**УДК 004.056.**

# УВЕЛИЧЕНИЕ СКОРОСТИ РАБОТЫ АЛГОРИТМА ШИФРОВАНИЯ «КУЗНЕЧИК» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA

**Удальцов В.А., Павлов В.Э.**

*ФГАОУ ВО Санкт-Петербургский Национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО)*

*E-mail: udalv1@rambler.ru, vladd001@yandex.ru*

Задача оптимизация работы алгоритмов шифрования является востребованной и актуальной в настоящее время. В данной статье рассматривается симметричный алгоритм блочного шифрования «Кузнечик» ГОСТ Р 34.12-2015 и технология NVIDIA CUDA, позволяющая использовать мощность графического процессора для увеличения вычислительной производительности. Особенности технологии CUDA используются для оптимизации работы алгоритма шифрования «Кузнечик». В статье используется эталонная программная реализация алгоритма шифрования с модифицированной для работы на графическом процессоре функцией encrypt_ecb. Авторами статьи произведены контрольные замеры скорости работы алгоритма шифрования на графическом процессоре и центральном процессоре, результаты которых представлены в виде графика. Представлен анализ полученных результатов, позволяющий оценить преимущество в скорости работы алгоритма шифрования использующего для расчетов мощности графического процессора. Результаты исследования, представленные в данной статье, в дальнейшем могут быть использованы для увеличения скорости процесса шифрования при применении отечественных стандартов шифрования, таких как ГОСТ Р 34.12-2015, ГОСТ 28147-89, а также при проведения различных исследований, направленных на выявление недостатков в области надежности шифров, с целью сокращения общего времени, затрачиваемого на исследование шифра.

**Ключевые слова:** криптографическое преобразование, криптография, ГОСТ, NVIDIA, CUDA, параллельные вычисления.

# OPERATION SPEED INCREASE OF THE CRYPTOALGORITHM “KUZNYECHIK” WITH THE USE OF CUDA TECHNOLOGY

**Udaltsov V.A., Pavlov V.E.**

The problem of encryption algorithm work optimization is actual and important nowadays. The following article is devoted to the symmetric block cipher «Kuznyechik” GOST R 34.12-2015 and NVIDIA CUDA technology, which enables the use of graphics processor energy to increase calculating capacity.

The peculiarities of the CUDA technology are used for optimization of the encryption algorithm “Kuznyechik". In the following article one uses the reference programm implementation of encryption algorithm with encrypt_ecb function modified for work on the graphics processor. The authors of the article have made check measurements of operation speed of cryptoalgorithm on graphic processing unit and central processing unit. The results are given in a graph. The analysis of the result is also given in the article. It enables benefits estimation of an operation speed which is used for capacity rating of a graphics processing unit. The results of research can be used in the future for increment of process speed of enciphering with the use of Russian encryption standard, such as GOST R 34.12-2015, GOST 28147-89, and for varied research concerning vices exposure in cipher reliability, with the purpose of retaining of total time spent on cipher research.

**Keywords:** cryptographic transformation, cryptography, NVIDIA, CUDA, parallel computations.

## Введение

В настоящее время обеспечение конфиденциальности информации является востребованной как в государственном, так и в коммерческом секторе услугой. Конфиденциальность информации, как одно из трех наиважнейших свойств информации, может обеспечиваться различными методами и способами. Один из таких методов — криптографические алгоритмы шифрования. В свою очередь существует проблема, заключающаяся в том, что реализация алгоритма шифрования требует значительных вычислительных ресурсов и времени для обработки больших массивов данных. Поэтому задача оптимизации работы алгоритмов шифрования является востребованной и актуальной. Статья посвящена решению задачи оптимизации работы алгоритма блочного шифрования ГОСТ 34.12-2015 «Кузнечик» при помощи математического аппарата технологии CUDA.

## Постановка задачи

Рассмотрим ГОСТ Р 34.12-2015 и технологию CUDA. Алгоритмы базовых блочных шифров, применяемые в криптографических методах обработки и защиты информации, в том числе для обеспечения конфиденциальности, аутентичности и целостности информации при ее передаче, обработке и хранении в автоматизированных системах определяются ГОСТ 34.12-2015. ГОСТ Р 34.12-2015 содержит описание двух базовых блочных шифров с длинами блоков n=128 («Кузнечик») бит и n=64 («Магма») бит и длинами ключей k=256 бит [1].

Complete Unified Device Architecture (CUDA) — архитектура параллельных вычислений, разработанная компанией NVIDIA, использующая графические процессоры для увеличения вычислительной производительности. Эффект увеличения производительности достигается за счет

того, что во время своей работы CUDA-программа задействует в производимых вычислениях как центральный процессор (далее — CPU), так и графический процессор (далее — GPU): на CPU выполняется последовательная часть кода программы, реализуются подготовительные элементы GPU-вычислений; на GPU, за счет использования так называемых нитей (threads), одновременно выполняются параллельные участки кода [2][3].

