Анализ аппаратной поддержки криптографии при построении информационной безопасности вуза

Егор Иванович Пристанских
студент
Московский государственный университет им. М.В. Ломоносова
Москва, Россия
79044097106@yandex.ru
0000-0000-0000-0000

Олег Александрович Кудрявцев
студент
Национальный исследовательский университет ИТМО
Москва, Россия
Ibezifooo@gmail.com
0000-0000-0000-0000

Даниил Евгеньевич Андреев
студент
Национальный исследовательский университет ИТМО
Москва, Россия
Poshackinfod@gmail.com
0000-0000-0000-0000

Юлия Владимировна Таран
студент
Национальный исследовательский университет ИТМО
Москва, Россия
Taran@mail.ru
0000-0000-0000-0000

Никита Сергеевич Соловьев
студент
Национальный исследовательский университет ИТМО
Москва, Россия
Nikki.Soloviev@gmail.com
0000-0000-0000-0000

Поступила в редакцию 29.04.2022
Принята 21.05.2022
Опубликована 20.06.2022

10.25726/h2048-6130-4735-p

Аннотация

Проблему имплементации традиционных криптоалгоритмов во встроенных системах сначала пытались решить на программном уровне, для чего максимально оптимизировали код за счет использования языка ассемблера и особенностей архитектуры процессора. Сформировался ряд криптобиблиотек, ориентированных на ВС и IoT, самые известные из которых WolfSSL, OpenSSL, GUARD TLS. Tiny / Toolkit, Cifra, содержат реализации как отдельных алгоритмов так и целых протоколов

с умеренными требованиями к ресурсам вычислителя. Вместе с тем и сама криптография пыталась подстроиться под требования ИОТ, и в начале 2000-х годов выделилось такое отдельное направление как легковесная или малоресурсная криптография (Lightweight Cryptography) для устройств с ограниченными ресурсами. При создании легковесных криптоалгоритмов на первое место выходит стоимость реализации при адекватном уровне защиты и нужной производительности, то есть важен компромисс между этими тремя параметрами, который зависит от конкретных требований к устройству. По сравнению с классическими алгоритмами, легковесные алгоритмы за счет уменьшения размера ключа, количества раундов, замены более сложных операций проще или отказа от них позволяют существенно увеличить производительность и снизить требования к ресурсам реализации. В качестве примера программно-ориентированных легковесных алгоритмов, получивших значительную популярность в последние годы можно указать шифры ChaCha20, Speck, хэш Blake2, Mac-функцию Poly1305 и др. Кроме того появились режимы работы шифров, призванные комплексно и с минимальными накладными расходами обеспечить конфиденциальность, целостность и аутентификацию пакетизированных данных.

Ключевые слова
данные, криптография, защита данных, вуз.

Введение

В первую очередь здесь стоит отметить режим GCM (Бауэр, 2007), что относится к режимам аутентифицированного шифрования с присоединенными данными (Authenticated Encryption with Associated Data, AEAD), который позволяет в одном алгоритме совместить операции шифрования (AES в режиме счетчика (CTR) и выработки MACC-кода на основе функции GHASH (умножение в поле Галуа), при этом часть данных (заголовок) остаются в открытой форме, однако весь пакет является полностью аутентифицируется.

Режим AEAD-AES-GCM стал стандартом де-юре во многих интернет-протоколах (в частности TLS, IPSec) и де-факто для многих криптобиблиотек и аппликаций. С целью совершенствования этого режима в 2013 г. анонсирован открытый конкурс CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) призван сформировать портфолио AEAD-шифров, которые бы превосходили по быстродействию AES-GCM и были пригодны для широкого использования (Бунин, 2003).

В финал вышли 7 алгоритмов из 57 и победители должны были быть объявлены в декабре 2017 г. Однако на момент написания статьи (март 2018 г.) этого еще не произошло.

Несмотря на то, даже высокопроизводительные микропроцессоры общего назначения (Intel, AMD) с высокими тактовыми частотами, большими объемами оперативной и кэш-памяти, мощной системой команд и поддержкой многопоточности столкнулись с проблемой недостаточной производительности при реализации криптоалгоритмов. Для решения этой проблемы производители стали перемещать криптографическую обработку данных в аппаратные блоки своей продукции – криптоакселераторы. (Алферов, 2001)

Ускоренная аппаратная криптографическая обработка вместо программного выполнения этих же алгоритмов позволяет существенно разгрузить центральный процессор.

