R.A. Gashin, Yu.V. Golchevskiy

Development of Cryptographic Library Based on Algorithms from Estream Project

Keywords: stream ciphers, cryptography, eSTREAM.
The research and realization of opportunities to improve program implementation of algorithms of eSTREAM project to increase their productivity are presented. The comparison between proposed implementations and author's implementations was made. The cryptographic library and software applications to demonstrate the capabilities of the library were developed.

Р.А. Гашин, Ю.В. Гольчевский

РАЗРАБОТКА КРИПТОБИБЛИОТЕКИ НА БАЗЕ АЛГОРИТМОВ ПРОЕКТА eSTREAM

Постановка проблемы

Для защиты конфиденциальной информации, передающейся по вычислительным сетям, весьма эффективным механизмом является использование потоковых криптографических алгоритмов. Их основные достоинства – высокая скорость шифрования и возможность работы с неограниченными потоками данных в режиме реального времени.

Интерес к подобным алгоритмам очень велик. Передовые разработки в области потоковых шифров ведутся в России, США, Европе и Японии. Например, в 2003 году в Японии завершился крупный проект по разработке криптографических алгоритмов CRYPTREC. Были выбраны три потоковых криптографических шифра для использования в системе электронного правительства Японии [1]. В Европе с 2004 по 2008 г. проводился научно-исследовательский проект eSTREAM по выявлению новых потоковых криптоалгоритмов. Последняя публикация документации шифров-победителей датируется 2012 годом. После этого изменения в криптографические алгоритмы не вносились.

Цель представленной работы – разработка потоковой криптографической библиотеки на основе алгоритмов-победителей проекта eSTREAM. В ходе работы проведены поиск и реализация возможностей для улучшения программной реализации алгоритмов проекта eSTREAM для повышения их производительности. Проведено сравнение предложенных реализаций с авторскими. Разработаны криптографическая библиотека estream.h и программные приложения для демонстрации возможностей этой библиотеки.

О проекте eSTREAM

В ноябре 2004 года начался прием претендентов на участие в конкурсе. К участникам предъявлялись следующие основные требования: длина ключа составляет максимум 128 бит, входной поток данных не ограничен по объему, выходной поток данных такого же объема, что и входной, и скорость работы выше, чем у применяемого в США стандарта шифрования AES-128 в режиме счетчика [2].

Все участники были тщательно исследованы на надежность, производительность, корректность работы, простоту и гибкость реализации шифров. Для изучения стойкости алгоритмов к современным методам атак была разработана платформа для анализа шифров (Testing framework) [3]. В апреле 2008 года были объявлены алгоритмы-

победители. В категории «Программно-ориентированные алгоритмы» победителями стали: HC128, Rabbit, Salsa, Sosemanuk, а среди «аппаратно-ориентированных алгоритмов»: Grain, MICKEY, Trivium и F-FCSR (позже он был исключен из финальной публикации из-за слабости, обнаруженной после завершения проекта) [2, 3]. Подробное техническое описание алгоритмов, результаты тестов, отчеты и протоколы конференций содержатся на ресурсе [3].

Ввиду высокой скорости шифрования входного потока и повышенной криптостойкости область применения шифров проекта eSTREAM достаточно разнообразна:

1) шифрование потоков данных в режиме реального времени (системы мгновенного обмена сообщениями, видеоконференции, защищенный канал передачи данных и другое);

2) шифрование информации в режиме блочных шифров (шифрование файлов, разделов жестких дисков и другое);

3) аппаратные комплексы шифрования для организации защищенных каналов связи.

