УДК 621.3.06

ОЦЕНКА КОЛИЧЕСТВА ДОПУСТИМЫХ ВНУТРЕННИХ СОСТОЯНИЙ В ПОТОЧНОМ АЛГОРИТМЕ MICKEY

Р.В.Олейников, кандидат технических наук, А.В.Казимиров

Рассматривается метод анализа количества ветвлений симметричного поточного шифра Mickey, основанный на обратных функциях. При помощи данного подхода оценивается количество возможных состояний, в которые может переходить алгоритм. Установлено, что возможны коллизии при различных вариантах вектора инициализации и ключа.

In this article number of stream cipher Mickey's branches is analyzed. Using this approach, which is built on the basis of inverse functions, it is possible to estimate number of states. Possibility of collisions between different initialization vectors and keys is shown.

ВВЕДЕНИЕ

На сегодняшний день поточные шифры являются наиболее производительными среди всех видов криптографических алгоритмов, которые используются для обеспечения конфиденциальности. Вместе с рядом преимуществ, этот класс алгоритмов имеет и ряд недостатков, среди которых одним из основных является ограниченный период гаммы, что приводит к необходимости периодической смены векторов инициализации или ключей шифрования.

В 2008 году закончился европейский конкурс eSTREAM [1]. По итогам конкурса все, представленные на конкурс, поточные шифры разделили на две категории [1]. К первой отнесли шифры, рекомендованные к программной реализации, ко второй – к аппаратной.

Алгоритм Mickey является одним из поточных шифров, рекомендованных к использованию защиты данных, и относится ко второй категории [1, 2]. Преимуществом данного алгоритма является высокая производительность и высокая криптостойкость. На сегодняшний день опубликовано достаточно малое количество работ, посвящённых криптоанализу данного алгоритма [3].

В настоящей статье получена среднее значение количества допустимых внутренних состояний конечного автомата, которым является шифр Mickey, на основе использования обратных тактовых функций.

1. ОПИСАНИЕ АЛГОРИТМА ШИФРОВАНИЯ MICKEY

Шифр mickey строится на основе линейного (*R*) и нелинейного (*S*) регистров.

Алгоритм принимает на вход два параметра: вектор инициализации (*IV*), длина которого может составлять от 0 до 80 бит, и сеансовый ключ (*K*) длиной 80 бит [2]. Существует 128-битная версия шифра [4], однако в дано работе рассматривается лишь 80 битный вариант, но аналогичные исследования можно провести и для 128 версии.

Пусть $r_0 \ldots r_{99}$ и $s_0 \ldots s_{99}$ биты регистров *R* и *S* соответственно. Тактирование линейного регистра (*CLOCK_R*(*R, IB_R, CB_R*)) происходит следующим образом:

- $FB\_R = r_{99} \oplus IB\_R$;
- For $1 \le i \le 99$, $r_i' = r_{i-1}$; $r_0' = 0$;
- For $0 \le i \le 99$, if $i \in RTAPS$, $r_i' = r_i' \oplus FB\_R$;
- If $CB\_R = 1$, For $0 \le i \le 99$, $r_i' = r_i' \oplus r_i$

где r_i' – биты регистра после тактирования, а *RTAPS* представляет собой

вектор константных бит, определённый в [2].

Тактирование нелинейного регистра (*CLOCK_S*(*S, IB_S, CB_S*)) описывается следующим алгоритмом:

- $FB\_S = s_{99} \oplus IB\_S$;
- For $1 \le i \le 98$, $\hat{s}_0 = 0; \hat{s}_{99} = s_{98}$;

$\hat{s}_i = s_{i-1} \oplus \left((s_i \oplus COMP0_i) \wedge (s_{i+1} \oplus COMP1_i) \right)$

- If $CB\_S = 0$, For $0 \le i \le 99$, $s_i^{'} = \hat{s}_i \oplus \left(FB0_i \wedge FB\_S \right)$;
- If $CB\_S = 1$, For $0 \le i \le 99$, $s_i^{'} = \hat{s}_i \oplus \left(FB1_i \wedge FB\_S \right)$,

где $s_i^{'}$ – состояние регистра *S* после тактирования, значения постоянных векторов *FB0*, *FB1*, *COMP0* и *COMP1* представлены в [2].

