УДК 681.0.245

СЛАЙДОВАЯ АТАКА НА КРИПТОГРАФИЧЕСКИЙ АЛГОРИТМ МАГМА И ЕЁ РЕАЛИЗАЦИЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ВЫЧИСЛЕНИЯ NVIDIACUDA

Ищукова Е.А., Богданов К.И., Бабенко Л.К.

Южный федеральный университет, Таганрог, e-mail: uaishukova@sfedu.ru

Данная статья посвящена общему анализу одного из методов проверки стойкости криптографических алгоритмов – слайдовой атаке, а также его применимости к новому криптографическому алгоритму Магма. Проанализировано выполнение поиска слайдовой пары с использованием технологии параллельного вычисления NVIDIACUDA для разных комбинаций ключа (для одинаковых подключей, для двух повторяющихся подключей, для четырех повторяющихся подключей). В статье имеются схемы основных элементов алгоритма поиска, приведены примеры отдельных ключевых участков кода. Кроме того, освещен временной аспект выполнения данной операции, сравнительный анализ и зависимость скорости выполнения кода от различных комбинаций элементов ключа и содержания вариантов слайдовой пары. Показано влияние количества выполняемых параллельно вычислений на общее время выполнения алгоритма поиска.

Ключевые слова: криптография, блочный шифр, параллельные вычисления, Cuda, Магма, ГОСТ 29147-89, фиксированные блоки замены, NVIDIA, GPGPU, слайдовая атака, сдвиговая атака, слайдовая пара

SLIDE ATTACK ON MAGMA CRYPTOGRAPHIC ALGORITHM AND ITS IMPLEMENTATION USING TECHNOLOGIES OF PARALLEL COMPUTING NVIDIA CUDA

Ischukova E.A., Bogdanov K.I., Babenko L.K.

Southern Federal University, Taganrog, e-mail: uaishukova@sfedu.ru

This article is focused on the overall analysis methods of testing the resistance of cryptographic algorithms – slide attack, as well as its applicability to a new cryptographic algorithm Magma. Assess the slide pair search technology with NVIDIA CUDA parallel computing for different key combinations (for the same subkey, for subkey repeated two to four repetitive connected). In the article are basic circuit elements of the search algorithm, are examples of some of the key sections of code. In addition, the illuminated temporal aspect of this operation, the comparative analysis of the dependence of the speed of execution of code from different combinations of the key elements and content options for slide couples. The influence of the number of simultaneously running calculations on the total execution time of the algorithm is reviewed here.

Keywords: cryptography, block cipher, parallel computing, Cuda, Magma, GOST 29147-89, fixed replacement blocks, NVIDIA, GPGPU, slide attack, shear attack, slide couple

В течение всей истории существования криптографии самым важным свойством шифра являлась и является до сих пор его стойкость – возможность алгоритма противостоять атакам. Чем больше времени и ресурсов требуется для взлома шифра, тем большей стойкостью он обладает. Существует несколько способов анализа стойкости алгоритма, одним из которых является слайдовая (сдвиговая) атака [4].

Слайдовая атака – криптографическая атака на основе подобранного открытого текста, которая позволяет проводить криптоанализ многораундовых блочных шифров [9]. Впервые такой вид атак был предложен Алексом Бирюковым и Дэвидом Вагнером в 1999 году.

Основой слайдовой атаки являются два понятия:

1. Идентичность раундов шифрования – для каждого раунда функция F() одинаковая;
2. Возможность нахождения ключа – зная текст на входе и на выходе любого раунда, можно найти K, зная P и F(P).

Простейшая слайдовая атака представлена на рис. 1.

Идея заключается в сопоставлении двух процессов шифрования между собой таким образом, что один из них будет отставать от другого на один раунд. После нахождения слайдовой пары можно вычислить некоторые биты подключа. Для нахождения оставшихся битов необходимо найти другу слайдовую пару и провести анализ уже с ее помощью. В результате нахождения нескольких слайдовых пар возможно нахождение всех битов ключа [1].

Рассмотрим, насколько применима слайдовая атака для алгоритма Магма.

Магма представляет собой симметричный блочный шифр, основанный на ГОСТ 29147-89, для которого зафиксированы блоки замены, с размером блока данных 64 бита, секретным ключом 256 бит и 32 раундами шифрования [6]. Магма основана на сети Фейстеля, модифицируется только одна половина сообщения, что позволяет сравнить левую часть P и правую часть P’.