Материалы и методы исследования

Примером такого подхода является расширение системы команд x86 шестью командами AES-NI (AESNew Instructions): AESENC, AESENCLAST, AESDEC, AESDECLAST, AESKEYGENASSIST, AESIMC с целью ускорения приложений, использующих AES-шифрование (Дошина, 2015). Сочетание AES-NI с инструкцией умножения в полях Галуа PCLMULQDQ, для эффективного вычисления функции GHASH, позволило существенно увеличить быстродействие в режиме AEAD-AES-GCM (Lieven, 2001).

Еще один способ ускорения криптографических операций благодаря параллельным вычислениям – это использование векторных инструкций, позволяющих выполнять несколько операций за один такт процессора (Комиссаренко, 2016).

Расширенный недетерминированный ГВЧ предназначен для того, чтобы сделать доступным сгенерированные в блоке подготовки зародыши для использования в других программных средствах. Данные поступают в буфер, с которого считываются инструкциями RDSEED.

Аналогичный ГВЧ также реализован в процессорах AMD в составе AMD Secure Processor.

В микропроцессорах фирмы Intel также присутствует еще один криптоакселератор: модуль Intel Secure Key – это условное название для новых инструкций RDRAND и RDSEED и встроенного в процессор аппаратного генератора случайных чисел, который реализует (Susan Decker, 2018). Intel называет его «цифровой генератор случайных чисел» (Digital Random Number Generator, DRNG).

DRNG можно разбить на три логических уровня:

1. Источник энтропии, которое производит случайные биты с недетерминированного аппаратного процесса с использованием теплового шума в полупроводниках и передает их блоку подготовки.
2. Блок подготовки данных по алгоритму AES-CBC-MAC, который осуществляет маскировку потенциальных статистических дефектов. Сгенерированное 256-битное значение используется как зародыш на следующем уровне для инициализации генератора псевдослучайных чисел DRBG.
3. Deterministic Random Bit Generator. Генерирует случайные данные большого объема с высоким быстродействием (до 6 Гбит/сек), используя стандартный алгоритм CTR-DRBG на базе AES. Данные поступают в буфер, с которого считываются инструкциями RDRAND.
4. Enhanced Nondeterministic Random Number Generator. Расширенный недетерминированный ГВЧ предназначен для того, чтобы сделать доступным сгенерированные в блоке подготовки зародыши для использования в других программных средствах. Данные поступают в буфер, с которого считываются инструкциями RDSEED.

Аналогичный ГВЧ также реализован в процессорах AMD в составе AMD Secure Processor.

Во встроенных системах криптоалгоритмы долгое время реализовались только программным образом, и лишь относительно недавно массово начали появляться интегрированные в микроконтроллеры криптоакселераторы, которые будут рассмотрены в статье. Использование криптоакселераторов дает следующие преимущества: более высокое быстродействие и энергоэффективность, разгрузка центрального процессора, экономия памяти, большая устойчивость к Side-Channel Attacks (в первую очередь CPA и DPA).

Чтобы иметь базу для сравнения с аппаратными ускорителями кратко рассмотрим программные реализации симметричных и асимметричных криптоалгоритмов в ВС. Оценке быстродействия и требований к памяти наиболее распространенных из них для различных микроконтроллерных архитектур посвящены многочисленные исследования (Шаньгин, 2012).

Результаты и обсуждение

Рассматриваются влияния как архитектуры, так и различных путей оптимизации на уровне алгоритма и компилятора, на производительность и объем необходимой памяти (Ященко, 2012).

Целью статьи является сравнительный анализ криптоакселераторов в наиболее распространенных 8/16/32-битных семьях микроконтроллеров, с точки зрения быстродействия и гибкости работы, что позволит обоснованно выбирать оптимальное решение при разработке механизмов защиты в IoT-устройствах.

Криптоакселераторы в 8-битных микроконтроллерах

AVR. Все микроконтроллеры AVR семейства XMega фирмы Atmel оснащены криптоакселераторами блочных симметричных шифров DES и AES (Nechvatal, 2001).

