Радиооптика

Сетевое научное издание
МГТУ им. Н.Э. Баумана

http://radiooptics.ru

Ссылка на статью:
// Радиооптика. МГТУ им. Н.Э. Баумана. Электрон. журн. 2016. № 06. С. 24–34.

DOI: **10.7463/rdopt.0616.0850899**

Представлена в редакцию: 10.10.2016
Исправлена: 24.10.2016

УДК 004.056.55

О производительности блочных шифров, основанных на клеточных автоматах, при их реализации на графических процессорах

Ключарёв П. Г.[1,*]

[*]pk.iu8@yandex.ru

[1]МГТУ им. Н.Э. Баумана, Москва, Россия

Статья посвящена вопросам реализации и тестированию производительности блочных шифров, основанных на обобщенных клеточных автоматах при программной реализации на графических процессорах AMD и NVIDIA. Использовался интерфейс OpenCL. Производительность для режима счетчика и режима ECB составила от 90 до 380 Мбит/с, в зависимости от параметров, что сопоставимо с производительностью традиционных блочных шифров на CPU. Учитывая, что криптоалгоритмы, основанные на обобщенных клеточных автоматах, предназначены для аппаратной реализации, такой уровень производительности на GPU существенно расширяет область применения данных блочных шифров – фактически это делает возможным их применение на различных вычислительных устройствах, имеющих GPU, в том числе персональных компьютерах, ноутбуках и др. Работа выполнена при финансовой поддержке РФФИ, в рамках научного проекта №16-07-00542 а.

Ключевые слова: клеточный автомат, блочный шифр, графический процессор

Введение

В настоящее время исключительную важность приобрело обеспечение информационной безопасности. Для решения связанных с информационной безопасностью задач используются криптографические алгоритмы. Широкое применение получили блочные шифры [6; 7; 8]. Методы построения симметричных блочных шифров на основе обобщенных клеточных автоматов, разработанные автором [1], позволяют построить блочные шифры, обладающие высокой производительностью при аппаратной реализации [4; 5]. Однако таким криптоалгоритмами присуще ограничение – низкая эффективность программной реализации на CPU. Это ограничение не является недостатком – оно показывает область применения таких алгоритмов. В то же время, учитывая особенности структуры клеточных автоматов со свойственной им высокой параллельностью, появляется гипотеза о том, что такие автоматы могут быть достаточно эффективно реализованы с помощью

графических процессоров. Возможность такой эффективной реализации существенно расширяет сферу применимости основанных на них алгоритмов.

Основной задачей этой работы является тестирование скорости реализации на графических процессорах блочных шифров, основанных на обобщенных клеточных автоматах.

Графические процессоры

Графические процессоры (GPU), имеющиеся в настоящее время практически в любом компьютере, создавались для ускорения графики и, прежде всего, трехмерной графики. Впоследствии, однако, оказалось, что они годятся и для вычислений общего назначения, а их параллельная архитектура позволяет достичь высокой скорости во многих хорошо распараллеливаемых задачах. Вычислениям на графических процессорах посвящено большое число источников, в том числе [11; 12; 13; 14; 15].

Для вычислений на графических процессорах используют специальные программные интерфейсы, такие как OpenCL. При этом, часть программы (ядро) выполняется на GPU, а другая часть (хост-программа) управляет работой ядра и выполняется на CPU. При этом одновременно исполняется достаточно большое число экземпляров ядра (рабочих элементов), каждый из которых имеет уникальный идентификатор. Рабочие элементы объединяются в рабочие группы. Для GPU фирмы AMD в рабочей группе не может быть более 256 рабочих элементов, а для GPU фирмы NVIDIA – не более 1024.

Обобщенные клеточные автоматы

Весьма перспективным криптографическим примитивом является обобщенный клеточный автомат. Кратко напомним его определение.

Будем называть *обобщённым клеточным автоматом* ориентированный мультиграф $A=(V,E)$, где $V=\{v_1,\ldots,v_N\}$ - множество вершин, а E -мультимножество ребер. С каждой вершиной v_i этого графа ассоциированы:

- булева переменная m_i, которая называется *ячейкой*;
- булева функция $f_i(x_1,\ldots,x_{d_i})$, которая называется локальной функцией связи i-й вершины.

При этом каждой паре (v,e), где v - вершина, а e - инцидентное ей ребро, будет соответствовать номер аргумента локальной функции связи, вычисляемой в вершине v (номер ребра e относительно вершины v).

