УДК 621.391

О.Г. Король[1], Л.Т. Пархуць[2], С.П. Евсеев[1]

[1] *Харьковский национальный экономический университет, Харьков*
[2] *Национальный университет "Львовская политехника", Львов*

ОБОСНОВАНИЕ ВЫБОРА ЦИКЛОВОЙ ФУНКЦИИ ДЛЯ ИТЕРАТИВНОГО ХЕШИРОВАНИЯ ИНФОРМАЦИИ

Обосновывается выбор цикловых функций в схеме доказуемо стойкого ключевого универсального хеширования, разрабатываются алгоритмы итеративного ключевого хеширования доказуемой стойкости на основе использования модулярных преобразований.

Ключевые слова: *модулярные преобразования, цикловые функции, ключевое хеширование.*

Введение

Применение преобразований с использованием модулярной арифметики позволяет строить универсальные и строго универсальные классы хеширующих функций, которые с одной стороны позволяют обеспечить высокие коллизионные свойства, с другой стороны, при выполнении определенных ограничений на значение модулярной экспоненты обеспечивают высокие показатели безопасности и применимость модели доказуемой стойкости. Проведенные исследования показали, что основными недостатками подобных конструкций являются: высокая сложность преобразований, которая обусловлена использованием в качестве цикловой функции модулярного возведения в степень; модель доказуемой безопасности к рассмотренным методам хеширования может быть применена только при выполнении определенных ограничений на значения экспоненты и модуля преобразования, т.е. при выполнении ограничений для RSA-подобных систем на параметры модульного возведения в степень.

Целью работы является обоснование выбора цикловых функций в схеме доказуемо стойкого ключевого универсального хеширования, разработка алгоритмов итеративного ключевого хеширования доказуемой стойкости на основе использования модулярных преобразований.

Основная часть

Обоснование выбора цикловой функции для итеративного хеширования информации

В основе предлагаемого метода ключевого универсального хеширования доказуемой стойкости лежит использование модулярных преобразований, обеспечивающих сведение задачи нахождения прообраза или секретного ключа в схеме хеширования к одной из известных теоретико-сложностных задач, например, к задаче факторизации, дискретного логарифмирования или задаче RSA. Подобное обоснование стойкости по классификации моделей безопасности NESSIE принято считать доказуемой безопасностью, подчеркивая тем самым сводимость задачи криптоанализа к одной из хорошо известных вычислительно неразрешимых за заданное время теоретико-сложностных задач [1 – 4].

Обоснование безопасности доказуемо стойких криптосистем базируется на принятии предположения о существовании т.н. односторонних функций. Односторонняя функция $f : x \to y$, заданная на множестве x и областью значений в множестве y обладает двумя свойствами:

– существует полиномиальный алгоритм вычисления $f(x)$;

– не существует полиномиального алгоритма инвертирования функции $f(x)$, т.е. решения уравнения $f(x) = y$.

Выполнение второго указанного свойства на сегодняшний день не доказано ни для одного из известных кандидатов на одностороннюю функцию, т.е. само существование односторонних функций не доказано. В то же время, на использовании некоторых из них строятся практически все известные несимметричные криптоалгоритмы [1, 3].

Рассмотрим наиболее известные примеры кандидатов на односторонние функции на предмет их использования для построения универсальных хеширующих функций, в частности, для построения цикловой функции при итеративном формировании хеш-кода (рис. 1). На вход хеш-функции h поступает исходное сообщение x – строка данных произвольной длины. Данная строка представляется в виде последовательности r-разрядных блоков x_i. Перед обработкой, по необходимости, последний блок дополняется до r битов. Кроме того, из соображений повышенной стойкости, исходная строка может быть дополнена новым r-разрядным блоком, который, например, может содержать двоичное представление числа, характеризующего длину исходного сообщения x.

Основой итеративной хеш-функции является так называемая цикловая функция $f(x, y)$ или функция сжатия, которая, в общем случае, берет на вход две строки x и y, длиной соответственно r и s разрядов и формирует на выходе s-разрядную строку. Каждый блок x_i служит входным аргументом для цикловой функции.

