УДК 621.3.06

А.В. КАЗИМИРОВ, Р.В. ОЛЕЙНИКОВ, канд. техн. наук

КРИПТОАНАЛИЗ ШИФРА MICKEY НА ОСНОВЕ АНАЛИЗА ВНУТРЕННИХ СОСТОЯНИЙ

Введение

Нелинейные регистры с обратной связью оказываются перспективными элементами для построения генераторов гаммы (поточных шифров). Такие криптопримитивы эффективны при аппаратной реализации и обеспечивают высокий уровень нелинейности, что имеет решающее значение для обеспечения криптографической стойкости. Тем не менее, работы [1, 2], посвященные анализу этих конструкций, основаны на ряде допущений и не имеют формальных оценок и доказательств. Ниже показано, что неудачный выбор параметров, которые, на первый взгляд, не влияют на стойкость, может кардинально снизить сложность криптоаналитической атаки.

Основными требованиями к гамме шифрующей являются большая длина периода, оптимальные корреляционные свойства и высокая нелинейность [3, 4]. Чтобы добиться этого, разработчики поточных шифров часто объединяют линейные и нелинейные регистры. При этом предполагается, что линейная часть гарантирует период гаммы шифрующей, а нелинейная обеспечивает стойкость к статистическим и алгебраическим атакам. Кроме того, функция обновления для обоих регистров может включать в себя состояние соседнего регистра и осуществлять так называемое взаимное управление. Современные генераторы гаммы, в дополнение к секретному ключу, часто используют открытый вектор инициализации (IV), что позволяет использовать один и тот же ключ в разных сеансах связи. Такой подход к построению был применен при разработке шифра Mickey [5, 6].

В предыдущей работе [7] были исследованы некоторые характеристики дерева обратных состояний алгоритма Mickey. В [8] представлена теоретическая оценка графа состояний. В представленной статье обобщаются предыдущие результаты и рассматриваются новые методы анализа шифра.

Краткое описание поточного шифра MICKEY

Существуют две версии шифра - Mickey-80 и Mickey-128 [5, 6]. Обе версии основаны на комбинации линейного (*R*) и нелинейного (*S*) регистров. Алгоритм имеет два входных параметра: вектор инициализации (IV) и сеансовый ключ (*K*). Общая структура алгоритма показана на рис. 1. Максимальная длина сообщения, которая может быть получена при использовании одной пары (*K, IV*), равна 2^{40}. Различия в параметрах между версиями представлены в табл. 1.

Ячейки регистров являются битовыми и обозначаются r_0, r_1, …, r_{RL-1} и s_0, s_1, …, s_{RL-1}. *R* и *S* тактируются в соответствии с управляющими битами $CB\_R = CB\_SL \oplus CB\_RR$ для регистра *R* и $CB\_S = CB\_SR \oplus CB\_RL$ для *S* – соответственно, где *CB_XY* означает определённый бит регистра *X* из табл. 1 (например, *CB_SL* = s_{34} для Mickey-80). Функция обновления регистров имеет дополнительный параметр – входной бит (*IB_S* и *IB_S*).

Таблица 1

Различия между параметрами шифров Mickey-80 и Mickey-128

Версия	Длина регистра (RL)	Длина ключа	Длина холостого хода	*CB_SL*	*CB_SR*	*CB_SM*	*CB_RL*	*CB_RR*
80	100	80	100	34	67	50	33	67
128	160	128	160	54	106	80	53	106

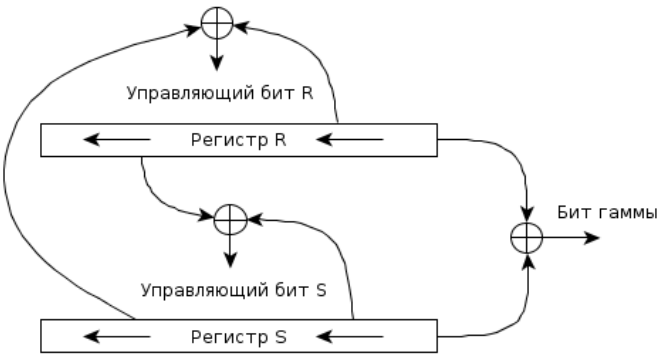

Рис. 1. Обобщённая структура шифра Mickey

Как видно из табл. 1, соотношение длины регистров ко всем остальным параметрам почти одинаково для обеих версий шифра. Несмотря на этот факт, дерево обратных состояний для Mickey-80 и Mickey-128 значительно отличается, что показано ниже.