Обобщенный клеточный автомат работает по шагам. Перед первым шагом каждая ячейка m_i, $i=1...N$, имеет начальное значение $m_i(0)\in\{0,1\}$. Далее, значения ячеек на шаге t вычисляются по формуле:

$$m_i(t)=f_i(m_{\eta(i,1)}(t-1),m_{\eta(i,2)}(t-1),\ldots,m_{\eta(i,d_i)}(t-1)), \qquad (1)$$

где $\eta(i,j)$ – номер вершины, из которой исходит ребро, заходящее в вершину i и имеющее относительно этой вершины номер j. Заполнением клеточного автомата на шаге t будем называть набор значений ячеек $(m_1(t), m_2(t), \ldots, m_N(t))$.

Обобщенный клеточный автомат будем называть *однородным*, если локальная функция связи для всех ячеек одинакова. Назовем обобщённый клеточный автомат *неориентированным*, если в графе для любого ребра (u,v) в существует и ребро (v,u). Граф такого автомата можно рассматривать как неориентированный, если заменить каждую пару ориентированных ребер (u,v) и (v,u) на неориентированное ребро $\{u,v\}$.

Здесь мы будем использовать неориентированные однородные обобщённые клеточные автоматы, называя их, для краткости, обобщёнными клеточными автоматами.

Пусть $F_t : \{0,1\}^n \to \{0,1\}^n$ - функция, аргументом которой является начальное заполнение данного обобщенного клеточного автомата, а значением – заполнение этого автомата через *t* шагов.

Блочные шифры

Здесь мы лишь очень кратко остановимся на схеме реализуемых блочных шифров. Подробную информацию о них можно найти в статье [1] .

S-блоки представляют собой функции вида $S_c : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ и задаются формулой:

$$S_c^A(key, x) = pr_m\big(F_A(x \parallel key \parallel c, r)\big).$$

Здесь $x \parallel y$ - конкатенация x и y,

r – число шагов обобщенного клеточного автомата,

$pr_m : \{0,1\}^* \to \{0,1\}^m$ – функция, возвращающая младшие m элементов аргумента;

A – обобщённый клеточный автомат;

$c \in \{0,1\}^t$ – некоторая константа, близкая к равновесной.

Как продемонстрировано в работе [3], такие функции нельзя отличить от псевдослучайных посредством статистических тестов из набора NIST, в случае правильного выбора параметров, в частности, графа клеточного автомата и локальной функции связи, параметров r и t .

Для построения блочного шифра используется схема Фейстеля, при этом раундовое преобразование определяется следующим образом:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus S_{c_i}^A(key_i, R_{i-1}),$$

где i – номер раунда;

L_0 – левая половина блока открытого текста;

R_0 – правая половина блока открытого текста;

L_i – левая половина блока после i-го раунда;

R_i – правая половина блока после i-го раунда;

key_i – раундовый ключ;

c_i – константа.

Раундовые ключи получаются из секретного ключа путем применения того или иного алгоритма разворачивания ключа, например алгоритма, определенного в работе [1].

При расшифровании выполняется преобразование, обратное приведенному выше.

Константы c_i попарно различаются, причем расстояние Хемминга между каждой парой констант близко к половине длины константы.

Алгоритм шифрования состоит из нескольких раундов, причем их число во всяком случае, должно быть не меньше трех. Из соображений удобства можно рекомендовать четное число раундов, причем проведенные в работе [1] исследования показали, что четырёх раундов достаточно для обеспечения криптографических свойств.

Реализация

Блочные шифры были реализованы для графических процессоров с использованием интерфейса OpenCL. Программа написана на языке C++ (использовался компилятор фирмы Microsoft, входящий в Microsoft Visual Studio 2013). Программа позволяла пользователю устанавливать различные параметры, в том числе, граф обобщенного клеточного автомата, локальную функцию связи, число раундов, число шагов клеточного автомата на раунде, константы, длину блока, длину ключа и т.д.

Обобщенный клеточный автомат представляет собой набор ячеек, над которыми на каждом шаге выполняются однотипные вычисления, что позволяет эффективно воспользоваться возможностями графического процессора по параллельной обработке данных.

При реализации на GPU для хранения графа клеточного автомата, раундовых ключей, а также ячеек клеточного автомата использовалась локальная память (local memory) элементарного вычислителя. Это дает следующие преимущества:

- Операции доступа к локальной памяти осуществляются значительно быстрее, чем операции доступа к глобальной памяти.
- Разные потоки имеют общий доступ к данным, хранящимся в локальной памяти.

