ОБЗОР АТАК НА АЛГОРИТМ A5/1

А.Н. Гайдук

Белорусский государственный университет, НИИ ППМИ
пр Независимости 4, Минск, Беларусь
телефон(ы): + (37517) 2095049; e-mail: GaidukAN@bsu.by

Приводится описание и характеристика известных атак на алгоритм A5/1, который используется в GSM сетях для шифрования информации между базовой и мобильной станциями.

Ключевые слова - криптоанализ, GSM, поточные шифры.

1 ОПИСАНИЕ АЛГОРИТМА A5/1

Поточный алгоритм A5 используется в системе GSM для шифрования канала связи между телефоном и базовой станцией. Существует две версии алгоритма A5: более сильная версия A5/1 и более слабая версия A5/2. Хотя описание алгоритмов не было официально опубликовано консорциумом GSM, в 1994 году Андерсен [1] привел описание алгоритма A5/1, которое было уточнено в работе Брисено [5] в 1999 году.

Криптоалгоритм A5/1 состоит из трех линейных регистров сдвига с обратной связью $R_1,\ldots,R_3$ длин $l_1=19, l_2=22, l_3=23$ бит. Обозначим через $g_i(x)$ полином обратной связи для регистра $R_i, i=1,\ldots,3$. В алгоритме A5/1 полиномы обратной связи имеют следующий вид: $g_1(x)=x^{19}+x^5+x^2+x+1$, $g_2(x)=x^{22}+x+1$, $g_3(x)=x^{23}+x^{15}+x^2+x+1$. Для управления движением в алгоритме A5/1 используется majority функция $F(x_1,x_2,x_3)=x_1x_2+x_1x_3+x_2x_3$. Регистр $R_i, i=1,\ldots,3$ алгоритма сдвигается в случае, если бит съема данного регистра $x_i, i=1,\ldots,3$ совпадает со значением функции $F(x_1,x_2,x_3)$, где x_1,x_2,x_3 значения битов съема каждого из регистров в текущий момент времени.

В GSM системах в качестве единицы передаваемой информации используется фрейм. Каждый фрейм состоит из 228 бит. Процедура инициализации алгоритма A5/1 запускается для каждого шифруемого фрейма. Входными параметрами функции инициализации является секретный ключ $K = (k_1, k_2, \ldots k_{64})$ (64 бита) и номер фрейма $F_n = (f_1, f_2, \ldots f_{22})$ (22 бита). Функция инициализации состоит из следующих 4 шагов:

1- Установка регистров в нулевое состояние: Биты регистров $R_i, i=1,\ldots,3$ устанавливаются равными 0. Алгоритм переходит в состоянии S^0.

2- Загрузка ключа K_C: Регистры $R_i, i=1,\ldots,3$ сдвигаются 64 раз и биты ключа k_j последовательно складываются по модулю 2 (операция XOR) с нулевым битом $R_i[0], i=1,\ldots,3$ каждого из регистров. Алгоритм переходит в состоянии S^{64}.

3- Загрузка номера фрейма F_n: Регистры $R_i, i=1,\ldots,3$ сдвигаются 22 раза и биты номера фрейма f_j последовательно складываются по модулю 2 (операция XOR) с нулевым битом $R_i[0], i=1,\ldots,3$ каждого из регистров. Алгоритм переходит в состоянии S^{86}.

4- 100 холостых тактов: Алгоритм выполняет 100 холостых тактов. Выходные биты игнорируются. Алгоритм переходит в состоянии S^{186}.

После функции инициализации алгоритм A5/1 выполняет 228 тактов, выходные биты складываются по модулю 2 (операция XOR) с битами данных фрейма F_n. В GSM каждый фрейм передается через 4.6 миллисекунды [4].

