ВСТРАИВАЕМЫЙ МОДУЛЬ ФОРМИРОВАНИЯ КЛЮЧА НА ОСНОВЕ ПАРОЛЯ ДЛЯ FPGA

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Санько Н.С

Качинский М.В. – к.т.н.,доцент

Защита информации — это деятельность по предотвращению утечки, хищения, утраты, подделки, несанкционированных и непреднамеренных воздействий на защищаемую информацию. Одной из наиболее очевидных причин нарушения системы защиты является умышленный несанкционированный доступ к конфиденциальной информации со стороны нелегальных пользователей и последующие нежелательные манипуляции с этой информацией. Существуют специализированные методы, которые защищают информацию от стороннего взлома на основе определённых криптографических алгоритмов шифрования данных.

PBKDF2 — стандарт формирования ключа на основе пароля. PBKDF2 использует псевдослучайную функцию для получения ключей.

Реализация алгоритм получения ключа PBKDF2:

- $P = Password$ (пароль);
- $S = Salt$(соль);
- i = Блок ID;
- c = Итерации;
- $hLen = |HMAC(S;P)|$, (160 бит для SHA1);
- $dkLen$ = полученная длинна ключа (в битах);
- $l = [dkLen/hLen]$;
- $r = (dkLen - (l - 1) * hLen)$;
- $U1 = HMAC(P, S \,||\, int(i))$ (первая итерация);
- $U2 = HMAC(P, U1)$ (вторая итерация);
- …
- $Uc = HMAC(P, U(c-1))$ (последняя итерация);

Для того, чтобы вычислить ключ необходимо совершить определённое количество итераций, которое в зависимости от решаемой задачи составляет от нескольких тысяч до десятков. Для вычисления хеш-значение с помощью функции HMAC-SHA1 необходимо последовательно 4 раза выполнить алгоритм SHA-1:

1. для вычисления хеш-значения блока ikeypad с использованием первого параметра функции HMAC-SHA1;
2. для вычисления хеш-значения блока okeypad с использованием первого параметра функции HMAC-SHA1;
3. для вычисления хеш-значения блока на основе второго параметра функции HMAC-SHA-1 с использованием хеш-значения блока ikeypad;
4. для вычисления итогового хеш-значения функции на основе хеш-значения, полученного на шаге 3, с использованием хеш-значения блока okeypad.

В качестве псевдослучайной функции PRF была выбрана функция HMAC-SHA1.

HMAC осуществляет проверку подлинности сообщений, тем самым обеспечивает защищённость передаваемых или хранящихся данных. Этот механизм направлен на обмен данными с использованием секретного ключа и хеш-функции, в данном случае хеш-функцией является SHA1.

Реализация HMAC:

- K = секретный ключ;
- M = сообщение;
- B = размер блока (байты);
- $ikeypad$ = Inner Padding (входнойблок);
- $okeypad$ =OuterPadding (выходной блок);
- $H = SHA1$;
- $B = 64$ (размер блока SHA1);
- $HMAC(K, M) = H((KXORokeypad) \,||\, H((KXORikeypad) \,||\, M))$;

Secure Hash Algorithm 1 — алгоритм криптографического хеширования. Для входного сообщения произвольной длинны алгоритм генерирует 160-битное хеш-значение, называемое также дайджестом сообщения. Применяется во многих криптографических приложениях и протоколах. Принципы, положенные в основу SHA-1, аналогичны тем, которые использовались Рональдом Ривестом при проектировании MD4.При сравнении SHA-1 с MD5 они являются улучшенными продолжениями MD4.

SHA-1 реализует хеш-функцию, построенную на идее функции сжатия. Входами функции сжатия являются блок сообщения длиной 512 бит и выход предыдущего блока сообщения. Выход представляет собой

значение всех хеш-блоков до этого момента. Хеш-значением всего сообщения является выход последнего блока.

Для вычисления хэша используются пять состояния (A, B, C, D, E). Для хранения этих переменных предусматривается буфер, который представляет собой набор 32-разрядных регистров.