Перед началом работы в массив ячеек клеточного автомата, размещенный в локальной памяти, загружаются открытый текст, раундовый ключ и раундовая константа.

Тестирование производительности

Тестирование проводилось для шифров со следующими параметрами:

- размер блока: 128 бит;
- количество раундов: 4;
- количество шагов обобщенного клеточного автомата в раунде: 8, 12, 16;

- использовались обобщенные клеточные автоматы с модифицированными графами Любоцкого-Филипса-Сарнака (с числом вершин 242, степени 6). Модификация проводилась таким образом, чтобы удалить из графа петли и кратные ребра, не влияя на его регулярность;
- использовалась локальная функция связи (из семейства, предложенного в работе [2]):

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_3x_5 \oplus x_3x_4 \oplus x_5x_6 \oplus x_3x_5 \oplus x_1x_5 \oplus x_1 \oplus x_2 \oplus 1.$$

При проведении тестирования использовалось видеоадаптеры, основанные на следующих графических процессорах:

- NVIDIA GTX 650;
- NVIDIA GTX 770;
- AMD R9 280X.

Основные параметры этих графических процессоров приведены в таблице 1.

Таблица 1. Параметры графических процессоров

Параметр	**Видеокарта**		
	NVIDIA GTX 650	**NVIDIA GTX 770**	**AMD R9 280X**
Количество вычислительных элементов(compute units)	4	8	32
Тактовая частота, МГц	1033	1137	1000
Максимальный размер рабочей группы	1024	1024	256
Размер глобальной памяти, МБ	1024	2048	2048
Размер локальной памяти, КБ	48	48	32
Тип памяти	GDDR5	GDDR5	GDDR5
Год появления на рынке	2012	2013	2014

Тестирование производилось для следующих классических режимов работы блочных шифров [8]:

- режима счетчика (Counter mode),
- режима ECB (Electronic code book, электронной кодовой книги, в отечественной литературе его иногда называют режимом простой замены),
- режима CBC (Cipher block chaining, режим сцепления блоков шифртекста).

Результаты произведенного тестирования производительности блочного шифра в режиме счетчика для различных значений числа шагов и различных графических процес-

соров приведены в табл. 2 и на рис. 1. Производительность в режиме ECB отличалась несущественно, поскольку режим счетчика практически не отличается от режима ECB с точки зрения вычислительной сложности.

Результаты тестирования производительности для режима CBC приведены в табл. 3 и на рис. 2.

Таблица 2. Производительность блочных шифров в режиме счетчика при реализации на различных графических процессорах, Мбит/с

Модель GPU	8 шагов	12 шагов	16 шагов
NVIDIA GTX 650	137	109	92
NVIDIA GTX 770	261	243	203
AMD R9 280X	382	274	227

