ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Март—апрель 2011. Том 18, № 2. С. 51–63

УДК 519.7

О СОКРАЩЕНИИ КЛЮЧЕВОГО ПРОСТРАНСТВА ШИФРА A5/1 И ОБРАТИМОСТИ ФУНКЦИИ СЛЕДУЮЩЕГО СОСТОЯНИЯ В ПОТОЧНОМ ГЕНЕРАТОРЕ *)

С. А. Киселёв, Н. Н. Токарева

Аннотация. Исследуются поточные шифры на регистрах сдвига с обратной связью. Для общего вида поточного генератора доказана теорема, позволяющая отождествить понятия обратимости функции следующего состояния и возвратности функции управления сдвигом. Отдельно рассматривается генератор для A5/1 — поточного шифра, используемого в стандарте GSM и обеспечивающего конфиденциальность переговоров. Для него подсчитано число состояний, которые могут быть получены за t тактов, начиная из состояний без предшественников, причём за меньшее число тактов в такие состояния перейти нельзя. Предложен способ экспоненциального сокращения ключевого пространства A5/1 при тактировании, который может быть использован в криптоанализе.

Ключевые слова: поточный шифр, регистр сдвига, A5/1.

Введение

GSM (Group Special Mobile) — глобальный цифровой стандарт для мобильной сотовой связи, разработанный ещё в конце 1980-х гг. Основу системы безопасности GSM составляют три алгоритма: A3 — алгоритм аутентификации, A8 — алгоритм генерации криптоключа и A5 — алгоритм шифрования оцифрованной речи для обеспечения конфиденциальности переговоров между абонентом и базовой станцией. Мобильные станции (телефоны) снабжены смарт-картой, содержащей алгоритмы A3 и A8, а в самом телефоне имеется ASIC-чип с алгоритмом A5. Базовые

*) Исследование выполнено при поддержке гранта Президента РФ для молодых российских ученых (грант МК–1250.2009.1). Исследование второго автора поддержано также Российским фондом фундаментальных исследований (проекты 09–01–00528, 10–01–00424, 11–01–00997) и ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № 02.740.11.0429).

станции также снабжены ASIC-чипом с алгоритмом A5 и центром аутентификации, использующим алгоритмы A3, A5, A8 для идентификации мобильного абонента и генерации сеансового ключа. В GSM используются две основные разновидности алгоритма: A5/1 — сильная версия шифра для избранных стран (в их числе и Россия) и A5/2 — ослабленная для всех остальных.

Криптоанализ A5/2 предложен в 1999 г. Вагнером и Голдбергом [8], сложность криптоанализа составила 2^{19} действий. Атаки на A5/1 начались в 1994 г. сразу после утечки информации из British Telecom [2]. В июне 1994 г. д-р Саймон Шеферд из Брэдфордского университета должен был представить на коллоквиуме IEEE в Лондоне свой корреляционный способ вскрытия A5/1 [7]. Однако в последний момент его выступление было запрещено штаб-квартирой правительственной связи. Двумя же наиболее известными современными работами являются статьи Голича [5] и Бирюкова, Шамира и Вагнера [3], использующие похожие идеи — парадокс дней рождения и балансировку время-память. Теоретическая сложность их методов составляет около 2^{40} действий. Ими частично исследован граф состояний A5/1: приведена доля состояний, имеющих 0, 1, 2, 3 и 4 предшественника соответственно, исследована структура случайно генерируемых деревьев (отдельно выделенных из графа всех состояний). Заметим, что из разных состояний A5/1 может получиться один и тот же поток гаммы произвольной длины. Такие ситуации называются коллизиями, примеры которых приводятся в [6], причём их поиск оказался довольно трудоёмким. Нами производится аналитическое исследование таких ситуаций.

В нашей статье исследуются поточные шифры на регистрах сдвига с обратной связью. Для общего вида поточного генератора доказывается теорема, которая позволяет отождествить понятия обратимости функции следующего состояния и возвратности функции управления сдвигом. Отдельно рассматривается генератор A5/1. Для него подсчитывается число состояний, которые могут быть получены совершением t тактов, начиная с состояний без предшественников, причём за меньшее число тактов в такие состояния перейти нельзя. Предложен способ экспоненциального сокращения ключевого пространства A5/1 при тактировании, который может быть использован в криптоанализе.

