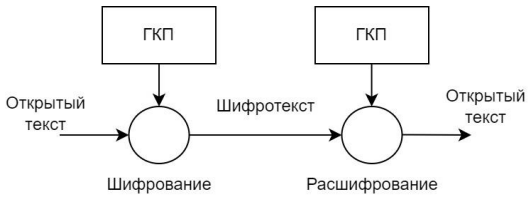


Рисунок 1 – Обобщенная схема работы поточного шифра

Альтернативой конечным автоматам в роли ГКП могут выступать, например, линейно-конгруэнтный метод, регистры сдвига и др. Но данные методы хорошо изучены и известны атаки на них. Поэтому была поставлена цель изучить альтернативный способ генерации ключевого потока, предложенный Агибаловым Геннадием Петровичем - двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$ [3]. Одной из рассматриваемых задач является задача криптоанализа данного генератора с различными вариантами ключа. Всего существует 63 варианта ключа, из которых ранее было рассмотрено 4 варианта [4]. В работе представлен криптоанализ для еще 6 вариантов.

Некоторые случаи криптоанализа сводились к наблюдению за выходами генератора и анализу полученных данных до момента, пока не будет найден ключ. Необходимо было ввести количество просматриваемых значений в строгие границы. В ходе наблюдений было выявлено, что генерируемая последовательность является периодической, соответственно длина периода является верхней границей для количества

просматриваемых элементов. Кроме того, периодические свойства генератора можно использовать в криптоанализе генератора, что может ускорить поиск ключевых параметров.

Данная работа посвящена криптоанализу двухкаскадного конечно-автоматного генератора, изучению его периодических свойств, а также влиянию конфигурации каждого из автоматов на период итоговой последовательности и применению полученных результатов в криптоанализе.

1 Определение генератора

Двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$ представляет собой последовательное соединение двух автоматов: автономного автомата $A_1 = (\mathrm{F}_2^n, \mathrm{F}_2, g_1, f_1)$ (с функцией переходов $g_1:\mathrm{F}_2^n \to \mathrm{F}_2^n$ и функцией выходов $f_1:\mathrm{F}_2^n \to \mathrm{F}_2$) и автомата $A_2 = (\mathrm{F}_2, \mathrm{F}_2^m, F_2, g_2, f_2)$ (с функцией переходов $g_2:\mathrm{F}_2\times\mathrm{F}_2^m \to \mathrm{F}_2^m$ и функцией выходов $f_2:\mathrm{F}_2\times\mathrm{F}_2^m \to \mathrm{F}_2$). На рисунке 2 представлена схема генератора G:

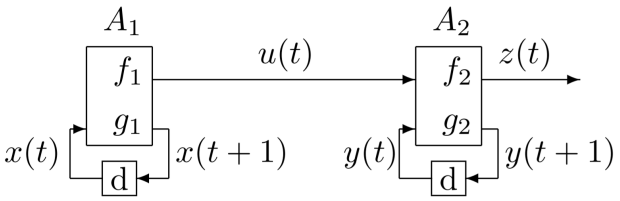


Рисунок 2 – Двухкаскадный конечно-автоматный криптографический генератор

Генератор работает в дискретном времени $t = 1, 2, \ldots$ В каждый момент времени t автомат A_1 в состоянии $x(t)\in\mathrm{F}_2^n$ выдает символ $u(t) = f_1(x(t))$, который поступает на вход автомата A_2, и переходит в новое состояние $x(t+1) = g_1(x(t))$, автомат A_2 в состоянии $y(t)\in\mathrm{F}_2^m$ принимает на вход символ $u(t)$, выдает символ $z(t) = f_2(u(t), y(t))$ и переходит в следующее состояние $y(t+1) = g_2(u(t), y(t))$. Последовательность $u(1)...u(k)$, где $k\in\mathrm{N}$, выходных символов автомата A_1 называется управляющей последовательностью автомата A_2, а последовательность $z(1)...z(k)$ выходных символов автомата A_2 - выходной последовательностью генератора G.

2 Криптоанализ генератора

Задача криптоанализа генератора состоит в определении ключа по его выходной последовательности. Ключом K генератора может быть любое непустое подмножество множества $\{x(1), g_1, f_1, y(1), g_2, f_2\}$. Если в ключевое множество K входят функции выходов первого и/или второго автомата, ключ называется функциональным. В случаях, когда ключ является функциональным, криптоаналитику дополнительно известен класс функций, которому принадлежит функция-ключ. Если в ключевое множество входит начальное состояние, функция переходов или функция выходов первого автомата, криптоаналитику неизвестна управляющая последовательность.

2.1 Случай $K=\{x(1)\}$

Так как ключом является начальное состояние первого автомата, криптоаналитику известны: выходная последовательность $z(1)...z(k)$; функции выходов f_1, f_2 обоих автоматов; функции переходов g_1, g_2 обоих автоматов; а также начальное состояние $y(1)$ автомата A_2. Управляющая последовательность $u(1)...u(k)$ криптоаналитику неизвестна.

Необходимо найти такое $x(1) \in F_2^n$, что

$$f_2(f_1(x(1)), y(1)) = z(1) \text{ и } z(t) = f_2(f_1(x(t)), y(t)),$$
$$x(t) = g_1(x(t - 1)), y(t) = g_2(f_1(x(t - 1), y(t - 1))$$

для $t = 2, \ldots, k$.

Знание управляющей последовательности упростит поиск ключа, так как

$$u(1) = f_1(x(1)).$$

Поиск начального состояния автомата A_1 разбивается на 2 этапа:

1) нахождение множества возможных управляющих последовательностей, при подаче которых на вход второго автомата первые k символов выходной последовательности второго автомата и соответственно генератора G равны $z(1)...z(k)$;
2) нахождение начального состояния первого автомата с использованием найденных управляющих последовательностей

Для решения задачи первого этапа используется метод DSS [3]. Построим граф, вершины которого расположены по ярусам с номерами $t \in \{1, 2, \ldots, l, l+1\}$ и помечены состояниями автомата A_2. Дуги графа помечаются значениями управляющей последовательности - 0 и 1. На первом ярусе расположена 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 $u(1)...u(l)$ дуг этого пути. Полученное множество управляющих последовательностей обозначим U.

Алгоритм нахождения начального состояния сводится к перебору всевозможных $x \in F_2^n$. Для этого для всех $x \in F_2^n$ находим символы управляющей последовательности u_x, пока существует такая $u \in U$, что $u_x(1)...u_x(l) = u(1)...u(l)$, где $l \le k$. Если $l = k$, то добавляем x в множество возможных ключей, иначе переходим к следующему состоянию. Сложность атаки равна $O(k2^n)$.

