УДК 004.056.55

А.С. Куликова, И.В. Лысенко

Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ», Харьков

РЕАЛИЗАЦИЯ МНОГОВЕРСИОННОГО ПОТОЧНОГО КРИПТОПРЕОБРАЗОВАНИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ БЕСКЛЮЧЕВЫХ ХЕШ-ФУНКЦИЙ НА ПРОГРАММИРУЕМОЙ ЛОГИКЕ

Предлагается разработка подхода к построению систем поточного шифрования данных, основанных на принципе диверсности, с использованием криптографических бесключевых хеш-функций. Описывается общая структурная схема предлагаемой многоверсионной системы, предлагаются варианты построения структур её шифрующих каскадов для формирования псевдослучайной последовательности. Описываются принципы выбора способа шифрования для каждого блока входного сообщения. Также рассматриваются особенности подхода к реализации данной системы на основе технологии ПЛИС типа FPGA.

Ключевые слова: *диверсность, конфиденциальность, поточное шифрование, криптографическая хеш-функция, ПЛИС, FPGA, IP-ядро.*

Введение

Постановка задачи. В связи с тем, что в современном мире практически вся информация передаётся по цифровым каналам связи, проблема обеспечения конфиденциальности данных стоит очень остро. Конфиденциальность информации – субъективно определяемая характеристика (свойство) информации, указывающая на необходимость введения ограничений на круг субъектов, имеющих доступ к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочий доступа к ней. Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней.

Задача повышения защищенности передаваемой информации решается путем разработки новых и совершенствования существующих криптографических алгоритмов. В сфере защиты данных, передаваемых по каналам связи в реальном времени, целесообразно использовать поточное шифрование.

Поточное шифрование (также известное как гаммирование) состоит в том, что биты открытого текста складываются по модулю 2 с гаммой шифра (вместо операции сложения по модулю два можно использовать любую обратимую операцию). Гамма шифра получается следующим образом: при помощи псевдослучайного генератора формируется начальная (предварительная) гамма, которая шифруется в режиме простой замены (таким образом получается основная гамма). Начальное значение псевдослучайного генератора (называемое *инициализационным вектором*) не является секретным и передается по каналу в открытом виде.

В рамках проектирования эффективных и криптостойких поточных алгоритмов существует несколько подходов [1]:

– системно-теоретический подход основан на создании для криптоаналитика сложной, ранее неисследованной проблемы;

– сложностно-теоретический подход основан на сложной, но известной проблеме (например, факторизация чисел или дискретное логарифмирование);

– информационно-технический подход основан на попытке утаить открытый текст от криптоаналитика – вне зависимости от того сколько времени потрачено на дешифрование, криптоаналитик не найдёт однозначного решения;

– рандомизированный подход основан на создании объёмной задачи; криптограф тем самым пытается сделать решение задачи расшифрования физически невозможной.

В рамках последнего варианта может быть использован многоверсионный подход (принцип диверсности), который используется для успешного решения задачи обеспечения заданного уровня надежности и гарантоспособности компьютерных систем.

В работах [2 – 4] рассмотрен подход и предложены модели к построению симметричных криптосистем на основе принципа диверсности, суть которого состоит в такой организации процесса защищённого взаимодействия пользователей, чтобы усложнить процесс криптоанализа за счёт создания неопределённости у криптоаналитика путём использования различных криптоалгоритмов в различных сеансах и внутри конкретного сеанса в соответствии с некоторым правилом.

Идея использования принципа диверсности состоит в том, что при разных сеансах для шифрова-

ния данных используются различные алгоритмы, скомбинированные различным образом. В свою очередь, выбор алгоритмов и их комбинации осуществляется в соответствии с результатом вычисления некоторой достаточно простой функции f, аргументом которой является некоторый неизвестный криптоаналитику параметр. По этой причине порядок выбора алгоритмов шифрования остаётся для криптоаналитика непредсказуемым. В этом случае на каждом такте в качестве элемента ключевой двоичной псевдослучайной последовательности, накладываемой на биты шифруемого документа, используется бит, генерируемый тем из алгоритмов, на который указывает результат вычисления некоторой функции, зависящей от номера такта.