2 ИЗВЕСТНЫЕ АТАКИ

Большинство атак на алгоритм A5/1 являются атаками открытого текста (plaintext attack) и различаются по следующим параметрам:

- Количество бит выходной последовательности алгоритма A5/1 доступных криптоаналитику. Может быть выражено в битах или секундах разговора.
- Вычислительная сложность подготовительной стадии.
- Необходимый объем памяти для выполнения атаки.
- Вычислительная сложность атаки.

В обзоре рассматриваются следующие атаки: атака Голича, атака Бирюкова, атака Келлера, атака Порнина, атака Бихэма, атака Эдхаля, атака Максимова.

Атака Голича. Голич в работе [7] предложил две атаки. Одна из атак основана на соотношении время-память, другая является атакой разделяй и властвуй.

Атака разделяй и властвуй (Divide-And-Conquer attack). Для атаки требуется 64 бита выходной последовательности алгоритма. Атака разделяй и властвуй определяет состояние S^{86} после загрузки номера фрейма.

Для каждого регистра перебирается $n \le 18$ бит управляющих движением регистров. На основе $3n$ бит составляется $3n$ линейно независимых уравнений относительно состояния S^{186}. В среднем $3n$ бит формируют $4n/3$ значений управляющей последовательности. Поэтому можно составить $1+4n/3$ дополнительных уравнений, где первое уравнение получается из первого бита выходной последовательности перед сдвигом алгоритма. Если n выбрано равным 10, тогда можно составить $1+3n+4n/3=44.3$ линейно независимых уравнений. Для составления остальных уравнений используется метод поиска по дереву. Каждая вершина в дереве содержит биты управляющие движением регистров. Среднее число ветвей для каждой вершины найдено равным 2.5. Глубина дерева должна быть $4m/3$, где $m \approx 19.02/3$ число неизвестных бит для каждого регистра. Поэтому обход дерева требует $2.5^{4m/3} \approx 2^{11.16}$ операций. Сложность первого этапа таким образом равна $2^{30+11.16} = 2^{41.16}$, или $2^{40.16}$ в среднем.

Следующий этап заключается в определении состояния S^{86} из состояния S^{186}. Для этого перебираются значения количества сдвигов для каждого из регистров необходимое для достижения состояния S_i^{186} из состояния S_i^{86}, где i это номер регистра сдвига. Количество сдвигов в среднем равно $4*101/3 \approx 76$. В каждом случае S_i^{86} вычисляется из S_i^{186} и проверяется на корректность путем старта алгоритма из этого начального состояния и генерации 100 выходных значений.

Атака основанная на соотношении время-память(Time-Memory Tradeoff attack).

Целью атаки является определение состояния S^{86} алгоритма для любой выходной последовательности длины 64 бит. Атака применима в случае выполнения соотношения $T*M \ge 2^{63.32}$, где T и M вычислительная сложность и объем памяти соответственно.

Предполагается, что криптоаналитику доступно $K \le 2^{22}$ фреймов выходной последовательности. Известная 228-битная выходная последовательность каждого фрейма разбивается на две отдельные последовательности каждая по 114 бит, которые в свою очередь разбиваются на 51 различных 64-битных блоков. Поэтому 228 битам выходной последовательности соответствует 102 64-битных блока.

Идея заключается в формирования таблицы с M значениями состояния алгоритма и соответствующих им 64-битных блоков выходной последовательности. Состояния выбираются равномерно и одинаково распределенными из множества состояний. Таблица сортируется по выходным 64-битным блокам. Результатом поиска в таблице по 102 64-битным блокам являются состояния, которые дают данную выходную последовательность. Пересечение между $102*K$ известными блоками и M выходными блоками происходит, если $102*K*M \ge 2^{63\,32}$. Время необходимое для определения общего блока равно $T = 102*K*log(M) \approx 102*K$. Поэтому атака основанная на соотношении время-память возможна, если $T*M \ge 2^{63.32}$ и требуется только один просмотр таблицы для определения состояния. Требуемое время и память необходимая для атаки определяются следующими 2 примерами:

Если $K=15$, тогда $T = 102*2^{15} = 2^{21.67}$ и $M = 2^{41.65} = 55210 Gbytes$.