Результаты проекта eSTREAM используются в различных программных разработках по всему миру. Например, в криптографической библиотеке Crypto++ (http://www.cryptopp.com/). Реализована преимущественно на языке программирования C++. Проект Crypto++ объединяет несколько типов криптоалгоритмов: потоковые, блочные, хэш-функции, ассиметричные. Раздел потоковые шифры содержит алгоритмы проекта eSTREAM Salsa и Sosemanuk. Шифр HC128 используется в кроссплатформенной криптографической библиотеке Bounce Castle, которая разрабатывается австралийскими исследователями на языках Java и C#. Последняя публикация проекта датируется мартом 2015 года ((https://www.bouncycastle.org/). Алгоритм Rabbit входит в состав вебприложения для шифрования информации Crypto Tools. Оно встраивается в браузер в виде дополнительного расширения и предназначено для шифрования небольших объемов информации (http://iblogbox.com/devtools/crypto/). Grain и Trivium используются в библиотеке AVR Crypto lib, которая ориентирована на работу на микроконтроллерах. Разработку ведет немецкая лаборатория Das Labor. Библиотека реализована на языке программирования C, распространяется под лицензией “GPL v3” (http://www.daslabor.org/wiki/AVR-Crypto-Lib/).

Исходные коды алгоритмов проекта eSTREAM в основном используются в том виде, в котором их предложили авторы шифров, без добавления каких-либо модификаций.

Библиотека estream.h

Организаторы представили результаты конкурса как разрозненные проекты, не зависящие друг от друга. Исключением является то, что функции, использующиеся в каждом шифре, называются одинаково. Однако не было представлено единого механизма, который позволил бы использовать конкурсные разработки в рамках единого проекта. Разработчикам предлагается выбрать нужный алгоритм и интегрировать в свои продукты.

В ходе выполнения представленной работы разработана криптографическая библиотека потокового шифрования информации, которая объединила все семь алгоритмов проекта eSTREAM. В библиотеке использованы модифицированные реализации шифров и предоставлен удобный интерфейс для их интеграции в более крупные проекты.

Библиотека насчитывает 16 файлов: девять заголовочных (header-файлы) и семь исходных файлов. Структура estream.h представлена на рис. 1.

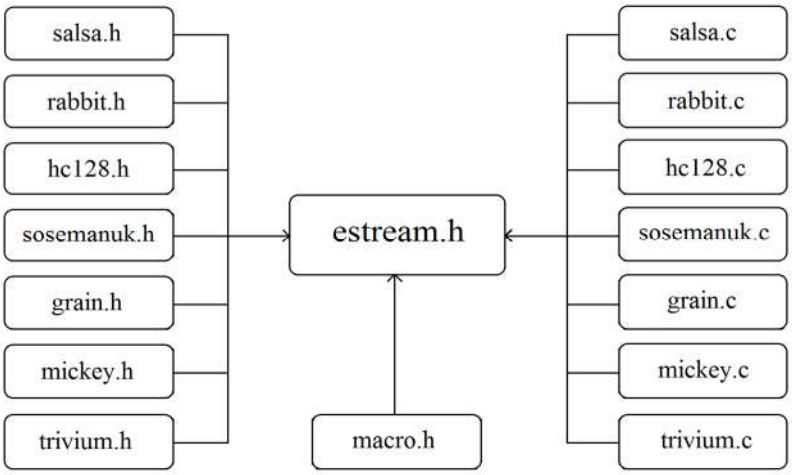

Рис. 1. Структура библиотеки estream.h

Основным файлом является estream.h, который разработчикам необходимо подключать к своей программе. Он содержит инструкции, которые подгружают остальные header-файлы.

Подгружаемые в estream.h файлы содержат базовый набор функций доступных разработчикам и специально сформированные структуры данных. Все подгружаемые заголовочные файлы называются по имени алгоритма шифрования, который они описывают (например, файл salsa.h соответствует криптографическому шифру Salsa). Библиотека estream.h поддерживает два основных порядка байтов: big-endian и little-endian. Кроме того, в библиотеке предусмотрены функции очистки оперативной памяти от оставшейся после работы информации (например, ключей шифрования).

Главное достоинство потоковых криптоалгоритмов – высокая скорость шифрования. Поэтому в процессе создания библиотеки исходные коды были изменены по сравнению с авторскими реализациями для увеличения скоростных показателей. Были использованы следующие механизмы модернизации алгоритмов.

1. Макросы. Они используются для уменьшения дублирования программного кода и облегчения понимания исходного кода, когда макросом заменяется часть сложной составной инструкции. Если препроцессор при обработке исходного кода встречает символьное имя, которое определено с помощью директивы #define, то он подставляет значение макроса вместо него. Это можно использовать для увеличения скорости работы алгоритма. Вместо многократного использования функции, при котором затрачивается время на ее вызов, можно использовать макрос. В ходе обработки файла препроцессор сам подставит необходимый программный код, что позволяет получить выигрыш по времени, особенно, если набирается большое количество вызовов функции.

