Заключение

Предложенные в работе принципы позволили создать программно-аппаратный комплекс, позволяющий проводить непрерывный мониторинг качества обслуживания в сети телефонной связи и своевременно устранять возникающие неполадки, не дожидаясь жалоб абонентов, что позволяет сократить расходы на обслуживание сети. Полученная система не уступает по характеристикам более дорогим аналогам и уже зарекомендовала себя как в практическом использовании на сетях операторов связи «Волгателеком» и «Сибирьтелеком», так и при обучении студентов в лабораториях телефонной связи Петербургского государственного университета путей сообщения и Петербургского Энергетического института повышения квалификации.

Литература

1. Павловский Е.А. Обзор систем сигнализации на цифровых сетях оперативно-технологической связи // Сборник трудов научной конференции «Шаг в будущее-2005». – СПб.: ПГУПС, 2005. – С. 183-185.

2. Ловягина О.Г. Эволюция распределенного мониторинга сети ОКС-7 // Вестник связи. – 2006. – № 12. – С. 26-32.

3. Villy B. Iversen. Teletraffic Engineering and Network Planning. – Technical University of Denmark, 2011. – 583 с.

4. Росляков А.В. ОКС № 7. Архитектура, протоколы, применение. – М.: Эко-Трендз, 2008. – 320 с.

Головашич С.А., Евсеев С.П., Король О.Г.

Эффективная реализация блочных симметричных шифров

(ООО «Криптомаш», ХНЭУ, Харьков, Украина)

Вступление. Наиболее распространённым подходом предварительного (экспресс) анализа современных криптоалгоритмов в целом и блочно-симметричных шифров (БСШ) в частности является оценка трёх показателей: устойчивости алгоритма к известным криптоаналитическим атакам, производительности программной реализации алгоритма на современных персональных компьютерах и «статистической безопасности» (формирование «хороших» псевдослучайных последовательностей). Однако данный подход оказывается недостаточным, когда речь идёт о создании/выборе алгоритма, претендующего на роль национального стандарта, а значит предполагающего его массовое внедрение в различных сферах применения. В таком случае, третьим (а иногда даже вторым) по значимости критерием, после криптостойкости и производительности, становится «стоимость» реализации алгоритма на различных программно-аппаратных платформах. При этом если не значительное увеличение производительности приводит существенному увеличению «стоимости», первой можно пренебречь. «Стоимость» реализации криптографического алгоритма можно разбить на: одноразовые интеллектуальные затраты и постоянные материальные. Первые определяются затратами конечных разработчиков на кодирование алгоритма и, при массовом производстве, могут быть проигнорированы. Вторые обусловлены теми требованиями, который предъявляет алгоритм к аппаратным ресурсам (вычислительным и/или памяти) для достижения приемлемой производительности. Особенно ощутимыми «постоянные материальные затраты» становятся при полностью аппаратной реализации алгоритма, либо в составе некоторого «встроенного» решения. Тогда увеличение требований криптоалгоритма приводит к сокращению ресурсов, доступных для реализации основной бизнес логики на интересуемой аппаратной платформе и необходимости перехода к более ресурсоёмкой, а следовательно, и более дорогой платформе.

Целью исследования является оценка эффективности реализации блочно-симметричных шифров на основе интегрального анализа оценок эффективности проектирования (показателя «стоимости» достижения необходимой криптостойкости при заданной производительности).

Задачу проектирования практического алгоритма блочно-симметричного шифра (БСШ) следует рассматривать как задачу минимизации «затрат» на реализацию криптопреобразования, обеспечивающего необходимые показатели криптостойкости.

