Санкт-Петербургский государственный университет

Прикладная математика и информатика

Исследование операций и принятие решений в задачах оптимизации, управления и экономики

Соловьева Дарья Георгиевна

Методы криптоанализа комбинирующих генераторов

Выпускная квалификационная работа

Научный руководитель:

к. ф.-м. н., доцент И. В. Агафонова

Рецензент:

к. ф.-м. н., доцент О. М. Дмитриева

Санкт-Петербург

2017

Saint Petersburg State University

Applied Mathematics and Computer Science

Operation Research and Decision Making in Optimization, Control and Economics Problems

Soloveva Daria

Methods of cryptanalysis of combination generators

Graduation Project

Scientific supervisor:

PhD, Associate Professor I. V. Agafonova

Reviewer:

PhD, Associate Professor O. M. Dmitrieva

Saint Petersburg

2017

Содержание

Введение

Криптоанализ — это наука о методах расшифровки зашифрованной информации без предназначенного для такой расшифровки ключа. Также криптоанализом называют взлом шифра.

Большинство существующих на данный момент поточных шифров основаны на регистрах сдвига с линейной обратной связью (РСЛОС). Для этого есть несколько причин:

— хорошие статистические свойства генерируемых последовательностей;

— легкость анализа их поведения с использованием алгебраических методов;

— простота программной реализации.

Но РСЛОС являются линейными устройствами, поэтому легко поддаются криптоанализу. Одним из ведущих способов повышения криптографической стойкости систем, основанных на регистрах сдвига, являются комбинирующие генераторы.

В этой работе будут рассмотрены методы их вскрытия. Основное внимание будет уделено корреляционным атакам, в частности корреляционной атаке Чжаня и Фэня.

Цели данной работы:

— Обзор методов вскрытия комбинирующих генераторов.

— Программная реализация многошаговой быстрой корреляционной атаки Чжаня и Фэня.

— Анализ работы этого алгоритма на специально построенных примерах.

Структурно работа состоит из введения, шести разделов, заключения и приложения.

Первый раздел содержит основные понятия, связанные с темой исследования.

Во втором разделе сформулирована задача криптоанализа комбинирующего генератора, а также конкретная задача, поставленная в этой работе.

Третий раздел посвящен основным методы криптоанализа поточных шифров, особенно корреляционным атакам.

В четвертом разделе подробно описана многошаговая быстрая корреляционная атака, предложенная Чжанем и Фэнем. Алгоритм детально проработан и подготовлен для программной реализации.

Пятый раздел посвящен расчетной части исследования, а именно, результатам, полученным при применении этой атаки. Он содержит выводы, сделанные на основе этих результатов.

Атака Чжаня и Фэня может применяться не ко всем входным данным. В последнем разделе предложена модификация данных, после которой мы можем применять к ним атаку.

В приложении содержатся подробно разобранные примеры применения атаки Чжаня и Фэня.

1 Регистры сдвига, комбинирующие генераторы и связанные с ними понятия

Здесь будет представлен ряд понятий, используемых в этой работе.

Регистр сдвига с линейной обратной связью длины L это устройство, состоящее из L битовых ячеек и *полинома обратной связи*:

$$c(x) = 1 + c_1 x + c_2 x^2 + \dots c_L x^L.$$

Начальное состояние $a_0, a_1, \dots, a_{L-1}$ записывается в ячейки регистра (с первой по L-ую). На Рисунке 1 ячейки пронумерованы справа налево.

На каждом шаге одновременно выполняется несколько действий:

— значение из первой ячейки отправляется на выход РСЛОС;

— содержимое i-ой ячейки записывается в $(i-1)$-ую для $i = 2, \dots, L$;

— в последнюю ячейку записывается $\oplus_{i=1}^{L} c_i a_{n+L-i}$, где n — номер шага (начиная с 0).

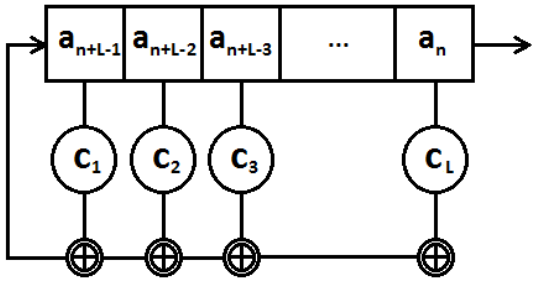

Рис. 1: Регистр сдвига с линейной обратной связью

Тогда выходная двоичная последовательность выглядит так:

$$a_0, a_1, \dots, a_{L-1}, a_L, a_{L+1}, \dots,$$

где $a_j = c_1 a_{j-1} \oplus c_2 a_{j-2} \oplus \dots \oplus c_L a_{j-L}$ для $j \geq L$.

Приведем формулы для вычисления некоторых битов выходной последовательности:

$$a_L = c_1 a_{L-1} \oplus c_2 a_{L-2} \oplus \dots \oplus c_L a_0,$$

$$a_{L+1} = c_1 a_L \oplus c_2 a_{L-1} \oplus \ldots \oplus c_L a_1 = c_1(c_1 a_{L-1} \oplus \ldots \oplus c_L a_0) \oplus c_2 a_{L-1} \oplus \ldots \oplus c_L a_1,$$

$$a_{L+2} = c_1 a_{L+1} \oplus c_2 a_L \oplus \ldots \oplus c_L a_2 =$$

$$= c_1(c_1(c_1 a_{L-1} \oplus \ldots \oplus c_L a_0) \oplus \ldots \oplus c_L a_1) \oplus c_2(c_1 a_{L-1} \oplus \ldots \oplus c_L a_0) \oplus \ldots \oplus c_L a_2.$$

Очевидно, что каждый бит выходной последовательности РСЛОС является линейной комбинацией битов его начального состояния. Пусть параметры $\{g_i^k\}$ — коэффициенты такой линейной комбинации для бита a_i. То есть

$$a_i = g_i^0 a_0 \oplus g_i^1 a_1 \oplus \ldots \oplus g_i^{L-1} a_{L-1} \text{ для } i \geq 0. \tag{1}$$

Для $i = 0, \ldots, L-1$ только один коэффициент не равен нулю: $g_i^i = 1$, а для $i \geq L$ коэффициенты линейной комбинации вычисляются через коэффициенты полинома обратной связи $c_1, \ldots, c_L$.

Обозначим как g_i вектор-столбец $(g_i^0, g_i^1, \ldots, g_i^{L-1})^T$. Теперь можно переписать равенство (1):

$$a_i = (a_0, \ldots, a_{L-1}) \cdot g_i. \tag{2}$$

Рассмотрим часть выходной последовательности длины N: $a_0, \ldots, a_{N-1}$. Ее можно выразить следующим образом:

$$(a_0, \ldots, a_{L-1}, \ldots, a_{N-1}) = (a_0, \ldots, a_{L-1}) \cdot G, \text{ где} \tag{3}$$

$$G = \begin{pmatrix} g_0^0 & g_1^0 & \cdots & g_{N-1}^0 \\ g_0^1 & g_1^1 & \cdots & g_{N-1}^1 \\ \vdots & \vdots & \cdots & \vdots \\ g_0^{L-1} & g_1^{L-1} & \cdots & g_{N-1}^{L-1} \end{pmatrix}. \tag{4}$$

Как уже говорилось выше, первые L столбцов матрицы G являются столбцами единичной матрицы $L \times L$. Остальные столбцы вычисляются через коэффициенты полинома обратной связи $c_1, \ldots, c_L$.