Шифр имеет следующие режимы работы:

– инициализация регистров *R* и *S* нулями;
– загрузка IV (*CLOCK_K_IV*);
– загрузка ключа (*CLOCK_K_IV*);
– холостой ход (*CLOCK_PRECLOCK*);
– генерация гаммы (*CLOCK_KG*).

Все режимы, кроме генерации гаммы, также работают в так называемом режиме смешивания. Это означает, что входной бит регистра *R* (*IB_R*) зависит от бита *CB_SM* регистра *S*. Более подробное описание приведено в работе [7].

Модель анализа нелинейного генератора гаммы шифрующей

В [9] уже была описана модель атаки, основанная на знании внутреннего состояния регистров. Далее приводится её краткое описание и применение к шифру Mickey.

Предполагается, что злоумышленник знает внутреннее состояние генератора гаммы (состояние регистров *R* и *S*) и сколько тактов было выполнено, чтобы попасть в это состояние из начального. Оценка стойкости шифра Mickey основана на построении дерева обратных состояний, которое рассчитывается с использованием обратных функций тактирования шифра $CLOCK\_PRECLOCK^{-1}$, $CLOCK\_K\_IV^{1}$ и $CLOCK\_KG^{-1}$ (примеры алгоритмов приведены в [7]). Получение обратных состояний сводится к полному перебору всех возможных значений входных параметров (бита обратной связи, управляющего и входного битов) и исключение невозможных состояний.

Стоит отметить, что предыдущее состояние не всегда определяется однозначно. Количество ветвлений может варьироваться в зависимости от известного криптоаналитику состояния и режима шифрования. В зависимости от режима, можно получить три различных дерева состояний. Любое из этих деревьев может рассматриваться как граф переходов конечного автомата. В общем случае дерево показано на рис. 2. В дальнейшем будут использоваться понятия, связанные с деревом, такие как уровень - множество обратных состояний, определяющие исходное состояние после определенного количества тактов и степень ветвления (CB) - число возможных предыдущих состояний, которые дают состояние *R* и *S* за один такт вперед. Точки ветвления в дереве обратных состояний хранят значение степени ветвления. В режиме загрузки ключа/IV рёбра помечаются соответствующими искомыми битами ключа. Используя приведенную структуру дерева, при достижении состояния со CB = 0 можно отсечь часть дерева, что приводит к уменьшениюпространственной сложности атаки.

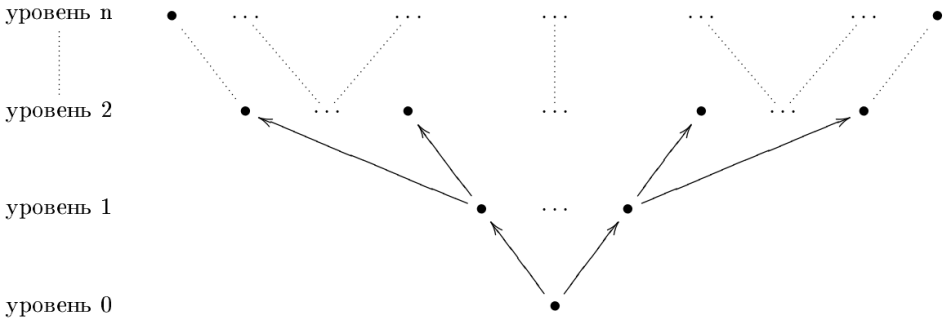

Рис. 2. Дерево обратных состояний

Предыдущая модель может быть сведена к более частному случаю в предположении, что злоумышленник знает внутреннее состояние генератора гаммы после определенного числа тактов в режиме загрузки ключа. Это количество тактов должно быть небольшим для отката состояния генератора до начала загрузки ключа (конец загрузки IV). Знание IV злоумышленником приводит к тому, что большая часть состояния дерева может быть удалена. На основании этого можно практически однозначно определить ключ, так как начальное состояние генератора фиксировано, и только один путь в дереве будет соответствовать реальному обновлению состояний регистра.