Под «затратами» будем понимать перечень и объём ресурсов целевой аппаратной платформы, необходимых для реализации на ней анализируемого алгоритма БСШ. Нас будут интересовать следующие аппаратные ресурсы:

- вычислительные ресурсы (единица измерения – количество тактов либо микроопераций вычислителя, необходимых для обработки единицы информации; в случае БСШ – это шифрование одного блока данных или «разворачивания» одного пользовательского ключа) – отражают производительность;
- ресурсы оперативной памяти (RAM, единица измерения – байт или КБайт) – необходимы для хранения, используемого в процессе вычислений, ключевого материала и промежуточных данных;
- ресурсы энергонезависимой перезаписываемой памяти (EEPROM, единица измерения – байт или КБайт) – необходимы для хранения исходных ключевых данных (пользовательских ключей), включая таблицы подстановки, если они являются сменными и представляют собой долговременный ключ шифрования;
- ресурсы постоянной памяти (ROM или PROM, единица измерения – байт или КБайт) – необходимы для хранения исполнимого машинного кода, реализующего алгоритм БСШ, а также всех фиксированных параметров алгоритма, т.е. всех констант, включая таблицы подстановок, если они фиксированные. В приведенном выше списке ресурсы расположены в порядке убывания «ценности» (для большинства аппаратных платформ).

Учитывая, что в настоящий момент не принято окончательное решение о том, должны ли таблицы подстановок быть сменным параметром алгорит-

ма или нет, и если – да, то в каком виде («только S-блоки» или «S-блоки + MBN-полином»), будем их учитывать как константные значения (т.е. в составе ресурса ROM).

При анализе требуемого объёма ресурсов необходимо выполнить оценку для каждого класса аппаратных платформ, на которых может потребоваться реализация заданного криптоалгоритма. Основные программно-аппаратные платформы, получившие широкое распространение сегодня можно разделить на три класса:

- 8/16-битные микроконтроллеры и смарт-карты;
- 32-битные микропроцессоры и микроконтроллеры (ARM, IA 32);
- 64-битные процессоры общего назначения (AMD64, Intel 64).

Реализация алгоритма БСШ на микроконтроллере узкоспециализированного назначения, как правило, накладывает ограничения на все перечисленные выше ресурсы. Так, наиболее технически защищённая и удобная аппаратная платформа массового применения – бесконтактная смарт-карта, обладает предельно ограниченными ресурсами. Наиболее современные представители характеризуются следующими показателями:

- CPU: 8/16 bit, с частотой до 30 MHz;
- RAM: 4-8 KB;
- ROM: 100-250 KB;
- EEPROM: 8-80 KB.

Однако применение защищённой SoC-технологии (System-on-Chip) позволяет обеспечить наибольшую степень защиты секретных ключей. Это достигается за счёт поддержки полного жизненного цикла секретных ключей исключительно в памяти чипа, интегрирующего в себе: CPU, RAM, EEPROM, FLASH/ROM и TRNG (сертифицированный физический датчик случайных чисел), и при этом технологически защищенного от пассивных и активных атак класса Side-Channel. Чипы смарт-карт получили широкое распространение в сфере финансовых услуг и персональных систем контроля доступа широкого назначения. Поэтому возможность эффективной реализации разрабатываемого алгоритма на смарт-карте является необходимым условием его массового применения.

В качестве опорных характеристик для оценки эффективности проектирования будем использовать:

1) уязвимость к известным криптоаналитическим атакам (значение показателя – да/нет);

2) теоретические недостатки конструкции алгоритма, негативно влияющие на его криптографические свойства, но не имеющие известного практического применения в виде криптоаналитической атаки (значение показателя – количество и перечень недостатков);

3) количественные характеристики каждого из необходимых аппаратных ресурсов;

4) конструктивные особенности построения БСШ, приводящие к избыточным накладным расходам некоторого из ресурсов вычислителя, и при этом не обоснованные повышением уровня криптостойкости алгоритма (значение показателя – количество и перечень «избыточных» конструкций).

Эффективность конструкции БСШ можно оценить соотношениями:

- «противодействие криптоаналитическим атакам» / «расходы на реализацию»;
- «противодействие криптоаналитическим атакам» / «быстродействие».