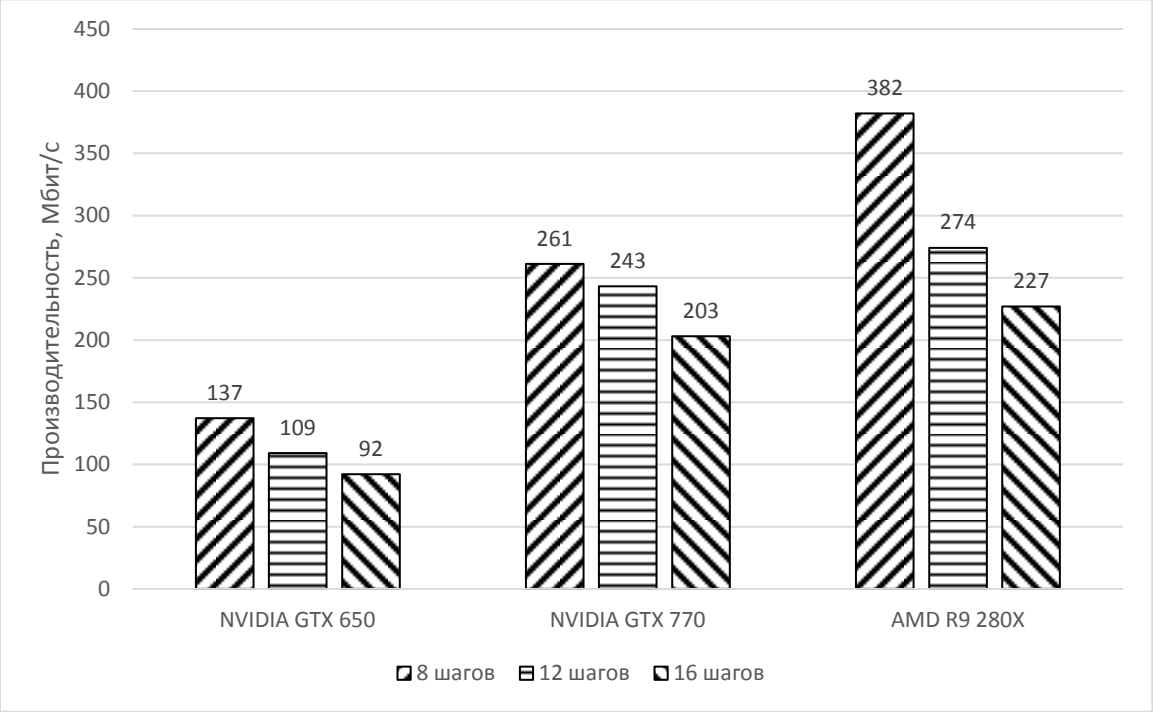

Рис. 1 – Производительность блочных шифров в режиме счетчика при реализации на различных графических процессорах

Таблица 3. Производительность блочных шифров в режиме CBC при реализации на различных графических процессорах, Мбит/с

Модель GPU	8 шагов	12 шагов	16 шагов
NVIDIA GTX 650	13	9	7
NVIDIA GTX 770	29	23	19
AMD R9 280X	20	17	14

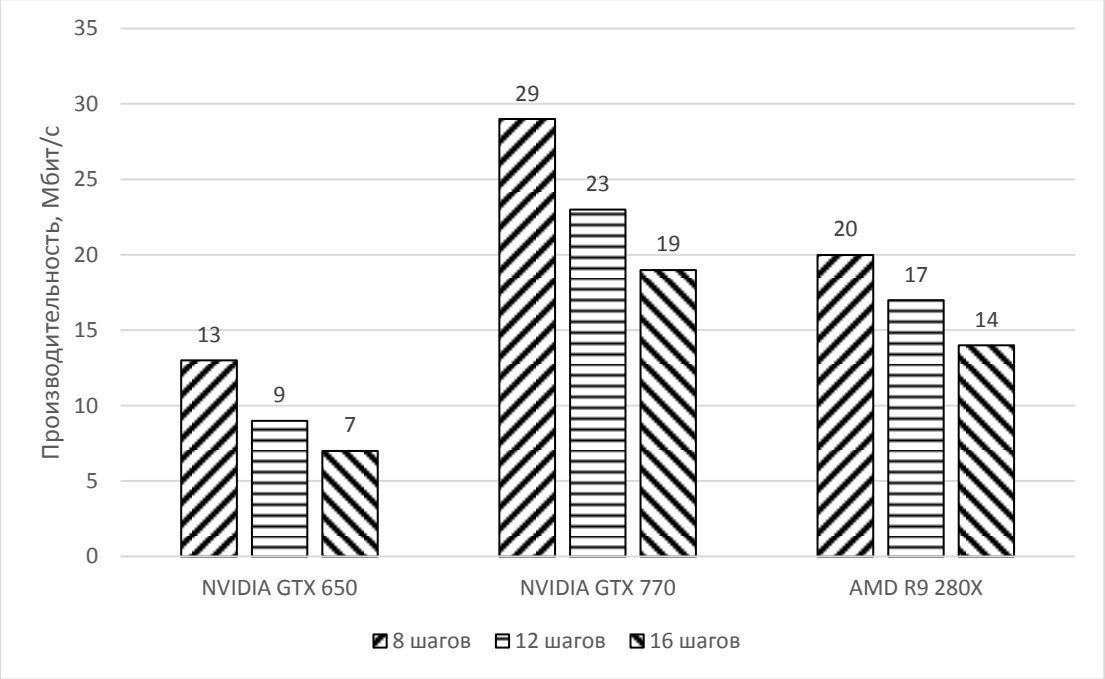

Рис. 2 – Производительность блочных шифров в режиме CBC при реализации на различных графических процессорах

Обсуждение результатов

Криптографические алгоритмы, основанные на обобщенных клеточных автоматах рассчитаны на аппаратную реализацию. Производительность их программной реализации (на базе традиционных CPU) невысока. Согласно полученным данным, реализация на графических процессорах таких блочных шифров, работающих в режиме ECB и режиме счетчика, обладает достаточно высокой производительностью, которая сравнима с производительностью реализации на CPU блочных шифров, рассчитанных на программную реализацию. Так, производительность блочных шифров DES, BLOWFISH, CAST, RC6, IDEA на процессоре Intel Core i3 составляет от 64 до 170 Мбит/с [9]. Производительность блочного шифра AES, являющегося стандартом шифрования США, при реализации на различных CPU Intel составляет порядка сотен Мбит/с [10] (данные о его производительности могут варьироваться в достаточно широких пределах, в зависимости от модели CPU, его тактовой частоты и количества ядер). В то же время, реализация на GPU блочных шифров, основанных на обобщенных клеточных автоматах, в режиме CBC показала более низкие скоростные характеристики.