Рассмотрим пример атаки. Пусть генератор G_1 задан следующим образом (рисунок 3).

$X = \{00, 01, 10, 11\}, x_1 = 00$ $\quad$ $Y = \{00, 01, 10, 11\}, y_1 = 01$

$g_1(00) = 11$	$f_1(00) = 1$	$g_2(01, 1) = 10$	$f_2(01, 1) = 1$
$g_1(11) = 01$	$f_1(11) = 0$	$g_2(01, 0) = 00$	$f_2(01, 0) = 1$
$g_1(01) = 10$	$f_1(01) = 0$	$g_2(10, 1) = 11$	$f_2(10, 1) = 0$
$g_1(10) = 00$	$f_1(10) = 1$	$g_2(10, 0) = 00$	$f_2(10, 0) = 0$
		$g_2(00, 1) = 11$	$f_2(00, 1) = 1$
		$g_2(00, 0) = 01$	$f_2(00, 0) = 1$
		$g_2(11, 0) = 10$	$f_2(11, 0) = 0$
		$g_2(11, 1) = 00$	$f_2(11, 1) = 1$

Рисунок 3 – Генератор G_1

Выходная последовательность $z = 101$. Начальное состояние $x(1)$ является ключом. Строим дерево, используя алгоритм DSS (рисунок 4).

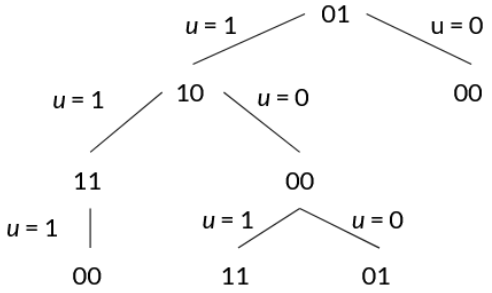


Рисунок 4 – Дерево, построенное при помощи алгоритма DSS для автомата G_1 и выходной последовательности $z = 101$

Тогда множество U будет состоять из 3 векторов: {111, 101, 100}. Так как все векторы из U начинаются с 1, имеет смысл рассматривать только состояния x, такие, что $f_1(x) = 1$, это 00 и 10. Если начальное состояние $x(1) = 00$, то управляющая последовательность $u_{00} = 100$ и $u_{00} \in U$, значит, включаем состояние 00 в множество решений. А если $x(1) = 10$, то управляющая последовательность $u_{10} = 110$, но $u_{10} \notin U$, следовательно, состояние 10 не включаем в множество решений. В итоге получается единственное решение $x(1) = 00$, что соответствует начальному состоянию автомата A_1 данного генератора G_1 (рисунок 3).

2.2 Случаи $K=\{y(1)\}$ и $K=\{y(1), x(1)\}$

Рассмотрим случай когда $K=\{y(1)\}$. Тогда криптоаналитику известны: выходная последовательность $z(1)...z(k)$; функции выходов f_1, f_2 обоих автоматов; функции переходов g_1, g_2 обоих автоматов; а также начальное состояние $x(1)$ автомата A_1. Так как про первый автомат криптоаналитику известна вся информация, ему также известна управляющая последовательность $u(1)...u(k)$.

Необходимо найти такое $y(1)\in F_2^m$, что

$$f_2(f_1(x(1)), y(1)) = z(1) \text{ и } y(t) = g_2(f_1(x(t-1), y(t-1)), z(t) = f_2(f_1(x(t)), y(t))$$

для $t = 2, \ldots, k$.

Сначала найдем множество $y\in F_2^m$, таких, что $f_2(u(1), y) = z(1)$. Данную операцию можно сделать разными способами в зависимости от того, в каком виде нам дана функция f_2: либо используя вектор значений, либо подставляя $u(1)$ и y в функцию f_2 и наблюдая за результатом. Полученное множество значений y обозначим Y.

Далее для каждого $y\in Y$ находим выходные символы z_y, пока $z_y(1)...z_y(l) = z(1)...z(l)$, где $l \le k$. Если $l = k$, то добавляем y в множество возможных ключей, иначе переходим к следующему. Сложность атаки будет равна $O(k2^n)$.

Рассмотрим пример атаки. Пусть генератор G_1 задан как на рисунке 2. Начальное состояние $y(1)$ является ключом. Выходная последовательность $z = 101$. Используя автомат A_1, найдем управляющую последовательность $u = 100$. Далее, используя функцию выходов f_2, находим множество $Y = \{00, 01, 11\}$. Для каждого $y\in Y$ находим выходную последовательность z_y (рисунок 5)

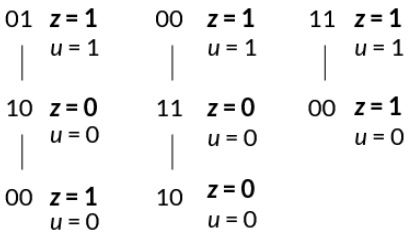


Рисунок 5 – Выходные последовательности для $y\in Y$

Из рисунка видно, что только для состояния $y = 01$ выходная последовательность z_y равна заданной выходной последовательности, следовательно, состояние 01 является ключом.

Данную задачу можно расширить, добавив к ключу начальное состояние первого автомата $x(1)$. Тогда управляющая последовательность $u(1)...u(k)$ будет неизвестной. В

этом случае для каждого $x \in F_2^n$ будем строить управляющую последовательность, дальнейшие действия сводятся к предыдущей задаче. Сложность атаки равна $O(k2^{n+m})$.

2.3 Случаи $K=\{f_2\}$, $K=\{y(1), f_2\}$ и $K=\{x(1), y(1), f_2\}$

Рассмотрим случай когда $K=\{f_2\}$. Тогда криптоаналитику известны: выходная последовательность $z(1)...z(k)$; функция выходов f_1 автомата A_1; функции переходов g_1, g_2 обоих автоматов; начальные состояния $x(1)$ и $y(1)$ обоих автоматов; а также класс функций C_2, к которому принадлежит функция f_2. Так как про первый автомат криптоаналитику известна вся информация, ему также известна управляющая последовательность $u(1)...u(k)$.

Необходимо найти такую функцию $f_2 \in C_2$, что

$$f_2(f_1(x(t)), y(t)) = z(t), \text{ где } y(t+1) = g_2(f_1(x(t)), y(t))$$

для $t = 1, \ldots, k$.

Задача нахождения функции выходов f_2 сводится к задаче доопределения функции до функции класса. Будем наблюдать за работой автомата и сопоставлять соответствующим значениям переменных значения функции, пока полностью не определим функцию f_2, либо пока не сможем применить алгоритм доопределения функции в классе C_2. В лучшем случае, когда не будет повторений значений переменных функции f_2, сложность задачи будет 2^{m+1}. Например, если C_2 - класс уравновешенных функций, то в лучшем случае, когда не будет повторений значений переменных функции f_2 и выходная последовательность z состоит из всех 1 или 0, сложность задачи будет 2^m.