2. Использование битовых операций позволяет увеличить производительность работы алгоритмов, так как все операции производятся непосредственно с битами машинного слова. В рамках разработки библиотеки использовались следующие операции: левый и правый битовый сдвиг, побитовое отрицание, побитовое сложение и побитовое умножение.

3. При изучении официальной документации и исходных кодов реализаций алгоритмов, предложенных их авторами и находящихся в открытом доступе, были найдены участки кода, изменение которых позволило увеличить производительность алгоритмов.

В алгоритмах Salsa и Sosemanuk была модифицирована функция шифрования/расшифровывания. Уменьшено количество обращений к входному потоку данных за счет использования 32-битных массивов машинных слов, генерируемых за одну итерацию алгоритма. После модернизации функция шифрования в алгоритме Salsa способна за один вызов обработать до 512 бит данных вместо 64-х бит, а в алгоритме Sosemanuk – до 640 бит за один проход вместо 80 проходов по 8 бит.

В алгоритме Trivium разработчиками предложено использовать один макрос для инициализации начального состояния алгоритма и генерации ключевой последовательности. При инициализации алгоритма происходит генерация 32-х битов ключевой последовательности, которая на данном этапе не используется. Также авторы применяли два макроса с использованием кроссплатформенных преобразований для загрузки и выгрузки битов секретного ключа и вектора инициализации при каждой итерации цикла. В реализации алгоритма Trivium в библиотеке estream.h исключена генерация ключевой последовательности при инициализации алгоритма, а также произведена замена макросов выгрузки битов на стандартную функцию языка программирования C – memcpy. Это, в частности, позволяет улучшить возможности работы с небольшими блоками данных (отдельными пакетами) в условиях потерь, когда чаще приходится выполнять инициализацию.

Увеличение скорости получено в алгоритме Grain путем применения для вычисления битов ключевой последовательности макросов, работающих с массивами регистров не через взаимодействие со структурой данных, а напрямую с битами регистров.

Описанные механизмов модернизации позволили увеличить скоростные показатели некоторых шифров, по сравнению с реализациями, предложенными авторами алгоритмов.

Время, затраченное на шифрование, измерялось с помощью UNIX-утилиты time, которая производит три измерения: реально затраченное пользовательское время, процессорное время, затраченное без учета времени вызова системных функций и процессорное время, затраченное процессом для вызова системных функций. Более точное реальное процессорное время, затраченное на шифрование, вычисляет второй режим [4].

Для измерений использовались объемы данных 0,5 МБ, 10 МБ и 1 ГБ. Измерения производились на различных процессорах, в частности Intel Atom N2600 Dual Core с тактовой частотой 1,66 ГГц, Intel Quad Core i7 4710HQ с тактовой частотой 2,5 ГГц, AMD Phenom Triple Core 8450 с тактовой частотой 2,1 ГГц.

Для сравнения результатов времени работы алгоритмов было использовано среднее арифметическое значение всех полученных измерений времени по каждому шифру.

На рис. 2 представлен пример некоторых полученных диаграмм с результатами измерений на объеме данных 1 ГБ для алгоритмов Salsa, Rabbit, HC128, Sosemanuk и Trivium. На рисунках введено обозначение estream.h – библиотека, разработанная в ходе выполнения данной работы, eStream – программные реализации алгоритмов, предложенные авторами шифров.