Рассмотрим алгоритм слайдовой атаки для различных комбинаций ключа.

В алгоритме Магма, так же, как и в стандарте ГОСТ 28147-89, не предусмотрена функция выработки подключей, что позволяет ослабить алгоритм, апеллируя определенными значениями ключа. Также не стоит забывать, что последние 8 элементов ключа инвертируются, но в данных примерах этот факт опускается.

Для начала рассмотрим случай, когда все подключи имеют **одинаковое** значение. Т.к. объем одного элемента ключа составляет 32 бита, то существует 2^{32} вариант ключа (из всего ключевого пространства 2^{256}, для которых все раундовые подключи будут иметь одинаковые значения. Опираясь на Парадокс Дней Рождений (ПДР), который гласит, что достаточно перебрать $2^{N/2}$ значений тестов для получения слайдовой пары с вероятностью 0,5, нам достаточно будет опробовать 2^{16} различных вариантов. Так как алгоритм Магма построен по схеме Фейстеля, то в соответствии с рис. 1, правая половина текста P (32 бита) будет совпадать с левой половиной текста P'.

Процесс шифрования будет выглядеть аналогично представленному на рис. 1. Пусть входные тексты P и P' будут представлены в виде двух половин текстов (PL, PR) и (PL1, PR1) соответственно, а выходные тексты E и E1 – из (EL, ER) и (EL1, ER1). В этом случае для нахождения слайдовой пары достаточно, чтобы выполнялось тождество PL = PR1.

Рассмотрим, как будет выглядеть поиск слайдовой пары при наличии **двух подключей,** которые циклически повторяются, то есть

$$K1 = K3 = K5 = K7 \text{ и } K2 = K4 = K6 = K8. \quad (1)$$

Всего существует 2^{64} варианта исходного секретного ключа из всего ключевого пространства 2^{256}, для которых будут выполняться равенства (1). В качестве допущения примем, что раундовые подключи не меняют порядок следования в последних раундах шифрования. В отличие от первого рассмотренного случая, в этом случае поиск необходимо вести по всему диапазону возможных входных значений в алгоритм шифрования. В соответствии с ПДР, можно ожидать, что слайдовая пара будет найдена при переборе всего 2^{32} вариантов открытых текстов.

На рис. 2 предоставлена схема сопоставления раундов шифрования с запаздыванием на 2 раунда.