Расширим задачу и добавим к ключу начальное состояние второго автомата $y(1)$. Тогда ключом будет являться множество $\{y(1), f_2\}$. Данный случай сводится к предыдущему, перебирая $y \in F_2^m$, будем наблюдать за работой и сопоставлять соответствующим значениям переменных значения функции. Отбрасывать y будем либо при нарушении свойства функциональности, то есть при попытке перезаписать значение функции на противоположное, либо при нарушении свойства функций класса C_2.

Добавим к множеству $\{y(1), f_2\}$ начальное состояние первого автомата $x(1)$. Задача поиска полученной тройки сводится к предыдущей, с добавлением перебора $x \in F_2^n$. В данном случае управляющая последовательность $u(1)...u(k)$ криптоаналитику не будет известна. Для каждого x будем строить управляющую последовательность $u(1)...u(k)$ и проводить аналогичные действия, как и в прошлой задаче.

Рассмотрим примеры атак. Пусть генератор G_2 задан следующим образом (рисунок 6).

$X = \{00, 01, 10, 11\}, x_1 = 00$

$g_1(00) = 11$ $f_1(00) = 1$
$g_1(11) = 01$ $f_1(11) = 0$
$g_1(01) = 10$ $f_1(01) = 0$
$g_1(10) = 00$ $f_1(10) = 1$

$Y = \{00, 01, 10, 11\}, y_1 = 01$

$g_2(01, 1) = 10$ $f_2(1, 01) = 0$
$g_2(01, 0) = 00$ $f_2(0, 01) = 1$
$g_2(10, 1) = 11$ $f_2(1, 10) = 0$
$g_2(10, 0) = 00$ $f_2(0, 10) = 0$
$g_2(00, 1) = 11$ $f_2(1, 00) = 1$
$g_2(00, 0) = 01$ $f_2(0, 00) = 1$
$g_2(11, 0) = 10$ $f_2(0, 11) = 0$
$g_2(11, 1) = 00$ $f_2(1, 11) = 1$

Рисунок 6 – Генератор G_2

Выходная последовательность z = 001100. Класс C_2 - класс уравновешенных функций. Функция выходов f_2 является ключом. Используя автомат A_1, найдем управляющую последовательность u = 100110. Наблюдая за работой генератора, получаем ситуацию, когда четырем различным значениям параметра соответствует значение функции равное 0, следовательно, наблюдения можно прекратить и оставшимся значениям параметра сопоставить значение функции равное 1 (рисунки 7, 8). В итоге получаем функцию, равную f_2.

U	*Y*	*Z*
1	01	0
0	10	0
0	00	1
1	11	1
1	10	0
0	11	0

Рисунок 7 – Имитация работы генератора G

U,Y	*Z*
000	1
001	1
010	0
011	0
100	1
101	0
110	0
111	1

Рисунок 8 – Функция выходов f_2 автомата A_2

Пусть генератор G_1 задан как на рисунке 2, выходная последовательность z = 100100011011. Класс C_2 - класс уравновешенных функций. Функция выходов f_2 и начальное состояние $y(1)$ является ключом. Используя автомат A_1, найдем управляющую последовательность u = 100110011001. Для каждого состояния $y \in F_2^m$ наблюдаем за работой генератора. Получаем, что при $y \in \{00, 10, 11\}$ нарушается свойство функциональности

получившейся функции (рисунок 9). В результате получаем начальное состояние $y(1) = 01$ и функцию выходов f_2, соответствующие заданным.

U	Z
1	1
0	0
0	0
1	1
1	0
0	0
0	0
1	1
1	1
0	0
0	1
1	1

y(1) = 00	
Y,U	Z
001	1
110	0
100	0
001	1
111	0
000	0
010	0
001	1
111	1
000	0
010	1

y(1) = 01	
Y,U	Z
011	1
100	0
000	0
011	1
101	0
110	0
100	0
001	1
111	1
000	0
010	1

y(1) = 10	
Y,U	Z
101	1
110	0
100	0
001	1
111	0
000	0
010	0
001	1
111	1
000	0
010	1

y(1) = 11	
Y,U	Z
111	1
000	0
010	0
001	1
111	0

Рисунок 9 – Пример поиска начального состояния y(1) и функции выходов f_2

3 Периодические свойства генератора

Одним из основных требований, предъявляемых к генераторам, является достаточно длинный период выходной последовательности. Иначе, например, в случае когда двухкаскадный генератор выступает в роли ГКП, генерируемая последовательность будет часто повторяться. Следовательно, открытый текст будет шифроваться большим числом одинаковых последовательностей, малой длины, что делает шифр уязвимым, и появляется необходимость чаще менять ключевые параметры генератора.

Дадим определение периода последовательности. Последовательность a имеет период k, если, начиная с n-го элемента, для всех следующих элементов последовательности выполняется условие $a(n + i) = a(n + i + k)$ для всех $i = 0, \ldots, \infty$. Стоить отметить, что число k - минимальное из возможных. Периодом автомата или генератора будем называть период его выходной последовательности.

3.1 Вспомогательные утверждения

Начнем рассмотрение генератора G с автомата A_1. Так как A_1 - автономный автомат, его выходная последовательность периодична. Причем стоит отметить несколько утверждений.

Утверждение 1: Если период автомата A_1 принимает максимальное значение, то функция его переходов g_1:$F_2^n \rightarrow F_2^n$ представляет собой полноцикловую подстановку.

Доказательство: Из условия следует, что мощность множества состояний автомата A_1 равна 2^n. Соответственно, очередное $(2^n + 1)$-е состояние автомата будет равно одному из состояний, пройденных ранее. Выход автомата A_1 - $u(t)$ зависит только от состояния $x(t)$, в котором он находится в данный момент времени t. Тогда если состояние $x(t)$ повторяет состояние $x(l)$, то выходной символ $u(t)$ повторяет выходной символ $u(l)$, $l \neq t$. Но и следующее состояние $x(t + 1)$ повторит состояние $x(l + 1)$, что приведет к повторению выходного символа $u(t + 1) = u(l + 1)$ и т.д. Получается, что значение периода выходной последовательности автомата равно или делит максимальную длину последовательности различных состояний автомата A_1, в которые он переходит во время своей работы. А максимальное количество достигается только в том случае, если g_1 - полноцикловая подстановка.

Утверждение 2: Если g_1 - подстановка, то период автомата A_1 начинается с первого элемента выходной последовательности.