Другой способ нахождения ключа основывается на атаке типа "встреча посередине". Пусть известно состояние регистра после *k*-тактов от начала режима загрузки ключа, что так же соответствует *k* битам ключа. Схема атаки состоит из трёх этапов и представлена на рис. 3.

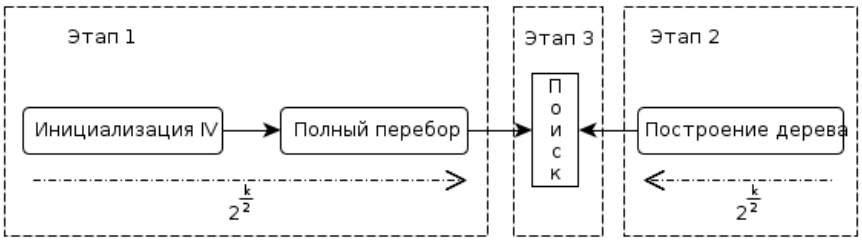

Рис. 3. Атака типа "встреча посередине" на шифр Mickey

На первом этапе для известного IV вычисляется состояние, которое эквивалентно входному значению в функцию инициализации ключа. Для полученного значения находятся все возможные состояния для $\frac{k}{2}$ битов ключа с применением функции *CLOCK_K_IV*. На втором этапе, на основе внутреннего состояния регистров, полученных после *k* тактов генератора, с использованием функции *CLOCK_K_IV*$^{-1}$ строится дерево обратных состояний вплоть до $\frac{k}{2}$ уровня. После этого, на последнем этапе находятся одинаковые состояния во множествах, сформированных на предыдущих этапах.

Сложность атаки полного перебора или построения дерева обратных состояний приблизительно равна 2^k. Однако комбинация этих двух методов в атаке типа "встреча посередине" является более практичной и даёт сложность

$$O_d(k) + O_i(k) + O_f(k) = O_d(2^{\frac{k}{2}}) + O_i(2^{\frac{k}{2}}) + O_f(2^{\frac{k}{2}}) \approx 2^{\frac{k}{2}+2},$$

где $O_d(k)$ – сложность атаки полного перебора для $\frac{k}{2}$ битов ключа, $O_i(k)$ – сложность построения дерева обратных состояний и $O_f(k)$ – сложность нахождения одинаковых состояний; используется хэш-таблица [10].

Результаты анализа шифра MICKEY

Количество допустимых предшествующих состояний

Как отмечено выше, предполагается, что криптоаналитик знает состояние регистров после определённого количества тактов работы шифра.

Вероятность получения степени ветвлений предшествующих состояний, определенная на основе анализа дерева обратных состояний на глубину до 18 уровней (обратных тактов) с помощью функции $CLOCK\_X^{-1}$, приведена в табл. 2 (случайное заполнение внутреннего состояния регистров) и табл. 3 (верное определённое состояние).

Таблица 2

Вероятность степени ветвления для случайного начального состояния

СВ	Загрузка ключа/IV		Холостой ход		Генерация гаммы	
	80	128	80	128	80	128
0	0.2982	0.198	0.2802	0.2825	0.3014	0.2718
1	0.00009	0.1031	0.4377	0.4590	0.4052	0.4281
2	0.4229	0.4022	0.2735	0.2294	0.2844	0.29
3	0.0001	0.1087	-	0.0256	-	-
4	0.2698	0.1703	0.0085	0.0035	0.0090	0.0101
6	0.00001	0.0177	-	-	-	-
8	0.0089	-	-	-	-	-