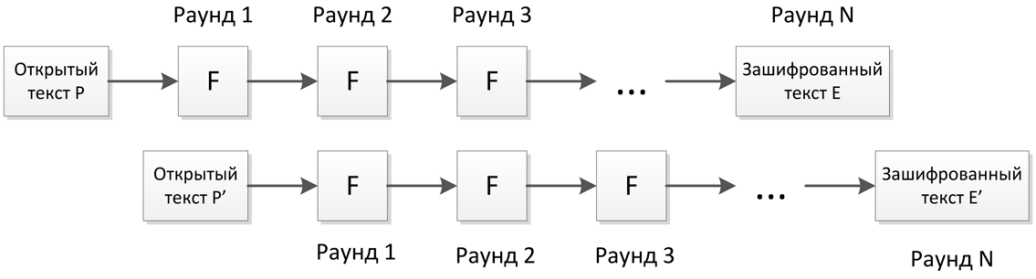

Рис. 1. Простейшая реализация слайдовой атаки

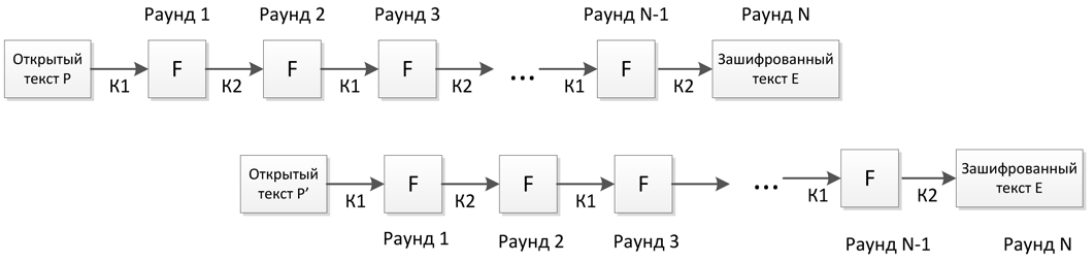

Рис. 2. Схема шифрования с запаздыванием на два раунда

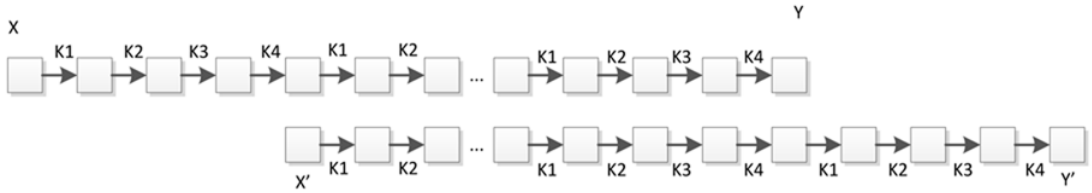

Рис. 3. Процесс зашифрования со сдвигом на 4 раунда

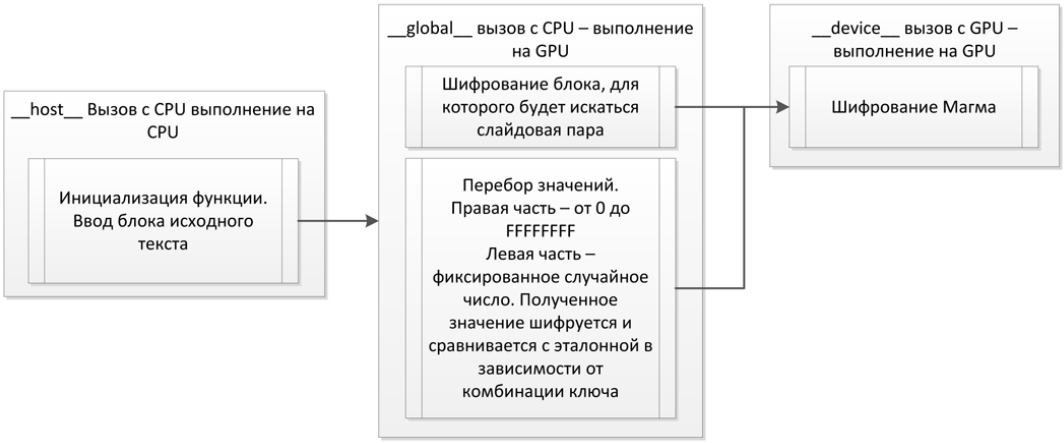

Рис. 4. Схема работы программы поиска слайдовой пары

Обозначим левую и правую часть первого открытого текста PL и PR соответственно, второго PL1 и PR2 соответственно. Сопоставив схему с рис. 2 и схему Фейстеля, получим следующие равенства:

$$PR \oplus PL1 = F(PR, K1);$$

$$PL1 \oplus PR = F(PR1, K2).$$

Аналогично определяется связь шифртекстов EL, ER и EL1, ER2:

$$EL1 \oplus ER = F(ER1, K2);$$

$$EL \oplus ER1 = F(ER, K1);$$

В заключении рассмотрим случай, когда исходный секретный ключ будет состоять из двух равных половин по 128 бит. Сделаем допущение о том, что раундовые подключи не меняют порядок следования в последних раундах шифрования. В этом случае будет четыре уникальных раундовых подключа, которые циклически повторяются. То есть будут выполняться следующие равенства:

$$K1 = K5,\ K2 = K6,\ K3 = K7,\ K4 = K8.$$

В данном случае полный перебор ключевого пространства будет составлять 2^{128} опробования. А для ПДР это количество будет равно 2^{65}, но с вероятностью нахождения слайдовой пары 0,5.

Сопоставим процессы зашифрования с запаздыванием на 4 раунда, как показано на рис. 3.

Связи между исходными и шифртекстами для первого (PL, PR, EL, ER) и второго (PL1, PR1, EL1, ER1) процессов шифрования будут выглядеть следующим образом:

$$PL \oplus F(PR, K1) \oplus PL1 =$$

$$= F((PR1 \oplus F(PL1, K4)), K3);$$

$$ER \oplus F(EL, K1) \oplus ER1 =$$

$$= F((EL1 \oplus F(ER1, K4)), K3).$$

В ходе данного исследования была разработана программа для поиска слайдовых пар с использованием технологии параллельных вычислений NVIDIACUDA [2, 3, 6]. За её основу была взята более ранняя программа «Шифрование криптографическим алгоритмом Магма с использованием технологии параллельных вычислений NVIDIACUDA». В новой программе считается, что для анализа мы используем модель Черного ящика. То есть данные шифруются одним и тем де ключом. Наша задача заключается в нахождении этого ключа. Мы можем подавать на вход открытые тексты и

считывать шифрованные. Анализируя их, найти слайдовую пару и сделать предположение о ключе. На рис. 4 представлена общая схема работы программы.