Другим аргументом является s разрядное промежуточное значение (переменная сцепления), полученное на предыдущем шаге хеширования. H_i обозначает частный результат хеширования на i-м шаге (итерации), IV - s-разрядный вектор инициализации (Initial Value). На выходе цикловой функции f формируется s-разрядный хеш-код. При необходимости он может быть усечен до заданной длины посредством дополнительной обработки $g(H_t)$. Чаще всего $g(H_t) = H_t$.

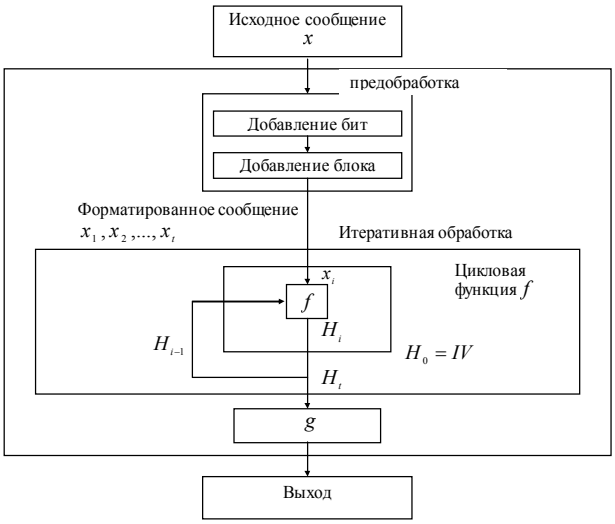

Рис. 1. Общая схема итеративной хеш-функции

1. *Проблема целочисленной факторизации*. Дано положительное целое число n, которое однозначно представимо в виде произведения различных степеней простых чисел (каноническое представление числа):

$$n = p_1^{m_1} \cdot p_2^{m_2} \cdot \ldots \cdot p_k^{m_k},$$

где p_i – i-е простое число в разложении n, $m_i \geq 1$. Существует полиномиальный алгоритм вычисления n по известным $p_i^{m_i}$, $i = 1, 2, \ldots, k$. На сегодняшний день не известно полиномиального алгоритма разложения числа n на простые сомножители.

2. *Проблема RSA*. Дано положительное целое число n, которое однозначно представимо в виде произведения двух простых чисел p и q, а также положительное целое число e, взаимопростое с числом $(p-1)(q-1)$ и целые числа c и m, причем $m^e \equiv (c) \bmod (n)$. Для заданных n и e существует полиномиальный алгоритм вычисления числа $c \equiv (m^e) \bmod (n)$. На сегодняшний день не известно полиномиального алгоритма вычисления такого числа m, что $m^e \equiv (c) \bmod (n)$.

3. *Проблема квадратичных вычетов*. Дано нечетное целое составное число n и целое число a, имеющее символ Якоби $\left(\frac{a}{n}\right) = 1$. Для заданных n и a существует полиномиальный алгоритм вычисления символа Якоби:

$$\left(\frac{a}{n}\right) = \begin{cases} +1, (a)^{(n-1)/2} = 1 \bmod (n), \\ -1, (a)^{(n-1)/2} \neq 1 \bmod (n). \end{cases}$$

На сегодняшний день не известно полиномиального алгоритма определения того, является или нет число a квадратичным вычетом по модулю n.

4. *Проблема извлечения квадратных корней по модулю* n. Дано составное целое число n и $a \in Q_n$ (Q_n – множество квадратичных вычетов по модулю n). Для заданных n и произвольного целого числа x существует полиномиальный алгоритм вычисления числа $a \equiv (x^2) \bmod (n)$. На сегодняшний день не известно полиномиального алгоритма извлечения квадратных корней по модулю n, т.е. нахождения такого x, что $x^2 \equiv (a) \bmod (n)$.

5. *Проблема дискретного логарифмирования*. Дано простое число p, генератор α кольца целых чисел Z_p и элемент $\beta \in Z_p$. Для заданных p, α и произвольного целого числа x существует полиномиальный алгоритм вычисления числа $\beta \equiv (\alpha^x) \bmod (p)$. На сегодняшний день не известно полиномиального алгоритма дискретного логарифмирования по модулю p, т.е. нахождения такого x, что $\alpha^x \equiv (\beta) \bmod (p)$.

