РЕАЛИЗАЦИЯ ПРОЦЕССОРА ФОРМИРОВАНИЯ КЛЮЧА PBKDF2 НА БАЗЕ FPGA

Качинский М. В., Станкевич А. В.
Кафедра электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: {kachinsky, stankevich}@bsuir.by

Рассматривается аппаратная реализация функции PBKDF2, позволяющая использовать эту функцию во встраиваемых системах с достаточно высокой производительностью. Получение ключа с помощью функции PBKDF2 организуется по итерационной схеме в виде последовательно работающего модуля, реализующего вычисление хэш-функции SHA-1. Приводятся аппаратные затраты реализации на базе кристалла ПЛИС XC6VLX240T-1FFG1759 фирмы Xilinx, оценивается производительность специализированного процессора.

В настоящее время для разрешения доступа пользователей к некоторым электронным данным широко используются пароли. Из-за плохой случайности этих паролей, опасности хранения их значений, а также их произвольной длины они не могут использоваться в качестве ключей криптографических алгоритмов шифрования. В большинстве приложений ключи получают на основе паролей с использованием криптографических хэш-функций.

Функция PBKDF2 (Password Based Key Derivation Function v2) является одним из важнейших криптографических примитивов [1], широко используемым в различных системах, таких как WiFi Protected Access (WPA/WPA2), Microsoft .NET framework, Apple OSX Operating System, Apple iOS, Android (v3.0 – v4.3), Blackberry, Cisco IOS type 4 и других. Характерной особенностью данной функции является значительное (обычно несколько тысяч или десятков тысяч) число повторений использования механизма HMAC (hash-based message authentication code – код проверки подлинности сообщений, использующий хэш-функции), в связи с чем процесс формирования ключа занимает значительное время. Еще одной важной особенностью функции PBKDF2 является отсутствие ограничения длины ключа.

В данной работе предлагается аппаратная реализация функции PBKDF2, позволяющая использовать эту функцию во встраиваемых системах с достаточно высокой производительностью.

Процессор реализует функцию получения ключа PBKDF2 в соответствии со спецификацией RFC 2898 PKCS #5: Password-Based Cryptography Specification Version 2.0 September 2000 [2]. В качестве псевдослучайной функции PRF используется функция HMAC на основе хэш-функции SHA-1 (HMAC-SHA-1), описанная в документе RFC 2104 HMAC: Keyed-Hashing for Message Authentication February 1997 [3].

Процедура получения ключа PBKDF2 (P, S, C, dkLen), реализуемая в предлагаемом процессоре, определяется следующим образом:

- параметр P – пароль для которого формируется значение ключа;
- параметр S – значение salt (соль) минимальной длины 128 бит;
- параметр C – количество итераций (положительное целое число);
- hLen – длина хэш-значения на выходе функции HMAC (20 байт для хэш-функции SHA-1);
- dkLen – длина формируемого ключа (для данной реализации выбрана равной 16 байт);
- b = [dkLen/hLen] = [16/20] = 1 – количество блоков длины hLen в ключе (округление вверх). Таким образом, для рассматриваемой реализации ключ состоит из одного блока длиной r = dkLen - (b-1)*hLen = 16 байт;
- генерируемый ключ DK = T_1<0...15> представляет собой первые 16 байт значения T_1 = F (P, S, C, 1) длины 20 байт, вычисляемого следующим образом:

  F (P, S, C, 1) = U_1 xor U_2 xor ... xor U_C;

  U_1 = HMAC-SHA-1 (P, S || x"00000001");

  U_2 = HMAC-SHA-1 (P, U_1);

  ...

  U_C = HMAC-SHA-1 (P, U_{C-1}).

В приведенных выше выражениях символ || означает операцию конкатенации.

Таким образом, для получения ключа необходимо выполнить C итераций, на каждой из которых вычисляется хэш-значение с помощью функции HMAC-SHA-1. Для вычисления хэш-значения с помощью функции HMAC-SHA-1 необходимо последовательно четыре раза выполнить алгоритм SHA-1:

1) для вычисления хэш-значения блока ikeypad с использованием первого параметра функции HMAC-SHA-1;

2) для вычисления хэш-значения блока okeypad с использованием первого параметра функции HMAC-SHA-1;

3) для вычисления хэш-значения блока на основе второго параметра функции HMAC-SHA-1 с использованием хэш-значения блока ikeypad;

4) для вычисления итогового хэш-значения функции на основе хэш-значения, полученного на шаге 3, с использованием хэш-значения блока okeypad.

Суммарно для получения ключа необходимо 4C раз выполнить алгоритм SHA-1. Однако, поскольку первый параметр у функции HMAC-SHA-1 на всех итерациях один и тот же, первые два шага можно выполнить однократно с запоминанием полученных хэш-значений. Тогда на каждой из C итераций необходимо только дважды выполнять алгоритм SHA-1 (шаги 3 и 4), что суммарно составляет 2C раз [4].