Однако в наше время злоумышленник может перехватить информацию не только путём криптоанализа алгоритма, но и путём атак на аппаратную часть защищаемой системы, такими, как: перегрузка системы запросами, прослушивание побочного канала, изучение потребления энергии или электромагнитного излучения системы, анализ временных откликов системы, обратное проектирование. С целью ослабления или предотвращения данных угроз в защищаемой системе целесообразно использование аппаратуры, обладающей стойкостью к вышеперечисленным возможным действиям злоумышленника.

Таким образом, **целью данной статьи** является разработка принципа многоверсионного поточного криптопреобразования данных с использованием бесключевых криптографических хеш-функций для последующей реализации в аппаратно-программном виде на основе технологии ПЛИС типа FPGA.

Диверсное поточное шифрование данных на основе использования бесключевых хеш-функций

Обобщенная структура модели диверсного поточного шифрования данных

В основу этой модели положена идея многократного использования бесключевых хеш-функций в каждом сеансе взаимодействия пользователей.

Обобщенная структура диверсной поточной системы шифрования на основе использования бесключевых хеш-функций изображена на рис. 1.

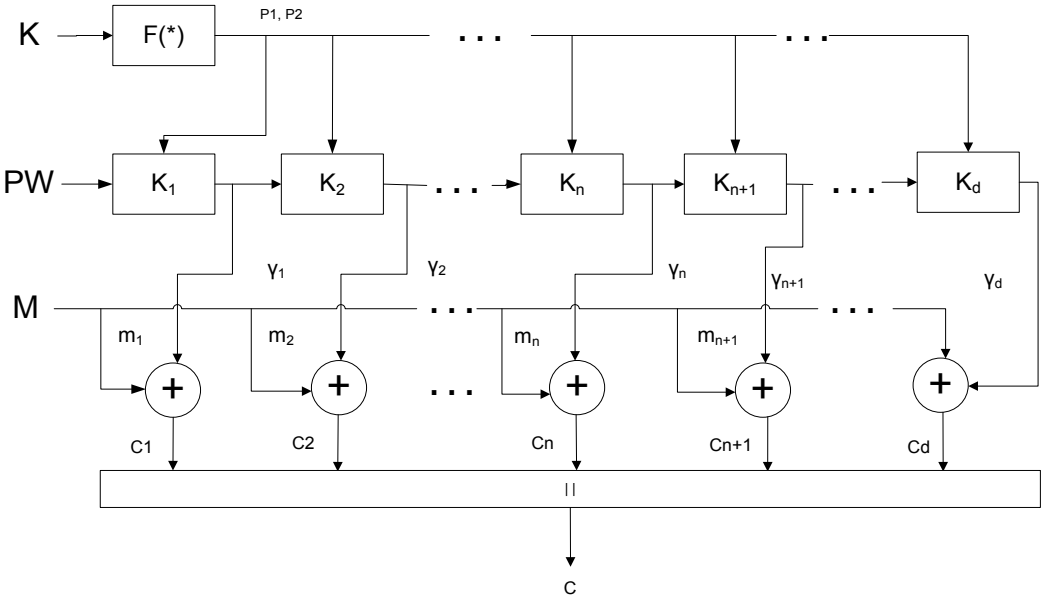

Рис. 1. Обобщенная структура диверсной поточной системы шифрования

На данном рисунке M – исходное сообщение длиной d, которое необходимо зашифровать. Оно представляется в виде последовательности блоков m_1, m_2,…, m_d. Для шифрования этого сообщения необходимо сформировать псевдослучайную последовательность, которая будет побитово складываться с исходным сообщением по модулю 2. Для формирования этой последовательности в данной схеме существуют различные комбинации (каскады) хеш-функций, обозначенные k_1,…, k_d.