Разберем подробнее схему, предоставленную на рис. 4. В функции main задается эталонный (для которого будем искать слайдовые пары) блок шифрования, выделяется память на GPU и вызывается функция с приставкой __global__ , обозначающая, что функция вызывается на CPU, а ее вычисления происходят на GPU. Внутри данной функции на нулевом потоке производится непосредственно шифрование эталонного блока с использованием функции, разработанной в более ранней программе, с той лишь разницей, что теперь функция имеет приставку __device__, которая обозначает, что функция вызывается и выполняется только на GPU. Поиск слайдовой пары основан на переборе требуемого набора значений. Левая часть блока заполняется фиксированным псевдослучайным числом, которое выработано на основе функции времени, а правая часть непосредственно перебирается. Затем эталонный шифр-тест сравнивается с перебираемым на основе требуемых условий.

Рассмотрим вышеописанный алгоритм на примерах из кода [8]. Память на GPU выделяется следующим образом:

```
intcnt = 0; //Host переменная
int *dev_cnt; //Device переменная
cudaMalloc( (void**)&dev_cnt, sizeof(int) ); // выделение памяти для переменной на device
cudaMemcpy(dev_cnt, &cnt, sizeof(int), cudaMemcpyHostToDevice); //копирование значения переменной с host на device
```

Выработка псевдослучайного числа происходит следующим образом:

```
srand ( time(NULL) ); //создание метки времени
intrandint = rand() % N; //получение случайного значение от 0 до N, где N = 2147483647 (2^31 – 1)
```

Непосредственный вызов __global__ функции выглядит довольно просто:

```
pair_search<<<1, 512>>>(dev_block, dev_cnt, dev_randint, dev_ar); // аргументами функции выступают эталонный блок, переменная для получения количества слайдовых пар, фиксированное псевдослучайное число, массив для получения найденных слайдовых пар.
```

В <<<>>> скобках представлены числа, обозначающие количества блоков и количества потоков. 512 выбрано в качестве оптимального числа потоков.

Теперь рассмотрим внутренность функции pair_search:

```
intLE = *EL>> 32; //получение LE и RE для шифрования эталонной пары
int RE = *EL& 0xffffffff;
int et = 0;
if ( 0 == threadIdx.x) //шифрование эталонного блока происходит на 0 потоке
{
et= encrypt(LE, RE, 0);
}
int it = 0
for (inti = 0; i < 65536 / blockDim.x; ++i) // перебор 2^16 значений
{
intPR1 = threadIdx.x + i * blockDim.x; //в ER1 записывается число от 0 до 65535, распараллеливаясь при этом на 512 потоков
ER1 = encrypt(*randint, PR1, 1); //шифрование перебираемого блока. Первым аргументом идет фиксированное псевдослучайное число, рассмотренное выше.\
if ( ER1 ==et) //проверка условия (данный пример для одинаковых подключей)
{
++*cnt; //инкрементирование переменной количества слайдовых пар
if (it< 5) //проверка и запись первых пяти найденных слайдовых пар
{
*ar[it] = PR1;
}
++it;
}
}
```

Таблица 1

Время вычисления для поиска пары при шифровании только одной части (PR1)

Количество потоков	Время, мс
1	8680
8	2357
16	2079
64	559
128	357
256	241
512	125
1024	120

Не стоит также забывать о том, что вычисления на GPU происходят асинхронно как между потоками, так и с GPU. Поэтому необходимо создать точку синхронизации:

```
cudaEvent_tstart; //инициализация точки синхронизации
cudaEventCreate(&start ); //создаем event
pair_search<<<1, 512>>>(dev_block, dev_cnt, dev_randint); //вызов функции
cudaEventRecord( start, 0 ) ; //записываем event
cudaEventSynchronize( start ) ; //синхронизируем event
cudaMemcpy(&ar, dev_ar, sizeof(int)*5, cudaMemcpyDeviceToHost); // получение массива найденных слайдовых пар с GPU на CPU
cudaMemcpy(&cnt, dev_cnt, sizeof(int), cudaMemcpyDeviceToHost); // получение количества найденных слайдовых пар с GPU на CPU
```