В частности, в системе команд микроконтроллеров XMega предусмотрена инструкция DES K, которая соответствует одному (K-му) из 16-ти раундов алгоритма DES. Входные данные для команды

располагаются в регистрах общего назначения (РЗП) R0-R7, ключ записывается в РЗП R8-R15. Флажок H регистра состояния SREG задает тип операции: H = 0 – зашифрования, H = 1 – расшифровка.

Кроме поддержки алгоритма DES на уровне системы команд в микроконтроллерах XMega реализована аппаратная поддержка алгоритма AES с помощью криптомодулей AES.

Криптомодуль AES является периферийным модулем, который шифрует данные блоками по 128 бит с помощью 128-битного ключа. Соответственно криптомодуль AES имеет память для хранения блока данных (AES State Memory) и ключа (AES Key Memory). Доступ к этим областям памяти осуществляется через регистры ввода-вывода AES_State и AES_Key. Управление и взаимодействие с модулем осуществляется через регистр управления CTRL и регистр статуса STATUS.

Поддерживаемые режимы работы алгоритма AES-ECB, CBC. Наличие DMA-контроллера прямого доступа в память (Direct Memory Access, DMA) позволяет выполнять пересылки входных и выходных данных без вмешательства центрального процессора (Яковлев, 2006).

i8051. Микроконтроллеры семейства C8051F96x фирмы Silicon Labs с процессорным ядром и8051 содержат криптоакселератор алгоритма AES с поддержкой ключей длиной 128, 192 и 256 бит и напрямую могут работать в режимах ECB, CBC, CTR (Susan Decker, 2018).

Криптоакселератор состоит из таких элементов:

- ядра – выполняет зашифровку, расшифровку и порождение ключа расшифровки;
- конфигурирующих регистров – задают длину ключа, начало преобразования и маршрут следования данных;
- регистров ключа, входных и выходных данных;
- входного и выходного мультиплексоров с блоками выполнения операции XOR;
- внутреннего конечного автомата.

STM8. В микроконтроллерах семей STM8L16 и STM8AL присутствует криптоакселератор алгоритма AES-128.

Непосредственно поддерживается только режим ECB. КА обеспечивает DMA-передачи, как для входных так и выходных данных, что разгружает центральный процессор от операций пересылки (Румянцев, 2015).

Криптоакселератор поддерживает четыре режима операций: зашифрование, порождение ключа расшифровки, расшифровка с предварительно вычисленным ключом, порождение ключа + расшифровка с использованием ключа шифрования. Режимы операций задаются битами регистра управления AES_CR.

Открытый текст, шифртекст или ключ записываются во входной регистр AES_DINR. После завершения вычислений устанавливается соответствующий флаг в регистре статуса AES_SR и может генерироваться прерывание. Считываются данные из исходного регистра AES_DOUT.

Заключение

На основании проведенного анализа можно отметить четкую тенденцию по аппаратной поддержке криптографических примитивов для ограниченных в ресурсах микроконтроллеров, широко используемых в ИОТ. Приведенные в работе данные обеспечивают лучшее понимание как оценивать, разрабатывать и имплементировать криптографическую защиту для нижнего и среднего сегментов микроконтроллерных IoT-устройств.

Использование криптоакселераторов позволяет поднять быстродействие шифрования AES в 10-20 раз для 8/16-битных МК и до 150 раз для 32-битных МК по сравнению с программными реализациями алгоритма. Рост быстродействия вычисления алгоритмов SHA-1, SHA-256 у 32-битных МК составляет более чем в 100 раз, а для Hmac приближается к 500. На сегодня, за счет использования криптоакселераторов, Ивт-устройства с 8/16-битными процессорами могут обеспечить производительность шифрования с учетом накладных расходов на уровне сотен Кбайт/с, тогда как для 32-битных микроконтроллерных ядер можно поддерживать скорость на уровне десятков-сотен Мбайт/сек.