1. Об обратимости функции следующего состояния

В этом разделе доказывается теорема, позволяющая отождествить понятия обратимости функции следующего состояния и возвратности функции управления сдвигом.

Регистр сдвига с обратной связью состоит из двух частей: собственно регистра сдвига и функции обратной связи. Регистр состоит из битов, его длина — их количество. *Состоянием* регистра называется произвольный набор значений его битов. При тактировании все биты текущего состояния регистра сдвигаются влево на одну позицию. Крайний левый бит заносится в гамму. Новый крайний справа бит определяется функцией от остальных битов.

Рассмотрим поточный генератор псевдослучайной последовательности, состоящий из k регистров сдвига с обратной связью. *Состоянием* генератора будем называть двоичный вектор x длины n, биты которого последовательно заполняют биты всех k регистров, где n — сумма длин всех регистров. *Ключевым пространством* генератора назовём множество всех его возможных состояний, т. е. множество двоичных векторов длины n. При каждом такте могут сдвигаться состояния не всех регистров. Сдвиг определяет следующая функция. *Функцией управления сдвигом* назовём векторную булеву функцию $c : \mathbb{Z}_2^n \to \mathbb{Z}_2^k$, $c(x) = (y_1, \ldots, y_k)$, такую, что если генератор находится в состоянии x, то при следующем такте состояние j-го регистра сдвигается тогда и только тогда, когда $y_j = 1$. Для каждого j-го регистра сдвига определена своя функция обратной связи f_j. *Функцией следующего состояния* называется векторная булева функция $\mathrm{next} : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, которая в соответствии с функцией управления сдвигом и функциями обратной связи определяет следующее состояние: $\mathrm{next}(x) = x'$. В таком случае говорят, что x является *предшественником* состояния x', или, наоборот, x' — *последователь* состояния x.

Представим состояние x в виде $x = (x_1, \ldots, x_k)$, где x_j — вектор состояния j-го регистра сдвига. Аналогично представим и вектор x'. Тогда функцию следующего состояния $\mathrm{next} : x \to x'$ удобно представить в виде набора k векторных булевых функций $\mathrm{next}_j : x_j \to x'_j$. Нетрудно показать, что справедлива

Лемма 1. *Функция* next_j *обратима тогда и только тогда, когда функция* f_j *линейна по последней переменной.*

Пусть $x' = \mathrm{next}(x)$. Будем говорить, что функция управления сдвигом обладает свойством *возвратности*, если по любому состоянию x' можно восстановить вектор $c(x)$ однозначно.

Теорема 1. *Пусть для любого* $j = 1, \ldots, k$ *функция* f_j *линейна по последней переменной. Тогда функция следующего состояния обратима тогда и только тогда, когда функция управления сдвигом обладает свой-*

ством возвратности.

Доказательство. Необходимость. Пусть функция next обратима. Тогда по x' восстанавливается состояние x, с помощью которого вычисляется $c(x)$. Таким образом, по x' восстановлено значение $c(x)$.

Достаточность. Пусть функция $c(x)$ обладает свойством возвратности. Покажем, что функция next обратима, т. е. что вектор x однозначно восстанавливается по x'. Предположим, что это не так. Тогда найдутся два состояния генератора x, z такие, что $x \neq z$ и $x' = z'$, где $x = \mathrm{next}(x)$, $z' = \mathrm{next}(z)$. Пусть без ограничения общности состояния x и z отличаются в первом регистре (всегда найдётся регистр, в котором они различаются, будем считать его первым), т. е. выполняется $x_1 \neq z_1$, при этом $x'_1 = z'_1$. Поскольку функция управления сдвигом обладает свойством возвратности, справедливо $c(x) = c(z)$. А значит, при переходе от x к x' и от z к z' состояния первого регистра одновременно либо сдвигались, либо нет. Если сдвига не было, то должны выполняться равенства $x_1 = z_1 = x'_1 = z'_1$, что неверно. Если произошёл сдвиг, то $x_1 = z_1$, поскольку функция next_1 согласно лемме 1 обратима. В обоих случаях получаем противоречие с предположением. Значит, функция next обратима. Теорема 1 доказана.