Если защищённость от криптоаналитических атак оценивать по двоичной шкале – «есть эффективная атака» или «нет», то оценка эффективности проектирования БСШ может быть сведена к сравнительному анализу указанных выше опорных характеристик для алгоритмов, не имеющих известных уязвимостей к атакам криптоанализа. Если алгоритм уязвим к некоторой атаке криптоанализа, то его дальнейший анализ целесообразен только после добавления противодействия этой атаке.

В качестве критерия «первичного отбора» предлагается использовать возможность эффективной реализации алгоритма на микроконтроллере или смарт-карте с ограниченными ресурсами. Т.е. алгоритм должен быть эффективно реализуем на устройстве при условии доступного объёма RAM не более 3КБ (37,5% от общего объёма RAM, присутствующего на самой совершенной бесконтактной смарт-карте – 8КБ) и доступного объёма ROM не более 40КБ (16% от общего объёма ROM для максимального случая – 250КБ). Отметим, что выбранные пороговые значения являются существенно завышенными, и их достижение, на практике, означает непригодность алгоритма для реализации на смарт-картах. *Анализ сложности реализации производительности.* Для оценки стоимости программной/аппаратной реализации каждого из алгоритмов предлагается определить общие требования к эффективной реализации алгоритма:

- Приоритеты оптимизации:
  1) максимальная производительность;
  2) минимальные аппаратные затраты.
- Анализируемые аппаратные ресурсы:
  1) энергонезависимая память;
  2) оперативная память;
  3) система команд арифметико-логического устройства (АЛУ) процессора, либо количество логических элементов (для FPGA).
- Общие техники оптимизации:
  1) все уникальные значения цикловых ключей формируются схемой разворачивания ключа, т.е. процедура шифрования не выполняет формирования цикловых ключей «налету»;
  2) общая табличная реализация S-блока и умножения на полином CMBN-кода;
  3) реализация байтовых перестановок, где это возможно, «явным» способом, т.е. с нулевыми вычислительными затратами;
  4) табличная реализации байтового сдвига строк CMBN-матрицы, для архитектур, где это оправдано;
  5) применение техник «разворачивания» циклов и «inline-подстановки» подпрограмм.

Таким образом, интересны следующие микрооперации ALU, используемые в алгоритмах БСШ:

- сложение по модулю 2 (XOR);
- сложение с переносом, т.е. по модулю 2^m, где m > 1 (ADD);
- вычитание с переносом, т.е. по модулю 2^m, где m > 1 (SUB);
- сдвиг вправо (ROTR);
- сдвиг влево (ROTL);
- пересылка машинного слова между регистрами либо регистром и памятью (MOV).

Изучив возможности наиболее распространённых современных аппаратных платформ, для оценки производительности данных операций, необходимо принять ряд допущений относительно вычислительных возможностей обобщённой модели микропроцессора:

- все элементарные операции ALU выполняются за 1 такт;
- ALU поддерживает косвенную адресацию (при этом смещение, относительно базового адреса, должно хранится в «индексном» регистре процессора);
- процессор поддерживает операции сложения и вычитания с учётом флага переноса, сформированного предыдущей командой (соответственно команды ADC и SBB);
- каждая команда ALU может оперировать не более чем двумя операндами (один операнд «получатель», второй – «источник», опционально);
- ALU не поддерживает операции типа «память-память», т.е. когда «источник» и «получатель» находятся в ОЗУ;
- если разрядность процессора составляет R байт, то каждая команда микропроцессора может манипулировать данными размером от 1 до R байт (в зависимости от разрядности обрабатываемых данных);
- процессор не имеет эффективной поддержки команд с произвольной адресацией отдельных битов или байтов, составляющих машинное слово в регистре процессора.

Затраты RAM для хранения цикловых ключей. Объём оперативной памяти (RAM), размещённый на кристалле, на ряду с FLASH/EEPROM-памятью, фактически, определяет стоимость миниатюрного микроконтроллера или смарт-карты, поэтому минимизация затрат RAM является весьма критичной для БСШ.