Для управления нитями применяется строгая иерархии, верхним ее уровнем является сетка - соответствующая всем нитям, которые выполняет данное ядро. Верхний уровень представляет из себя одномерный или двухмерный массив блоков. Каждый блок - это одномерный, двухмерный или трехмерный массив потоков. При этом все блоки, образующие сетку, имеют одинаковую размерность и размер [4].

Современные блочные шифраторы, использующие для своих вычислений CPU, обрабатывают данные со скоростью от 1 Мбайт/с до 5 Мбайт/с. Современные GPU, использующие технологию CUDA, за счет использования большого количества потоков обработки информации, способны обрабатывать данные с большей скоростью.

Перед нами поставлены следующие задачи:

- рассмотреть особенности работы GPU, использующих технологию CUDA;
- произвести оптимизацию работы программной реализации алгоритма шифрования «Кузнечик» за счет использования технологии CUDA;
- измерить скорость работы алгоритма шифрования «Кузнечик» на CPU и GPU, сравнить полученные результаты.

**Применение модели**

Для оптимизации алгоритма под работу на GPU была взята официальная реализация ГОСТ Р 34.12-2015 Техническим комитетом по стандартизации «Криптографическая защита информации» [5].

С целью увеличения скорости расчетов были произведены модификации алгоритма, описанные ниже.

Загрузка данных в память GPU осуществляется единым для всех потоков массивом данных, его размер определяется по формуле 1, где x – количество используемых потоков, b - размер блока шифрования.

$$S = x * b \ (1)$$

На GPU доступ осуществляется по номеру потока, формула 2, где d – номер потока в блоке, n – номер блока, b –размер блока.

$$F = d + n * b \ (2)$$

Для ускорения вычислений данные загружаются в распределенную память. Так как адресация для данной памяти одинакова для всех потоков в блоке, выделяется массив, равный размеру блока, индексация осуществляется по номеру потока в блоке.

После окончания расчётов, потоки синхронизируются, данные по индексу записываются в выходной массив и выгружаются из видеопамяти. Ниже приведена модифицированная для работы на GPU функция encrypt_ecb:

```
__global__ void encrypt_ecb(void *ctx, unsigned char *indataG, unsigned char *outdata, unsigned int length, int t)
{
 long thread = blockIdx.x * blockDim.x + threadIdx.x;
 if(thread < t)
 {
  __shared__ unsigned char indata[1024][16];
  __shared__ unsigned char *indataPTR[1024];
  __shared__ Context_ecb* context;
  __shared__ unsigned char block[1024][16];
  __shared__ unsigned char *blockPTR[1024];
  context = (Context_ecb*)ctx;
  memcpy(indata[threadIdx.x], indataG + (16 * thread), 16 );
  indataPTR[threadIdx.x] = &indata[threadIdx.x][0];
  size_t i;
  memcpy(block[threadIdx.x], outdata + (16 * thread), 16 );
  blockPTR[threadIdx.x] = &block[threadIdx.x][0];
  for(i = 0; i < (length / context->BlockLen) ; ++i)
  {
   context->EncryptFunc(indataPTR[threadIdx.x], blockPTR[threadIdx.x], context->Keys, context->PrintByteArray, context->PrintUIntArray);
   indataPTR[threadIdx.x] += context->BlockLen;
   blockPTR[threadIdx.x] += context->BlockLen;
  }
  memcpy(outdata + (16 * thread), block[threadIdx.x], 16 );
 }
 __syncthreads();
}
```

Нахождения оптимального количества потоков для корректной работы GPU приведено ниже.

Время работы данного программного обеспечения будет определяться с учетом скорости вычисления и времени загрузки данных на GPU, определяется по формуле 3, где c – время одного вычисления, k количество вычислений S – время загрузки данных.

$$T - c * k + S \ (3)$$

Так как S зависит от количества данных и скорости их загрузки, а, следовательно, является константой для конкретного объема данных на конкретной машине, то T будет минимальным при минимальном произведении c и k.

Количество вычислений определяется отношением всего объема данных к количеству данных, обрабатываемых за один шаг вычислений, формула 4, где N – весь объем данных, $N_i$ – среднее количество данных обрабатываемых за один шаг вычислений.

$$k = \frac{N}{N_i} \ (4)$$

Количество данных, обрабатываемых за один шаг вычислений, для вышеописанной функции определяется по формуле 5, где b – размер шифруемого блока, x – количество потоков, следовательно, формула 4 преобразуется в формулу 6.

$$N_i = b * x \ (5)$$

$$k = \frac{N}{b * x} \ (6)$$

Из формулы 6 следует, что график зависимости k от x является гиперболой; так как гипербола не имеет смещения по осям, то координаты ее вершины по x равна:

$$\boldsymbol{x = \sqrt{\frac{N}{b}}} \ \boldsymbol{(7)}$$

**Анализ полученных результатов**

С учетом формулы 7, приведенной в предыдущем пункте, были произведены измерения скорости работы данного программного обеспечения на CPU и GPU результаты представлены на рисунке 1.