6. *Проблема Диффи-Хеллмана*. Дано простое число p, генератор α кольца целых чисел Z_p и элементы $(\alpha^a) \bmod (p)$ и $(\alpha^b) \bmod (p)$. Для известных целых чисел a и b существует полиномиальный алгоритм вычисления числа $(\alpha^{ab}) \bmod (p)$. На сегодняшний день не известно полиномиального алгоритма вычисления числа $(\alpha^{ab}) \bmod (p)$ по известным $(\alpha^a) \bmod (p)$ и $(\alpha^b) \bmod (p)$.

7. *Проблема суммы подмножества*. Дано множество положительных целых чисел $\{a_1, a_2, \ldots, a_n\}$ и положительное целое число s. Для известного подмножества чисел a_j существует полиномиальный алгоритм вычисления их суммы $\sum_j a_j = s$. На сегодняшний день не известно полиномиального алгоритма определения

такого подмножества чисел a_j, что их сумма равна s, т.е. нахождения таких a_j, что $\sum_j a_j = s$.

В табл. 1 приведены результаты исследований цикловых функций: в первой колонке указана теоретико-сложностная задача, положенная в основу ее построения, во второй колонке приведена аналитическая запись цикловой функции, в третьей колонке – оценка сложности вычисления значения цикловой функции, в четвертой – оценка вычислительной сложности ее инвертирования (оценка стойкости). Проведенные исследования показывают, что наиболее целесообразным решением следует, очевидно, считать использование цикловой функции, задача инвертирования которой сопряжена с решением теоретико-сложностной задачи извлечения квадратных корней по модулю n.

Таблица 1

Кандидаты на построение цикловой функции итеративного хеширования информации

Теоретико-сложностная задача	Кандидаты на построение цикловой функции схемы хеширования	Оценка сложности вычисления	Оценка сложности инвертирования
Проблема целочисленной факторизации	$f(x_i, H_{i-1}) = x_i H_{i-1}$, Функция определена над большими простыми числами $x_i = p$ и $H_{i-1} = q$	$O(n^2)$, где $n = \lceil \log_2 p \rceil + \lceil \log_2 q \rceil$	$L_N(\alpha, \beta) = \exp\left((\beta + o(1))(\log N)^{\alpha} (\log\log N)^{1-\alpha}\right)$. Для поля чисел общего вида сложность инвертирования составляет $L_N\left(\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right)$. Для поля чисел специального вида $N = a^b + c$ сложность инвертирования составляет $L_N\left(\frac{1}{3}, \sqrt[3]{\frac{32}{9}}\right)$
Проблема RSA	$f(x_i, H_{i-1}) = (x_i \oplus H_{i-1})^e \bmod(N)$, $\gcd(e, \varphi(p,q)) = 1$, $N = pq$	$O(\log_2 e)$ умножений, алгоритм быстрого возведения в степень	
Проблема дискретного логарифмирования	$f(x_i, H_{i-1}) = \left(\alpha^{x_i \oplus H_{i-1}}\right) \bmod(p)$, α – генератор Z_p	$O(\log_2 n)$ умножений, алгоритм быстрого возведения в степень, $O(n^3)$ для $\alpha = 2$, где $n = \lceil \log_2 p \rceil$	$\min\left\{\sqrt{p}, L_N(\alpha, \beta)\right\}$, где $L_N(\alpha, \beta) = \exp\left((\beta + o(1))(\log N)^{\alpha} (\log\log N)^{1-\alpha}\right)$. Для примитивного поля $GF(p)$ сложность инвертирования составляет $\min\left\{\sqrt{p}, L_N\left(\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right)\right\}$, Для расширенного поля $GF(2^m)$ сложность инвертирования составляет $L_N\left(\frac{1}{3}, 1, 4\right)$
Проблема Диффи-Хеллмана	$f(x_i, H_{i-1}) = \left(\alpha^{x_i \oplus H_{i-1}}\right) \bmod(p)$, α – генератор Z_p	$O(n^3)$ для $\alpha = 2$, где $n = \lceil \log_2 p \rceil$	