РСЛОС *порождает* последовательность, если она является его выходной последовательностью.

Линейной сложностью последовательности a называется число

$$L = \begin{cases} 0, \text{ если } a = 000\ldots, \\ \infty, \text{ если не существует РСЛОС, порождающего } a, \\ \min\{\text{длины РСЛОС, порождающих } a\}. \end{cases}$$

Пусть $B = \{0, 1\}$. B^n — множество всех двоичных векторов длины n. Произвольное отображение из B^n в B называется *булевой функцией от n переменных*.

Возьмем несколько РСЛОС, работающих параллельно, и будем комбинировать их выходные биты с помощью некоторой булевой функции f, как показано на Рисунке 2. Если мы использовали n регистров, то функция должна быть от n переменных. Мы получили *комбинирующий генератор*.

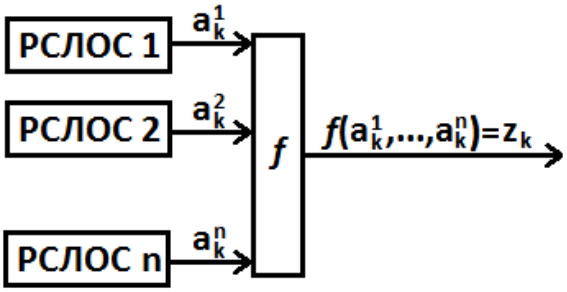

Рис. 2: Комбинирующий генератор

Здесь a_k^i $(i = 1, \ldots, n,\ k \geq 0)$ — бит выходной последовательности i-го регистра сдвига на k-ом такте. Выходная последовательность комбинирующей функции $z = z_0, z_1 \ldots$ называется *гаммой*.

2 Постановка задачи

Задачу криптоанализа комбинирующего генератора поставим так, как ее обычно ставят для аналитических атак. Такая задача предполагает, что аналитику известно описание комбинирующего генератора:

— количество используемых регистров сдвига n,

— их длины L_i, $i = 1, \ldots, n$,

— их полиномы обратной связи

$$c^{(i)}(v) = 1 + c_1^{(i)} v + c_2^{(i)} v^2 + \ldots c_{L_i}^{(i)} v^{L_i},\ i = 1, \ldots, n,$$

— комбинирующая булева функция f.

Найти при этом надо начальное заполнение каждого РСЛОС —

$$a_0^i, a_1^i, \ldots, a_{L-1}^i,\ i = 1, \ldots, n.$$

Мы подробно рассмотрим многошаговую быструю корреляционную атаку, предложенную Чжанем и Фэнем [8], и исследуем работу этого алгоритма на примерах с различными параметрами.

3 Общий обзор методов криптоанализа поточных шифров

Методы криптоанализа поточных шифров принято делить на *силовые* (основанные на принципе полного перебора всех возможных комбинаций ключа), *статистические* (основанные на оценке статистических свойств шифрующей гаммы) и *аналитические* (основанные на аналитических принципах вскрытия криптосхемы). Среди последних основным видом атак являются *корреляционные атаки*, основная идея которых состоит в нахождении корреляции между гаммой и различными линейными комбинациями ключа (регистра сдвига) [3].

Пусть $p_i = P(f(x_1, \ldots, x_n) = x_i)$ — вероятность того, что значение булевой функции f совпадет со значением ее i-ой переменной (так называемая *корреляционная вероятность*).

Если $p_i \neq 1/2$, то говорят, что функция f *коррелирует* со своей i-й переменной.

Если комбинирующая функция допускает корреляцию между выходной последовательностью генератора и выходными последовательностями регистров сдвига, то процесс криптоанализа упрощается. Впервые это отметили Блэйзер и Хайнцманн [4] на рубеже 70-80-х годов, а в 1984 г. Зигенталер развил идею и ввел понятие корреляционно-иммунной функции [7], не дающей возможности для такого упрощения.

Принцип Керкгоффса — это правило разработки криптографических систем, согласно которому засекреченным является только определенный набор параметров алгоритма, называемый ключом, а сам алгоритм шифрования должен быть открытым. То есть при оценке надежности шифрования необходимо предполагать, что противник знает об используемой системе шифрования все, кроме применяемых ключей.

Поэтому все аналитические атаки происходят при допущении, что криптоаналитику известно описание генератора, а его задачей является определение применяемого ключа (начального состояния регистра сдвига).

То есть задача криптоанализа комбинирующего генератора предполагает нахождение начальных заполнений n регистров сдвига с длинами L_i, $i = 1, \ldots, n$. При этом нам известны полиномы обратной связи $c^{(i)}$ и комбинирующая функция f. Возможных начальных состояний $\prod_{i=1}^{n} 2^{L_i}$ штук, и их полный перебор недостижим в большинстве вычислительных систем. Но если есть корреляция между выходной последовательностью генератора

и выходными последовательностями отдельных РСЛОС, то можно определять начальное состояние каждого регистра отдельно.

Пусть $p_i = P(f(a^1, \ldots, a^n) = a^i)$ — вероятность того, что цифра в гамме совпадет с цифрой выходной последовательности i-го РСЛОС. Для каждого i при заданой функции f мы можем вычислить величину p_i, сравнивая столбец значений функции и столбец значений i-ой переменной в таблице истинности функции f.

Чтобы выделить влияние отдельного РСЛОС на гамму z, можно представить остальную часть генератора как двоичный симметричный канал с вероятностью ошибки $(1 - p_i)$, как показано на Рисунке 3.

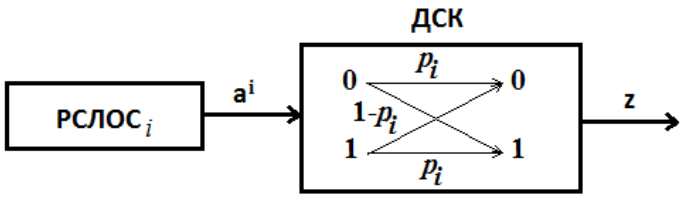

Рис. 3: Модель «двоичный симметричный канал»

Теперь перебрать нужно только $\sum_{i=1}^{n} 2^{L_i}$ начальных состояний. Этот подход называется базовой корреляционной атакой Зигенталера [7]. Однако, несмотря на эффективность метода, количество перебираемых состояний все еще велико, и эту атаку можно применять, только если длины РСЛОС меньше 50.

В 1989 Майер и Штаффельбах [5] положили начало быстрым корреляционным атакам. Они основаны на том, что биты выходной последовательности РСЛОС удовлетворяют некоторым соотношениям, которые называются *уравнениями проверки четности*. Майер и Штаффельбах показали, что если $t < 10$, где t — количество ненулевых коэффициентов в полиноме обратной связи (называемых *отводами*), то можно определять начальное состояние регистра сдвига с помощью итерационного алгоритма со сложностью меньшей, чем при поиске полным перебором.