Доказательство: Т.к. g_1 - подстановка, то ее можно записать в виде произведения независимых циклов. Пусть начальное состояние автомата A_1 содержится в одном из циклов длиной l. Тогда $x(1 + i) = x(1 + i + l)$ для всех $i = 0, \ldots, \infty$. Но, исходя из доказательства *Утверждение 1*, $u(1 + i) = u(1 + i + l)$ для всех $i = 0, \ldots, \infty$. По определению периода, данного в п.3, следует, что последовательность u имеет период, который начинается с первого ее элемента.

В дальнейшем будем подразумевать, что функция переходов автомата A_1 - полноцикловая подстановка.

Начальное состояние $x(1)$ автомата A_1 не влияет на длину периода выходной последовательности. При изменении $x(1)$ на $x(1)'$ выходная последовательность будет циклически сдвигаться влево на n элементов, где n - число шагов от $x(1)$ до $x(1)'$.

Выходные функции обоих автоматов будем считать отличными от констант, иначе:

1. Если функция выходов первого автомата равна const, то автомат A_2 становится автономным и генератор G становится эквивалентным подавтомату A_2;
2. Если функция выходов второго автомата равна const, то выходная последовательность также является константой.

Далее рассмотрим, какое максимальное значение может принимать период генератора и при каких условиях это значение достигается.

3.2 Теорема о максимальности периода генератора

Теорема 1: Пусть задан двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$, где функция переходов первого автомата $g_1:F_2^n \rightarrow F_2^n$, а функция переходов второго автомата $g_2:F_2 \times F_2^m \rightarrow F_2^m$. Тогда максимальное значение периода генератора равняется 2^{n+m}.

Доказательство: Из условия теоремы следует, что мощность множества состояний автомата A_1 равна 2^n, а мощность множества состояний автомата A_2 равна 2^m. Всего вариантов пар $(x(i), y(j))$ (где $i = 1, \ldots, 2^n$, $j = 1, \ldots, 2^m$) будет 2^{n+m}. Соответственно очередная $(2^{n+m} + 1)$-я пара состояний будет равна одной из пар, пройденных ранее. А так как выход автомата A_1 - $u(t)$ зависит только состояния $x(t)$, в котором он находится в данный момент времени t, то если пара $(x(t), y(t))$ повторяет пару $(x(l), y(l))$, то пара $(y(t), u(t))$ повторяет пару $(y(l), u(l))$, $l \neq t$. В свою очередь повторение пары $(y(t), u(t))$ влечет повторение выходного символа генератора $z(t)$, т.е. $z(t) = z(l)$. Но следующая пара $(x(t + 1), y(t + 1))$ повторит пару $(x(l + 1), y(l + 1))$, что приведет к повторению выходного символа $z(t + 1) = z(l + 1)$ и т.д.

Из теоремы можно выделить следующие следствия, которые являются необходимыми условиями максимальности периода генератора.

Следствие 1: Если период генератора максимален, то функция переходов автомата A_1 является полноцикловой подстановкой.

Доказательство: Докажем от обратного, допустим, функция переходов автомата A_1 не образует полноцикловую подстановку. Тогда по *Утверждению 1* ее максимальный период меньше 2^n, и количество различных состояний автомата A_1, в которые переходит первый автомат во время работы генератора G, меньше 2^n. Следовательно, количество различных пар состояний $(x(i), y(j))$, которые пройдёт генератор G, начиная с $(x(1), y(1))$, до повторения меньше 2^{n+m}. В итоге, исходя из рассуждений, представленных в доказательстве *Теоремы 1*, период генератора G будет меньше 2^{n+m}, что противоречит условию.

Следствие 2: Если период генератора максимален, то в первых 2^n элементах выходной последовательности автомата A_1 нет периода.

Доказательство: Докажем от обратного, пусть выходная последовательность автомата A_1 имеет период l, причем $2^n \vdots l$ и $l \neq 2^n$. Тогда построим автомат A_1`, количество

состояний которого будет равно l, а выходная последовательность равна первым l элементам выходной последовательности автомата A_1. Тогда генератор $G` = A_1` \cdot A_2$ будет эквивалентен автомату G, так как через 2^n тактов автомата $A_1`$ выходная последовательность $G`$ будет равна выходной последовательности G. Следовательно, последовательности генераторов будут одинаковые и при дальнейшем наблюдении. Но тогда максимальный период G и $G`$ будет равен $l2^m$, что меньше 2^{n+m}, т.к. $l < 2^n$.

Рассмотрим подстановку следующего вида:

$$\left(\frac{01234567}{12345670}\right)$$

В данной подстановке i-й элемент переходит в i+1 для i = 0, …, 2^n-2, а (2^n-1)-й элемент переходит в 0. Назовем такую подстановку последовательной. Последовательная подстановка является полноцикловой.

По *Следствию 1*, будем считать, что функция переходов первого автомата это последовательная подстановка. Если дан генератор, функция переходов первого автомата которого не последовательная подстановка, то построим автомат $A_1`$, эквивалентный A_1: переобозначим состояния первого автомата так, чтобы подстановка стала последовательной, соответственно заменив начальное состояние и функцию выходов. Полученный генератор будет эквивалентен заданному.

3.3 Утверждения, полученные в результате экспериментов

В ходе практики была разработана программа, реализующая двухкаскадный конечно-автономный генератор и находящая период в выходной последовательности. В результате экспериментов сформулирован ряд гипотез, некоторые из которых доказаны, другие требуют доказательства.

Утверждение 3: Если период генератора максимален, то изменение начального состояния первого автомата не изменяет период.

Доказательство: Пусть начальное состояние автомата A_1 было $x(i)$, а начальное состояние автомата A_2 - $y(j)$. Из рассуждений, представленных в доказательстве *Теоремы 1*, следует, что генератор переберет все возможные пары, начиная с пары $(x(i), y(j))$. Поменяем начальное состояние первого автомата на $x(k)$, теперь первой парой будет $(x(k), y(j))$. Но эта пара достижима из пары $(x(i), y(j))$ за l тактов генератора, $l \leq 2^{n+m}-1$, и в следующих $2^{n+m}-1$ парах, пара $(x(k), y(j))$ не встретится, иначе период будет не максимальным, что противоречит условию. Но тогда получается, что начиная с пары $(x(k), y(j))$ генератор также переберет 2^{n+m} всевозможных различных пар, следовательно, исходя из рассуждений, представленных в доказательстве *Теоремы 1*, период останется максимальным.

Утверждение 4: Если период генератора максимален, то изменение начального состояния второго автомата не изменяет период.

Доказательство: Пусть начальное состояние автомата A_2 было $y(i)$, а начальное состояние автомат A_1 - $x(j)$. Из рассуждений, представленных в доказательстве *Теоремы 1*, следует, что генератор переберет все возможные пары, начиная с пары $(x(j), y(j))$. Поменяем начальное состояние второго автомата на $y(k)$, теперь первой парой будет $(x(k), y(j))$. Дальнейшее доказательство аналогично доказательству *Утверждения 3*.