При определенных ограничениях на значения составного модуля n эта задача по вычислительной сложности инвертирования сопоставима с проблемами факторизации и дискретного логарифмирования. В тоже время прямое вычисление значения функции $a \equiv \left(x^2\right) \bmod(n)$ требует значительно меньшего числа операций. Следует, однако, отметить, что использование квадратичной цикловой функции не приводит к построению универсального хеширования. Следующей по вычислительной сложности идет цикловая функция

$$f(x_i, H_{i-1}) = (x_i \oplus H_{i-1})^e \bmod(N), \qquad (1)$$

задача инвертирования которой сопряжена с решением теоретико-сложностной задачи RSA, где

$$\gcd(e, \varphi(p,q)) = 1,\ N = pq,$$

$\gcd(x, y)$ – наибольший общий делитель чисел x и y.

Оценки вычислительной сложности нахождения значений этой цикловой функции, а также сложности ее инвертирования, приведенные в табл. 1, свидетельствуют о том, что при выполнении соответствующих ограничений на значение экспоненты (условие (2)):

$$\gcd(e, j(N)) = 1, \qquad (2)$$

удается строить доказуемо стойкое преобразование, используемое для шифрования/расшифрования и/или формирования/проверки цифровой подписи в криптосистеме RSA.

Использование подобного преобразования в целях ключевого хеширования применено в алгоритме MASH-2, однако выполнение условия (2) спецификацией алгоритма не гарантировано, что и подтверждают результаты проведенных в данной работе исследований.

Так, например, при выборе следующих начальных параметров $p = 1543$, $q = 263$, $N = 405809$ цикловой функции (1) (или в схеме, построенной на основе алгоритма MASH-2 при смене значений вектора инициализации секретным ключом) значение функции Эйлера $\gcd(e, \varphi(p,q)) = 257$ и условие (2) выполняться

не будет. Следовательно, коллизионные характеристики формируемых хеш-кодов не будут удовлетворять свойствам универсального хеширования.

Таким образом, применение цикловой функции (1) на основе модулярного возведение в степень позволяет строить доказуемо безопасное универсальное хеширование только при выполнении ограничений на значение модульной экспоненты и значения модуля преобразований.

Еще одним кандидатом на цикловую функцию в итеративной схеме хеширования является функция вида:

$$f\left(x_i, H_{i-1}\right) = \left(\alpha^{x_i \oplus H_{i-1}}\right) \bmod (p), \qquad (3)$$

задача инвертирования которой сопряжена с решением теоретико-сложностной задачи дискретного логарифмирования, где α – генератор мультипликативной группы чисел Z_p, а p – большое постое целое число. Использование такой цикловой функции обеспечивает построение доказуемо безопасного хеширования, коллизионные свойства которого удовлетворяют условиям универсальности.

Таким образом, проведенные исследования показали, что для построения универсального хеширования информации с доказуемым уровнем безопасности следует использовать цикловую функцию вида (1) или вида (3).

При выполнении соответствующих ограничений итеративное формирование хеш-кодов по схеме, приведенной на рис. 1, позволяет с одной стороны обеспечить выполнение условий модели доказуемой безопасности, т.е. обеспечить высокую криптографическую стойкость, с другой стороны – обеспечить выполнение условий универсального хеширования, т.е. обеспечить высокие коллизионные свойства схемы хеширования.

Платой за достижение таких свойств хеширования является сравнительно высокая вычислительная сложность формирования хеш-кода (исследования вычислительной сложности хеширования проведены ниже).

Разработка алгоритмов итеративного ключевого хеширования доказуемой стойкости на основе использования модулярных преобразований

В основу алгоритмов итеративного ключевого хеширования доказуемой стойкости на основе использования модулярных преобразований положен алгоритм MASH-1, при условии смены векторов инициализации и использовании рассмотренных выше цикловых функций, удовлетворяющих определенным ограничениям на применяемые модулярные преобразования.

Схема итеративного ключевого хеширования с использованием цикловой функции (1), разработанная по аналогии со схемой NH хеширования (алгоритм UMAC), представлена на рис. 2.