Таким образом, можно свести такой сложный вопрос, как обратимость достаточно громоздкой функции next, к исследованию свойства возвратности более простой функции управления сдвигом. Например, для поточного генератора A5/1 обратимость функции $\mathrm{next} : \mathbb{Z}_2^{64} \to \mathbb{Z}_2^{64}$ сводится к исследованию функции $c : \mathbb{Z}_2^3 \to \mathbb{Z}_2^3$.

2. Исследование поточного генератора A5/1

2.1. Описание. A5/1 — это поточный алгоритм шифрования, используемый для обеспечения конфиденциальности передаваемых данных между телефоном и базовой станцией в европейской системе мобильной цифровой связи GSM. В A5/1 псевдослучайная последовательность реализуется на основе трёх регистров сдвига с линейной обратной связью. Регистры имеют длины 19, 22 и 23 битов соответственно. Сдвигами управляет функция большинства (также известная как majority: $m(a_1, a_2, a_3) = a_1a_2 \vee a_2a_3 \vee a_1a_3$) — в каждом регистре есть контрольный бит: восьмой в первом регистре (обозначим a_1), десятый во втором (обозначим a_2) и в третьем (обозначим a_3). Биты нумеруются справа налево. При очередном такте сдвигаются состояния только тех регистров, у которых значения контрольных битов совпадают со значением функции m.

Функция управления сдвигом и $m(a_1, a_2, a_3)$ связаны соотношением

$$c(x) = (a_1 \equiv m, a_2 \equiv m, a_3 \equiv m).$$

Последние биты регистров суммируются по модулю два. Результат сложения становится новым битом гаммы. Гамма накладывается на открытый текст, вследствие чего получается шифротекст. На одном ключе генерируется 114 битов гаммы.

Линейные функции обратной связи удобно представлять с помощью полиномов, сопоставляя каждому биту регистра соответствующую степень переменной x. В шифре A5/1 функции обратной связи задаются следующими полиномами:

$$\begin{aligned} &x^{19} + x^{18} + x^{17} + x^{14} + 1 && \text{для R1},\\ &x^{22} + x^{21} + 1 && \text{для R2},\\ &x^{23} + x^{22} + x^{21} + x^8 + 1 && \text{для R3}. \end{aligned}$$

Например, в первом регистре суммируются биты с номерами 18, 17, 16 и 13 (биты нумеруются справа налево, начиная с 0). Результат становится новым значением крайнего правого бита (рис. 1).

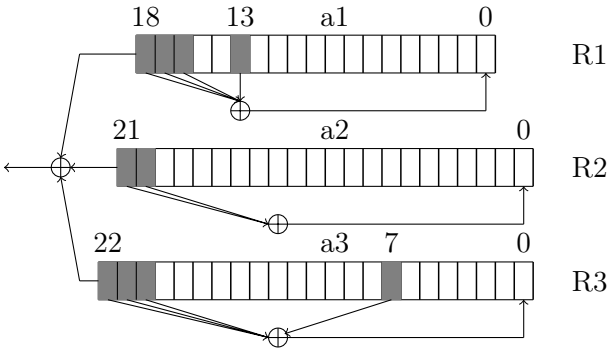

Рис. 1

На рис. 1 символами a1, a2, a3 отмечены контрольные биты, а темным цветом выделены биты, от которых существенно зависят функции обратной связи.

2.2. Анализ функции управления сдвигом. Введём следующее определение. *Шаблоном состояния* назовём такое множество состояний генератора, в которых часть битов зафиксирована конкретными значениями, а часть — нет. Например, если выбираем и фиксируем четыре

бита, то такой шаблон состояния состоит из 2^{60} состояний — ровно столькими способами можно означить оставшиеся шестьдесят битов.