Если $K=21$, тогда $T = 102*2^{21} = 2^{27.67}$ и $M = 2^{35.65} = 862 Gbytes$.

Следующий этап состоит в определении S^{86} из S^{186}.

После того, как получено S^{86} либо с помощью первой либо второй атаки, последний этап заключается в определении секретного ключа из S^{86}.

Достоинства.

Для атаки разделяй и властвуй требуется только 64 бита.

Для атаки разделяй и властвуй не требуется подготовительная стадия и дополнительная память.

Недостатки.

Используемое описание алгоритма A5/1 не является полностью правильным [5].

В атаке разделяй и властвуй каждая операция основана на решении систем линейных уравнений, т.е. вычислительная сложность $2^{40.16}$ решения линейных систем эквивалентна вычислительной сложности в 2^{47} операций равных одному такту работы алгоритма A5/1 [3].

Атака основанная на соотношении время память требует около 10 часов известной выходной последовательности.

Атака Бирюкова.

В работе [4] рассматриваются две атаки на алгоритм A5/1 (*biased birthday attack* и *the random subgraph attack*). Обе атаки являются модификациями атаки время-память предложенной Голичем. Для каждой из них требуется подготовительная стадия заключающаяся в создании таблицы большого объема содержащей состояния алгоритма A5/1 и выходные последовательности соответствующие данным состояниям. В таблице хранятся только такие состояния, которым соответствует выходная последова-

тельность имеющая определенный шаблон префикса α длины K. В качестве такого префикса авторы работы выбрали следующий шаблон 1000000000000000 (единица и 15 нулей). Количество состояний, которые могут сгенерировать α приблизительно равно $2^{64} \cdot 2^{-16} = 2^{48}$. Авторы предлагают специальный методы для эффективного хранения состояний и префикса выходной последовательности в таблице (префикс, состояние) упорядоченной по префиксу.

Общий этап для атак

В обеих атаках ищется коллизия между множеством состояний через которые проходит алгоритм A5/1 при генерации бит выходной последовательности и множеством состояний сохраненными в таблице. Для каждого фрейма (длины 228 бит) выходной последовательности осуществляется поиск известного префикса. Если префикс найден, тогда осуществляется поиск следующих 35 бит выходной последовательности в таблице (префикс, состояние). Результатом поиска являются состояния в интервале от S^{186} до S^{363}.

Следующий шаг заключается в определении состояния S^{86} алгоритма A5/1. Для этого для найденного состояния $S^i, i \in \overline{186,363}$ вычисляются возможные предыдущие состояния и отбрасываются те из них, которые не приводят в S^i. После определения состояния S^{86} восстанавливается ключ.

Достоинства.

1. Вычислительная сложность обеих атак, позволяет отнести их к классу атак реального времени.
2. Для обеих атак требуется сравнительно небольшое количество ($2^{20.5}$) бит выходной последовательности алгоритма A5/1.

Недостатки.

Подготовительная стадия требует больших вычислительных затрат (2^{48}) и большой объем памяти.

Атака Келлера (Hardware-based attack).

Атака предложенная Келлером в работе [8] опирается на идеи изложенные в предыдущих атаках. Для атаки требуется 64 бита выходной последовательности алгоритма A5/1. Атака выполняется в три этапа.

Первый этап заключается в определении множества состояний S^{186}, которым соответствует известная выходная последовательность длины 64 бита. Эта трудоемкая часть атаки была реализована на FPGA. Идея этого этапа заключается в переборе 19 бит регистра R_1, и 22 бит регистра R_2 (всего 2^{41} бит) и определении состояний регистра R_3 по известной выходной последовательности длины 64 бита. После того, как восстановлено одно из состояний S^{186} алгоритма A5/1, генерируется выходная последовательность из этого состояния и сравнивается с известной выходной последовательностью.