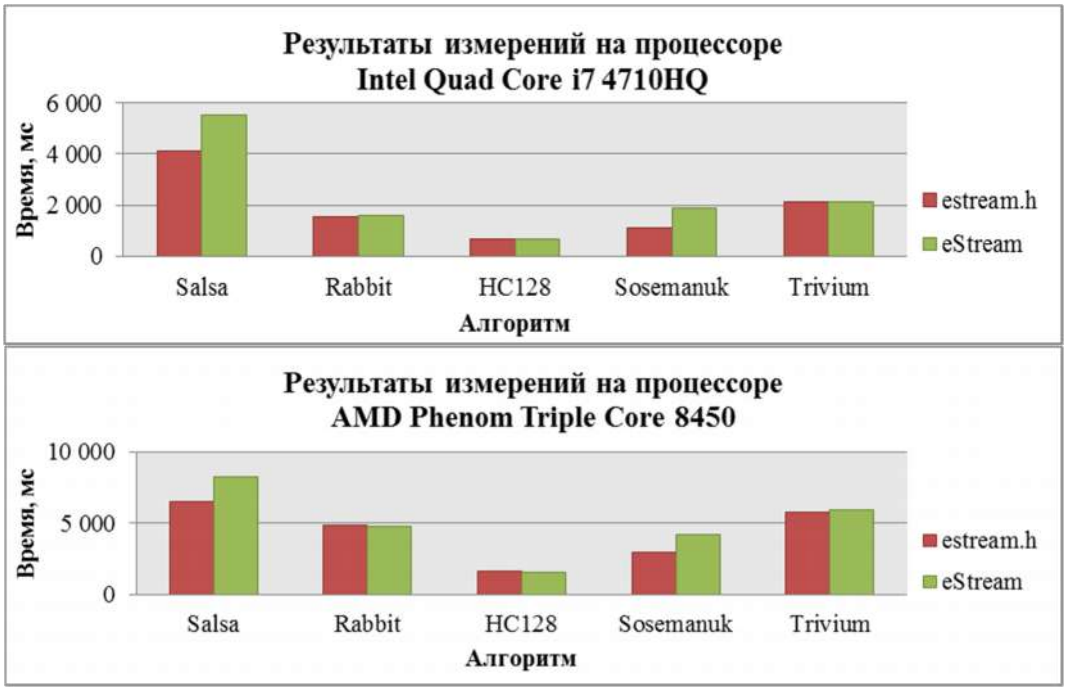

Рис. 2. Результаты измерений на различных процессорах (объем данных 1 ГБ)

Аппаратно-ориентированные шифры Mickey и Grain тестировались на объемах данных 0,5 и 10 МБ, так как их программные реализации значительно уступают остальным алгоритмам. Результаты тестирования для двух процессоров представлены на рис. 3.

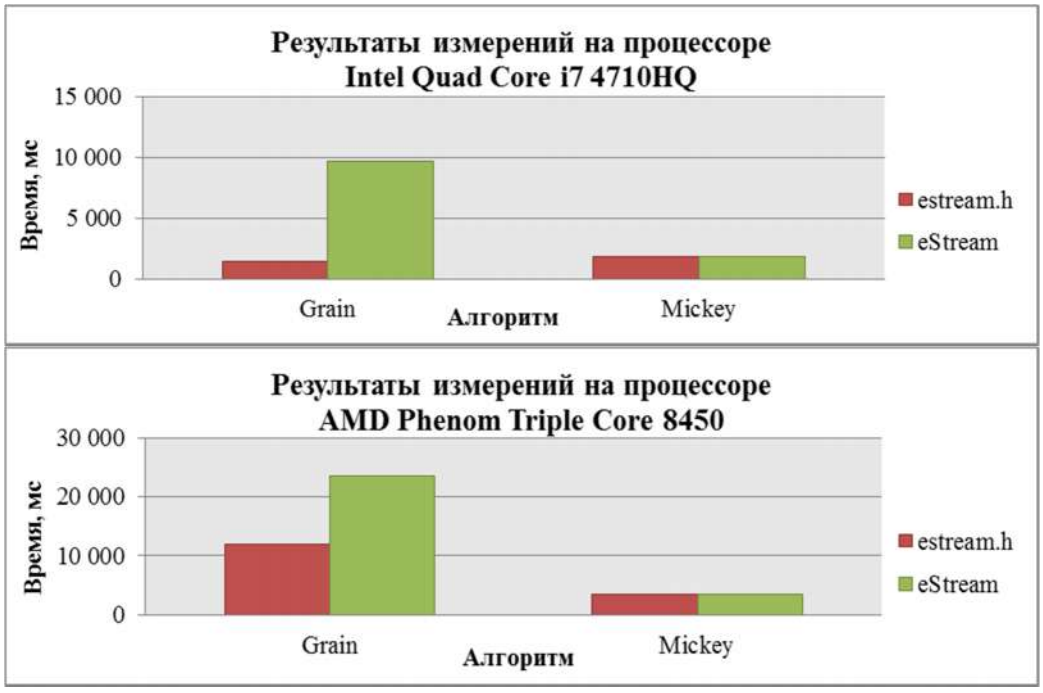

Рис. 3. Результаты измерений для процессоров Intel Quad Core i7 4710HQ и AMD Phenom Triple Core 8450.