Учитывая, что практически все современные персональные компьютеры, ноутбуки, планшеты, смартфоны и многие другие вычислительные устройства снабжены достаточно высокопроизводительными графическими процессорами, представленные в настоящей статье результаты существенно расширяют область применения рассматриваемых шифров, по крайней мере, в некоторых режимах работы.

Заключение

В статье было продемонстрировано, что блочные шифры, основанные на обобщенных клеточных автоматах, из семейства, разработанного автором [1] могут быть эффективно реализованы на GPU, показывая при этом в режимах счетчика и ECB производительность, сравнимую с производительностью блочных шифров AES, DES, BLOWFISH, CAST, RC6, IDEA на CPU. Этот факт позволяет использовать блочные шифры из семейства, разработанного автором, на обычных компьютерах, а также других вычислительных устройствах, имеющих графический процессор, в том числе, планшетах, сматфонах и др.

Работа выполнена при финансовой поддержке РФФИ, в рамках научного проекта №16-07-00542 а.

Список литературы

1. Ключарев П.Г. Блочные шифры, основанные на обобщённых клеточных автоматах // Наука и образование. Электронное научно-техническое издание. 2012. № 12.
2. Ключарев П.Г. Обеспечение криптографических свойств обобщённых клеточных автоматов // Наука и образование. Электронное научно-техническое издание. 2012. № 3.
3. Ключарев П.Г. Построение псевдослучайных функций на основе обобщённых клеточных автоматов // Наука и образование. Электронное научно-техническое издание. 2012. № 10.
4. Ключарев П.Г. Производительность и эффективность аппаратной реализации поточных шифров, основанных на обобщенных клеточных атоматах // Наука и образование. Электронное научно-техническое издание. 2013. № 10. — С. 299-314.
5. Ключарёв П.Г. Реализация криптографических хэш-функций, основанных на обобщенных клеточных автоматах, на базе ПЛИС: производительность и эффективность // Наука и образование. Электронное научно-техническое издание. 2014. № 1.
6. Панасенко С.П. Алгоритмы шифрования: классификация алгоритмов шифрования и методов их вскрытия, новейшая история симметричного шифрования, описание алгоритмов : специальный справочник. — Санкт-Петербург: БХВ-Петербург, 2009. — 564 с.
7. Смарт Н. Криптография: [для специалистов, работающих в обл. защиты информ., специалистов-разраб. програм. обеспечения]. — М. : Техносфера, 2005. — 525 с.
8. Алферов А.П. Основы криптографии. — М. : Гелиос АРВ, 2002. — 480 с.
9. Aggarwal K., Saini J.K., Verma H.K. Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers // International Journal of Computer Applications. 2013. Т. 68. № 25.
10. Bernstein D.J., Schwabe P. New AES software speed records. Springer, 2008. — 322-336.
11. Eberly D.H. GPGPU Programming for Games and Science. Taylor & Francis, 2014.

12. Gaster B., Howes L., Kaeli D.R., Mistry P., Schaa D. Heterogeneous Computing with OpenCL: Revised OpenCL 1.2 Edition. : Elsevier Science, 2012.
13. Kaeli D.R., Mistry P., Schaa D., Zhang D.P. Heterogeneous Computing with OpenCL 2.0. : Elsevier Science, 2015.
14. Kowalik J., Puźniakowski T. Using OpenCL: Programming Massively Parallel Computers. : IOS Press, 2012.
15. Scarpino M. OpenCL in Action: How to Accelerate Graphics and Computation. : Manning, 2012.

Radiooptics
Electronic journal of the Bauman MSTU
http://radiooptics.ru

Radiooptics of the Bauman MSTU, 2016, no. 06, pp. 24–34.