Автор предложил реализовать этот этап на 1000 платах ASIC с 0.1 μ технологией. Перебор 2^{41} бит занимает 70 секунд. В среднем требуется около 35 секунд, что позволяет отнести эту атаку к атакам реального времени. Стоимость оборудования на момент написания статьи составляла от 1 до 2 миллионов долларов США.

На втором этапе восстанавливается состояние S^{86} перед 100 холостыми тактами.

Третий этап заключается в определении S^{64} из S^{86}.

Достоинства

Для атаки требуется только 64 бита выходной последовательности алгоритма.

Распределенные вычисления на 1000 интегральных схемах ASIC восстанавливают ключ менее чем за минуту.

Для атаки не требуется подготовительная стадия

Недостатки

Высокая стоимость оборудования.

Атака Порнина (Software-Hardware Trade-offs).

Атака предложенная в работе [10] основана на разделении вычислительных ресурсов между обычным персональным компьютером и быстрым аппаратном комплексом.

Программная версия атаки является модифицированной версией предложенной Голичем атаки разделяй и властвуй. Она основана на переборе управляющей последовательности для заданного числа тактов. Для каждого выходного бита составляется линейное уравнение относительно состояния алгоритма A5/1. Также составляется два уравнения относительно битов управляющих движением регистров. Данные уравнения задают СЛАУ относительно неизвестного состояния алгоритма A5/1. Число тактов управляющей последовательности необходимое для получения системы полного ранга (64) может варьироваться в зависимости от движения регистров по циклу внутренних состояний., Авторы предлагают использовать оценку для данного числа тактов равной 64/3. После того, как получена система линейных уравнений полного ранга авторы предлагают проверять полученную систему путем добавления к ней новых уравнений для следующих тактов, что приведет ее к несовместности в случае если управляющая последовательность была выбрана не верно. Если же все добавляемые уравнения будут линейно выражаться через данную систему, то управляющая последовательность восстановлена правильно и из системы восстанавливается значение состояния S^{186} алгоритма A5/1. Сложность атаки равна $2^{45.3}$ операций исключений одного уравнения. Время необходимое для программной реализации равно 400 дням на компьютере Compaq XP-1000 (21264 Alpha процессор с частотой 500 MHz).

Аппаратная версия атаки выполнена на PCI плате Pamette содержащей элементы Xilinx 4010E FPGA.

Характеристики данной платы следующие:

- Для каждого такта выполняется, 1 такт работы алгоритма A5/1.

- Реализована возможность загрузки новых состояний в регистры за один такт.

- Карта может работать на частоте 50 MHz.

- На карте могут быть размещены до 48 версий реализаций алгоритма A5/1.

За одну секунду на карте Pamette может быть проверено до 37 миллионов состояний алгоритма A5/1. Для атаки полного перебора потребуется 15 800 лет вычислений на одной карте Pamette.

Программно-аппаратная версия атаки использует достоинства программной и аппаратной версий атаки и авторами предлагается использовать две платы Pamette для одного ПК, в этом случае среднее время атаки равно 2.5 дням.

Достоинства.

Для атаки требуется 64 бита выходной последовательности алгоритма A5/1.

Недостатки.

Вычислительная сложность атаки высокая и данная атака не является атакой реального времени.

Атака Бихэма (Cryptanalysis of the A5/1).

Главная идея атаки предложенной в работе [3] заключается в ожидании специального события, которое состоит в том, что на протяжении 10 последовательных тактах работы алгоритма A5/1 регистр R_3 простаивает, в то время как регистры R_1 и R_2 сдвигаются.

Сложность атаки равна 2^{27}, если известно положения начиная с которого регистр R_3 простаивает. Поскольку данное положение неизвестно, требуется проверить 2^{20} таких положений. Поэтому для атаки требуется 2^{20} бит выходной последовательности и вычислительная сложность атаки равна 2^{47}.