Рассмотрим на примере. Пусть генератор G_3 задан следующим образом (рисунок 10).

$X = \{00, 01, 10, 11\}, x_1 = 00$	$Y = \{00, 01, 10, 11\}, y_1 = 00$
$g_1(00) = 01$ $f_1(00) = 1$	$g_2(0, 00) = 01$ $f_2(0, 00) = 0$
$g_1(01) = 10$ $f_1(01) = 0$	$g_2(0, 01) = 10$ $f_2(0, 01) = 0$
$g_1(10) = 11$ $f_1(10) = 1$	$g_2(0, 10) = 00$ $f_2(0, 10) = 0$
$g_1(11) = 00$ $f_1(11) = 1$	$g_2(0, 11) = 11$ $f_2(0, 11) = 0$
	$g_2(1, 00) = 00$ $f_2(1, 00) = 1$
	$g_2(1, 01) = 01$ $f_2(1, 01) = 1$
	$g_2(1, 10) = 11$ $f_2(1, 10) = 1$
	$g_2(1, 11) = 10$ $f_2(1, 11) = 0$

Рисунок 10 – Генератор G_3

Проимитируем работу генератора (рисунок 11).

x u y z	x u y z
0 - 1 - 0 - 1	0 - 1 - 2 - 1
1 - 0 - 0 - 0	1 - 0 - 3 - 0
2 - 1 - 1 - 1	2 - 1 - 3 - 0
3 - 1 - 1 - 1	3 - 1 - 2 - 1
0 - 1 - 1 - 1	0 - 1 - 3 - 0
1 - 0 - 1 - 0	1 - 0 - 2 - 0
2 - 1 - 2 - 1	2 - 1 - 0 - 1
3 - 1 - 3 - 0	3 - 1 - 0 - 1

Рисунок 11 – Работа генератора G_3

Выходная последовательность z = 1011 1010 1001 0011 1011 1010 1001 0011… Период выходной последовательности равен 16. Поменяем теперь начальное состояние второго автомата на 01. Теперь выходная последовательность z’ = 1010 1001 0011 1011 1010 1001 0011 1011… Период последовательности остался так же равен 16. Видно, что вторая последовательность получена сдвигом первой на 4 позиции.

Аналогичная ситуация при изменении начального состояния первого автомата на 01. Выходная последовательность станет равной z’ = 0111 0101 0010 0111 0111 0101 0010 0111… Выходная последовательность получена сдвигом первой на 1 позицию.

В результате экспериментов была выявлена следующая закономерность. Если заменить функцию выходов первого автомата на такую функцию выходов, при которой выходная последовательность первого автомата циклически сдвигается, то это приведет только к циклическому сдвигу его выходной последовательности, а период генератора останется тем же.

Утверждение 5: Если период генератора максимален, то замена функции выходов первого автомата на функцию выходов, при которой выходная последовательность первого автомата циклически сдвигается, не приведет к изменению периода генератора.

Доказательство: Пусть нам дан генератор, у которого функция переходов автомата A_1 задана последовательной подстановкой (если это не так, то можно перенумеровать его состояния и прийти к такому виду). Тогда доказательство сводится к случаю, описанному в *Утверждении 3*. Изменение начального состояния автомата будет подразумевать циклический сдвиг его вектора функции выхода, следовательно, и самой выходной последовательности (за счет функции переходов).

Рассмотрим на примере. Возьмем генератор G_3 такой же, как на рисунке 10. Выходная последовательность данного генератора z = 1011 1010 1001 0011 1011 1010 1001 0011… Период выходной последовательности равен 16. Теперь изменим функцию выходов первого автомата на функцию, вектор значений которой циклически сдвинут на 1 единицу влево, то есть равен 0111. Проимитируем работу полученного генератора *G'* (рисунок 12).

x u y z	*x u y z*
0 - 0 - 0 - 0	0 - 0 - 3 - 0
1 - 1 - 1 - 1	1 - 1 - 3 - 0
2 - 1 - 1 - 1	2 - 1 - 2 - 1
3 - 1 - 1 - 1	3 - 1 - 3 - 0
0 - 0 - 1 - 0	0 - 0 - 2 - 0
1 - 1 - 2 - 1	1 - 1 - 0 - 1
2 - 1 - 3 - 0	2 - 1 - 0 - 1
3 - 1 - 2 - 1	3 - 1 - 0 - 1

Рисунок 12 – Работа генератора *G'*

Теперь выходная последовательность *z*' = 0111 0101 0010 0111 0111 0101 0010 0111… Период последовательности остался так же равен 16. Видно, что вторая последовательность получена сдвигом первой на 1 позицию.

Данные утверждения позволяют при дальнейшем изучении генераторов с максимальным периодом сократить количество экспериментов в 2^n раз (т.к. чтобы двоичный вектор длины 2^n вернулся в исходное состояние, нужно сделать 2^n циклических сдвигов в одну сторону). Кроме того, можно зафиксировать начальные состояния обоих автоматов, например они всегда будут равны 0, и функцию переходов первого автомата, использовав последовательную подстановку.

Также в результате экспериментов было выявлено, что подфункции переходов второго автомата (функции переходов для $u = 1$ и $u = 0$) должны быть подстановками, чтобы период генератора был максимален. При выполнении данного условия докажем следующее утверждение.

Утверждение 6: Если функция переходов первого автомата и подфункции переходов второго автомата являются подстановками, то период генератора G начинается с первого элемента выходной последовательности.

Доказательство: Предположим, что период начинается с k-го элемента выходной последовательности, где $k > 0$, автомат A_1 находится в состоянии $x(i)$, а автомат A_2 - в состоянии $y(j)$. Через число тактов равное периоду, автомат A_1 также будет находится в состоянии $x(i)$, а автомат A_2 - в состоянии $y(j)$. Тогда в состояние $x(i)$ можно попасть из двух состояний $x(q)$ и $x(p)$, причем $x(q) \neq x(p)$, т.к. иначе выход автомата A_1 был бы одинаковым, следовательно, одинаковым был бы и выходной символ генератора, следовательно период начинался бы не с k-го элемента а с (k-1)-го. Получаем, что $x(i) = g_1(x(q)) = g_1(x(p))$. Но, по условию, функция переходов первого автомата представляет подстановку, получаем противоречие. Получается, что $x(q) = x(p)$ и $u(q) = u(p)$. Аналогично рассматривается второй автомат. В состояние $y(j)$ автомата A_2 также можно перейти из двух состояний $y(c)$ и $y(r)$ под действием управляющего бита u. Получаем, что $y(j) = g_2(y(c), u) = g_2(y(r), u)$, что противоречит условию.