Они соединены последовательно. Для своей работы хеш-функции используют парольную фразу PW и на выходе генерируют псевдослучайную последовательность γ. Выбор хеш-функций, которые будут использоваться в текущем сеансе шифрования, осуществляется с помощью некоторой функции F. Она принимает в качестве входного параметра исходный ключ k, а на выходе этой функции мы получаем случайные параметры P_1 и P_2, которые для каждого каскада определяют хеш-функцию и тип внутрикаскадной схемы хеширования. Соответственно, значения этих параметров лежат в границах множества используемых хеш-функций и множества типов структур каскада. После определения схемы хеширования и формирования псевдослучайной последовательностью (ПСП) блок битов входного

сообщения m_1 объединяется с ПСП γ_1 логической операцией XOR и получается первый фрагмент шифртекста C_1. Блок m_2 таким же способом объединяется с γ_2, и т.д. Все полученные фрагменты шифртекста C_1, C_2,…,C_n объединяются операцией конкатенации и на выходе получается результирующий шифртекст исходного сообщения C.

Схемы формирования ПСП

Идея схемы формирования ПСП сходна с описанной в работе [5]. Она состоит в следующем: имеется множество хеш-функций MH={H_1, H_2,…, H_N} и множество вариантов структурных комбинаций этих хеш-функций MS = {S_1, S_2,…, S_L}. На каждом раунде шифрования, используя процедуру выбора хеш-функции, которая является секретной для криптоаналитика, и которая будет описана ниже, из множества MH выбирается одна хеш-функция H_{ij}, а из множества MS – один вариант комбинации (комбинация). В каждом каскаде может быть две хеш-функции. Варианты внутрикаскадных схем представлены на рис. 2 – 4. В статье [5] предлагается следующий подход к формированию ПСП: хеш-функции также выбираются с помощью простой процедуры выбора, на первом раунде шифрования на вход H_{ij} подается ключ k и выполняется хеширование, на выходе получается хеш-значение h_1. Далее выполняется конкатенация для k и h_1. На втором раунде выполняется хеширование результата конкатенации и получается значение h_2, на третьем – h_3 и т.д. Таким образом, диверсность достигается путём комбинирования различных входных данных для выбранных хещ-функций. В данной же работе это достигается посредством комбинирования выбранных хеш-функций в различные логические структуры типа каскад, отличающиеся для каждого блока шифруемых данных.

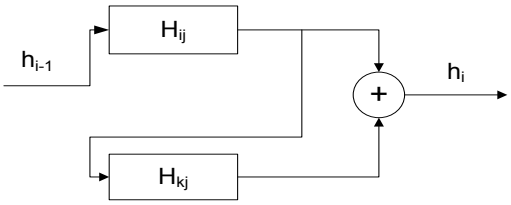

Рис. 2. Структура первого типа

В первой схеме хеширования, изображённой на рисунке 2, на первом раунде шифрования на вход H_{ij} подается ключ h_{i-1} (который в то же время является результатом работы предыдущего каскада или парольной фразой PW, в случае, если это первый каскад) и выполняется хеширование, результат которого подаётся на вход хеш-функции H_{kj}. На втором раунде выполняется хеширование функцией H_{kj}. Далее результаты работы обеих хеш-функций объединяются логической операцией XOR, и на выходе получается хеш-значение h_i.

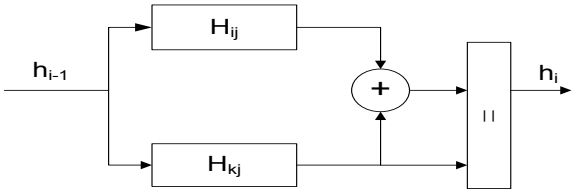

Рис. 3. Структура второго типа

В структуре второго типа, изображённой на рис. 3, на первом раунде шифрования ключ h_{i-1} подается на входы H_{ij} и H_{kj} и выполняется хеширование обеими функциями. На втором раунде результаты их работы объединяются логической операцией XOR. Далее происходит объединение этого результата с результатом хеширования функцией H_{kj} с помощью логической операции ИЛИ, и на выходе получается хеш-значение h_i.

В структуре третьего типа, изображённой на рис. 4, на первом раунде шифрования ключ h_{i-1} подается на входы H_{ij} и логического элемента XOR, в котором ключ h_{i-1} объединяется с результатом работы функции H_{ij}. На втором раунде выполняется хеширование результата работы логического элемента XOR функцией H_{kj}. Далее результаты работы функций H_{ij} и H_{kj} объединяются логической операцией XOR, и на выходе получается хеш-значение h_i.