Авторы предложили использовать таблицы хранимые на диске для уменьшения вычислительной сложности атаки. Вычислительная сложность модифицированная версия атаки составляет $2^{39.91}$ операций равных 1 такту работы алгоритма A5/1. Однако подготовительная стадия составляет 2^{37} операций равных 1 такту работы алгоритма A5/1 и объем памяти необходимый для хранения таблиц равен 32Gbyte.

Достоинства.

Относительно небольшой объем памяти для выполнения атаки (32Gbyte).

Вычислительная сложность подготовительной стадии относительно небольшая 2^{37} операций равных 1 такту работы алгоритма A5/1.

Недостатки.

Атака основана на специальном событии, которое как предполагают авторы может произойти среди 2^{20} бит выходной последовательности алгоритма A5/1.

Атака Эдхаля (Correlation attack).

Эдхаль и Йохансен в работе [6] предложили корреляционную атаку. Атака основана на слабости функции инициализации, поскольку загрузка ключа и номера фрейма производится линейным образом. Введем обозначения:

- u_t^i: выходной бит в момент времени t регистра $R_i, i = 1,...,3$ начиная из состояния S^{186}.

- Z: Выходная последовательность алгоритма A5/1, где $Z = (z_1, z_2, \ldots z_{228})$.

Состояния S^{186} является линейной функцией от ключа K и номера фрейма F_n. Выходной бит в момент времени t регистра $R_i, i = 1,...,3$ равен $u_t^i = \sum_{j=1}^{64} c_{jt}^i k_j + \sum_{j=1}^{22} d_{jt}^i f_{ji}$, где c_{jt}^i и d_{jt}^i некоторые константы. Пусть $s_t^i = \sum_{j=1}^{64} c_{jt}^i k_j$ and $\hat{f}_t^i = \sum_{j=1}^{22} d_{jt}^i f_j$. Тогда, $u_t^i = s_t^i + \hat{f}_t^i$, где s_t^i линейная комбинация битов ключа и $\hat{f}_t^i$ линейная комбинация битов номера фрейма. Обозначим сдвиг регистров $R_i, i = 1,...,3$ на заданное число шагов через тройку (cl_1, cl_2, cl_3), тогда $u_{cl1}^1 + u_{cl2}^2 + u_{cl3}^3 = z_k$ для $k \in \overline{1,228}$. Подставляя значения $u_t^i = s_t^i + \hat{f}_t^i$, получим:

$$s_{cl1}^1 + s_{cl2}^2 + s_{cl3}^3 = \hat{f}_{cl1}^1 + \hat{f}_{cl2}^2 + \hat{f}_{cl3}^3 + z_k \quad (1)$$

Обозначим правую часть уравнения (1) через $O_{(cl1,cl2,cl3,k)}^{Fn}$, где F_n это номер фрейма. Пусть $P_{(cl1,cl2,cl3,k)}^{Fn}$ вероятность того, что 3 регистра сдвинутся на (cl_1, cl_2, cl_3) за $100 + k$ тактов. Согласно [6] уравнение (1) будет выполняться со следующей вероятностью: $P(s_{cl1}^1 + s_{cl2}^2 + s_{cl3}^3 = O_{(cl1,cl2,cl3,k)}^{Fn}) = ½ + ½P_{(cl1,cl2,cl3,k)}^{Fn}$.

Для сокращения необходимого числа фреймов авторы предложили следующий метод: пусть E(k) это событие того, что сдвиг регистров (cl_1, cl_2, cl_3) произошел в $k \in I$ позиции, где I интервал для которого $P(E(k)) > 0$. Тогда согласно [6]:

$$P_{(cl1,cl2,cl3)}^{Fn} = \sum_{k \in I} P(E(k)).[O_{(cl1,cl2,cl3,k)}^{Fn} = 0] +$$