Вспомним, что регистр сдвига порождается полиномом

$$c(x) = 1 + c_1 x + c_2 x^2 + \ldots c_L x^L,$$

а выходная последовательность $\{a_i\}$ задается следующим соотношением:

$$a_j = c_1 a_{j-1} \oplus c_2 a_{j-2} \oplus \ldots \oplus c_L a_{j-L}, \quad j \geq L.$$

Пусть только коэффициенты с номерами $i_1, \ldots, i_t$ не равны нулю. Тогда можно переписать это соотношение следующим образом:

$$a_j \oplus a_{j-i_1} \oplus \ldots \oplus a_{j-i_t} = 0.$$

Мы получили уравнение проверки четности. Если в этих уравнениях заменить неизвестные биты последовательности $\{a_i\}$ на известные биты гаммы $\{z_i\}$, то часть равенств перестанет выполняться. Алгоритм постепенно заменяет биты z_i, до тех пор, пока все проверки четности не будут выполняться. Когда это происходит, последовательность $\{z_i\}$ становится равной искомой последовательности $\{a_i\}$.

Количество отводов сильно влияет на вычислительную сложность алгоритма, поэтому важно находить низковесные уравнения проверки четности, то есть те, в которых мало ненулевых членов.

Существуют различные модификации исходного алгоритма. В 1996 Пенцхорн [6] разработал метод вычисления низковесных уравнений проверки четности. Он находит проверки четности веса 3 и 4. Этот алгоритм можно использовать в рамках атаки Майера и Штаффельбаха, и его вычислительная сложность не зависит от количества отводов.

Опубликовано множество статей, в которых исследуются различные быстрые корреляционные атаки. В этой работе мы подробно рассмотрим одну из них — многошаговую быструю корреляционную атаку Чжаня и Фэня.

4 Многошаговая быстрая корреляционная атака Чжаня и Фэня

4.1 Основная идея атаки

Нам известно число РСЛОС, которые используются в комбинирующем генераторе, их полиномы обратной связи и комбинирующая булева функция. Найти надо начальные заполнения регистров сдвига. Будем искать начальное состояние каждого РСЛОС отдельно.

Мы разделяем неизвестное нам начальное состояние рассматриваемого РСЛОС на несколько частей:

$$(\underbrace{a_0, \ldots, a_{k^{(1)}-1}}_{k^{(1)}}, \underbrace{a_{k^{(1)}}, \ldots, a_{k^{(1)}+k^{(2)}-1}}_{k^{(2)}}, \underbrace{a_{k^{(1)}+k^{(2)}}, \ldots}_{\ldots}, a_{L-1})$$

и восстанавливаем их по очереди, используя разные проверки четности на разных шагах.

Сначала мы восстанавливаем первые $k^{(1)}$ битов, используя только гамму, потом следующие $k^{(2)}$ битов, используя гамму и уже восстановленные биты начального состояния a_i, и так далее.

Как и в других быстрых корреляционных атаках, сначала мы должны вычислить уравнения проверок четности. Для многошаговой быстрой корреляционной атаки мы будем конструировать уравнения проверки четности следующего вида:

$$a_{i_1} \oplus a_{i_2} \oplus \ldots \oplus a_{i_{t_1}} = \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} a_i \text{ — на первом шаге,}$$

$$a_{j_1} \oplus a_{j_2} \oplus \ldots \oplus a_{j_{t_2}} \oplus \sum_{i=0}^{k^{(1)}-1} x'^{(1)}_i a_i = \sum_{i=k^{(1)}}^{k^{(1)}+k^{(2)}-1} x_i^{(2)} a_i \text{ — на втором шаге,}$$

$$\ldots$$

$$a_{l_1} \oplus a_{l_2} \oplus \ldots \oplus a_{l_{t_m}} \oplus \sum_{i=0}^{\delta-1} y_i a_i = \sum_{i=\delta}^{\delta+k^{(m)}-1} x_i^{(m)} a_i \text{ — на } m\text{-ом шаге,}$$

где $\delta = \sum_{i=1}^{m-1} k^{(i)} (m \geq 2)$.

В статье [8] авторы алгоритма накладывают на параметры t_i ограничения: $t_i \leq t_1 (i = 2, \ldots, m)$.

4.2 Детальное описание атаки и теоретический анализ

Пусть f — это комбинирующая функция от n переменных и величина

$$p_j = P(f(v_1, \ldots, v_n) = v_j) = 0.5 + \varepsilon \ (\varepsilon > 0) \tag{5}$$

— вероятность того, что цифра в гамме совпадет с цифрой выходной последовательности j-го РСЛОС.

Как уже упоминалось, для заданной функции f и каждого номера j можно вычислить p_j. Например, пусть таблица истинности функции f выглядит следующим образом:

v_1	v_2	v_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Сравним столбцы значений первой переменной и функции. Значения совпадают 7 раз из 8. Следовательно, $p_1 = 7/8$.

Аналогично можно вычислить, что $p_2 = 5/8$ и $p_3 = 5/8$.

Мы будем рассматривать один регистр сдвига, поэтому зафиксируем номер j и для краткости будем писать не p_j, а p.

Пусть многочлен c степени L — полином обратной связи рассматриваемого РСЛОС, $(a_0, \ldots, a_{L-1})$ — его начальное состояние, тогда по формулам (1)-(4) биты его выходной последовательности можно выразить следующим образом:

$$a_i = (a_0, \ldots, a_{L-1}) \cdot g_i,$$

где g_i — столбцы матрицы G, чьи элементы вычисляются через коэффициенты полинома c. Вспомним, что первые L столбцов матрицы G являются столбцами единичной матрицы $L \times L$.

4.2.1 Первый шаг

Уравнения проверки четности

Рассмотрим $(g_{i_1}, \ldots, g_{i_{t_1}})$ — набор из t_1 столбцов матрицы G. Таких наборов $\binom{N}{t_1}$ штук. Сложим по модулю 2 векторы в каждом наборе. Для построения уравнений проверки четности нам надо, чтобы часть наборов удовлетворяла следующему равенству:

$$g_{i_1} \oplus \ldots \oplus g_{i_{t_1}} = (x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)}, \underbrace{0, \ldots, 0}_{L-k^{(1)}})^T, \tag{6}$$

где $(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)})$ — некий вектор, полученный при суммировании, разный для разных наборов $(g_{i_1}, \ldots, g_{i_{t_1}})$.

Так как первые столбцы матрицы G — это столбцы единичной матрицы $L \times L$, то мы можем ограничить t_1 и $k^{(1)}$ так, чтобы у нас обязательно были наборы, удовлетворяющие (6). Столбец матрицы $(0, \ldots, 0, 1, \underbrace{0, \ldots, 0}_{L-k^{(1)}})^T$ имеет номер $k^{(1)}$. Если из столбцов с номерами $i \leq k^{(1)}$ выбрать t_1 произвольных столбцов, то такой набор будет удовлетворять равенству (6). Следовательно, для того, чтобы построить уравнения проверки четности, достаточным условием является выполнение неравенства: $t_1 \leq k^{(1)}$.