Основные затраты RAM приходятся на хранение «развёрнутых» цикловых ключей, кроме того дополнительная память требуется для хранения текущего состояния шифратора (LB/8) и «вспомогательный» буфер для эффективной реализации F-функции (размер буфера соответствует размерности F-функции). Указанный «вспомогательный» буфер необходим для реализации байтовых перестановок с «нулевыми» вычислительными затратами. Для хранения текущего (промежуточного) состояния шифра может использоваться буфер вызывающего кода. Учитывая, что вызывающая шифратор процедура должна обеспечить передачу исходного блока и приём результирующего блока в некоторой области памяти, целесообразно использовать выходной

интерфейсный буфер для хранения промежуточных данных (текущего состояния шифруемого блока).

Таким образом, затраты RAM непосредственно шифратора будут включать только память для хранения цикловых ключей и «вспомогательного» буфера F-функции.

Затраты памяти для хранения вспомогательных констант. Все алгоритмы БСШ используют некоторые константные значения. Эти данные относятся к постоянным и, безусловно, размещаются в ROM, в отличие от таблиц S-блоков и SL-преобразования, которые могут размещаться в EEPROM, в случае сменного S-блока / SL-преобразования. Следует отметить, что абсолютное значение данной оценки незначительно влияет на общую стоимость программной реализации алгоритма, т.к. указанные затраты приходятся на самый «дешёвый» ресурс – ROM, однако избыточное использование констант свидетельствует о недостаточной продуманности архитектуры шифратора и попытке «залатать» бреши в структурных свойствах алгоритма БСШ.

Для оценки сложности функций шифрования предлагается ввести понятие сложности KSL-преобразования, включающего в себя последовательность из четырёх базовых преобразований:

- сложение блока данных с ключом (операция ADD либо XOR);
- перестановку байтов («Pbyte» или «ShiftRows»);
- табличную подстановку каждого байта, блока данных по таблице S-блока;
- линейное смешивание байтов (MBN-преобразование).

Последние два преобразования на практике объединяются в одно и реализуются посредством общей таблицы подстановок, а байтовая перестановка реализуется без накладных вычислительных затрат. Следует отметить, что вычислительная сложность объединённого SL-преобразования зависит от длины CMBN-полинома, только если длина полинома больше разрядности процессора. Затраты на реализацию SL-преобразования всегда кратны количеству байтов, которое оно охватывает. Отметим, что реализация преобразования PSL_8×8 на 8-битном микроконтроллере будет очень сильно зависеть от его архитектуры, а именно – количества доступных регистров общего назначения и способа адресации блоков памяти размером более 256 байт. При этом разброс в сложности может достигать нескольких раз. Учитывая, что алгоритмы БСШ могут на каждой итерации манипулировать полублоком, в качестве единицы измерения сложности алгоритмов выберем сложность преобразования HKSL = KSL(LB/2), т.е. сложность KSL-преобразования (AES-подобного циклового преобразования) для блока половинной длины. Отметим, что применение относительной оценки сложности (HKSL) позволяет:

- получить сравнительную оценку сложности алгоритмов, независящую от деталей реализации подобных преобразований на конкретном CPU;
- с минимальными усилиями «переносить» сравнительную оценку сложности с одной архитектуры на другую (за счёт уже выявленной общности алгоритмов);

• анализировать зависимость сложности преобразования от L_B и L_K, без «жёсткой» привязки к архитектуре;

• выполнять сравнительную оценку не только производительности алгоритмов, но также и объёма исполнимого кода, необходимого для реализации, как отдельных преобразований, так и всего алгоритма; а также, косвенно, оценивать энергозатраты на выполнение криптопреобразования. Отметим, что в случае полностью 32-битной реализации, для самого последнего SL-преобразования, в пределах каждой F-функции, финальная загрузка результата в память может не выполняться.