Утверждение 1. *Функция управления сдвигом в генераторе A5/1 не обладает свойством возвратности.*

Доказательство. Достаточно привести некоторый общий контрпример. Рассмотрим шаблон состояния с шестью фиксированными битами, как это сделано на рис. 2.

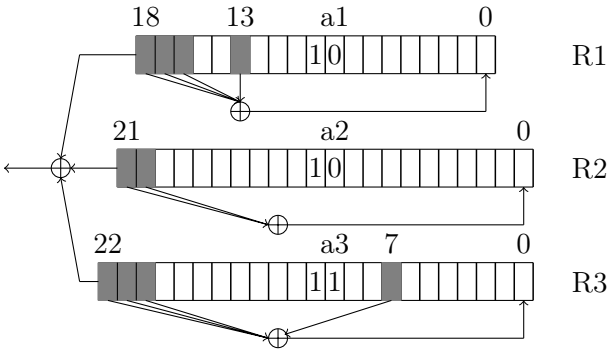

Рис. 2

Нетрудно понять, что для любого состояния из этого шаблона нельзя определить, состояния каких регистров сдвигались на последнем такте. Могли сдвигаться состояния первого и третьего, либо второго и третьего, либо всех трёх регистров. Это означает, что функция управления сдвигом не обладает свойством возвратности. Утверждение 1 доказано.

Из утверждения 1 и теоремы 1 следует, что в генераторе A5/1 функция следующего состояния необратима.

2.3. Подсчёт числа t-простых состояний. Введём следующие определения. *Состоянием без предшественников* назовём такое состояние генератора, в которое нельзя перейти ни из какого другого состояния. *t-Простым состоянием* генератора, где t — целое число, $t \geqslant 0$, назовём такое его состояние, которое получено совершением t тактов, начиная с состояния без предшественника, причём за меньшее число тактов в такое состояние перейти нельзя. Очевидно, что 0-простые состояния — это и есть состояния без предшественников. Количество t-простых состояний обозначим через N_t.

Обозначим контрольные биты регистров через a_1, a_2 и a_3, биты, находящиеся слева от контрольных, — через b_1, b_2, b_3.

Утверждение 2. *Справедливо* $N_0 = 3 \cdot 2^{61}$.

Доказательство. Будем фиксировать b_1, b_2, b_3 различными значениями, а остальные биты полагать произвольными. При этом множество всех состояний генератора разобьётся на восемь соответствующих шаблонов. Каждое состояние из шаблонов, определяемых векторами (000) и (111), очевидно, имеет предшественника. Действительно, оно могло быть получено сдвигом состояний всех трёх регистров.

Рассмотрим шаблоны, для которых в указанных битах присутствуют как нули, так и единицы. Например, рассмотрим шаблон, определяемый вектором (101). Этот шаблон разделим на два подшаблона, фиксируя значение четвертого бита, а именно бита a_2. Если $a_2 = 0$, то все состояния этого подшаблона имеют предшественников. Действительно, эти состояния можно получить, сдвинув на последнем такте состояния первого и третьего регистров. Рассмотрим подшаблон, в котором $a_2 = 1$. Он представлен на рис. 3.

Докажем, что у состояний этого подшаблона нет предшественников. Предположим, что для некоторого состояния s' из этого подшаблона существует предшественник s. Пусть при переходе от s к s' сдвигалось состояние второго регистра, т. е. $a_2(s) = 0$. Так как $b_1(s') = 1$ и $b_3(s') = 1$, состояния ни первого, ни третьего регистра при этом не могли сдвигаться, что невозможно, ведь в генераторе должны были сдвигаться состояния как минимум двух регистров. Следовательно, состояние второго регистра не сдвигалось. Тогда должны были сдвигаться состояния первого и третьего регистров, но в силу того, что $a_1(s) = 1$, $a_3(s) = 1$ и $a_2(s) = 1$, состояние второго регистра также должно было сдвигаться; противоречие. Значит, любое состояние s' из указанного подшаблона мощности 2^{60} не имеет предшественника.