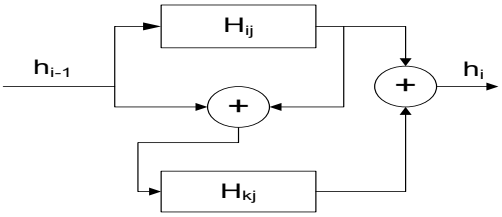

Рис. 4. Структура третьего типа

Различные варианты комбинирования различных хеш-функций в каскадах представляют собой возможные модели схем формирования ПСП.

Процедура выбора хеш-функции

Выбор хеш-функций, с помощью которых будет осуществляться шифрование, а также выбор типа структуры каскада является неизвестным для криптоаналитика, в отличие от множества всех хеш-функций MH = = {H_1, H_2,…,H_N} и множества всех структур каскадов MS = {S_1, S_2,…, S_L}, и может осуществляться по какому-либо правилу. Это правило будет разным для выбора хеш-функции и выбора структуры каскада, так как в формуле должно присутствовать модулярное преобразование по числу всех хеш-функций/каскадов для того, чтобы не превысить границ этих множеств. Для первого раунда хеширования это правило может быть, например, таким:

$$k(\mathrm{mod}\ N) \rightarrow H_{i1}, \qquad (1)$$

или несколько усложнённым, например, таким или любым другим образом:

$$(k \| (k>>5))(\mathrm{mod}\ N) \rightarrow H_{i1}, \qquad (2)$$

где «<>>5» обозначает циклический сдвиг вправо на 5 бит, а N – число всех хеш-функций, для формулы выбора типа каскада N заменяем на L – число всех типов внутрикаскадных схем.

Так как ключ для хеширования k представляет собой последовательность бит, то необходимо представить его в десятичном виде, а результат модулярного преобразования будет соответствовать порядковому номеру одной из хеш-функций из множества MH или одной из структур каскада из множества MS. Очевидно, что поскольку результатом модулярного преобразования может быть 0, то H_1 (S_1) будет соответствовать 0, H_2 (S_2) будет соответствовать 1 и т.д.

Для выбора хеш-функций, с помощью которых будет осуществляться шифрование на всех последующих раундах, могут использоваться хеш-значения предыдущих раундов и, например, следующие правила:

$$\forall j = 2,\ldots,n: (k \oplus h_{j-1}) \bmod N \rightarrow H_{ij,}, \quad (3)$$

$$\forall j = 2,\ldots,n: (k \oplus (h_{j-1}>>5)) \bmod N \rightarrow H_{ij,} \quad (4)$$

где h_{j-1} – хеш-значение, полученное на предыдущем раунде. И снова для вычисления номера типа каскада N заменяем на L.

Суть данного преобразования такая же, как и (1) или (2).

При программной реализации данного шага имеется возможность воспользоваться встроенной функцией языка типа random с указанием пределов генерации случайных чисел от 0 до (N–1) или (L–1).

Особенности реализации на ПЛИС

ПЛИС широко используется для построения различных по сложности и по возможностям цифровых устройств. Для применения в области криптографии ПЛИС обладает многими преимуществами. Например, высокое быстродействие, которое является одной из важнейших характеристик поточного шифрования. Довольно высокая сложность осуществления обратного проектирования плат ПЛИС не даст злоумышленнику легко раскрыть схему формирования ПСП, если устройство попадёт к нему в руки. Возможность перепрограммирования устройства «на лету» облегчает обновление или замену криптоалгоритма в схеме или изменение самой схемы.

В настоящее время существует достаточно много ПЛИС-реализаций криптографических алгоритмов, в том числе и хеш-функций. Основой таких реализаций является трансляция алгоритма, описанного в стандарте (например, FIPS) в код на языке описания аппаратуры (Verilog, VHDL, AHDL). Поскольку одним из требований к криптографии на ПЛИС является возможность обновления и реконфигурации криптосистемы, то собственно код алгоритма обычно представляет собой модуль, который можно повторно использовать, иногда оформленный в виде т.н. IP-ядра (библиотеки с программным кодом). Эти IP-ядра разрабатываются в основном производителями ПЛИС-микроконтроллеров или отдельными коммерческими фирмами по производству IP-ядер (в частности, такие фирмы, как ALMA Technologies, Helion Technology, Ocean Logic, SafeNet ориентированы на криптографические IP-ядра), однако существуют и открытые проекты. Реализации алгоритмов SHA-1, SHA-2 и MD-5, которые будут использоваться в данной разработке, особенно распространены. Для этих реализаций существует множество описаний (см. публикации таких авторов, как N. Sklavos [10], D. Zibin [11], J.M. Diez [12], Y.K. Kang [13], S. Dominikus).