Суммарная сложность *реализации БСШ.* Суммарная оценка будет отражать общий объём исполнимого кода программной реализации БСШ, при условии, что используется стратегия оптимизации, направленная на достижение максимальной производительности всех трёх процедур (зашифрования, расшифрования и установки ключа). Отметим, что для инволютивных шифров данная оценка также будет отражать суммарную сложность зашифрования одного блока с предварительной установкой ключа, а для не инволютивного шифра – сложность зашифрования и расшифрования по одному блоку с предварительной установкой ключа.

Производительность «эталонных» реализаций. Для верификации теоретических оценок сложности алгоритмов предлагается провести экспериментальное исследование показателей производительности эталонных реализаций БСШ и решить три задачи:

• выполнить сравнение производительности рассматриваемых алгоритмов на современной суперскалярной архитектуре процессора, позволяющей задействовать заложенные в алгоритм БСШ возможности распараллеливания;

• сравнить производительность алгоритмов как на 32-битной программно-аппаратной платформе (наиболее распространённой на сегодняшний день, но уже морально устаревшей), так и на перспективной 64-битной платформе;

• оценить пиковую производительность алгоритмов в серверных приложениях, ориентированных на многопользовательское обслуживание.

Все массово выпускаемые сегодня процессоры архитектуры x86 для рабочих станций, ноутбуков и серверов поддерживают 64-битное расширение системы команд, разработанное компанией AMD (архитектура AMD64) и лицензированное компании Intel (архитектура Intel 64). С другой стороны, серверные процессоры конкурирующей архитектуры UltraSPARC сегодня также являются 64-битными. Более того, даже процессоры, ориентированные на сектор «энергосберегающих» вычислений, уже поддерживают 64-битную систему команд – VIA Nano (архитектура Isaiah). В связи с этим для тестирования алгоритмов БСШ предлагается выбирать именно 64-битную программно-аппаратную платформу.

Выводы. Таким образом, представленный подход позволяет выполнить сравнительный анализ показателя эффективности проектирования для достижения необходимой криптостойкости при заданной производительности алгоритмов БСШ.

Литература

1. Report on the Development of the Advanced Encryption Standard (AES): http://csrc.nist.gov/archive/aes/.

2. New European Schemes for Signatures, Integrity, and Encryption: http://cryptonessie.org/.

3. Pascal Junod, Serge Vaudenay. FOX: a New Family of Block Ciphers. http://crypto.junod.info/sac04a.pdf

4. Federal Information Processing Standards Publication 197 (FIPS PUB 197). Specification for the Advanced Encryption Standard (AES) // Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory. November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips – 197.pdf

Евсеев С.П., Дорохов А.В.

Обзор механизмов обеспечения безопасности и достоверности данных в информационных системах

(ХНЭУ, Харьков, Украина)

Проблема защиты информационных систем от несанкционированного доступа в современных условиях приобрела особую остроту.

Стремительное развитие коммуникационных и вычислительных технологий позволяет строить сложные информационные системы с распределенной архитектурой, объединяющие большое количество сегментов, расположенных на значительном удалении друг от друга.

Все это вызывает увеличение числа узлов сетей и количества различных линий связи между ними, что, в свою очередь, повышает риск несанкционированного подключения к информационной системе и доступа к конфиденциальной информации [1–3].

Увеличение объемов обрабатываемых и передаваемых данных в компьютерных системах и сетях, прежде всего в банковских системах, в системах управления крупными финансовыми и промышленными организациями, предприятиями энергетического сектора, транспорта, в системах управления и связи военного назначения требует новых подходов в организации протоколов и механизмов обеспечения безопасности передаваемых данных.

Естественное требование к безопасности и достоверности обрабатываемой и передаваемой информации в таких системах стоит очень остро, поскольку отказ системы или выход за установленные ограничения указанных параметров может привести к значительным финансовым и материальным потерям, снижению обороноспособности, ущербу экологии, жизни, здоровью людей.