Умножив равенство (6) слева на $(a_0, \ldots, a_{L-1})$, получим уравнение проверки четности:

$$a_{i_1} \oplus \ldots \oplus a_{i_{t_1}} = \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} a_i. \tag{7}$$

Наборов, удовлетворяющих равенству (6), может быть несколько, и каждому из них соответствует свое уравнение проверки четности и свой вектор $(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)})$. Но может быть и так, что разным наборам соответствуют одинаковые векторы.

Предполагаемое начальное состояние

Перепишем (7) как

$$z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} = \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} a_i \oplus \sum_{j=1}^{t_1} e_{i_j}, \tag{8}$$

где z_j — биты гаммы, а $e_j = a_j \oplus z_j$ $(j = i_1, \ldots, i_{t_1})$ — случайный шум со следующим распределением: $P(e_j = 0) = 0.5 + \varepsilon$ и $P(e_j = 1) = 0.5 - \varepsilon$.

Для краткости будем называть восстанавливаемую на данном шаге часть начального состояния просто начальным состоянием. Мы будем перебирать все возможные начальные состояния и искать среди них правильное.

Предположим, что начальное состояние — это вектор $(a'_0, \ldots, a'_{k^{(1)}-1})$. Перепишем равенство (8) следующим образом:

$$z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} a'_i = \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} (a_i \oplus a'_i) \oplus \sum_{j=1}^{t_1} e_{i_j}. \tag{9}$$

Введем обозначения

$$\Delta(i_1, \ldots, i_{t_1}) = \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} (a_i \oplus a'_i) \oplus \sum_{j=1}^{t_1} e_{i_j}, \tag{10}$$

$$q^{(1)} = P\left(\sum_{j=1}^{t_1} e_{i_j} = 0\right) = 0.5 + 2^{t_1 - 1} \varepsilon^{t_1}. \tag{11}$$

$\Omega^{(1)}$ — количество наборов $(g_{i_1}, \ldots, g_{i_{t_1}})$, удовлетворяющих (6), то есть количество уравнений проверки четности. Чжань и Фэнь [8] приводят формулу для вычисления этой величины: $\Omega^{(1)} = \binom{N}{t_1} / 2^{L - k^{(1)}}$.

В статье [8] указано, что если начальное состояние угадано правильно, то величина $\sum_{i=1}^{\Omega^{(1)}} (\Delta(i_1, \ldots, i_{t_1}) \oplus 1)$ имеет биномиальное распределение с параметрами $(\Omega^{(1)}, q^1)$, а если неправильно, то с параметрами $(\Omega^{(1)}, 1/2)$. Но применять такой критерий на практике слишком сложно, поэтому надо найти другой.

Критерий правильности начального состояния

Для фиксированного вектора $(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)})$ определим вспомогательную функцию

$$h(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)}) = \sum_{(i_1, \ldots, i_{t_1}) \in (6)} (-1)^{z_{i_1} \oplus \ldots \oplus z_{i_{t_1}}}$$

(суммируем по наборам $(i_1, \ldots, i_{t_1})$ таким, что выполняется (6)).

Если вектор $(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)})$ не соответствует ни одному из $\Omega^{(1)}$ уравнений проверки четности, то $h(x_0^{(1)}, x_1^{(1)}, \ldots, x_{k^{(1)}-1}^{(1)}) = 0$.

Тем самым мы определили функцию $h : GF(2)^{k^{(1)}} \to \mathbb{R}$.

Пусть векторы $u, x \in GF(2)^{k^{(1)}}$, тогда

$$u \cdot x = \sum_{i=0}^{k^{(1)}-1} u_i x_i.$$

Рассмотрим *преобразование Уолша-Адамара первого рода* для функции h:

$$H(u) = \sum_{x \in GF(2)^{k^{(1)}}} h(x)(-1)^{u \cdot x} = \sum_{\Omega^{(1)}} (-1)^{z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} u_i} = \Omega_0^{(1)} - \Omega_1^{(1)}.$$

Здесь

$$\Omega_0^{(1)} = |\{(i_1, \ldots, i_{t_1}) \in (6) : \ z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} u_i = 0\}|,$$

$$\Omega_1^{(1)} = |\{(i_1, \ldots, i_{t_1}) \in (6) : \ z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} u_i = 1\}|,$$

$$\Omega_0^{(1)} + \Omega_1^{(1)} = \Omega^{(1)}.$$

Пусть вектор u равен $(a'_0, \ldots, a'_{k^{(1)}-1})$, тогда по формулам (9) и (10) можно получить следующее:

$$\sum_{i=1}^{\Omega^{(1)}} (\Delta(i_1, \ldots, i_{t_1}) \oplus 1) = \sum_{i=1}^{\Omega^{(1)}} \left(z_{i_1} \oplus \ldots \oplus z_{i_{t_1}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i^{(1)} a'_i \oplus 1 \right) = \Omega_0^{(1)}.$$

Заметим, что

$$\frac{H(u) + \Omega^{(1)}}{2} = \frac{\Omega_0^{(1)} - \Omega_1^{(1)} + \Omega_0^{(1)} + \Omega_1^{(1)}}{2} = \Omega_0^{(1)}.$$

Таким образом, мы получили равенство:

$$\sum_{i=1}^{\Omega^{(1)}} (\Delta(i_1, \ldots, i_{t_1}) \oplus 1) = \frac{H(u) + \Omega^{(1)}}{2}.$$

Следовательно, чтобы определить, является ли начальное состояние $(a'_0, \ldots, a'_{k^{(1)}-1})$ правильным, надо вычислить значение H в этой точке. Значит, нам надо вычислить значение H для каждого возможного начального состояния. Мы можем эффективно сделать это, используя *быстрое преобразование Уолша-Адамара* (FWT). Этот алгоритм подробно изложен в шестой главе книги *Ортогональные преобразования при обработке цифровых сигналов*[1].

Мы будем использовать пороговое значение $T^{(1)}$ как границу для принятия решения, то есть, если

$$\frac{H((a'_0, \ldots, a'_{k^{(1)}-1})) + \Omega^{(1)}}{2} \geq T^{(1)}, \tag{12}$$

то мы считаем, что правильно угадали начальное состояние $(a'_0, \ldots, a'_{k^{(1)}-1})$. Ниже рассказано, как именно мы определяем значение порога $T^{(1)}$.

Пороговое значение

Вероятность того, что правильное начальное состояние пройдет тест (12), равна

$$P_1^{(1)} = \sum_{i=T^{(1)}}^{\Omega^{(1)}} \binom{\Omega^{(1)}}{i} \left(q^{(1)}\right)^i \left(1-q^{(1)}\right)^{\Omega^{(1)}-i},$$

(здесь $q^{(1)}$ вычисляется по формуле (11)), а вероятность того, что неправильное начальное состояние пройдет его, равна

$$P_2^{(1)} = \sum_{i=T^{(1)}}^{\Omega^{(1)}} \binom{\Omega^{(1)}}{i} \left(\frac{1}{2}\right)^{\Omega^{(1)}}.$$

Очевидно, что мы получим лучшие результаты, если вероятность $P_1^{(1)}$ будет близка к 1, а вероятность $P_2^{(1)}$, наоборот, очень мала. Рассмотрим четыре случая, предложенные Чжанем и Фэнем:

1. $P_1^{(1)} \geq 0.99$ и $P_2^{(1)} < 2^{-k^{(1)}}$. Правильное начальное состояние пройдет тест почти наверняка, и ни одно неправильное состояние его не пройдет.