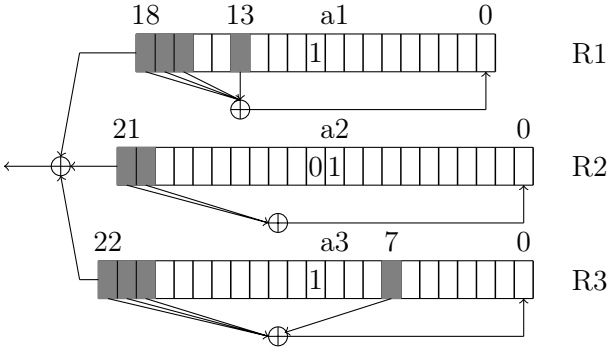

Рис. 3

Аналогичные рассуждения можно провести для остальных пяти шаблонов, определяемых битами b_1, b_2, b_3. Таким образом, получаем число N_0 состояний без предшественников: $N_0 = 6 \cdot 2^{60} = 3 \cdot 2^{61}$. Утверждение 2 доказано.

Исходя из полученных данных и самой конструкции генератора, вычислим число 1-простых состояний. Для этого используем метод включений-исключений. Согласно доказательству утверждения 2 все состояния без предшественников можно представить в виде шести шаблонов мощности 2^{60} каждый. Сделаем один такт для состояний из каждого шаблона. Полученное множество состояний можно представить в виде объединения новых 24 шаблонов меньшей мощности: из каждого начального шаблона, фиксируя новые биты и соответствующим образом сдвигая «картинку» влево, получаем четыре новых шаблона. Каждый раз фиксируем ровно два бита, так что шаблоны получаются мощности 2^{58}.

На рис. 4 приведены первые 12 шаблонов (квадрат 3×3 — это записанные друг под другом биты трёх регистров, начиная с контрольных). Еще 12 шаблонов получаются из приведённых заменой единиц нулями и нулей единицами. Далее необходимо вычислить мощности пересечения получившихся шаблонов состояний.

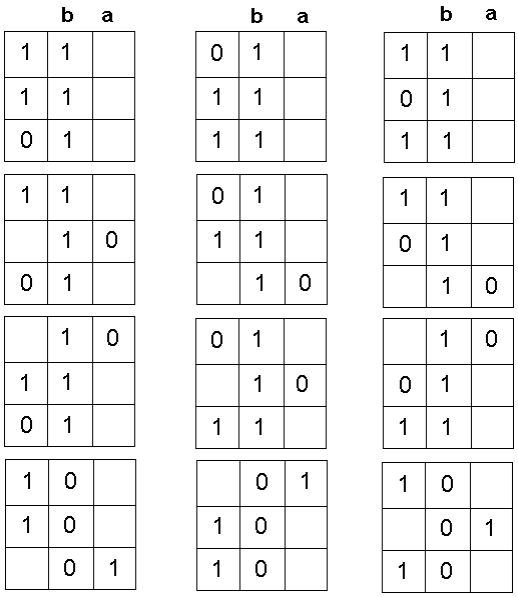

Рис. 4

Утверждение 3. *Число* 1*-простых состояний равно* $N_1 = 13 \cdot 2^{58}$, *что приблизительно составляет* $2^{61.7004}$.

Доказательство. Число попарных пересечений шаблонов 1-простых состояний мощности 2^{57} равно 12. Число попарных пересечений мощности 2^{56} равно 30. Число тройных пересечений мощности 2^{56} равно 6. Число тройных пересечений мощности 2^{55} равно 8. Непустых пересечений других кратностей и других мощностей нет. По формуле включений-исключений получаем число 1-простых состояний:

$$N_1 = 24 \cdot 2^{58} - 12 \cdot 2^{57} - 30 \cdot 2^{56} + 6 \cdot 2^{56} + 8 \cdot 2^{55} = 13 \cdot 2^{58}.$$

Утверждение 3 доказано.