В данной работе реализация собственно алгоритмов хеш-функций может осуществляться посредством использования криптографических IP-ядер на языке описания аппаратуры (Verilog, VHDL). IP-ядра содержат хеш-функции из множества MH, описанного ранее. В символьном представлении (в графической среде разработки) они являются блоками с запрограммированными входами и выходами. Программирование логических структур для формирования ПСП происходит посредством комбинирования этих блоков (соединения их шинами данных) между собой и с логическими элементами в соответствии с вариантами логических структур, описанными в разделе «Схемы формирования ПСП». Выходное значение блока выбора хеш-функции и типа каскада определяет, с какой из комбинаций мы будем работать для обработки данного блока сообщения. Блоки входного и зашифрованного сообщений хранятся в отдельных элементах памяти. После завершения обработки сообщения они, в соответствии с выбранной структурой диверсного шифрования, объединяются с помощью логической операции ИЛИ и записываются в результирующий элемент памяти. Таким образом, основным аппаратным ограничением при разработке данной системы является объём памяти, встроенной в плату и количество логических элементов на плате. Эти параметры определяют максимальное возможное количество используемых каскадов формирования ПСП, хеш-функций и длину шифруемого сообщения.

Результирующая схема шифрования, закодированная на языке описания аппаратуры, также может быть представлена в виде IP-ядра, что сделает возможным её дальнейшее использование в других проектах на ПЛИС.

Ввод исходных данных и вывод результатов работы схемы осуществляется средствами подключаемой периферии (клавиатура, дисплей). Следовательно, реализация данного проекта на ПЛИС требует также написания модулей интерфейсных взаимодействий с подключаемой периферией.

Анализ быстродействия разработанной схемы может быть выполнен посредством реализации таймера штатными средствами ПЛИС. Впоследствии

возможно сравнение полученных результатов с быстродействием программной реализации разработанной схемы на стандартном ПК. Проверка работы системы осуществляется с помощью симуляции стандартными программными средствами.

Заключение

Использование рассмотренного подхода, как предполагается, приведет к возникновению неопределенности для криптоаналитика, так как это не позволит ему собрать необходимый материал для криптоанализа. Иными словами, он не сможет определить, какая именно схема применялась при использовании того же самого ключа в разных сеансах передачи сообщений. Кроме того, данный подход позволяет избежать необходимости генерирования уникальных для каждого сеанса ключей.

Направлением дальнейшей работы является разработка аппаратной системы, корректно реализующей предложенный подход. Перспективной является ее реализация на FPGA, поскольку в этом случае не только улучшаются технические характеристики схемы, а и увеличивается безопасность системы благодаря особенностям этой технологии, обеспечивающей улучшение криптозащиты по сравнению с программными реализациями.

Список литературы

1. Rueppel R.A. Analysis and Design of Stream Ciphers / R.A. Rueppel // Springer Communications And Control Engineering Series. – 1986. – № 1. – P. 244-260.

2. Волковой А.В. Многоверсионные системы и технологии для критических приложений: Лекц. материал; Под ред. В.С. Харченко / А.В. Волковой, И.В. Лысенко, В.С. Харченко, О.В. Шурыгин. – Министерство образования и науки Украины, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2008. – 224 с.

3. Лысенко И.В. Использование принципа диверсности для обеспечения конфиденциальности сообщений в рамках современной криптографии / И.В. Лысенко // Радіоелектронні і комп'ютерні системи. – Х.: НАКУ «ХАІ». – 2006. – № 5(17). – С. 118-121.

4. Лысенко И.В. Модели обеспечения конфиденциальности сообщений средствами криптографии на основе принципа диверсности / И.В. Лысенко, Д.А. Филиппов // Системи обробки інформації. – Х.: ХУ ПС, 2006. – Вип. 2(51). – С. 76-80.