2. $P_1^{(1)} \geq 0.99$ и $P_2^{(1)} \approx 2^{-k_1^{(1)}}$, где $0 < k_1^{(1)} < k^{(1)}$. Правильное начальное состояние пройдет тест с высокой вероятностью, но несколько неправильных тоже его пройдут, то есть у нас будет список начальных состояний, которые могут быть правильными (состояний-кандидатов). Если $k_1^{(1)}$ больше 1, то список состояний-кандидатов, скорее всего, будет коротким. Если же $k_1^{(1)}$ близко к 0, то список может быть слишком длинным, и тогда желательно изменить параметры.

3. $P_1^{(1)} < 0.99$ и $P_2^{(1)} < 2^{-k^{(1)}}$. Ни одно неправильное состояние тест не пройдет, а правильное может быть пройдет, а может быть — нет. В этом случае мы должны поменять значения параметров, чтобы достичь успеха.

4. $P_1^{(1)} < 0.99$ и $P_2^{(1)} \approx 2^{-k_1^{(1)}}$, где $0 < k_1^{(1)} < k^{(1)}$. Мы получим список состояний-кандидатов, но среди них может не быть правильного, поэтому желательно поменять параметры.

Следовательно, нужно подбирать $T^{(1)}$ так, чтобы выполнялся первый или второй случай.

4.2.2 Второй и последующие шаги

После того как мы закончили первый шаг, переходим ко второму, на котором мы можем определить следующие $k^{(2)}$ битов начального состояния, используя гамму и уже восстановленную часть. Аналогично первому шагу мы перебираем все возможные наборы из t_2 столбцов матрицы G и находим те, для которых выполняются равенства следующего вида:

$$g_{j_1} \oplus \ldots \oplus g_{j_{t_2}} = (x_0'^{(1)}, \ldots, x_{k^{(1)}-1}'^{(1)}, x_{k^{(1)}}^{(2)}, \ldots, x_{k^{(1)}+k^{(2)}-1}^{(2)}, \underbrace{0, \ldots, 0}_{L-k^{(1)}-k^{(2)}})^T, \tag{13}$$

где $(x_0'^{(1)}, \ldots, x_{k^{(1)}-1}'^{(1)}, x_{k^{(1)}}^{(2)}, \ldots, x_{k^{(1)}+k^{(2)}-1}^{(2)})$ — вектор, полученный при суммировании.

Для того чтобы у нас обязательно были наборы, удовлетворяющие (13), нужно, чтобы выполнялось неравенство: $t_2 \leq k^{(1)} + k^{(2)}$.

Уравнения проверки четности на этом шаге будут выглядеть следующим образом:

$$a_{j_1} \oplus \ldots \oplus a_{j_{t_2}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i'^{(1)} a_i = \sum_{i=k^{(1)}}^{k^{(1)}+k^{(2)}-1} x_i^{(2)} a_i', \tag{14}$$

где $(a_{k^{(1)}}', \ldots, a_{k^{(1)}+k^{(2)}-1}')$ — предполагаемое начальное состояние.

Пусть всего будет $\Omega^{(2)}$ уравнений. Преобразуем их и получим $\Omega^{(2)}$ равенств следующего вида:

$$\Delta(j_1, \ldots, j_{t_2}) = z_{j_1} \oplus \ldots \oplus z_{j_{t_2}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i'^{(1)} a_i \oplus \sum_{i=k^{(1)}}^{k^{(1)}+k^{(2)}-1} x_i^{(2)} a_i' =$$

$$= \sum_{i=k^{(1)}}^{k^{(1)}+k^{(2)}-1} x_i^{(2)} (a_i \oplus a_i') \oplus \sum_{j=1}^{t_2} e_{i_j}.$$

Как и на первом шаге, определим вспомогательную функцию

$$h^{(2)}(x_{k^{(1)}}^{(2)}, \ldots, x_{k^{(1)}+k^{(2)}-1}^{(2)}) = \sum_{(j_1,\ldots,j_{t_2}) \in (13)} (-1)^{z_{j_1} \oplus \ldots \oplus z_{j_{t_2}} \oplus \sum_{i=0}^{k^{(1)}-1} x_i'^{(1)} a_i}.$$

Если вектор $(x_{k^{(1)}}^{(2)}, \ldots, x_{k^{(1)}+k^{(2)}-1}^{(2)})$ не соответствует ни одному из уравнений проверки четности, то $h^{(2)} = 0$ в этой точке.

Используем быстрое преобразование Уолша-Адамара, чтобы вычислить значение

$$\sum_{i=1}^{\Omega^{(2)}} (\Delta(j_1, \ldots, j_{t_2}) \oplus 1) = \frac{H^{(2)}(u) + \Omega^{(2)}}{2},$$

где $H^{(2)}$ — это преобразование Уолша-Адамара для функции $h^{(2)}$ и вектор u равен $(a'_{k^{(1)}}, \ldots, a'_{k^{(1)}+k^{(2)}-1})$. Как и прежде, введем пороговое значение $T^{(2)}$ для принятия решения о том, правильно ли угадано начальное состояние. Подбор этого параметра выполняется так же, как и на первом шаге (с заменой всех верхних индексов 1 на 2).

Теперь у нас есть $k^{(1)}+k^{(2)}$ битов начального состояния. На следующих шагах действуем аналогично, пока у нас не закончатся неизвестные биты начального состояния, или пока их не останется настолько мало, что их будет быстрее найти полным перебором.

4.3 Схема атаки

Приведем краткую схему атаки на один РСЛОС длины L.

Параметры:

— m — количество частей, на которые разбиваем начальное состояние,

— $k^{(i)}$, $i = 1, \ldots, m$ — количество битов начального состояния в i-ой части,

— t_i, $i = 1, \ldots, m$ — количество битов гаммы, участвующих в уравнениях проверки четности на i-ом шаге,

— $T^{(i)}$, $i = 1, \ldots, m$ — пороговое значение на i-ом шаге.

Input: фрагмент гаммы $\{z_i\}_{i=0}^{N-1}$, полином обратной связи $c(x)$, комбинирующая функция f.

0. Вычислить корреляционную вероятность p, используя функцию f.

1. **for** $i = 1, \ldots, m$ **do**

   Определить

$$h^{(i)}(x_{\delta_i}^{(i)}, \ldots, x_{\delta_i + k^{(i)} - 1}^{(i)}) = \sum (-1)^{z_{j_1} \oplus \ldots \oplus z_{j_{t_i}} \oplus \sum_{j=0}^{\delta_i - 1} y'_j a_j},$$

   где $\delta_i = \sum_{j=1}^{i-1} k^{(j)}$, $k^{(0)} = 1$, для $i = 1$ положим $y'_0 = 0$, а для $i > 1$ вектор $(y'_0, \ldots, y'_{\delta_i - 1})$ равен вектору $(x_0'^{(1)}, \ldots, x_{k^{(1)}-1}'^{(1)}, \ldots, x_{\delta_{i-1}-1}'^{(i-1)}, \ldots, x_{\delta_i - 1}'^{(i-1)})$.