Следующие результаты были выявлены экспериментальным путем, но не доказаны.

Предположение 1: Если период генератора максимален, то подфункции переходов автомата A_2 (функции переходов для $u = 1$ и $u = 0$) должны быть подстановками.

Предположение 2: Если период генератора максимален, то вес функции выходов первого автомата - нечетный.

Предположение 3: Если вес функции выходов первого автомата четное число, то максимальный период генератора равен $2^{n+m} - 2^n$.

Предположение 4: Если период генератора больше 2^n, то он кратен 2^n.

Предположение 5: Для любого автомата A_1, у которого g_1 - полноцикловая подстановка, а $w(f_1)$ - нечетный, можно сгенерировать автомат A_2 так, чтобы период генератора был максимальным.

3.4 Определение максимальности периода автомата

В ходе проведения экспериментов возникала задача определения периода генератора. Так, например, для изучения требований к каждому из автоматов A_1 и A_2 генератора с максимальным периодом необходимо уметь отличать такие генераторы от остальной массы случайным образом сгенерированных генераторов. Так как нам известна величина максимально возможного периода для генератора с заданной конфигурацией, можно для каждого генератора искать период его выходной последовательности, сравнивать с максимальным и таким образом отсеивать.

Изначально для данной задачи использовался трудозатратный алгоритм, при котором перебираются все возможные длины периодов и определяется, имеется ли в последовательности период с текущей длиной. Помимо трудозатратности одной из проблем такого алгоритма является необходимость подавать на его вход последовательность длины $2k$, чтобы определить, имеется ли в последовательности период равный k. Поэтому предложен менее затратный алгоритм.

Вместо элементов выходной последовательности будет запоминать состояния второго автомата. Введем новое понятие - траектория автомата. Траекторией первого автомата назовем последовательность состояний, в которые переходит автомат за 2^n тактов. Траекторией второго автомата назовем последовательность состояний, в которые переходит автомат за 2^n тактов под действием элементов управляющей последовательности. Длина траектории в обоих случаях будет одинаковой 2^n. Для удобства будем записывать состояния в матрицу размером $k \times 2^n$, где $k = 1, \ldots, 2^m$. Строками матрицы являются траектории второго автомата. Для определения максимальности периода достаточно построить матрицу размером $2^m \times 2^n$, то есть генератор должен отработать 2^{n+m} тактов. Такую матрицу назовем матрицей пути генератора. В итоге суть построения матрицы выделим в утверждение.

Утверждение 7: Если период генератора максимален, то в каждом столбце матрицы пути генератора все состояния различны.

Доказательство: Доказательство аналогично доказательству *Теоремы 1*, т.к. каждой траектории второго автомата соответствует траектория первого автомата. Получается набор пар из декартового произведения состояний первого и второго автомата. А если в

какой-то момент пара повторится, то период не будет максимален, т.к. в матрице пути всего 2^{n+m} элементов. Получаем противоречие.

Рассмотрим еще одно утверждение, которое поможет упростить поставленную задачу.

Утверждение 8: Если период генератора максимален, то если в первом столбце матрицы пути генератора все состояния различны, то и в других столбцах состояния также различны.

Доказательство: Для доказательства воспользуемся *Утверждением 4* и *Утверждением 5*. Возьмем первую траекторию матрицы пути генератора и будем изменять начальное состояние второго автомата на следующее в траектории, а вектор функции выходов первого автомата циклически сдвигать влево на один символ. После этого построим матрицу пути нового генератора. Период нового генератора останется таким же - максимальным, а столбцы матрицы пути генератора циклически сдвинутся влево на один. Таким образом, по *Утверждению 7* в каждом столбце новой матрицы пути отсутствуют одинаковые значения. Повторив операцию 2^n раз, проверим все столбцы матрицы, что доказывает выдвинутое утверждение.

Проиллюстрируем на примере. В качестве примера возьмем генератор из рисунка 10 и построим его матрицу пути (рисунок 13).

00	00	01	01
01	01	10	11
10	11	11	10
11	10	00	00

Рисунок 13 – Матрица пути генератора *G*

Видно, что в каждом столбце отсутствуют одинаковые значения. Теперь попробуем циклически сдвинуть матрицу на один столбец влево. Для этого сдвинем вектор функции значений первого автомата на одну единицу влево, получим вектор 0111, а состояние второго автомата оставим неизменным, т.к. следующее состояние в первой траектории второго автомата равно 00. Получим генератор *G'* и соответствующую матрицу пути (рисунок 14).

00	01	01	01
01	10	11	10
11	11	10	11
10	00	00	00

Рисунок 14 – Матрица пути генератора *G'*

Можно заметить, что столбцы матрицы пути циклически сдвинулись на один, а значения зеленого столбца циклически сместились на одну позицию вверх (но это не влияет на уникальность его значений). Повторим процедуру еще 2 раза, каждый из столбцов окажется на первом месте.

По *Утверждению 8* можно не строить всю матрицу полностью, а только запоминать каждый $i2^n$-й элемент, где $i = 0, \ldots, 2^m - 1$. Тогда для определения уникальности чисел в полученном массиве воспользуемся следующим алгоритмом. Будем записывать в битовый вектор длины 2^m, изначально заполненный нулями, единицу в тот бит, номер которого равен очередному элементу в полученном массиве. Если в очередном бите при попытке записи уже стоит единица, то алгоритм заканчивает работу с ответом, что период генератора не максимален. Если же преждевременного выхода не было, то алгоритм заканчивает работу с ответом, что период генератора максимален. Сложность такого алгоритма $O(2^m)$.

3.5 Композиция автоматов

Для ускорения работы генератора можно построить эквивалентный автономный автомат $C = (F_2^{n+m}, F_2, g, f)$ (с функцией переходов $g: F_2^{n+m} \to F_2^{n+m}$ и функцией выходов $f: F_2^{n+m} \to F_2$), который будет производить меньше операций, чем исходный генератор. Построение композиции позволяет сократить количество операций до двух: перейти в новое состояние и выдать очередной символ последовательности. Для построения сделаем ряд подготовительных операций:

1. Запишем функцию переходов автомата A_1 в виде массива длины 2^n. На i-ом месте расположен элемент $x(j)$, если $x(j) = g_1(x(i))$, $i = 0, \ldots, 2^n - 1$;
2. Функцию переходов автомата A_2 запишем в виде массива длины 2^{m+1}. В первой части расположены переходы при $u = 0$, во второй - при $u = 1$. Построение массива происходит аналогичным образом, как и для автомата A_1. На i-м месте расположен элемент $y(j)$, если $y(j) = g_2(y(i), 0)$, а на $(i+2^m)$-м $y(j)$, если $y(j) = g_2(y(i), 1)$, $i = 0, \ldots, 2^m - 1$;
3. Аналогичным способом представляем функции выходов обоих автоматов, только вместо состояний записывает выходные значения автоматов в состояниях $x(i)$ и $y(i)$.