В алгоритме хеширования SHA-1 используется раундовая система расчёта значений. Для каждого раунда определяются своя нелинейная раундовая операция F и раундовая константа K.

Блок сообщения преобразуется из 16 32-разрядных слов M(t) в 80 32-разрядных слов W(t) по определённому правилу. Далее значения сохраняются во временные регистры A, B, C, D, E и производятся запланированные преобразования. После этого текущие значения регистров прибавляются к их исходным значениям соответственно, далее идёт обработка и конечным значением будет объединение пяти 32-разрядных слов в одно 160-разрядное хеш-значение. В случае, когда сообщение состоит из одного SHA-1 блока, полученная в регистрах A, B, C, D, E сумма представляет собой финальный хеш. Для вычисления хеша блок обработки используется в цикле 80 раза, так что финальное значение получается за 80 тактов.

В докладе рассматриваются вопросы построения встраиваемого модуля формирования ключа на основе пароля для FPGA. Архитектурным вариантом аппаратной реализации алгоритма SHA-1 наиболее подходит итеративная архитектура, так как конвейерную архитектуру довольно сложно реализовать из-за последовательности выполнения операций в алгоритме SHA-1 при условии, что сообщение будет больше одного блока. Таким образом итеративная архитектура обеспечивает минимальное использование ресурсов FPGA, однако, и невысокое быстродействие.

Список использованных источников:

1. Т. В. Кузьминов. Криптографические методы защиты информации. Новосибирск: Сибирское предприятие РАН, 1998.
2. Х. К. А. ван Тилборг. Основы криптологии. Профессиональное руководство и интерактивный учебник. Мир, 2006.
3.Ященко В.В. Введение в криптографию. Издание 4-е, дополненное. Москва: Изд-во МЦНМО, 2012.

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ ПРОЦЕССОР АДАПТИВНОГО ФИЛЬТРА ВИНЕРА НА РАСПРЕДЕЛЕННОЙ АРИФМЕТИКЕ ДЛЯ СИСТЕМ МУЛЬТИМЕДИА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Андреев И.Д.

Петровский Н.А. – к.т.н.

В рамках данной работы было разработано VHDL-описание процессора адаптивного фильтра на распределенной арифметике для платформы Xinix Zynq 7010. Данная платформа представляет собой систему на кристалле со встроенным блоком ПЛИС, что значительно упрощает тестирование разрабатываемых аппаратных модулей. Процессор предназначен для подавления акустического эха и обеспечивает обработку аудио в режиме реального времени.

Одной из проблем, препятствующих комфортному общению людей посредством голосовой связи, является эхо – вместо того, чтобы слышать только собеседника, человек слышит еще и самого себя с некоторой задержкой. В таких условиях продолжение беседы порой невозможно.

Акустическое эхо является шумом. Удаление любых шумов из полезного сигнала в компьютерной технике осуществляется с помощью цифровой фильтрации. Наиболее эффективными с точки зрения соотношения производительность\энергопотребление являются аппаратные фильтры. Они позволяют работать в масштабе реального времени потребляя относительно небольшую мощность.

Основной операцией цифровой фильтрации является умножение с накоплением. Однако аппаратный умножитель занимает большую площадь кристалла процессора, а также оказывает негативное влияние на энергопотребление. Особо острой проблема питания становится в условиях полностью автономной работы вычислительной системы.

Поэтому был найден способ замены операции умножения на операцию суммирования, который взял на вооружение распределенную арифметику: числа в двоичной системе счисления представляются в виде суммы произведений разрядов числа и соответствующих их положению в числе степеней двойки. Тогда каждый из разрядов можно использовать в качестве адреса некоторой памяти, в которой хранятся все возможные суммы коэффициентов фильтра. С развитием технологий память стала дешевой, объемной и крайне энергоэффективной, что делает реализацию фильтра с использованием распределенной арифметики на памяти еще более привлекательной.

Проблема фильтрации эха заключается в том, что подобный шум не стационарен: его импульсная характеристика меняется во времени, а значит коэффициенты фильтра не могут быть постоянными. Задачу удаления таких сигналов призван решить адаптивный фильтр: его коэффициенты меняются на основании входных данных.