   Применить FWT для вычисления величины $\sum_{j=1}^{\Omega^{(i)}}(\Delta \oplus 1)$ для всех $2^{k^{(i)}}$ возможных начальных состояний

   if $(\sum_{j=1}^{\Omega^{(i)}}(\Delta \oplus 1) \geq T^{(i)})$ **then**

   {считать соответствующее начальное состояние правильным}

   else {отклонить его}

   end if

   if (не найдено ни одного подходящего начального состояния) **then**

   {прекратить выполнение цикла и считать параметры неподходящими для правильной работы алгоритма}

end if

end for

2. **if** $(\sum_{i=1}^{m} k^{(i)} < L)$ **then**

   {использовать полный перебор для поиска все еще неизвестных битов начального состояния}

   end if

Output: Начальное состояние-кандидат $(a_0, \ldots, a_{L-1})$, короткий список состояний-кандидатов или вывод о непригодности параметров.

5 Реализация атаки и анализ результатов

В рамках работы была программно реализована многошаговая быстрая корреляционная атака Чжаня и Фэня, подробно описанная в четвертом разделе. Для тестирования программы-алгоритма был промоделирован комбинирующий генератор. Листинги этих программ находятся по адресу https://yadi.sk/d/pstEP6aB3JRUdY/2017. Обе программы были написаны на языке C#.

Атака Чжаня и Фэня была применена мной к примерам с различными параметрами. На основе полученных результатов были сделаны некоторые выводы, которые приведены далее. Характерные результаты приведены в приложении.

Авторы метода утверждают, что их атака работает хорошо, даже если корреляционная вероятность близка к 1/2, но полученные результаты этого не подтверждают.

Пусть корреляционная вероятность $p = 0.5 + \varepsilon$. Вспомним формулу для вычисления значения $q^{(i)}$:

$$q^{(i)} = 0.5 + 2^{t_i - 1}\varepsilon^{t_i}.$$

Учитывая, что $0 < \varepsilon < 0.5$, можно записать это так: $\varepsilon = 2^{-d}$, где $d > 1$. Тогда

$$q^{(i)} = 0.5 + 2^{t_i - 1 - dt_i} = 0.5 + 2^{t_i(1-d)-1}.$$

Вероятности того, что правильное начальное состояние пройдет тест, описанный на с. 17, и того, что неправильное состояние пройдет его, вычисляются по формулам, которые мало отличаются друг от друга:

$$P_1^{(i)} = \sum_{j=T^{(i)}}^{\Omega^{(i)}} \binom{\Omega^{(1)}}{i} \left(q^{(i)}\right)^j \left(1 - q^{(i)}\right)^{\Omega^{(1)} - j},$$

$$P_2^{(i)} = \sum_{j=T^{(i)}}^{\Omega^{(i)}} \binom{\Omega^{(1)}}{i} \left(\frac{1}{2}\right)^{\Omega^{(1)}}.$$

Если $q^{(i)}$ близко к 0.5, то значения этих вероятностей тоже близки. А нам надо, чтобы одна из них была близка к 1, а другая — к 0. На рассмотренном мной диапазоне длин регистров сдвига (от 15 до 23) если $d \geq 2$, то значение t_i должно быть не больше 2, что плохо сказывается на работе алгоритма. Но даже если $d < 2$, нам все еще стоит ввести ограничение на t_i. Для рассмотренного диапазона длин оно выглядит так: $t_i \leq 5$.

Лучше всего атака работает для $q^{(i)}$ далекого от 0.5, так что значение ε должно быть от 0.25 до 0.5.

Как уже упоминалось, Чжань и Фэнь вводят дополнительное ограничение на параметры:

$$t_i \leq t_1 \ (i = 2, \ldots, m). \tag{15}$$

Они не обосновывают этот шаг теоретически, поэтому, видимо, ограничение было выведено экспериментально. Проведенные эксперименты подтверждают, что лучшие результаты действительно достигались на таких параметрах.

Наряду с ограничениями (15) можно предложить и другие ограничения:

$$t_i \leq \sum_{j=1}^{j=i} k^{(j)} \ (i = 1, \ldots, m), \tag{16}$$

выводящиеся аналогично тем, которые были обоснованы нами на с. 15, 19. Но надо отметить, что иногда лучший результат можно было получить, нарушив такое ограничение на первом шаге, как показано в примере 1 в приложении. В этом примере в четвертом случае не было выполнено ограничение $t_1 \leq k^{(1)}$ и можно было подобрать пороговое значение так, чтобы список состояний-кандидатов на первом шаге был покороче.

Очевидно также, что если соблюдать ограничения вида (15) и ограничение $t_1 \leq k^{(1)}$, то ограничения вида (16) выполняются автоматически.

Чжань и Фэнь особо отмечают, что этап вычисления уравнений проверки четности является несложным и быстро выполняется. Но на рассмотренных примерах это не подтвердилось. Элементы матрицы G вычисляются по сложным формулам, и выполнение этой части алгоритма занимает значительную часть времени работы программы. Далее нам нужно перебирать все возможные наборы столбцов этой матрицы, складывать столбцы в каждом наборе и проверять равенства вида (6) и (13). С увеличением длины регистра сдвига эта часть начинает преобладать по времени.

При этом следующий этап, на котором мы вычисляем значения функции h и ее преобразование Уолша-Адамара, требует меньшего времени счета, чем предыдущий.

6 Модификация входных данных

Атака Чжаня и Фэня может быть использована для вскрытия только тех регистров сдвига, у которых корреляционная вероятность выше 0.5. Но мы можем модифицировать входные данные так, чтобы использовать атаку и для вскрытия РСЛОС с корреляционной вероятностью меньше 0.5.

Пусть нам дана функция f, и корреляционная вероятность ее i-ой переменной $p_i = r < 0.5$. В таблице истинности f заменим все 0 на 1 и все 1 на 0. Обозначим полученную функцию $\overline{f}$. Сравним столбцы значений i-ой переменной и функции $\overline{f}$. В тех строках, где раньше значения совпадали, теперь они не совпадают, и наоборот. Тогда корреляционная вероятность p_i стала равна $1 - r > 0.5$.

Мы знаем фрагмент гаммы z, которая получилась при шифровании со старой комбинирующей функцией f. Если бы при шифровании использовалась функция $\overline{f}$, то получилась бы гамма $\overline{z}$. Каждый ее бит является инверсией соответствующего бита z (то есть все 1 заменены на 0, а 0 на 1).

Таким образом, если нам надо вскрыть регистр сдвига с корреляционной вероятностью $p_i = r < 0.5$, мы должны инверсией из гаммы z получить гамму $\overline{z}$ и считать, что корреляционная вероятность регистра сдвига равна $1 - r$. К таким входным данным мы уже можем применить атаку Чжаня и Фэня, как показано в примере 2 в приложении.