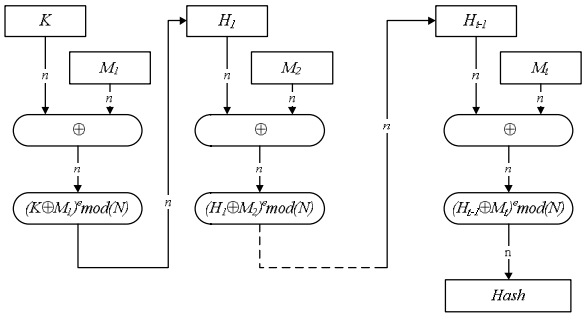

Рис. 2. Схема итеративного ключевого хеширования с использованием цикловой функции (1)

Алгоритм вычисления значения хеш-кода на основе цикловой функции (1) отличается от алгоритма MASH-2, в основном, системными установками и определением констант.

Вход. Двоичная строка x длиной $0 \le b \le 2^{n/2}$.

Выход. n-разрядный хеш-код от строки x, длиной приблизительно равный длине модуля N.

1. *Системные установки и определение констант*.

Установить RSA-подобный модуль $N = pq$ длиной m бит, где p и q случайно выбранные большие простые числа, которые сохраняются в секрете.

Установить значение модульной экспоненты равным e, причем $gcd(e, \varphi(p,q)) = 1$.

Определить двоичную длину n хеш-кода, как наибольшее произведение числа 16, т.е. длина хеш-кода определяется из условия $n = 16 \times s < m$, где s – наибольшее целое, удовлетворяющее указанному ограничению.

Как вектор инициализации выбрать $H_0 = Key$ (секретный ключ хеширования). Определить n-битное целое число как константу $A = f00...00_x$.

2. *Предобработка.* Дополнить, если необходимо, строку x нулевыми битами, для того, чтобы получить двоичную строку длиной $t \times n/2$ для наименьшего $t \ge 1$. Разделить дополненный текст на $n/2$-розрядные блоки $x_1 ..., x_t$ и прибавить последний блок x_{t+1}, который содержит $n/2$-розрядное представление числа b.

3. *Расширение.* Расширить каждый x_i блок в n-разрядный блок y_i путем вставки между 4- разрядными полубайтами блока x_i комбинации из четырех единиц (1111). Последний блок y_{t+1} формируется аналогичным способом за исключением того, что вставляется комбинация 1010.

4. *Цикловая функция.* Для всех $1 < i \le t + 1$ отобразить два n-разрядных входных блока (H_{i-1}, y_i) в один n-разрядный блок в соответствии с выражением (1).

5. *Окончание*. В качестве хеш-кода принимается n-разрядный блок H_{t+1}.

Используя цикловую функцию (3), задача инвертирования которой базируется на решении теоретико-сложностной задачи дискретного логарифмирования, построим следующую схему хеширования (рис. 3).

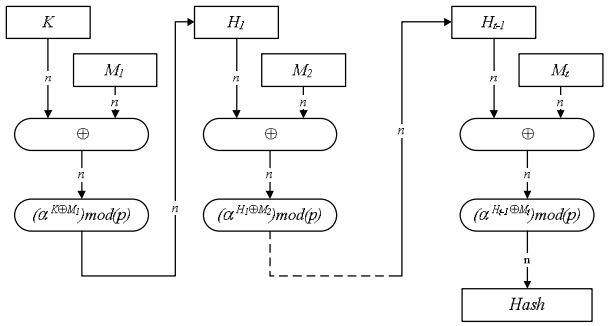

Рис. 3. Схема итеративного ключевого хеширования с использованием цикловой функции (3)

Алгоритм вычисления значения хеш-кода на основе цикловой функции (3) следующий.

Вход. Двоичная строка x длиной $0 \le b \le 2^{n/2}$.

Выход. n-разрядный хеш-код от строки x, длиной приблизительно равный длине модуля N.

1. *Системные установки и определение констант*.

Установить модуль p длиной m бит, где p случайно выбранное большое простое число, которое хранится в секрете.

Установить значение числа α, равным генератору кольца целых чисел Z_p.