DOI: **10.7463/rdopt.0616.0850899**

Received: 10.10.2016
Revised: 24.10.2016

On the Performance of GPU-Implemented Block Ciphers Based on Generalized Cellular Automata

P.G. Klyucharev[1,*]

[*]pk.iu8@yandex.ru

[1]Bauman Moscow State Technical University, Moscow, Russia

Keywords: cellular automata, block cipher, GPU

Block ciphers have found extensive use when solving the tasks of information security. The article considers implementation and testing of the performance of symmetric block ciphers based on generalized cellular automata, which were constructed using the techniques the author developed earlier in the software implementation on the GPUs NVIDIA GTX 650, NVIDIA GTX 770, AMD R9 280X.

The implementation used the OpenCL interface. There were used codes with 4 rounds, the number of steps of the generalized cellular automata in each round was chosen from the list: 8, 12, 16; the block length was 128 bits. As a graph of the cellular automata, the Lubotzky-Phillips-Sarnak graph was used. Performance of obtained implementation for the counter and ECB modes was in the range from 90 to 380 Mbit/s, depending on parameters, which is comparable with the performance of CPU-based traditional block ciphers, such as AES, DES, BLOWFISH, CAST, RC6, IDEA. At the same time, performance in the CBC mode ranged from 7 to 29 Mbit /s.

Given that the cryptographic algorithms based on generalized cellular automata, are designed for hardware implementation, the achieved level of performance in ECB and counter modes significantly broadens the application scope of these block ciphers is actually enabling their use in any computing device that has a GPU, including personal computers, laptops, tablets, smartphones, etc.

The work was implemented under support of the Russian Federal Property Fund, as part of a research project and №16-07-00542.

References

1. Kliucharev P.G. Blochnye shifry, osnovannye na obobshchennykh kletochnykh avtomatakh. *Nauka i obrazovanie = Science and education*. Electronic scientific and technical publication. 2012. No. 12.

2. Kliucharev P.G. Obespechenie kriptograficheskikh svoistv obobshchennykh kletochnykh avtomatov. *Nauka i obrazovanie = Science and education*. Electronic scientific and technical publication. 2012. No. 3.
3. Kliucharev P.G. Postroenie psevdosluchainykh funktsii na osnove obobshchennykh kletochnykh avtomatov. *Nauka i obrazovanie = Science and education*. Electronic scientific and technical publication. 2012. No. 10.
4. Kliucharev P.G. Proizvoditel'nost' i effektivnost' apparatnoi realizatsii potochnykh shifrov, osnovannykh na obobshchennykh kletochnykh atomatakh. *Nauka i obrazovanie = Science and education*. Electronic scientific and technical publication. 2013. No. 10. P. 299-314.
5. Kliucharev P.G. Realizatsiia kriptograficheskikh khesh-funktsii, osnovannykh na obobshchennykh kletochnykh avtomatakh, na baze PLIS: proizvoditel'nost' i effektivnost'. *Nauka i obrazovanie = Science and education*. Electronic scientific and technical publication. 2014. No. 1.
6. Panasenko S.P. Algoritmy shifrovaniia: klassifikatsiia algoritmov shifrovaniia i metodov ikh vskrytiia, noveishaia istoriia simmetrichnogo shifrovaniia, opisanie algoritmov: spetsial'nyi spravochnik. *BKhV-Peterburg*. Saint Petersburg, 2009. 564 p.
7. Smart N. Kriptografiia: [dlia spetsialistov, rabotaiushchikh v obl. zashchity inform., spetsialistov-razrab. program. obespecheniia]. *Tekhnosfera*, Moscow, 2005. 525 p.
8. Alferov A.P. Osnovy kriptografii. *Gelios ARV*. Moscow, 2002. 480 p.
9. Aggarwal K., Saini J.K., Verma H.K. Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers. *International Journal of Computer Applications*. 2013. Vol. 68. No. 25.
10. Bernstein D.J., Schwabe P. New AES software speed records. *Springer*, 2008. 322-336 pp.
11. Eberly D.H. GPGPU Programming for Games and Science. *Taylor & Francis*, 2014.
12. Gaster B., Howes L., Kaeli D.R., Mistry P., Schaa D. Heterogeneous Computing with OpenCL. Revised OpenCL 1.2 Edition. *Elsevier Science*, 2012.
13. Kaeli D.R., Mistry P., Schaa D., Zhang D.P. Heterogeneous Computing with OpenCL 2.0. *Elsevier Science*, 2015.
14. Kowalik J., Puźniakowski T. Using OpenCL: Programming Massively Parallel Computers. *IOS Press*, 2012.
15. Scarpino M. OpenCL in Action: How to Accelerate Graphics and Computation. *Manning*, 2012.