Ниже представлены этапы тактирование всего шифра (*CLOCK_KG*(*R, S, MIXING, IB*)):

- $CB\_R = s_{34} \oplus r_{67}$;
- $CB\_S = s_{67} \oplus r_{33}$;
- If $MIXING = TRUE$, $IB\_R = IB \oplus$ $\oplus s_{50}$; Else $IB\_R = IB$;
- $IB\_S = IB$;
- *CLOCK_R*(*R*, *IB_R*, *CB_R*);
- *CLOCK_S*(*S*, *IB_S*, *CB_S*).

Инициализация ключа и *IV* происходит в четыре этапа:

а) *R* и *S* регистры инициализируются нулями;

б) For $0 \le i \le |IV| - 1$,

$CLOCK\_KG(R, S, TRUE, iv_i)$;

в) For $0 \le i \le 79$,

$CLOCK\_KG(R, S, TRUE, k_i)$;

г) сто тактов холостого хода при $MIXING = TRUE$ и $IB = 0$,

где iv_i и k_i *i*-е биты вектора инициализации и ключа соответственно, $|IV|$ – длина вектора инициализации в битах.

Таким образом, весь алгоритм можно представить в виде конечного автомата, состояниями которого будут выступать значения *R* и *S* регистров. При данной интерпретации количество возможных состояний составляет 2^{200}. Далее будет показано, что на практике таких состояний гораздо меньше.

2 АЛГОРИМ НАХОЖДЕНИЯ ВЕТВЛЕНИЙ

Как было показано выше, шифр Mickey имеет 3 основные функции для тактирования регистров. Идея нахождения ветвлений основана на переборе всех возможных значений *FB_S*, *IB_S*, *CB_S*, *FB_R*, *IB_R*, *CB_R* и анализе состояний регистра после обратного тактирования.

Предположим, что мы знаем текущее состояние регистров, тогда для нахождения предыдущих состояний необходимо вычислить функции $CLOCK\_S^{-1}$, $CLOCK\_R^{-1}$, $CLOCK\_KG^{-1}$, которые являются обратными к *CLOCK_S*, *CLOCK_R*, *CLOCK_KG* соответственно. На рисунках 1, 2 и 3 представлены алгоритмы этих функций.

Пусть X_1 известное состояние конечного автомата, тогда под веткой понимается *m* состояний вида:

$$X_1 \overset{такт}{\rightarrow} X_2 \overset{такт}{\rightarrow} \ldots \overset{такт}{\rightarrow} X_m,$$

где *такт* – для алгоритма Mickey эквивалентен *CLOKC_KG* ($CLOCK\_KG^{-1}$).

Под циклом понимается *n* состояний следующего вида:

$$X_1 \overset{такт}{\rightarrow} X_2 \overset{такт}{\rightarrow} \ldots \overset{такт}{\rightarrow} X_i \overset{такт}{\rightarrow} \ldots \overset{такт}{\rightarrow} X_{n=i}$$

Из описания алгоритмов обратных функций видно, что в определенных случаях вообще не возможны предыдущие состояния. Таким образом, существуют состояния, ветки или циклы, в которые конечный автомат никогда не перейдёт, при любых значениях *K* и *IV*.

