УДК 004.056.55

РЕАЛИЗАЦИЯ РОССИЙСКИХ СТАНДАРТОВ СИММЕТРИЧНОГО ШИФРОВАНИЯ НА ОСНОВЕ OPENCL

© Борисов А.Н., Мясников Е.В.

e-mail: borisovalexey1996@gmail.com

Самарский национальный исследовательский университет имени академика С.П. Королёва, г. Самара, Российская Федерация

Криптографическая защита информации в настоящее время является неотъемлемой частью IT-инфраструктуры. Постоянно возрастающие объемы обрабатываемых данных приводят к повышению требований к производительности средств криптографической защиты информации.

Основным средством криптографической защиты информации являются симметричные шифры. В России стандартами симметричного шифрования являются алгоритмы «Кузнечик» и «Магма», определяемые ГОСТ Р 34.12-2015 [1].

«Кузнечик» является алгоритмом блочного шифрования с длиной ключа 256 бит и длиной блока 128 бит. Число раундов алгоритма равно 10. По своей структуре является SP-сетью, в чем схож с алгоритмом AES. Раунд «Кузнечика» состоит из побайтового нелинейного преобразования, линейного преобразования и сложения с раундовым ключом по модулю 2.

«Магма» является блочным шифром, построенным на основе сети Фейстеля с 32 раундами. Длина ключа – 256 бит, длина блока – 64 бита.

Для ускорения шифрования возможно использовать графические процессоры. Архитектура графических процессоров нацелена на массивно параллельные вычисления с использованием сотен и тысяч потоков, поэтому наибольшее ускорение от их использования получают задачи, которые могут предоставить необходимую степень параллельности.

В настоящей работе для написания кода, исполняемого на графическом процессоре, используется фреймворк OpenCL [2]. Код, написанный на OpenCL, может исполняться на любом совместимом устройстве, будь то центральный процессор (ЦП) или графический процессор (ГП).

Алгоритм «Магма» переносится на OpenCL без специфических модификаций, в силу своей простоты.

Реализация «Кузнечика» на OpenCL вместо прямого вычисления всех функций использует заранее подготовленные таблицы поиска, получение которых подробно описано в работе [3]. При этом вычисление функции заменяется на ряд чтений из таблицы и операций XOR между полученными значениями, что требует существенно меньшего количества времени. В настоящей работе рассмотрено 2 варианта таблиц – таблица 16x256 и таблица 32x16, размер одного элемента в обоих случаях равен 16 байтам. Первая таблица совмещает одновременно линейное и нелинейное преобразование, что упрощает код и уменьшает число обращений в память. Размеры второй таблицы позволяют размещать ее в выделенном регионе сверхбыстрой памяти, которая имеется в графических процессорах (локальная память, в терминологии OpenCL). Вариант алгоритма с первой таблицей обозначен как «Кузнечик-1», со второй таблицей – «Кузнечик-2».

Для проведения исследования использовалась следующая конфигурация:

1. ЦП: Intel Core i7-4510U.
2. ГП1: Intel HD Graphics 4400.

3. ГП2. NVIDIA GeForce 850M, 4 ГБ DDR3.

4. ОЗУ: 12 ГБ DDR3-1600.

Массив раундовых ключей перед началом исполнения копировался в память устройства. Все данные хранились в закрепленном буфере в ОЗУ, без копирования в память устройства. Размер шифруемого блока данных составлял 256МБ. Сам шифруемый блок представлял собой случайную последовательность символов. Результаты исследования представлены в таблице.

Таблица. Скорость шифрования

Устройство	Скорость шифрования, МБ/с		
	Кузнечик-1	Кузнечик-2	Магма
i7-4510U	61	198	66
HD Graphics 4400	354	213	624
GeForce 850M	2758	1893	7842

Из результатов экспериментов следует, что для центральных процессоров наилучшим вариантом является реализация «Кузнечик-2», поскольку при этом OpenCL-компилятор способен использовать SIMD-регистры. Встроенный графический процессор IntelHDGraphics 4400 оказывается быстрее двухъядерного процессора, что потенциально позволяет его использовать в качестве выделенного устройства для шифрования данных, освобождая тем самым центральный процессор для других задач. Мобильный графический процессор GeForce 850M позволяет развивать скорость более 2 ГБ/с для «Кузнечика» и более 7 ГБ/с для «Магмы», что соответствует пропускной способности современных NVMe интерфейсов, и позволит шифровать большие объемы данных без потерь производительности.

Библиографический список

1. ГОСТ 34.12 – 2015. Криптографическая защита информации. Блочные шифры [Текст] – Москва: Стандартинформ, 2015. – 21 с.

2. OpenCLReferencePages [Электронный ресурс] / KhronosGroup. – https://www.khronos.org/registry/OpenCL/sdk/1.2/docs/man/xhtml/ (дата обращения: 23.04.2019)

3. Ищукова Е.А. Разработка и реализация высокоскоростного шифрования с использованием алгоритма "Кузнечик" [Текст]/ Е.А Ищукова , Р.А. Кошуцкий, Л.К. Бабенко // Журнал Auditorium. – 2015. – Вып. № 4(8). "Общие и комплексные проблемы технических и прикладных наук и отраслей народного хозяйства".