5. Авдеева А.В. Модели диверсного поточного криптопреобразования данных / А.В. Авдеева, И.В. Лысенко // Системи управління, навігації та зв'язку. – К.: ДП «Центральний науково-дослідний інститут навігації і управління», 2011. – Вип. 3 (19). – С. 189-191.

6. Молдовян А.А. Криптография: скоростные шифры / А.А. Молдовян, Н.А. Молдовян, Н.Д. Гуц, Б.В. Изотов. – СПб.: БХВ-Петербург, 2002. – 496 с.

7. Littlewood B. Redundancy and diversity in security / B. Littlewood, L. Strigini, B. Littlewood // Proc. 9th European Symposium on Research in Computer Security (ESORICS'2004), France. – 2004. – P. 117-126.

8. Многоверсионные системы, технологии, проекты; под ред. В.С. Харченко / В.С. Харченко, В.Я. Жихарев, В.М. Илюшко и др. – Х.: НАУ «ХАИ», 2002. – 486 с.

9. Максфилд К. Проектирование на ПЛИС. Курс молодого бойца / К. Максфилд. – М.: Изд. дом «Додэка XXI», 2007. – 408 с.

10. Sklavos N. Implementation of the SHA-2 Hash Family Standard Using FPGAs / N. Sklavos, O. Koufopavlou // Journal of Supercomputing, Springer-Verlag. 2005. – Vol. 31, No 3. – P. 227-248.

11. Zibin D. FPGA Implementation of SHA-1 Algorithm / D. Zibin, Z. Ning // In Proceedings of the 5 International Conference on ASIC. – 2003. – P. 1321-1324.

12. Diez J.M. Hash Algorithms for Cryptographic Protocols: FPGA Implementations / J.M. Diez, S. Bojanic, C. Carreras, O. Nieto-Taladriz // Proc. Telecomm. Forum (TELEFOR), 2002.

13. Kang Y.K. An Efficient Implementation of Hash Function Processor for IPSEC / Y.K. Kang, D.W. Kim, T.W. Kwon, J.R. Choi // Proceedings of the IEEE Asia-Pacific Conference on ASIC. – 2002. – P. 93-96.

Поступила в редколлегию 30.08.2012

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

РЕАЛІЗАЦІЯ БАГАТОВЕРСІЙНОГО ПОТОКОВОГО КРИПТОПЕРЕТВОРЕННЯ ДАНИХ З ВИКОРИСТАННЯМ БЕЗКЛЮЧОВИХ ХЕШ-ФУНКЦІЙ НА ПРОГРАМУЄМІЙ ЛОГІЦІ

А.С. Кулікова, І.В. Лисенко

Пропонується розробка підходу до побудови систем потокового шифрування даних на основі принципу диверсності, з використанням криптографічних безключових хеш-функцій. Описується загальна структурна схема запропонованої багатоверсійної системи, пропонуються варіанти побудови структур її шифрувальних каскадів для формування псевдовипадкової послідовності. Описуються принципи вибору способу шифрування для кожного блоку вихідного повідомлення. Також розглядаються особливості підходу до реалізації даної схеми на основі технології ПЛІС типу FPGA.

Ключові слова: *диверсність, безпека, конфіденційність, потокове шифрування, криптографічна хеш-функція, ПЛІС, FPGA, IP-ядро.*

REALIZATION OF DIVERSE STREAM DATA ENCRYPTION WITH KEYLESS HASH FUNCTIONS ON THE BASIS OF PROGRAMMABLE LOGIC

A.S. Kulikova, I.V. Lysenko

Approach development to creation of systems of stream encryption of the data based on a principle of a diversity, with use of cryptographic keyless hash functions is offered. The general structural schema of offered diverse system is described, also options of creation of encoding stages structures, used for formation of pseudorandom sequence are offered. Principles of a choice of a method of encoding for each unit of the input message are described. Also features of an approach to implementation of this system on the basis of technology of FPGA technology are considered here.

Keywords: *diversity, security, privacy, stream encryption, cryptographic hash function, CPLD, FPGA, IP-core.*