В 32-битных микроконтроллерах наблюдается тренд к внедрению комплексных решений безопасности, которые бы не только ускоряли широкий круг симметричных и асимметричных алгоритмов и протоколов, но и предоставляли возможность защищенного хранения и генерации ключей, безопасной загрузки и обновления кода, поддержки цифровых подписей и сертификатов. Заявленные производителями характеристики позволяют использовать традиционные криптоалгоритмы и протоколы без существенных ограничений, оставляя легковесовую криптографию для 8/16-битных процессоров и ультрамалоресурсных устройств типа RFID-меток и смарт-карт.

Производители микроконтроллеров все чаще уделяют внимание защите криптографических блоков от атак на реализацию, в первую очередь таких как анализ энергопотребления, что очень характерны и опасны для встраиваемых систем. Вполне естественно для этого выбраны методы сокрытия (hiding), как простейших в реализации.

Поддержка микроконтроллерных криптоакселераторов уже присутствует в некоторых легковесных SSL / TLS криптобиблиотеках, ориентированных на встроенные системы, ИОТ и RTOS, например, в wolfSSL. Учитывая, что большинство программных реализаций в известных криптобиблиотеках уязвимы к sidechannel атак, то переход к аппаратному выполнению криптопримитивов дополнительно повышает их защищенность к атакам на реализацию.

Представленный в статье описание характеристик криптоакселераторов призван помочь разобраться с программированием прикладных задач по защите информации для микроконтроллерных устройств Интернета вещей.

Список литературы

1. Алферов А.П., Кузьмин А.С., Черемушкин А.В., Зубов А.Ю. Основы криптографии: учебное пособие. М.: Гелиос АРВ, 2001. 479 с.
2. Бауэр Ф. Расшифрованные секреты. Методы и принципы криптологии. М.: Мир, 2007. 550 с.
3. Бунин О. Занимательное шифрование / отдел «Мир ПК». 2003. https://www.osp.ru/pcworld/2003/07/166048
4. Дошина А.Д., Михайлова А.Е., Карлова В.В. Криптография. Основные методы и проблемы. Современные тенденции криптографии // Современные тенденции технических наук: материалы IV Междунар. науч. конф. Казань: Бук, 2015. С. 10-13.
5. Комиссаренко В.В. Современные тенденции развития средств и методов криптографической защиты информации. В кн.: 2-я конф-ия. «Технологии защиты информации и информационная безопасность организаций», Минск, 2016.
6. Румянцев К. Е., Плёнкин А. П., Синхронизация системы квантового распределения ключа в режиме однофотонной регистрации импульсов для повышения защищенности. // Радиотехника. 2015. № 2. С. 125-134.
7. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. М.: ДМК Пресс, 2012. 593 с.
8. Яковлев А.В., Безбогов А.А., Родин В.В., Шамкин В.Н. Криптографическая защита информации: учебное пособие. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2006. С. 11-15.
9. Ященко В.В. Введение в криптографию. Издание 4 дополненное. МЦНМО: Москва, 2012.
10. Lieven M. K. et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance// Nature 414. 20-27 Dec. 2001. - pp. 883-887.
11. Nechvatal J. Report on the Development of the Advanced Encryption Standard (AES). / J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback, - Journal of Research of the National Institute of Standards and Technology, Volume 106, Number 3, May-June 2001. -pp. 511-577.
12. Susan Decker, Christopher Yasiejko, Forget the Trade War. China Wants to Win the Computing Arms Race. Bloomberg, apr 09, 2018 [Electronic resource]. - Mode of access: https://www.industryweek.com/technology-and-iiot/article/22025445/forget-the-trade-war-china-wants-to-win-the-computing-arms-race

Analysis of hardware support for cryptography in the construction of information security of the university

Egor I. Pristanskov
student
Lomonosov Moscow State University
Moscow, Russia
79044097106@yandex.ru
0000-0000-0000-0000

Oleg A. Kudryavtsev
student
ITMO National Research University
Moscow, Russia
Ibezifooo@gmail.com
0000-0000-0000-0000

Daniil E. Andreev
student
ITMO National Research University
Moscow, Russia
Poshackinfod@gmail.com
0000-0000-0000-0000

Juliya V. Taran
student
ITMO National Research University
Moscow, Russia
Taran@mail.ru
0000-0000-0000-0000