Шаблоны 1-простых состояний удобно привести к единому виду. А именно, заполнить пустые клетки квадрата 3×3 всеми возможными значениями (при этом из одного шаблона получается восемь новых шаблонов мощности 2^{55}). Таким образом, получаем $24 \cdot 8 = 192$ шаблона. После исключения совпадающих остаётся ровно 104 различных шаблона. С этими шаблонами удобнее работать, так как они не имеют пересечений между собой. Как показано выше, $N_1 = 104 \cdot 2^{55} = 13 \cdot 2^{58}$.

Найдём число t-простых состояний, $t = 2, \ldots, 8$. Для соответствующих подсчётов написана программа, работающая согласно следующему алгоритму.

Пусть $t = 2$. Производим один такт над всеми 104 шаблонами 1-простых состояний. При этом в каждом шаблоне происходит сдвиг состояний как минимум двух регистров. Например, сдвигаются состояния двух первых регистров, так что если заполнение до сдвига таково:

1	1	0
1	1	0
0	1	1

, то после будет:

1	1	0	
1	1	0	
	0	1	1

.

Заполним три новые пустые клетки всевозможными значениями. Так, от квадрата 3×3, задающего шаблон 1-простого состояния, переходим к восьми прямоугольникам 3×4, определяющим шаблоны состояний, полученных за два такта. Их общее число составляет $104 \cdot 8$. Поскольку набор фиксирующих битов в каждом шаблоне один и тот же, шаблоны либо не пересекаются, либо совпадают. После исключения совпадающих остаётся 744 шаблона. Теперь необходимо проверить, могут ли состояния некоторых из них быть получены за меньшее число тактов (т. е. за один такт). Другими словами, не покрываются ли они шаблонами 1-простых состояний. Для этого в прямоугольнике 3×4 достаточно выделить справа

квадрат 3×3 и проверить, не является ли он определяющим квадратом для какого-нибудь шаблона 1-простого состояния. После таких проверок остаётся ровно 454 шаблона мощности 2^{52} каждый. Эти шаблоны и содержат все 2-простые состояния. Таким образом, получено следующее

Утверждение 4. *Справедливо* $N_2 = 227 \cdot 2^{53} \approx 2^{60.8265}$.

При $t = 3$ действуем аналогичным образом: совершаем такт над шаблонами 2-простых состояний, фиксируя биты, получаем новые шаблоны, исключаем повторения и проверяем, не покрываются ли некоторые из них шаблонами 1- и 2-простых состояний. Получаем, что множество всех 3-простых состояний является объединением $2568 = 3 \cdot 107 \cdot 2^3$ непересекающихся шаблонов мощности 2^{49}. Каждый шаблон задаётся прямоугольником 3×5, в котором крайний правый столбец является столбцом контрольных битов.

Утверждение 5. *Число* 3*-простых состояний равно* $N_3 = 321 \cdot 2^{52}$, *что приблизительно составляет* $2^{60.3264}$.

Аналогично, множество 4-простых состояний можно определить с помощью $15\ 266 = 17 \cdot 449 \cdot 2$ шаблонов мощности 2^{46}. Каждый шаблон определяется значениями 18-ти битов.

Утверждение 6. *Выполняется* $N_4 = 7\,633 \cdot 2^{47} \approx 2^{59.8980}$.

5-Простые состояния задаются $91\,468 = 13 \cdot 1759 \cdot 2^2$ шаблонами мощности 2^{43}.

Утверждение 7. *Число* 5*-простых состояний равно* $N_5 = 22\,867 \cdot 2^{45}$, *что приблизительно составляет* $2^{59.4809}$.

Все 6-простые состояния можно представить в виде объединения $548\,694 = 3^5 \cdot 1129 \cdot 2$ шаблонов мощности 2^{40}.

Утверждение 8. *Справедливо* $N_6 = 274\,347 \cdot 2^{41} \approx 2^{59.0656}$.

Множество 7-простых состояний определяется $3\,292\,064 = 102\,877 \cdot 2^5$ шаблонами мощности 2^{37}.

Утверждение 9. *Выполняется* $N_7 = 102\,877 \cdot 2^{42} \approx 2^{58.6505}$.