Таблица 3

Вероятность степени ветвления для верного определённого начального состояния

СВ	Загрузка ключа/IV		Холостой ход		Генерация гаммы	
	80	128	80	128	80	128
0	0.2773	0.2186	0.3052	0.29	0.3041	0.3038
1	0.00001	0.1047	0.4345	0.4534	0.4323	0.4154
2	0.4331	0.3753	0.2523	0.2256	0.2558	0.2698
3	0.00002	0.1029	-	0.0289	-	-
4	0.28	0.1783	0.008	0.0021	0.0079	0.0111
6	0.00007	0.0203	-	-	-	-
8	0.0095	-	-	-	-	-

Как видно из табл. 2 и 3, вероятности для случайного и верного определённого заполнения регистров примерно одинаковы и соответствуют теоретическим (вычисленным по методике, приведенной в [8]).

Математическое ожидание числа степени ветвлений для режима загрузки ключа/IV приблизительно равно 2, в то время как для режима холостого хода и генерации гаммы - 1. Таким образом, получив значение регистров на любом из этапов, криптоаналитик может выполнить обратное тактирование и получить состояние регистров после инициализации ключа. Сложность этапа приблизительно равна $O(k)$, где k – количество тактов, сделанных регистром после завершения этапа инициализации ключа.

В табл. 4 - 6 показано среднее число допустимых состояний на каждом уровне для режимов загрузки ключа/IV, холостого хода и генерации гаммы соответственно. Усреднение проводилось на 1000 случайных состояний, чтобы исключить зависимость от начального состояния.

Таблица 4

Количество допустимых предшествующих состояний
в зависимости от уровня дерева обратных состояний в режиме загрузки ключа/IV

Уровень	Количество состояний	
	80	128
0	1	1
1	3	3
2	9	7
3	25	18
4	45	39
5	143	82
6	247	171
7	523	347
8	1183	703
9	2221	1435
10	5075	2904
11	9453	5849
12	18694	11834
13	37702	23801

Таблица 5

Режим холостого хода

Уровень	Количество состояний	
	80	128
0	1	1
1	1	1
2	1	2
3	2	3
4	3	4
5	4	6
6	5	8
7	7	8
8	5	11
9	5	14
10	10	15
11	12	15
12	13	15
…	…	…
99	25	30
100	32	27
…	…	…
159	-	59

Таблица 6

Режим генерации гаммы

Уровень	Количество состояний	
	80	128
0	1	1
1	1	2
2	1	3
3	2	4
4	3	6
5	3	6
6	4	4
7	5	3
8	3	4
...	...	...
125	17	64
126	17	81
127	16	91
128	20	97

Как видно из табл. 4 - 7, количество допустимых предшествующих состояний увеличивается в соответствии с математическим ожиданием индекса степени ветвлений для соответствующего режима.

Определение битов ключа на основе дерева обратных состояний

Дерево обратных состояний, описанное выше, позволяет криптоаналитику вычислять значение некоторых битов ключа. Их количество и расположение определяется структурой дерева и зависит от конкретного значения вектора инициализации и ключа шифрования (табл. 7).

С увеличением уровня увеличивается и вероятность появления тупикового состояния (для которого отсутствуют допустимые предшествующие). Каждый обратный шаг увеличивает вероятность отсечения поддерева с множеством неверных значений битов ключа.

Как видно из колонки для Mickey-128, биты ключа могут быть однозначно определены на уровнях 1 и 3 ("1" и "0" соответственно). Однако, зная эти данные, невозможно найти однозначно обратное состояние на этих уровнях, следовательно, они не уменьшают дерево и сложность определения всех битов ключа. Более того, возможность нахождения бита ключа напрямую зависит от начального состояния. Например, для Mickey-80 невозможно определить биты ключа с вероятностью, равной 1.

Таблица 7

Распределение вероятностей значений битов ключа на пяти уровнях дерева

Уровень	Вероятность			
	80		128	
	1	0	1	0
1	0.5	0.5	1	0
2	0.5	0.5	0.5	0.5
3	0.5	0.5	0	1
4	0.5	0.5	0.5	0.5
5	0.4857	0.5143	0.5	0.5

Нахождение одинаковой гаммы для различных пар ключа и IV

Функции, используемые в различных режимах шифра Mickey, допускают генерацию гаммы шифрующей, сдвинутой на несколько бит для различных пар ключа и IV (перекрытие гаммы).