Таблица 2

Примеры найденных слайдовых пар

Исходное значение	Ключ	Слайдовая пара
P – 0xfedcba9876543210 E – 0xe9ec4191bad969b9	0xffeeddccffeeddccffeeddccffeeddccffeeddccffeeddccffeeddccffeeddcc	P1 – 0x7654321028da3b14 E1 – 0xbad969b9ef04eacb
P – 0x1111111111111111 E – 0xe52d6d84bcf784ec	0x11111111111111111111111111111111111111111111111111111111111111111111	P1 – 0x11111111b8382606 E1 –0xbcf784ec6552d11e
P – 0xffffffffffffff E – 0x8727ffdcd2f88b5	0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff	P1 – 0xffffff9a7806c7 E1 – 0xcd2f88b5eb81ab2a

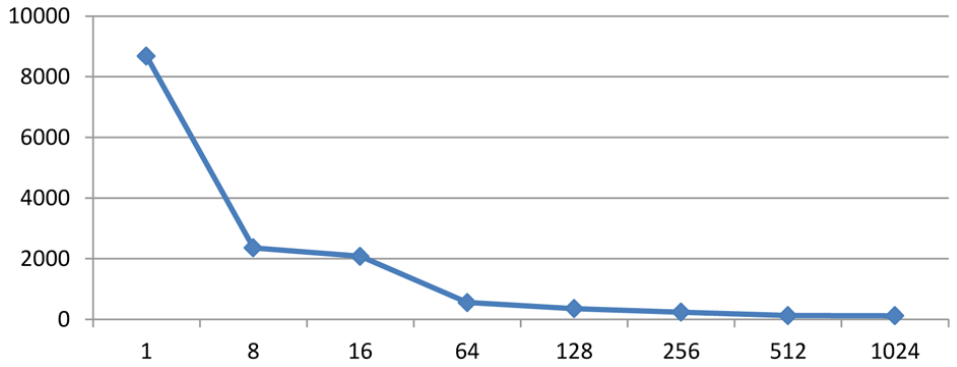

Рис. 5. Зависимость времени выполнения от количества параллельных потоков

Одной из целей данного исследования был временной аспект поиска слайдовой пары [1]. Вычисления производились на ПК Inteli5 2400 NVIDIAGTS 459 8 GbRAM. Поиск слайдовых пар (по сути это перебор значений от 0 до 65525 с использованием их в качестве входных данной функции) занимает порядка 100–150 мс.

На табл. 1 представлена зависимость времени поиска от количества потоков.

На рис. 5 приведен график по данным из табл. 1.

Как видно из вышеприведенного графика, при увеличении количества потоков, время уменьшается, что является довольно закономерным результатом.

Результаты найденных слайдовых пар представлены в табл. 2.

Работа выполнена при поддержке гранта РФФИ № 15-37-20007-мол-а-вед.

Список литературы

1. Бабенко Л.К., Ищукова Е.А. Анализ симметричных криптосистем // Известия ЮФУ. Технические науки. Тематический выпуск «Информационная безопасность». – Таганрог: Изд-во ТТИ ЮФУ, 2012. – № 11 – С. 136–147.

2. Бабенко Л.К. Ищукова Е.А. Сидоров И.Д. Параллельные алгоритмы для решения задач защиты информации. – М.: Горячая линия Телеком, 2014. – 304 с.

3. Бабенко Л.К., Ищукова Е.А., Сидоров И.Д. Применение параллельных вычислений при решении задач защиты информации // Программные системы: теория и приложения. – 2013. – Т. 4, № 3–1 (17). – С. 25–42.

4. Бабенко Л.К., Ищукова Е.А. Современные алгоритмы шифрования и методы их анализа. Учебное пособие. – М.: «Гелиос АРВ», 2006.

5. Криптографическая защита информации Блочные шифры // https://www.tc26.ru/standard/gost/GOST_R_3412-2015.pdf.

6. Параллельные вычисления Cuda | Что такое Cuda | NVIDIA. [Электронный ресурс] URL: http://www.nvidia.ru/object/cuda-parallel-computing-ru.html (дата обращения: 15.10.2015).

7. Примеры использования CUDA. [Электронный ресурс] URL: http://sporgalka.blogspot.ru/2011/10/cuda.html (дата обращения: 16.10.2015.

8. Слайдовая атака. [Электронный ресурс] URL: http://wiki.org/Слайдовая_атака. (дата обращения: 01.11.2015).

9. Cuda: аспекты производительности при решении задач. [Электронный ресурс] URL: http://habrahabr.ru/post/119435/ (Дата обращения: 18.10.2015).