Все 8-простые состояния представляются как объединение 19 752 298 шаблонов мощности 2^{34}. Каждый шаблон определяется фиксацией уже 30-ти битов в прямоугольнике размера 3×10. На рис. 5 биты, определяющие шаблон, отмечены точками.

Утверждение 10. *Справедливо* $N_8 = 9\,876\,149 \cdot 2^{35} \approx 2^{58.2355}$.

Для вычисления двух последних значений потребовались уже серьёз-

ные затраты. Для вычисления N_7 понадобилось около 1Gb оперативной памяти и 25 секунд работы процессора Core i7 3.0Ghz 12 Gb. Параметр N_8 был вычислен на этом же процессоре за 3 минуты с использованием 6 Gb оперативной памяти.

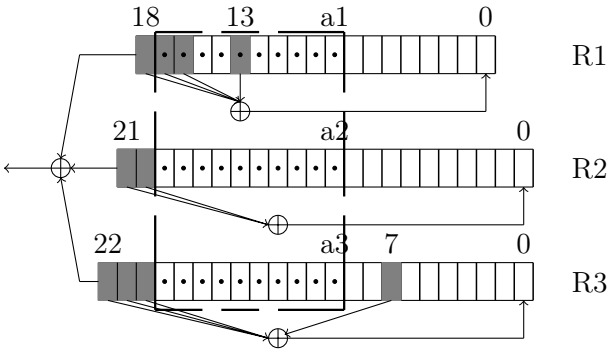

Рис. 5

3. Сокращение ключевого пространства A5/1

Рассмотрим следующую атаку на поточный шифр A5/1. Пусть, начиная с некоторого неизвестного нам состояния, генератор вырабатывает гамму. С каждым новым битом гаммы сокращается множество состояний, в которые генератор мог перейти после выработки ещё и этого бита. Наша задача — анализируя гамму, определить текущее состояние генератора. *Теоретической стойкостью* к такой атаке (или *расстоянием единственности*) называется наименьшая длина L начального отрезка гаммы, достаточного для однозначного восстановления того состояния генератора, в котором он окажется после выработки этого отрезка гаммы.

Напомним, что N_t обозначает число состояний генератора, в которые он может перейти за t тактов работы и не может перейти за меньшее число тактов из всевозможных состояний без предшественников. Если K — число всех состояний генератора (в нашем случае равное 2^{64}), то после совершения t тактов генератор может находиться только в одном из K_t состояний, где $K_t = K - \sum\limits_{i=0}^{t-1} N_i,\ t \geqslant 1$. Удобно считать, что $K_0 = K$. Пусть t^* — наименьшее число тактов, при котором $K_{t^*} = 1$. Тогда справедливо неравенство $L \leqslant \min\{t^*, P+T\}$, где P и T — длины предпериода и периода гаммы соответственно.

Нетрудно проверить, что в генераторе A5/1 все многочлены обратной связи примитивны над $GF(2)$. Из этого следует, что каждый из трёх

регистров имеет максимальный период $2^\ell - 1$, где ℓ — длина регистра. В генераторе A5/1 вероятность движения состояния каждого из регистров равна 3/4. А значит, справедливо

Утверждение 11 [4]. *Минимальный период генератора A5/1 равен* $4/3 \cdot (2^{19} - 1)$.

Таким образом, при любом начальном состоянии генератора выполняется $P + T \geqslant 4/3 \cdot (2^{19} - 1)$. Это число является достаточно большим. Естественно предположить, что теоретическая стойкость L будет оцениваться скорее числом t^*. Исследуем продвижения в вычислении этого параметра.

Нами определены значения $N_0, \ldots, N_8$, составляющие в сумме достаточно большое число. В следующей таблице собраны полученные результаты. В ней приводятся также округлённые значения $(K_t/K) \cdot 100\%$ при $t = 0, \ldots, 9$. Эти числа показывают, какой объём возможных состояний генератора (в процентах от первоначального) останется после t произведённых тактов.