Пусть z_i^h – i-й бит гаммы для h-го набора (K_h, IV_h), где ключ имеет вид $K_1 = \{k_0, k_1, \ldots, k_{n-1}\}$, а n – длина ключа в битах (табл. 1). Тогда можно найти такие K_2, IV_1 (значение первого вектора инициализации) и IV_2, для которых состояние регистров будут отличаться на один шаг, а выходная гамма иметь свойство $z^2{}_i = z^1{}_{i+1}$.

Допустим, что $IV_1 = \{iv_0, iv_1, \ldots, iv_j\}$, $IV_2 = \{iv_0, iv_1, \ldots, iv_j, k_0\}$ и $K_2 = K_1 << 1 = \{k_1, k_2, \ldots, k_{n-1}, 0\}$. Относительное размещение битов для различных наборов ключа/IV приведены в табл. 8.

Таблица 8

Отличия параметров для различных наборов ключа и IV

IV_1				K_1						Холостой ход				Генерация гаммы					
iv_0	iv_1	…	iv_j	k_0	k_1	k_2	…	k_{n-2}	k_{n-1}	0	0	0	…	0	0	0	0	0	…
iv_0	iv_1	…	iv_j	k_0	k_1	k_2	…	k_{n-2}	k_{n-1}	0	0	0	…	0	0	0	0	0	…
IV_2				K_2						Холостой ход				Генерация гаммы					

Очевидно, что состояния регистров отличаются на один шаг, что, в свою очередь, приводит к формированию одинаковой гаммы шифрования, сдвинутой на один бит. Это обусловлено тем, что режим холостого хода эквивалентен режиму инициализации ключа/IV, когда "входной бит" равен 0.

Отличия состояния регистров на один шаг означает, что гамма шифрования будет иметь те же свойства, то есть $z^2{}_i = z^1{}_{i+1}$. Перекрытие в шифре Mickey присутствует из-за того, что режим холостого хода эквивалентен режиму инициализации ключа/IV, когда "входной бит" равен 0. Должно выполняться дополнительное условие: $s^1{}_{50} = 0$ в момент перехода между режимами холостого хода и генерации гаммы. Более того, длина IV_2 может быть увеличена на определенное количество бит, и в результате гамма шифрования будет сдвинута на такое же значение (при условии соответствия ключа K_2 вектору инициализации, см.табл. 8).

Более того, длина IV_2 может быть изменена на другое количество бит, в результате K_2 и гамма шифрования будут сдвинуты на такое же значение.

Аналогичные рассуждения могут быть применены к 128-разрядной версии Mickey.

Ниже представлены примеры одинаковых гамм для различной длины IV. Все значения имеют шестнадцатеричное представление, кроме первого значения Z_1 и последнего Z_2 – они являются битами.

Значения для Mickey-80:

K_1 = { d3, ec, f0, 84, 8a, 1d, b1, b7, 4a, dd }
IV_1 = { 58, e5, 77, 0a, 9c, a2, 34, c7, cd, 5e } (79 bits)
K_2 = { a7, d9, e1, 09, 14, 3b, 63, 6e, 95, ba }
IV_2 = { 58, e5, 77, 0a, 9c, a2, 34, c7, cd, 5f } (80 bits)
Z_1 = { 0, B7, 61, 27, 92, C5, 85, 91, 51, 18, 2A, D6, 7C, 8C, C8, C7, 04 }
Z_2 = { B7, 61, 27, 92, C5, 85, 91, 51, 18, 2A, D6, 7C, 8C, C8, C7, 04, 1 }

Значения для Mickey-128:

K_1 = { c9, 55, e7, 7a, 80, 13, 1a, ad, 40, 45, d9, 6c, 71, 04, 97, 9c }
IV_1 = { 4e, db, 6e, 01, 05, 98, 2b, 30, c3, 56, 5a, ed, 80, 85, 18, aa } (127 bits)
K_2 = { 92, ab, ce, f5, 00, 26, 35, 5a, 80, 8b, b2, d8, e2, 09, 2f, 38 }
IV_2 = { 4e, db, 6e, 01, 05, 98, 2b, 30, c3, 56, 5a, ed, 80, 85, 18, ab } (128 bits)
Z_1 = { 0, 79, 23, 91, 05, E1, DD, 2D, 9D, 83, 3E, B4, 78, 52, E5, A6, 66 }
Z_2 = { 79, 23, 91, 05, E1, DD, 2D, 9D, 83, 3E, B4, 78, 52, E5, A6, 66, 1 }

Выводы

Шифр Mickey является генератором гаммы, основанном на одновременном использовании линейного и нелинейного регистров. Проведенный анализ алгоритма показал, что криптоанализ на основе обратного тактирования возможен во всех режимах, включая режим загрузки ключа/IV. Несмотря на то, что переходы между внутренними состояниями осуществляются псевдослучайным образом, в некоторых случаях, возможно восстановление битов ключа шифрования.

Особенности функций тактирования дают возможность находить пары ключа и IV, использование которых приводит к сдвигу гамма шифрования на определенное количество позиций. Ограничение, накладываемое в спецификации шифра на длину вектора инициализации при использовании одного ключа, не позволяет реализовать практическую атаку.

Корректность известной теоретической методики [8] для оценки вероятности степени ветвлений шифра Mickey дополнительно подтверждается нашими практическими результатами, что позволяет обосновать выбор параметров Mickey-подобных алгоритмов шифрования на стадии проектирования.

Список литературы: 1. *Dubrova, E., Maxim,T., Hannu, T.* On Analysis and Synthesis of (n,k)-Non-Linear Feedback Shift Registers. DATE 2008. – pp. 1286-1291. 2. *Dubrova, E.* A List of Maximum Period NLFSRs. Electronic resource. Available at http://eprint.iacr.org/2012/166.pdf. 3. *Rueppel, R.* Analysis and Design of Stream Ciphers. - Springer-Verlag, 1986. - 244 p. 4. *Грушо, А. А.* Теоретические основы компьютерной безопасности : учеб. пособие для студентов высш. учеб. заведений / А. А. Грушо, Э. А. Применко, Е. Е. Тимонина. – М. : Изд. центр «Академия», 2009. – 272 с. 5. *Babbage, S., Dodd, M.* The Stream Cipher Mickey 2.0. eSTREAM, ECRYPT Stream Cipher Project, End of 3rd Phase, September 2009. http://www.ecrypt.eu.org/stream/ p3ciphers/mickey/mickey_p3.pdf. 6. *Babbage, S., Dod, M.* The stream cipher MICKEY-128 2.0. eSTREAM, ECRYPT Stream Cipher Project, End of 3rd Phase, September 2009. http://www.ecrypt.eu.org/ stream/p3ciphers/mickey/mickey128_p3.pdf. 7. *Казимиров, А.В., Олейников, Р.В.* Оценка количества допустимых внутренних состояний в поточном алгоритме Mcikey // Прикладная радиоэлектроника. – 2011. – Т.10, №2. - С. 112-115. 8. *Jansen, C.J.A.* The State Space Structure of the Mickey Stream Cipher. In Horlin, F., ed. // 32nd Symposium on Information Theory in the Benelux, Brussels, Werkgemeenschap voor Informatien Communicatietheorie (2011). 9. *Golic, J. D.* Cryptanalysis of alleged A5 stream cipher, in Advances in Cryptology – EUROCRYPT'97 (W. Fumy, ed.), vol. 1233 of Lecture Notes in Computer Science, Springer-Verlag, 1997. – P. 239-255. 10 *Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К.* Хеш-таблицы. Гл. 11. // Алгоритмы: построение и анализ ; под ред. И. В. Красикова. – 2-е изд. – М. : Вильямс, 2005. – 1296 с.

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 21.09.2012