Определить двоичную длину n хеш-кода, как наибольшее произведение числа 16, т.е. длина хеш-кода определяется из условия $n = 16 \times s < m$, где s – наибольшее целое, удовлетворяющее указанному ограничению.

Как вектор инициализации выбрать $H_0 = Key$ (секретный ключ хеширования). Определить n-битное целое число как константу $A = f00...00_x$.

2. *Предобработка.* Дополнить, если необходимо, строку x нулевыми битами, для того, чтобы получить двоичную строку длиной $t \times n/2$ для наименьшего $t \ge 1$. Разделить дополненный текст на $n/2$-розрядные блоки $x_1 ..., x_t$ и прибавить последний блок x_{t+1}, который содержит $n/2$-розрядное представление числа b.

3. *Расширение.* Расширить каждый x_i блок в n-разрядный блок y_i путем вставки между 4- разрядными полубайтами блока x_i комбинации из четырех единиц (1111). Последний блок y_{t+1} формируется аналогичным способом за исключением того, что вставляется комбинация 1010.

4. *Цикловая функция.* Для всех $1 < i \le t + 1$ отобразить два n-разрядных входных блока (H_{i-1} , y_i) в один n-разрядный блок

5. *Окончание*. В качестве хеш-кода принимается n-разрядный блок H_{t+1}.

Разработанные вычислительные алгоритмы отличаются от алгоритмов бесключевого хеширования MASH-1 и MASH-2, в основном, системными установками и определением констант. Кроме того, предлагаемые схемы хеширования являются ключевыми, в качестве секретных ключевых данных используются сменные вектора инициализации $H_0 = Key$. На применяемые модулярные преобразования в цикловой функции ключевого хеширования накладываются рассмотренные выше ограничения.

Выводы

Таким образом, предлагаемый метод универсального хеширования с использованием модулярных преобразований позволяет реализовать формирование аутентификаторов (хеш-кодов) с обеспечением требуемых показателей безопасности. Разработанные алгоритмы позволяют практически реализовать предлагаемые схемы хеширования, как в программном, так и в аппаратном виде. Перспективным направлением дальнейших исследований является разработка предложений по реализации итеративного ключевого хеширования доказуемой стойкости с использованием модулярных преобразований.

Список литературы

1. Кузнецов О.О. Захист інформації в інформаційних системах / О.О. Кузнецов, С.П. Євсеєв, О.Г. Король. – Х.: Вид. ХНЕУ, 2011. – 504 с.

2. Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption, April 19, 2004 – V/ 0.15 (beta), Springer-Verlag.

3. Столлингс В. Криптография и защита сетей: принципы и практика, / В. Столлингс. – М.: Вильямс, 2001. – 672 с.

4. Король О.Г. Исследование методов обеспечения аутентичности и целостности данных на основе односторонних хеш-функций / О.Г. Король, С.П. Евсеев // Захист інформації. – 2008. – Спецвипуск (40).– С. 50-55.

Поступила в редколлегию 3.07.2013

Рецензент: д-р техн. наук**,** проф. В.А. Хорошко, Национальный авиационный университет, Киев.

ОБГРУНТУВАННЯ ВИБОРУ ЦИКЛОВОЇ ФУНКЦІЇ ДЛЯ ІТЕРАТИВНОГО ГЕШУВАННЯ ІНФОРМАЦІЇ

О.Г. Король, Л.Т. Пархуць, С.П. Євсеєв

Обґрунтовується вибір циклових функцій у схемі доказовою стійкого ключового універсального гешування, розробляються алгоритми ітеративного ключового гешування доказовою стійкості на основі використання модулярних перетворень.

Ключові слова: *модулярні перетворення, циклові функції, ключове гешування.*

RATIONALE FOR SELECTION ROUND FUNCTION FOR THE ITERATIVE HASHING INFORMATION

O.G. Korol, L.T. Parkhuts, S.P. Evseev

The choice of cycle functions in the scheme provably secure key universal hashing algorithms developed iterative hash key demonstrable resistance through the use of modular transformations.

Keywords: *modular transformation, cyclic function key hashing.*