Теперь опишем сам алгоритм построение композиции автоматов.

Алгоритм

Дано: g_1, g_2 - массивы, подготовленные по функциям переходов автоматов A_1 и A_2 соответственно; f_1, f_2 - массивы, подготовленные по функциям выходов автоматов A_1 и A_2; $x(1)$, $y(1)$ - начальные состояния автоматов A_1 и A_2.

1. Начальное состояние итогового автомата вычисляется по формуле $x = 2^n x(1) + y(1)$.
2. Функции переходов и выходов будут заданы в массивах длиной по 2^{n+m}.
3. $i := 0$, пока $i < 2^n - 1$:
   - 3.1. $j := 0$, пока $j < 2^m - 1$:
     - 3.1.1. $g[i2^n + j] = 2^n g_1[i] + g_2[j + 2^m f_1[i]]$.
     - 3.1.2. $f[i2^n + j] = f_2[j + 2^m f_1[i]]$.
     - 3.1.3. j ++.
   - 3.2. i ++.

Сложность алгоритма построения $O(2^{n+m})$.

Ниже приведена конфигурация композиции C (рисунок 15) генератора G из рисунка 10.

$X = \{0000, 0001, ..., 1110, 1111\}$ $x = 0000$

$g(0000) = 0100$	$g(0100) = 1001$	$f(0000) = 1$	$f(0100) = 0$
$g(0001) = 0101$	$g(0101) = 1010$	$f(0001) = 1$	$f(0101) = 0$
$g(0010) = 0111$	$g(0110) = 1000$	$f(0010) = 1$	$f(0110) = 0$
$g(0011) = 0110$	$g(0111) = 1011$	$f(0011) = 0$	$f(0111) = 0$
$g(1000) = 1100$	$g(1100) = 0000$	$f(1000) = 1$	$f(1100) = 1$
$g(1001) = 1101$	$g(1101) = 0001$	$f(1001) = 1$	$f(1101) = 1$
$g(1010) = 1111$	$g(1110) = 0011$	$f(1010) = 1$	$f(1110) = 1$
$g(1011) = 1110$	$g(1111) = 0010$	$f(1011) = 0$	$f(1111) = 0$

Рисунок 15 – Конфигурация композиции C генератора G

Проверим композицию на эквивалентность генератору. Выходная последовательность генератора z = 1011 1010 1001 0011 1011 1010 1001 0011… Период выходной последовательности равен 16. Выходная последовательность композиции z' = 1011 1010 1001 0011 1011 1010 1001 0011…, и ее период также равен 16.

4 Использование периодических свойств в криптоанализе

Как было сказано в п.3, одним из основных требований, предъявляемых к генераторам, является достаточно большой период выходной последовательности. Но не всегда максимальность периода генератора означает его криптостойкость. В некоторых случаях, если криптоаналитику известно, что рассматриваемый им генератор имеет максимальный период, то он может значительно сократить задачу поиска ключа. Продемонстрируем это на некоторых ключевых наборах, которые впоследствии не рекомендуется использовать в качестве ключа. В криптоанализе используется *Теорема 1* о максимальности периода, а также ряд утверждений.

4.1 Случаи $K=\{y(1)\}$ и $K=\{y(1), f_2\}$

Так как ключом является начальное состояние второго автомата, криптоаналитику известны: выходная последовательность $z(1)...z(k)$, для того, чтобы гарантированно найти ключ $k = 2^{n+m}$; функции выходов f_1, f_2 обоих автоматов; функции переходов g_1, g_2 обоих автоматов; а также начальное состояние $x(1)$ автомата A_1. Так как про первый автомат криптоаналитику известна вся информация, ему также известна управляющая последовательность $u(1)...u(k)$.

Необходимо найти такое $y(1) \in F_2^m$, что

$$f_2(f_1(x(1)), y(1)) = z(1) \text{ и } y(t) = g_2(f_1(x(t - 1), y(t - 1)),$$
$$z(t) = f_2(f_1(x(t)), y(t))$$

для $t = 2, ..., k$.

Алгоритм поиска ключа основан на *Утверждении 3* и состоит из следующих шагов:

1. Фиксируем $y(1)$ в любом состоянии и запоминаем его. Заведем переменную $l := 0$;
2. Находим выходную последовательность генератора такой же длины, как заданная последовательность, параллельно запоминая в массив первый столбец матрицы пути (то есть каждое $i2^n$, $i = 0, \ldots, 2^m - 1$, состояние, в которое второй автомат переходит во время работы генератора);
3. Если полученная последовательность совпала с заданной, то начальное состояние второго автомата будет равно l-му по модулю 2^m элементу массива, сформированного на шаге 2. Выход из алгоритма;
4. Иначе сдвигаем полученную последовательность на 2^n элементов влево и увеличиваем l на 1. Переход на шаг 3.

Сложность данного алгоритма $O(\lfloor k / 2^n \rfloor)$, что много меньше сложности алгоритма атаки без знания о максимальности периода, которая равна $O(k2^m)$.

Добавив к ключевому множеству функцию выходов второго автомата f_2, сначала воспользуемся атакой из п.3.3 и найдем f_2, а затем, используя алгоритм выше, найдем начальное состояние второго автомата $y(1)$.

Рассмотрим действие алгоритма на примере. Пусть дан генератор G из рисунка 10. Его выходная последовательность равна z = 1011 1010 1001 0011. Зафиксируем $y(1)$

например в 01, $l := 0$. Выходная последовательность полученного генератора будет равна z' = 1010 1001 0011 1011, а первый столбец матрицы пути равен [01, 10, 11, 00]. Сдвинув последовательность z' на 4 элемента влево, получим z' = 1001 0011 1011 1010, $l := 1$. Последовательность не совпала, значит, идем дальше. Повторив процедуру еще два раза, получим, что выходная последовательность z' = 1011 1010 1001 0011 будет равна заданной, а $l := 3$. Тогда $y(1)$ будет равен 3-му элементу построенного массива, то есть 00.

В итоге получаем, что больше нет необходимости перебирать все 2^m состояний второго автомата, достаточно выбрать одно состояние и, используя его, найти ключ.

4.2 Случаи $K=\{x(1)\}$ и $K=\{x(1), f_2\}$

Так как ключом является начальное состояние первого автомата, криптоаналитику известны: выходная последовательность $z(1)...z(k)$, для того, чтобы гарантированно найти ключ $k = 2^{n+m}$; функции выходов f_1, f_2 обоих автоматов; функции переходов g_1, g_2 обоих автоматов; а также начальное состояние $y(1)$ автомата A_2. Управляющая последовательность $u(1)...u(k)$ криптоаналитику неизвестна.