Структурная организация вычислительного ядра процессора функции PBKDF2 показана на рис. 1.

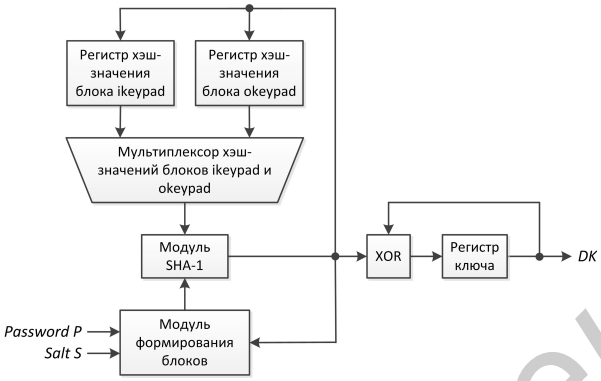

Рис. 1 – Структура вычислительного ядра процессора функции PBKDF2

В предлагаемом процессоре получение ключа с помощью функции PBKDF2 организуется по итерационной схеме в виде последовательно работающего модуля, реализующего вычисление хэш-функции SHA-1.

Входные блоки для модуля SHA-1 формируются в модуле формирования блоков из входных значений пароля password и параметра salt, а также промежуточных хэш-значений с выхода модуля SHA-1. Модуль формирования блоков реализуется в виде двухступенчатого мультиплексора, который в зависимости от цикла работы коммутирует на вход модуля SHA-1 слово соответствующего блока.

На выходе модуля SHA-1 последовательно формируются значения U_1, U_2 ... U_C. Эти значения поступают на схему XOR, где в свою очередь последовательно получаются значения U_1, U_1 xor U_2, ..., U_1 xor U_2 xor ... xor U_C, которые фиксируются в выходном регистре ключа. После выполнения 2C циклов в регистре ключа получаются первые 16 байт значения T_1, что представляет собой искомый ключ PBKDF2 DK = T_1<0...15>. Начальные значения для модуля SHA-1 выбираются с помощью входного мультиплексора. В качестве начальных значений используются вычисленные на первых двух циклах хэш-значения блоков ikeypad и okeypad.

Разработанный процессор функции PBKDF2 описан на языке VHDL с использованием среды проектирования Xilinx ISE 14.7 для ПЛИС XC6VLX240T-1FFG1759 фирмы Xilinx для значения параметра C равного 4096. Аппаратные затраты на реализацию процессора составляют: триггеры секций (slice registers) – 1172; просмотровые таблицы (slice LUTs) – 1390; логические секции (slices) – 509. Анализ результатов реализации (implement design) процессора показывает, что аппаратные затраты ресурсов кристалла по занятым секциям (slice) ПЛИС составляют около одного процента ресурсов кристалла. Предельная рабочая тактовая частота по оценкам процедуры синтеза составляет 262 МГц.

Общее количество циклов работы процессора равно 8194: два для вычисления хэш-значений блоков ikeypad и okeypad и 8192 – для получения ключа. Цикл работы включает однократный запуск модуля SHA-1 для обработки одного входного блока данных и состоит из 85 тактов, необходимых для реализации алгоритма SHA-1. Время вычисления искомого ключа составляет 850172 такта, что при тактовой частоте синхронизации 262 МГц равно 3,24 мс. При этом производительность процессора составляет 308 ключей/с.

Рассматриваемая реализация может использоваться и для обратных задач криптографии. В этом случае задача формирования ключа может распараллеливаться на несколько процессорных ядер для разных значений пароля, что значительно повышает производительность подобной системы.

1. NIST SP 800-132. Recommendation for Password-Based Key Derivations. Desember 2010. [Электронный ресурс] – Режим доступа: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf. – Дата доступа: 06.09.2017.
2. RFC 2898 PKCS #5: Password-Based Cryptography Specification Version 2.0. September 2000. [Электронный ресурс] – Режим доступа: https://www.ietf.org/rfc/rfc2898.txt. – Дата доступа: 06.09.2017.
3. RFC 2104 HMAC: Keyed-Hashing for Message Authentication. February 1997. [Электронный ресурс] – Режим доступа: https://www.ietf.org/rfc/rfc2104.txt. – Дата доступа: 06.09.2017.
4. Andrew Ruddick, Jeff Yan. Acceleration Attacks on PBKDF2: Or, what is inside the black-box of oclHashcat. WOOT'16. 10th USENIX Workshop on Offensive Technologies. August 8-9, 2016. [Электронный ресурс] – Режим доступа: https://www.usenix.org/system/files/conference/woot16/woot16-paper-ruddick.pdf. – Дата доступа: 06.09.2017.