Заключение

В ходе данного исследования были решены следующие задачи:

1. Составлен общий обзор методов криптоанализа комбинирующих генераторов.
2. Подробно рассмотрена многошаговая быстрая корреляционная атака Чжаня и Фэня.
3. Промоделирован комбинирующий генератор с целью тестирования программы-алгоритма.
4. Реализована и протестирована программа-алгоритм.
5. Проведена серия расчетов на различных примерах.
6. Проанализированы результаты с точки зрения влияния параметров на работоспособность алгоритма.
7. Предложена модификация входных данных для случая, при котором корреляционная вероятность меньше 0.5.

Обе программы написаны на языке C#. Их листинги находятся по адресу https://yadi.sk/d/pstEP6aB3JRUdY/2017.

Отметим, что многошаговая быстрая корреляционная атака Чжаня и Фэня имеет достаточно много ограничений на параметры и требует их тщательного подбора. Ее программная реализация не слишком сложна, но работа над каждым примером занимает много времени за счет подбора параметров, который приходится осуществлять вручную.

Время работы алгоритма сильно зависит от длин используемых РСЛОС. В связи с работой на обычном компьютере модельные примеры проводились для регистров сдвига длины не более чем 23.

Изначально в рамках выполнения работы предполагалось подробно рассмотреть несколько методов криптоанализа комбинирующих генераторов. Но, в связи с объемностью работы над атакой Чжаня и Фэня, был изучен только один метод.

Достоинства и недостатки многошаговой корреляционной атаки могли бы больше выразиться, если бы мы рассмотрели другие атаки и сравнили их с этой.

Список литературы

[1] Ахмед, Н. Ортогональные преобразования при обработке цифровых сигналов / Н. Ахмед, К. Р. Рао ; пер. с англ. под ред. И. Б. Фоменко. — М.: Связь, 1980. — 248 с., ил.

[2] Поточные шифры. Результаты зарубежной открытой криптологии [Электронный ресурс]. — Режим доступа: https://kiwiarxiv.files.wordpress.com/2016/02/potochnye_shifry_stream_ciphers_ru_1997.pdf, свободный (дата обращения: 25.04.2017).

[3] Стасев, Ю. В. Исследование методов криптоанализа поточных шифров [Электронный ресурс] / Ю. В. Стасев, А. В. Потий, Ю. А. Избенко - . — Режим доступа: http://www.nrjetix.com/fileadmin/doc/publications/articles/stasev_potiy_izbenko_ru.pdf, свободный (дата обращения: 10.05.2017).

[4] Blaser, W. New cryptographic device with high security using public key distribution / W. Blaser, P. Heinzmann // IEEE Student Paper Contest. — 1979-80. — P.145-153.

[5] Meier, W. Fast correlation attacks on certain stream ciphers / W. Meier, O. Staffelbach // Journal of Cryptology. — 1989. — Vol. 1, No. 3. — P. 159-176.

[6] Penzhorn, W. Correlation Attacks on Stream Ciphers: Computing Low-Weight Parity Checks Based on Error-Correcting Codes / W. T. Penzhorn // Fast Software Encryption — Third International Workshop, Cambridge. — 1996. — P. 159-172.

[7] Siegenthaler, T. Correlation-immunity of nonlinear combining functions for cryptographic applications / T. Siegenthaler // IEEE Trans. Inform. Theory. — 1984. — Vol. IT-30. — P. 776-780.

[8] Zhang, B. Multi-pass fast correlation attack on stream ciphers / B. Zhang, D. Feng // Biham E., Youssef A. M. (eds.) SAC 2006. Lecture Notes in Computer Science. — 2006. — Vol. 4356. — P. 234-248.

Приложение

Примеры применения атаки Чжаня и Фэня

Напомним некоторые обозначения, используемые в работе:

— m — количество частей, на которые мы разбиваем начальное состояние,

— $k^{(i)}$ — количество битов начального состояния в i-ой части,

— t_i — количество битов гаммы, участвующих в уравнениях проверки четности на i-ом шаге,

— $T^{(i)}$ — пороговое значение на i-ом шаге,

— $\Omega^{(i)}$ — количество уравнений проверки четности,

— $P_1^{(i)}$ — вероятность того, что правильное начальное состояние пройдет тест,

— $P_2^{(i)}$ — вероятность того, что неправильное начальное состояние пройдет тест.

Пример 1

Входные данные:

— Гамма z равна последовательности 1010101010010001000011000000101100101000 длины $N = 40$.

— Комбинирующая функция f от трех переменных имеет вектор значений $(0, 0, 0, 0, 0, 1, 1, 1)$. Есть три регистра сдвига с длинами 15, 10 и 19. Рассматриваем первый. Его корреляционная вероятность $p = 0.875$.

— Вектор коэффициентов полинома обратной связи $(c_1, \ldots, c_L)$ равен вектору $(1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1)$ $(L = 15)$.

Разбиваем начальное состояние на $m = 4$ частей. Рассматриваем четыре разных набора значений остальных параметров (t_i, $k^{(i)}$, $T^{(i)}$) и вычисляемых по ним величин ($\Omega^{(i)}$, $P_1^{(i)}$, $P_2^{(i)}$):

1).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	4	4	25	46	0.9615	0.3293
2	4	5	1000	1429	0.9891	≈ 0
3	3	3	800	1236	0.9756	≈ 0
4	2	3	500	781	0.9999	≈ 0

2).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	4	4	25	46	0.9615	0.3293
2	4	5	1200	1429	0.9543	10^{-6}
3	3	3	800	—	—	—
4	2	3	500	—	—	—

3).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	4	4	25	46	0.9615	0.3293
2	4	5	1000	1429	0.9891	≈ 0
3	3	3	800	1236	0.9756	≈ 0
4	2	3	600	781	0.8221	≈ 0

4).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	5	4	200	322	0.4883	10^{-5}
2	4	5	1000	1429	0.9891	≈ 0
3	3	3	800	1236	0.9756	≈ 0
4	2	3	600	781	0.8221	≈ 0

Результаты

В таблицах указаны списки состояний-кандидатов (биты от 0 до $k^{(1)} + \ldots + k^{(i)} - 1$), полученные на i-ом шаге. Таблица номер j соответствует j-му набору параметров.

1).

i	$a_0 \ldots a_{k^{(1)}+\ldots+k^{(i)}-1}$
1	0101 1010
2	010101010 101010101
3	101010101001
4	1010101010010 01 1010101010011 00

2).

i	$a_0 \ldots a_{k^{(1)}+\ldots+k^{(i)}-1}$
1	0101 1010
2	Не найдено ни одного начального состояния (так как пороговое значение $T^{(2)} = 1200$ слишком велико)

3).

i	$a_0 \ldots a_{k^{(1)}+\ldots+k^{(i)}-1}$
1	0101 1010
2	010101010 101010101
3	101010101001
4	1010101010010 01

4).

i	$a_0 \ldots a_{k^{(1)}+\ldots+k^{(i)}-1}$
1	1010
2	101010101
3	101010101001
4	1010101010010 01

Правильное начальное состояние $a = 101010101001001$, как и получено в третьем и четвертом случаях. В первом случае пороговое значение на четвертом шаге мало, поэто-

му получен список состояний-кандидатов. А во втором случае на втором шаге пороговое значение слишком велико, поэтому нет ни одного состояния-кандидата.