Входные значения: биты IB_S, CB_S, FB_S и регистр S
Выходные значения: TRUE, если ветвление существует, в противном случае FALSE
$S' \leftarrow S$
if FB_S = TRUE **then**
  if CB_S = TRUE **then**
    for i=0 to 99 **do**
      $\hat{s}_i \leftarrow s'_i \oplus FB1_i$;
    end for
  else
    for i=0 to 99 **do**
      $\hat{s}_i \leftarrow s'_i \oplus FB0_i$;
    end for
  end if
else
  for i=0 to 99 **do**
    $\hat{s}_i \leftarrow s'_i$;
  end for
end if
if $\hat{s_0} \neq 0$ **then**
  return FALSE
end if
$s_{99} \leftarrow$ FB_S $\oplus$ IB_S;
$s_{98} \leftarrow \hat{s}_{99}$;
for i=98 to 1 **do**
  $s_{i-1} \leftarrow \hat{s}_i \oplus ((s_i \oplus COMP0_i) \wedge (s_{i+1} \oplus COMP1_i))$;
end for
return TRUE

Рисунок 1 – алгоритм функции $CLOCK\_S^{-1}$

Входные значения: биты IB_R, CB_R, FB_R и регистр R
Выходные значения: TRUE, если ветвление существует, в противном случае FALSE
$R' \leftarrow R$
if CB_R = TRUE **then**
  $r_0 \leftarrow r_0 \oplus$ (FB_R $\wedge$ $RTAPS_0$);
  for i=1 to 99 **do**
    $r_i \leftarrow r_i \oplus r_{i-1} \oplus$ (FB_R $\wedge$ $RTAPS_i$);
  end for
  if $r_{99} \neq$ (FB_R $\oplus$ IB_R) **then**
    return FALSE
  end if
else
  if $(r_0 \oplus$ (FB_R $\wedge$ $RTAPS_0$)) $\neq 0$ **then**
    return FALSE
  end if
  for i=1 to 99 **do**
    $r_{i-1} \leftarrow r_i \oplus$ (FB_R $\wedge$ $RTAPS_i$);
  end for
  $r_{99} \leftarrow$ FB_R $\oplus$ IB_R;
end if
return TRUE

Рисунок 2 – алгоритм функции $CLOCK\_R^{-1}$

Входные значения: регистры R и S
Выходные значения: все возможные ветвления
for $s = 0$ to 64 **do**
  FB_S $\leftarrow (s \gg 0) \wedge 1$;
  IB_S $\leftarrow (s \gg 1) \wedge 1$;
  CB_S $\leftarrow (s \gg 2) \wedge 1$;
  FB_R $\leftarrow (s \gg 3) \wedge 1$;
  IB_R $\leftarrow (s \gg 4) \wedge 1$;
  CB_R $\leftarrow (s \gg 5) \wedge 1$;
  TS $\leftarrow$ S;
  TR $\leftarrow$ R;
  if CLOCK_S^{-1}(TS,IB_S,CB_S,FB_S) $\neq$ TRUE **then**
    continue;
  end if
  if CLOCK_R^{-1}(TR,IB_R,CB_R,FB_R) $\neq$ TRUE **then**
    continue;
  end if
  INPUT_BIT_S $\leftarrow$ IB_S;
  if MIXING = TRUE **then**
    INPUT_BIT_R $\leftarrow$ IB_R $\oplus s_{50}$;
  else
    INPUT_BIT_R $\leftarrow$ IB_R;
  end if
  CONTROL_BIT_S $\leftarrow s_{67} \oplus r_{33}$;
  CONTROL_BIT_R $\leftarrow s_{33} \oplus r_{67}$;
  if CONTROL_BIT_R = CB_R **and** CONTROL_BIT_S = CB_S **and**
  INPUT_BIT_R = IB_R **then**
    branch $\leftarrow$ Append(branch,[S;R;INPUT_BIT_S]);
  end if
end for
return branch

Рисунок 3 – алгоритм функции $CLOCK\_KG^{-1}$

3 РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ

Используя описанные выше алгоритмы обратных функций, можно восстановить все предыдущие состояния, из которых могло получиться текущее состояние регистров *R* и *S*. Зная данную информацию можно посчитать невозможные состояния и количество возможных ветвлений.

В таблице 1 приведены результаты практического поиска ветвлений для шифра Mickey. Результаты были получены для начального состояния, после инициализации 80 тактов *IV* и 20 тактов ключа. В данном случае каждое новое состояние получалось их предыдущего, что соответствует реальному поиску начального состояния.