Проведенный анализ показывает, что предложенные в работе реализации алгоритмов Salsa и Sosemanuk работают на 20–35 % эффективнее реализаций, предложенных авторами, Trivium – на 3–5 % эффективнее, Grain – примерно в 2,9 раза быстрее

обрабатывает информацию, чем реализация из проекта eSTREAM. HC128 и Mickey демонстрируют результаты, совпадающие в пределах погрешности с результатами реализаций авторов, представленных в eSTREAM. Реализация Rabbit из библиотеки estream.h немного уступает авторской из-за использования в последней ассемблерных вставок. Программные реализации Grain и Mickey значительно уступают программным реализациям других шифров.

Ключевым моментом разработки криптобиблиотеки является проверка алгоритмов на соответствие стандартам. Для тестирования корректности реализации алгоритмов была разработана терминальная программа, подающая на вход библиотеки тестовые векторы для каждого реализованного алгоритма. Результаты проверки показали, что алгоритмы генерируют одинаковую с авторскими реализациями ключевую последовательность.

Кроме того, тестируемые файлы шифровались с использованием разработанной библиотеки, а затем расшифровывались с помощью программ, скомпилированных из исходных кодов авторов шифров. Затем сравнивались их хэш-суммы. Совпадение подтверждает соответствие реализации алгоритмов из библиотеки estream.h стандарту.

В ходе выполнения работы, все реализации алгоритмов шифрования, представленные в библиотеке estream.h, были проверены на соответствие стандартам описанными способами.

Для демонстрации возможностей библиотеки разработана программа для шифрования/расшифровывания файлов и потоков данных в режиме реального времени с графическим интерфейсом для операционных систем семейства Windows и терминальная версия для операционных систем семейства *nix.

Заключение

Была разработана криптобиблиотека потокового шифрования estream.h на основе алгоритмов проекта eSTREAM. Она включает модифицированные реализации семи алгоритмов: Salsa, Rabbit, HC128, Sosemanuk, Grain, Mickey, Trivium. Библиотека estream.h готова к использованию в составе более крупных проектов, либо как самостоятельный инструмент для шифрования потоков информации.

СПИСОК ЛИТЕРАТУРЫ

1. Cryptography Research and Evaluation Committees. [Электронный ресурс] URL: http://www.cryptrec.go.jp/english/index.html (дата обращения: 13.07.2015).
2. Robshaw M., Billet O. New Stream Cipher Designs: The eStream Finalists (Lecture Notes in Computer Science). Paris: Springer Science+Business Media, 2008.
3. eSTREAM: the ECRYPT Stream Cipher Project [Электронный ресурс] URL: http://www.ecrypt.eu.org/stream/index.html (дата обращения: 13.07.2015).
4. Проект OpenNet: справочное руководство по утилите time. [Электронный ресурс] URL: http://www.opennet.ru/man.shtml?topic=time&category=1 (дата обращения: 05.05.2015).

REFERENCES:

1. Cryptography Research and Evaluation Committees. URL: http://www.cryptrec.go.jp/ english/index.html (access date: 13.07.2015).
2. Robshaw M., Billet O. New Stream Cipher Designs: The eStream Finalists (Lecture Notes in Computer Science). Paris: Springer Science+Business Media, 2008.
3. eSTREAM: the ECRYPT Stream Cipher Project. URL: http://www.ecrypt.eu.org/ stream/index.html (access date: 13.07.2015).
4. OpenNet Project: MAN time. URL: http://www.opennet.ru/man.shtml?topic=time&category=1 (access date: 05.05.2015).