Необходимо найти такое $x(1) \in F_2^n$, что

$$f_2(f_1(x(1)), y(1)) = z(1) \text{ и } z(t) = f_2(f_1(x(t)), y(t)),$$
$$x(t) = g_1(x(t - 1)), y(t) = g_2(f_1(x(t - 1), y(t - 1)),$$

Алгоритм поиска ключа основан на *Утверждении 4* и состоит из следующих шагов:

1. Фиксируем $x(1)$ в любом состоянии и запоминаем его. Заведем переменную $l := 0$;
2. Находим выходную последовательность генератора такой же длины, как заданная последовательность, параллельно запоминая в массив первые 2^n состояний, в которые первый автомат переходит во время работы генератора;
3. Если полученная последовательность совпала с заданной, то начальное состояние второго автомата будет равно l-му по модулю 2^n элементу массива сформированного на шаге 2. Выход из алгоритма;
4. Иначе сдвигаем полученную последовательность на 1 элемент влево и увеличиваем l на 1. Переход на шаг 3.

Сложность данного алгоритма $O(k)$, что много меньше сложности алгоритма атаки без знания о максимальности периода, которая равна $O(k2^n)$.

Добавив к ключевому множеству функцию выходов второго автомата f_2, сначала воспользуемся атакой из п.3.3 и найдем f_2, а затем, используя алгоритм выше, найдем начальное состояние первого автомата $x(1)$.

Рассмотрим действие алгоритма на примере. Пусть нам дан генератор G из рисунка 10. Его выходная последовательность равна z = 1011 1010 1001 0011. Зафиксируем $x(1)$ например в 10, $l := 0$. Выходная последовательность полученного генератора будет равна z' = 1110 1110 1010 0100, а последовательность состояний, в которые переходит первый автомат, равна [10, 11, 00, 01]. Сдвинув последовательность z' на 1 элемента влево,

получим z' = 1101 1101 0100 1001, l := 1. Последовательность не совпала, значит, идем дальше. Повторив процедуру еще раз, получим, что выходная последовательность z' = 1011 1010 1001 0011 будет равна заданной, а l := 2. Тогда $x(1)$ будет равен 2-му элементу построенного массива, то есть 00.

В итоге получаем, что больше нет необходимости перебирать все 2^n состояний первого автомата, достаточно выбрать одно состояние и, используя его, найти ключ.

ЗАКЛЮЧЕНИЕ

Предложены способы криптоанализа генератора, когда его ключом является одно из множеств $\{x(1)\}$, $\{y(1)\}$, $\{x(1), y(1)\}$, $\{f_2\}$, $\{f_2, y(1)\}$, $\{f_2, y(1), x(1)\}$. Далее представлена таблица 1 с количествами перебираемых значений для каждого варианта ключа.

Таблица 1 - Сложность алгоритмов атак для рассмотренных вариантов ключа

K	$x(1)$	$y(1)$	$x(1), y(1)$	f_2	$f_2, y(1)$	$f_2, x(1), y(1)$
Сложность атаки	$O(k2^n)$	$O(k2^m)$	$O(k2^{n+m})$	-	-	-

Сложность атаки для $K = \{f_2\}$ сводится к сложности алгоритма доопределения булевой функции. На данный момент для случаев, когда ключом является $\{f_2, y(1)\}$ или $\{f_2, y(1), x(1)\}$, не удалось построить атак со сложностью меньше полного перебора, что говорит о криптографической стойкости генератора.

Написана программа, реализующая двухкаскадный конечно-автоматный криптографический генератор $G = A_1 \cdot A_2$ и находящая период автомата по выходной последовательности, а также по состояниям автомата A_2. Проведены исследования свойств периодичности генератора, выведены и доказаны вспомогательные утверждения.

Предложены способы криптоанализа для ключевых множеств $\{x(1)\}$, $\{y(1)\}$, $\{x(1), f_2\}$, $\{y(1), f_2\}$ в случае, если известно, что рассматриваемый генератор имеет максимальный период. Знание о максимальности периода позволяют сократить сложность атак (таблица 2).

Таблица 2 - Сравнение сложности атак с учетом знаний о максимальности периода генератора и без них

K	$x(1)$	$y(1)$
Сложность атаки	$O(k2^n)$	$O(k2^m)$
Сложность атаки с учетом знаний о максимальности периода генератора	$O(k)$	$O(\lfloor k / 2^n \rfloor)$

ЛИТЕРАТУРА

1. *Агибалов Г. П.*, Конечные автоматы в криптографии // Прикладная дискретная математика. Приложение. – 2009. – № 2. – С. 43–73.
2. *Фомичёв В. М*. Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003. 400 с.
3. *Агибалов Г. П., Панкратова И. А.,* О двухкаскадных конечно-автоматных криптографических генераторах и методах их криптоанализа, Прикладная дискретная математика, 2017, № 35, 38–47.
4. *Боровкова И. В., Панкратова И. А., Семенова Е. В.*, Криптоанализ двухкаскадного конечно-автоматного генератора с функциональным ключом, Прикладная дискретная математика, 2018, № 42, 48–56.

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: petr5542@gmail.com / ID: 10421003
Проверяющий:

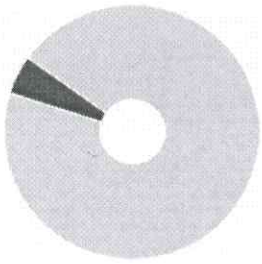
Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 2
Начало загрузки: 18.01.2023 09:52:55
Длительность загрузки: 00:00:01
Имя исходного файла: Дипломная работа Обухов П.К.pdf
Название документа: Дипломная работа Обухов П.К
Размер текста: 47 кБ
Символов в тексте: 47875
Слов в тексте: 6068
Число предложений: 297

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 18.01.2023 06:52:57
Длительность проверки: 00:00:06
Комментарии: не указано
Модули поиска: Интернет Free



СОВПАДЕНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
5,37%	0%	0%	94,63%

Совпадения — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Совпадения, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[01]	3,99%	Скачать полнотекстовую версию http://journals.tsu.ru	19 Июл 2020	Интернет Free	
[02]	0,46%	Cryptanalysis of 2-cascade finite automata generator with functional key.pdf http://journals.tsu.ru	19 Июл 2020	Интернет Free	
[03]	0,57%	Скачать полнотекстовую версию http://journals.tsu.ru	01 Сен 2017	Интернет Free	

Еще источников: 7
Еще совпадений: 0,34%

С результатами ознакомлена.
Руководитель И.А. Панкратова