Измерения проводились на персональной ЭВМ со следующими характеристиками:

- CPU Intel Core I7-3537U;
- GPU GeForce GT750M;
- актуальная версия видеодрайвера 378.49 от 24.01.2017.

На рисунке 1 представлена зависимость времени работы алгоритма шифрования, исполняемого на GPU и CPU, от объема исходного файла.

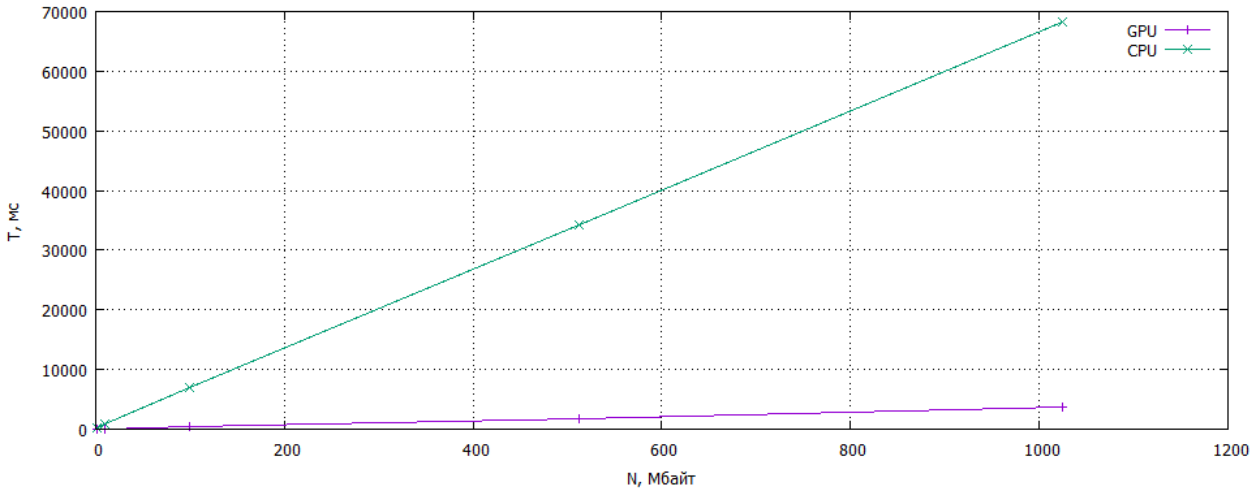


*Рисунок 1 — Сравнение времени шифрования по алгоритму «Кузнечик» на CPU и GPU*

Для эксперимента были взяты файлы объемом 10, 100, 512 и 1024 Мбайт. В таблице 1 приведены данные о скорости работы алгоритма шифрования «Кузнечик» на CPU и GPU.

*Таблица 1 — Время шифрования файлов на CPU и GPU*

| *Размер файла, Мбайт* | *Время обработки файла на CPU, мс* | *Время обработки файла на GPU, мс* |
|---|---|---|
| 1 | 282 | 4 |
| 10 | 882 | 41 |
| 100 | 7073 | 383 |
| 512 | 34342 | 1845 |
| 1024 | 68243 | 3684 |

Проведем анализ полученных результатов, сравнив скорость обработки входных данных на GPU и CPU. С учетом данных таблицы 1 видно, что входные файлы обрабатывались на CPU в среднем со скоростью 120.4 Мбит/с, на GPU — 2.2 Гбит/с. За счет использования большого количества одновременно работающих потоков, а также особенностей при работе с памятью, графические процессоры, использующие технологию CUDA, обрабатывают файлы при использовании алгоритма шифрования «Кузнечик» в 18 раз быстрее, чем это делает CPU.

**Заключение**

В рамках проведенного исследования была применена модель оптимизации работы блочного шифра «Кузнечик» с использованием технологии CUDA. Были получены результаты, анализ которых подтвердил, правильность выбранного подхода к решению поставленной задачи. Данная разработка может быть использована в дальнейшем для ускорения процесса шифрования при применении отечественных стандартов шифрования, таких как ГОСТ 34.12-2015, ГОСТ 28147-89, а также с целью проведения исследований, направленных на выявление недостатков в области

надежности шифров с применением различных методов анализа, с целью сокращения общего времени, затрачиваемого на исследование шифра.

**Список литературы**

1. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. — Введ. 19.06.2015. — М.: Стандартинформ. — 25с.

2. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах. Введение в программирование графических процессов. — М.: ДМК Пресс, 2011. — 232с.

3. Боресков А. В., Харламов А. А. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие. — М.: Издательство Московского Университета, 2012. — 336с.

4. Боресков А. В. Основы работы с технологией CUDA. — М.: ДМК Пресс, 2016. — 232с.

5. Технический комитет по стандартизации «Криптографическая защита информации» [Электронный ресурс]. — Режим доступа: https://www.tc26.ru, свободный. Яз. рус. (дата обращения 10.09.2016).