Nikita S. Solovev
student
ITMO National Research University
Moscow, Russia
Nikki.Soloviev@gmail.com
0000-0000-0000-0000

Received 29.04.2022
Accepted 21.05.2022
Published 20.06.2022

10.25726/h2048-6130-4735-p

Abstract

At first, they tried to solve the problem of implementing traditional crypto algorithms in embedded systems at the software level, for which they optimized the code as much as possible by using assembly language and processor architecture features. A number of crypto libraries focused on VS and IoT have been formed, the most famous of which are wolfSSL, OpenSSL, GUARD TLS. Tiny/ Toolkit, Cifra, contain implementations of both individual algorithms and entire protocols with moderate requirements for computing

resources. At the same time, cryptography itself tried to adapt to the requirements of IOT, and in the early 2000s, such a separate direction as lightweight or low-resource cryptography (Lightweight Cryptography) for devices with limited resources stood out. When creating lightweight crypto algorithms, the cost of implementation comes first with an adequate level of protection and the necessary performance, that is, a compromise between these three parameters is important, which depends on the specific requirements for the device. Compared to classical algorithms, lightweight algorithms by reducing the key size, the number of rounds, replacing more complex operations with simpler ones or abandoning them can significantly increase performance and reduce the requirements for implementation resources. As an example of software-oriented lightweight algorithms that have gained considerable popularity in recent years, you can specify the ciphers ChaCha20, Speck, hash Blake2, Mac function Poly1305, etc. In addition, there are modes of operation of ciphers designed to comprehensively and with minimal overhead ensure the confidentiality, integrity and authentication of packaged data.

Keywords
data, cryptography, data protection, university.

References

1. Alferov A.P., Kuz'min A.S., Cheremushkin A.V., Zubov A.Ju. Osnovy kriptografii: uchebnoe posobie. M.: Gelios ARV, 2001. 479 s.
2. Baujer F. Rasshifrovannye sekrety. Metody i principy kriptologii. M.: Mir, 2007. 550 s.
3. Bunin O. Zanimatel'noe shifrovanie / otdel «Mir PK». 2003. https://www.osp.ru/pcworld/2003/07/166048
4. Doshina A.D., Mihajlova A.E., Karlova V.V. Kriptografija. Osnovnye metody i problemy. Sovremennye tendencii kriptografii // Sovremennye tendencii tehnicheskih nauk: materialy IV Mezhdunar. nauch. konf. Kazan': Buk, 2015. S. 10-13.
5. Komissarenko V.V. Sovremennye tendencii razvitija sredstv i metodov kriptograficheskoj zashhity informacii. V kn.: 2-ja konf-ija. «Tehnologii zashhity informacii i informacionnaja bezopasnost' organizacij», Minsk, 2016.
6. Rumjancev K. E., Pljonkin A. P., Sinhronizacija sistemy kvantovogo raspredelenija kljucha v rezhime odnofotonnoj registracii impul'sov dlja povyshenija zashhishhennosti. // Radiotehnika. 2015. № 2. C. 125-134.
7. Shan'gin V.F. Zashhita informacii v komp'juternyh sistemah i setjah. M.: DMK Press, 2012. 593 s.
8. Jakovlev A.V., Bezbogov A.A., Rodin V.V., Shamkin V.N. Kriptograficheskaja zashhita informacii: uchebnoe posobie. Tambov: Izd-vo Tamb. gos. tehn. un-ta, 2006. S. 11-15.
9. Jashhenko V.V. Vvedenie v kriptografiju. Izdanie 4 dopolnennoe. MCNMO: Moskva, 2012.
10. Lieven M. K. et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance// Nature 414. 20-27 Dec. 2001. - pp. 883-887.
11. Nechvatal J. Report on the Development of the Advanced Encryption Standard (AES). / J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback, - Journal of Research of the National Institute of Standards and Technology, Volume 106, Number 3, May-June 2001. -pp. 511-577.
12. Susan Decker, Christopher Yasiejko, Forget the Trade War. China Wants to Win the Computing Arms Race. Bloomberg, apr 09, 2018 [Electronic resource]. - Mode of access: https://www.industryweek.com/technology-and-iiot/article/22025445/forget-the-trade-war-china-wants-to-win-the-computing-arms-race