$$+ ½.(1 - \sum_{k \in I} P(E(k)))$$

где $[O^{F_n}_{(cl1,cl2,cl3,k)} = 0]$ индикатор события: равен 1, если $O^{F_n}_{(cl1,cl2,cl3,k)} = 0$, 0 иначе. Если атакующему доступно m фреймов выходной последовательности, тогда вводя логарифмическое отношение правдоподобия $\Lambda_{cl1,cl2,cl3} = \sum_{n=0}^{m} \log_2\left(\frac{P^{F_n}_{(cl1,cl2,cl3)}}{1 - P^{F_n}_{(cl1,cl2,cl3)}}\right)$, возможно получить оценку для битов ключа:

$$s^1_{cl1} + s^2_{cl2} + s^3_{cl3} = \begin{cases} 0, \Lambda_{cl1,cl2,cl3} \geq 0 \\ 1, \Lambda_{cl1,cl2,cl3} < 0 \end{cases}.$$

В работе [6] авторы исследовали разные стратегии для эффективного восстановления битов ключа.

Достоинства.

Для атаки требуется подготовительная стадия вычислительная сложность которой оценивается в 15 минут работы на ПК с процессором Intel Pentium 4, 1.8 Ghz и 512 MByte оперативной памяти.

Объем памяти для атаки маленький (2Mb).

Сложность атаки не зависит от длины регистров сдвига.

Вычислительная сложность атаки оценивается в 5 минут.

Недостатки.

Для атаки требуется большое количество (2^{24}) бит выходной последовательности алгоритма A5/1.

Атака Максимова (An improved correlation attack).

Максимов, Йохансен и Бабаж в работе [9] предложили улучшенную корреляционную атаку основанную на работе [6]. Было сделано два предположения, которые позволили уменьшить необходимое для атаки количество фреймов.

1. Предположение относительно сдвига (cl_1,cl_2,t): для фрейма с номером F_n регистры R_1 и R_2 сдвинулись cl_1 и cl_2 раз за t тактов работы алгоритма.

2. Предположение относительно t+1 такта: для фрейма с номером F_n регистры R_1 и R_2 сдвинулись в момент t+1 такта, а регистр R_3 простаивал.

Согласно этим двум предположениям с регистра R_3 за такт t и такт t+1 снимается один и тот же бит для формирования выходной последовательности. Поэтому можно исключить бит регистра R_3 и вычислить смещение для уравнения относительно битов ключа:

$$P(s^1_{cl_1} + s^2_{cl_2} + s^1_{cl_1+1} + s^2_{cl_2+1} = \\ = O^{Fn}_{(cl_1,cl_2,cl_3,t)} + O^{Fn}_{(cl_1+1,cl_2+1,cl_3,t)}) = 1/2 + 1/2\,1/4\,P^{Fn}_{(cl1,cl2)},$$

где $P^{Fn}_{(cl1,cl2)}$ вероятность сдвига регистров R_1 и R_2 на cl_1 и cl_2 шагов. Значение $1/8\,P^{Fn}_{(cl1,cl2)}$ в два-три раза выше, чем $1/2\,P^{Fn}_{(cl1,cl2,cl3)}$ указанное в работе [6]. Данная модификация позволила сократить в 4-10 раз количество необходимых фреймов для атаки.

Достоинства.

Вычислительная сложность атаки оценивается в 10 минут работы на обычном ПК.

Недостатки.

Для атаки требуется большое количество (2^{21}) бит выходной последовательности алгоритма A5/1.

3 ЗАКЛЮЧЕНИЕ

В статье рассмотрены 7 различных атак на алгоритм A5/1. Для каждой из атаки указано необходимое количество бит выходной последовательности алгоритма A5/1, вычислительная сложность подготовительной стадии, необходимый объем памяти для выполнения атаки, вычислительная сложность атаки. В таблице 1 приведены основные характеристики каждой из атак.

В обзоре указано, что часть атак основаны только на программной реализации (software-only) в то время как другие являются аппаратно-программными атаками. Некоторые атаки основаны на соотношении время-память, в то время как другие являются алгебраическими (основаны на решении систем линейных уравнений) или корреляционными.