В первых трех случаях соблюдены ограничения (15) и (16):

$$t_i \le t_1, \quad t_i \le \sum_{j=1}^{j=i} k^{(j)} \quad (i = 1, \ldots, m).$$

В последнем случае нарушено одно из этих ограничений: $t_1 > k^{(1)}$. Это дает возможность подобрать пороговое значение такое, чтобы получить только одно состояние-кандидат на первом шаге.

Пример 2

Входные данные:

— Гамма z равна последовательности 0100011010011111011101011011111010011101 длины $N = 40$.

— Комбинирующая функция f от трех переменных имеет вектор значений $(1, 1, 1, 1, 0, 0, 0)$. Есть три регистра сдвига с длинами 20, 15 и 24. Рассматриваем первый. Его корреляционная вероятность $p = 0.125$.

— Вектор коэффициентов полинома обратной связи $(c_1, \ldots, c_L)$ равен вектору $(1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1)$ $(L = 20)$.

Так как $p < 0.5$, сначала инверсией битов получим из гаммы z гамму $\overline{z}$, равную 10111001011000001000101001000000101100010. Теперь можно применить атаку.

Рассматриваем два разных набора значений параметров $(t_i,\ k^{(i)},\ T^{(i)})$ и вычисляемых по ним величин $(\Omega^{(i)},\ P_1^{(i)},\ P_2^{(i)})$:

1).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	5	7	50	81	0.5590	0.0224
2	4	5	200	358	0.9999	0.0150
3	3	4	500	619	0.9002	≈ 0
4	2	4	600	781	0.5023	≈ 0

2).

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	5	10	400	644	0.4661	10^{-10}
2	3	6	450	619	0.8202	≈ 0
3	2	4	650	781	0.5023	≈ 0

Результаты

В таблицах ниже указаны списки состояний-кандидатов (биты от 0 до $k^{(1)} + \ldots + k^{(i)} - 1$), полученные на i-ом шаге. Таблица номер j соответствует j-му набору параметров.

Правильное начальное состояние $a = 10111001011010001001$, как и получено во втором случае. В первом случае пороговое значение на последнем шаге мало, поэтому получен список состояний-кандидатов.

1).

i	$a_0 \dots a_{k^{(1)}+\dots+k^{(i)}-1}$
1	1011100
2	101110000100
	101110000110
	101110000111
	101110010010
	101110010100
	101110010110
	101110010111
	101110011110
3	1011100101100000
	1011100101100010
	1011100101101000
4	10111001011000101000
	10111001011010001001

2).

i	$a_0 \dots a_{k^{(1)}+\dots+k^{(i)}-1}$
1	1011100001
	1011100101
2	1011100001100000
	1011100001100010
	1011100101100000
	1011100101100001
	1011100101100010
	1011100101100100
	1011100101101000
	1011100101110000
3	10111001011010001001

Пример 3

Гамма z равна последовательности 00000110010001010001101010000110001001 00 длины $N = 40$. Комбинирующая функция f от четырех переменных имеет вектор значений $(0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0)$. Есть три регистра сдвига с длинами 8, 8, 9 и 10.

Пробуем расшифровать все регистры сдвига по очереди.

Первый регистр

Его корреляционная вероятность $p = 0.3125$. Вектор коэффициентов полинома обратной связи $(c_1, \ldots, c_L)$ равен вектору $(1, 0, 0, 1, 1, 0, 0, 1)$.

Пусть $m = 2$, остальные параметры и вычисляемые по ним величины записаны в таблице:

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	3	5	660	1236	0.6530	10^{-10}
2	3	3	5080	9881	0.8799	10^{-13}

В таблице указаны состояния-кандидаты (биты от 0 до $k^{(1)} + \ldots + k^{(i)} - 1$), полученные на i-ом шаге:

i	$a_0 \ldots a_{k^{(1)}+\ldots+k^{(i)}-1}$
1	10011
2	10011101

Получено правильное начальное состояние $a = 100111011$.

Второй регистр

Его корреляционная вероятность $p = 0.6875$. Вектор коэффициентов полинома обратной связи $(c_1, \ldots, c_L) = (0, 1, 0, 1, 1, 0, 0, 1)$.

Возьмем $m = 2$. Остальные параметры и вычисляемые по ним величины записаны в таблице:

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	3	5	649	1236	0.8452	10^{-9}
2	3	3	5031	9881	0.9623	10^{-12}

В таблице указаны списки состояний-кандидатов полученные на i-ом шаге:

i	$a_0 \dots a_{k^{(1)}+\dots+k^{(i)}-1}$
1	10011
	10100
2	10011111
	10100111

Получен список состояний-кандидатов, среди них есть правильное начальное состояние $a = 10011111$.

Третий регистр

Его корреляционная вероятность $p = 0.6875$. Вектор коэффициентов полинома обратной связи $(c_1, \dots, c_L) = (1, 0, 1, 0, 1, 1, 0, 0, 1)$.

Для вскрытия этого регистра понадобилась только половина гаммы: 00000110010001010001.

Пусть $m = 3$. Остальные параметры и вычисляемые по ним величины записаны в таблице:

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	3	4	25	37	0.4783	0.0235
2	2	3	28	9881	0.5537	0.1958
3	2	2	91	9881	0.9963	0.7652

В таблице указаны списки состояний-кандидатов полученные на i-ом шаге:

i	$a_0 \dots a_{k^{(1)}+\dots+k^{(i)}-1}$
1	1001
2	1001111
3	100111100
	100111110
	100111111

Получен список состояний-кандидатов, среди них есть правильное начальное состояние $a = 100111110$.

Четвертый регистр

Его корреляционная вероятность $p = 0.3125$. Вектор коэффициентов полинома обратной связи $(c_1, \dots, c_L) = (1, 0, 1, 0, 1, 1, 0, 1, 0, 0)$.

Возьмем $m = 2$. Остальные параметры и вычисляемые по ним величины записаны в таблице:

i	t_i	$k^{(i)}$	$T^{(i)}$	$\Omega^{(i)}$	$P_1^{(i)}$	$P_2^{(i)}$
1	2	5	16	25	0.3109	0.1147
2	3	3	5249	9881	0.7695	10^{-8}

В таблице указаны списки состояний-кандидатов полученные на i-ом шаге:

i	$a_0 \dots a_{k^{(1)}+\dots+k^{(i)}-1}$
1	00000
	00100
	01011
	01100
	10011
	10100
	11011
	11111
2	0101100110
	0101110100
	1001100110
	1001110100
	1101100110
	1101110100

Получен список состояний-кандидатов, среди них есть правильное начальное состояние $a = 1001100110$.

Вообще говоря, чтобы определить, какие именно состояния-кандидаты являются правильными начальными состояниями, надо перебрать все возможные их комбинации и сравнить получаемую гамму с известной. В данном примере надо перебрать $1 \cdot 2 \cdot 3 \cdot 6 = 36$ комбинаций.