Число тактов, t	$\log_2(N_{t-1})$	$\log_2(K_t)$	$(K_t/K) \cdot 100\%$
0	–	64.0000	100%
1	62.5849	63.3219	62.5000%
2	61.7004	62.7548	42.1875%
3	60.8265	62.3151	31.1035%
4	60.3264	61.8963	23.2666%
5	59.8980	61.4807	17.4430%
6	59.4809	61.0656	13.0815%
7	59.0656	60.6505	9.8110%
8	58.6505	60.2355	7.3583%
9	58.2355	59.8204	5.5187%

Например, в начале работы возможны все 2^{64} состояния генератора. После первого такта генератор будет находиться в одном из $0.625 \cdot 2^{64}$ состояний, после второго — в одном из $0.421875 \cdot 2^{64}$ и т. д. После девятого такта возможны лишь около 5.5% его состояний, которые нетрудно получить в явном виде, действуя так, как показано в п. 2.3. Этот факт можно непосредственно использовать в криптоанализе A5/1.

Отметим, что наблюдается экспоненциальное сокращение ключевого пространства с каждым новым тактом. Остаётся открытым вопрос, сохранится ли такая тенденция в дальнейшем. Необходимы дальнейшие вычисления. Однако, увеличивая число тактов, нужно учитывать и то, что фиксируемые биты в шаблонах вскоре достигнут левого края регистров. Когда это произойдёт, технику вычисления t-простых состояний необходимо будет изменить.

Представляется интересным получить верхнюю оценку числа t^*, а также определить номер такта, после которого ключевое пространство сокращаться не будет (или почти не будет).

Часть результатов работы анонсирована в [1].

Авторы выражают глубокую благодарность Г. П. Агибалову и И. А. Панкратовой за ценные замечания, способствовавшие улучшению статьи, за указание на целый ряд неточностей, а также за предложенное более простое доказательство теоремы 1.

ЛИТЕРАТУРА

1. **Киселёв С. А.** О сокращении ключевого пространства поточного шифра A5/1 при тактировании // Прикл. дискрет. математика. Прил. № 3. — 2010. — С. 21–23.
2. **Anderson R.** Subject: A5 // posting to Newsgroups: sci.crypt, alt.security, uk.telecom; 17 June 1994.
3. **Biryukov A., Shamir A., Wagner D.** Real time cryptanalysis of A5/1 on a PC // Fast Software Encryption Workshop — FSE'2000. (New York, April 10–12, 2000): Proc. — Berlin: Springer–Verl., 2001. — P. 1–18. (Lect. Notes Comput. Sci.; Vol. 1978).
4. **Chambers W. G.** On random mappings and random permutations // Fast Software Encryption Workshop — FSE'1994. (Leuven, December 14–16, 1994): Proc. — Berlin: Springer-Verl., 1995. — P. 22–28. (Lect. Notes Comput. Sci.; Vol. 1008).
5. **Golic J.** Cryptanalysis of alleged A5 stream cipher // Adv. Cryptology. Workshop on the theory and application of cryptographic techniques — EUROCRYPT'97 (Konstanz, May 11–15, 1997): Proc. — Berlin: Springer-Verl., 1997. — P. 239–255. (Lect. Notes Comput. Sci.; Vol. 1233).
6. **Semenov A., Zaikin O., Bespalov D., Posypkin M.** Parallel algorithms for SAT in applocation to discrete functions inversion problems // arXiv.org, Preprint 1102.3563v1.
7. **Shepherd S.J.** Cryptanalysis of the GSM A5 cipher algorithm // IEEE Colloquium on Security and Cryptography Applications to Radio Systems, Digest No. 1994/141 (Savoy Place, London, June 3, 1994).
8. **Wagner D. et al.** The real-time cryptanalysis of A5/2 // Crypto'99 (Santa Barbara, August 15–19, 1999): Proc. — Berlin: Springer–Verl., 1999 — P. 12–21.

Киселёв Семён Александрович,
e-mail: kiselev.senya@gmail.com

Токарева Наталья Николаевна,
e-mail: tokareva@math.nsc.ru

Статья поступила
24 июня 2010 г.
Переработанный вариант —
19 февраля 2011 г.