Количество бит выходной последовательности алгоритма A5/1 является решающим параметром для сравнительного анализа атак. Сравнительный анализ показывает, что часть атак не осуществимы на практике, так как требуют большой объем открытого текста, большой объем памяти или больших вычислительных ресурсов.

ТАБЛИЦА 1

ХАРАКТЕРИСТИКА АТАК

Атака	Кол-во бит выход. послед	Сложность подготовит. стадии	Объем памяти	Сложность атаки
01[Golic] Divide-And-Conquer	64	0	0	2^{47}
02[Golic] Time-Memory Tradeoff	$2^{28.8}$	$2^{35.65}$	862 Gb	$2^{27.65}$
03[Biryukov] Biased Birthday	$2^{20.5}$	2^{42}	292 Gb	1 секунда ($\cong$ 0)*
04[Biryukov] Random Subgraph	$2^{14.7}$	2^{48}	146 Gb	неск. минут ($\cong$ 0)*

05[Keller] Hardware-Based	64	0	0	1 минута ($\cong$ 0)*
06[Pornin] Software-Hardware Trade-offs	64	0	0	5 дней ($\cong 2^{42}$)*
07[Biham] Cryptanalysis of A5/1	2^{20}	2^{38}	32 GB	$2^{39.91}$
08[Ekdahl] Correlation Attack	2^{24}	15 минут	2 MB	5 минут
09[Maximov] An Improved Correlation attack	2^{21}	$2^{37.6}$	2 MB	10 минут

* Значение в скобках является приближенной величиной относительно значений для других атак.

ЛИТЕРАТУРА

[1] Anderson, R. A5(Was: HACKING DIGITAL PHONES)/ R. Anderson// Usenet communication on sci.crypt, alt.security and uk.telecom. — 1994.

[2] Barkan, E. Instant Ciphertext Only Cryptanalysis of GSM Encrypted Communication/ E. Barkan, E. Biham, N. Keller.// In Proc. Of Crypto 2003. — 2003. P. 600–616.

[3] Biham, E. Cryptanalysis of the A5/1 GSM Stream Cipher/ E. Biham, O. Dunkelman.// In Proc. Of Indocrypt 2000. — 2000. P. 43–51.

[4] Biryukov, A. Real Time Cryptanalysis of A5/1 on A PC/ A. Biryukov, A. Shamir, D. Wagner.// Lecture Notes in Computer Science 1978, in Proc. of FSE2000. — 2000. P.. 1-18,2000.

[5] Briceno, M. A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms/ M. Briceno, I. Goldberg, D. Wagner. // [Electronic resource]. — Mode of access: http://www.gsm-security.net/papers/a51.shtml. — Date of access: 09.10.2009.

[6] Ekdahl, P. Another attack on A5/1/ P. Ekdahl, T Johansson. // IEEE Transactions on Information Theory. — 2003. —Vol. 49. P. 284–289.

[7] Golic, J. Cryptanalysis of Alleged A5 Stream Cipher/ J. Golic. // Lecture Notes in Computer Science 1233, Advances in Cryptology, in Proc. of EUROCRYPT'97, — 1997. P. 239-255.

[8] Keller, J. Hardware-Based Attack on the A5/1 Stream Cipher/ J. Keller, B. Seitz. // [Electronic resource]. — Mode of access: http://pv.fernuni-hagen.de/docs/apc2001-final.pdf. — Date of access: 08.10.2009.

[9] Maximov, E. An Improved Correlation Attack on A5/1/ E. Maximov, T. Johansson, S. Babbage S. // [Electronic resource]. — Mode of access: http://it.lth.se/movax/Publications/2004/AttA51.pdf. — Date of access: 09.10.2009.

[10] Pornin, T. Software-hardware Trade-offs: Application to A5/1 Cryptanalysis./ T. Pornin, J. Stern// In Proc. of CHES'00. — 2000. P. 318–327.