В таблице 2 представлены результаты для случайных состояний, т.е. каждое новое состояние генерировалось случайным образом.

Таблица 1 – Результаты расчёта количества ветвлений для детерминированного состояния.

Ветвления	Количество	Количество, %
0	3078948	28,664
1	969	0,009
2	4723204	43,971
3	1019	0,009
4	2859210	26,618
6	159	0,001
8	78045	0,727

Таблица 2 – Результаты расчёта количества ветвлений для случайного состояния.

Ветвления	Количество	Количество, %
0	3017259	28,090
1	7402	0,069
2	4710004	43,848
3	6072	0,057
4	2912621	27,115
6	17	0,000
8	88179	0,821

Как видно из таблиц 1 и 2, результаты для случайно сгенерированных значений и взятых для определённого значения *IV* и *K* практически совпадают.

Интересным является тот факт, что отсутствую ветвления 5 и 7. Это может быть связано с внутренней структурой регистров *S* и *R* или маленькой выборке проверяемых значений. Вероятность появлений данных ветвлений приближается к 0, и не будет существенно влиять на результат.

Существует приблизительно 2^{198} состояний, в которые конечный автомат не сможет перейти, из-за того, что приблизительно 28 % состояний не имеют предыдущих.

В таблице 3 приведена зависимость количества состояний конечного автомата от тактов $CLOKC\_KG^{-1}$. Очевидно, что количество возможных состояний приблизительно равно 2^{n+2}. Данные свойство проявляется за счёт того, что ветвления «3» и выше компенсируются за счёт «0» ветвлений.

Таблица 3 – Зависимость количества состояний от тактов обратной функции

Такты	Количество состояний	Количество состояний, $\log_2$
8	1200	10,23
10	4814	12,23
12	18458	14,17
14	83959	16,36
16	337064	18,36
18	1317142	20,33
20	5410768	22,37

Высокие вероятности «2» и «4» ветвлений говорят о том, что возможны различные комбинации *IV* и ключ шифрования, приводящие к одному и тому же состоянию.

ВЫВОДЫ

В работе представлены результаты, которые позволяют на практике анализировать состояния шифра Mickey и выполнять поиск возможных битов ключа или *IV*, при знании текущего состояния.

Авторы алгоритма рекомендуют использовать шифр для генерации 2^{40} бит гаммы на одном векторе инициализации. Такая длина гораздо меньше 2^{199}, что позволяет говорить о том, что шифр Mickey остаётся криптографически стойким.

В дальнейших исследованиях планируется разработка аналитической модели и поиск длин невозможных веток (циклов), что позволит более точно оценить количество допустимых состояний шифра Mickey.

Литература

1. *Babbage S. "The eSTREAM portfolio" [Electronic resource] / S. Babbage, C. Cannire, A. Canteaut, C. Cid, H. Gilbert, T. Johansson, M. Parker, B. Preneel, V. Rijmen, M. Robshaw. Mode of access : WWW.URL: http://www.ecrypt.eu.org/ stream/portfolio.pdf – Last access: 2011. – Title from the screen.*

2. *Babbage S. "The stream cipher MICKEY 2.0" [Electronic resource] / S. Babbage, M. Dodd. Mode of access : WWW.URL: www.ecrypt.eu.org/stream/ p3ciphers/mickey/mickey_p3.pdf – Last access: 2011. – Title from the screen.*

3. *Hong J., Kim W. "TMD-Tradeoff and State Entropy Loss Considerations of Streamcipher MICKEY" [Text] / Hong J., Kim W. Lecture Notes in Computer Science, 2005, Volume 3797/2005. pp. 169-182.*

4. *Babbage S. "The stream cipher MICKEY-128 2.0" [Electronic resource] / S. Babbage, M. Dodd. Mode of access : WWW.URL: http://www.ecrypt.eu.org/ stream/p3ciphers/mickey/mickey128_p3.pdf – Last access: 2011. – Title from the screen.*