УДК 004.021+681.3.05

А.В. Антонов

Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков

РАЗВИТИЕ МЕТОДА ПОСТРОЕНИЯ ХЕШ-ФУНКЦИИ НА ОСНОВЕ ХАОТИЧЕСКИХ ОТОБРАЖЕНИЙ С ПЕРЕМЕННЫМИ ПАРАМЕТРАМИ И ПАРАЛЛЕЛЬНОЙ ОРГАНИЗАЦИЕЙ ВЫЧИСЛЕНИЙ

Предлагается вариант усиления стойкости хеш-функции, построенной на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений. Показана его стойкость к атакам, использующим алгоритм поиска близких коллизий и особенности реализации хаотических систем на конечном множестве состояний в цифровых вычислительных системах.

Ключевые слова: *хеш-функция, хаотическое отображение, распараллеливание, коллизия.*

Введение

Стремительное и повсеместное внедрение информационных технологий во все сферы человеческой деятельности привело к актуализации задач обеспечения функциональной безопасности и живучести информационных и информационно-управляющих систем. Отказы в работе таких систем в ряде областей их применения могут приводить к недопустимому ущербу, а подчас и к катастрофическим последствиям. Одним из аспектов их функциональной безопасности и живучести является обеспечение достоверности информации, недопущение как предумышленного, так и непреднамеренного ее разрушения или искажения. Эта задача в современных информационных системах может решаться с помощью хеш-функций.

В последнее время в качестве одного из альтернативных подходов к конструированию сжимающих функций алгоритмов хеширования все чаще предлагается использовать достижения теории динамического хаоса. Из работ, выполненных в данном направлении, следует выделить метод построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений, предложенный в публикации [1]. Большой практический интерес к этому методу обусловлен его способностью в значительной мере компенсировать низкую эффективность расчета траекторий хаотических отображений путем их распараллеливания.

Однако детальное исследование, представленное в работах [2] и [3], показало его уязвимость к поиску и обнаружению близких коллизий, а так же коллизий первого и второго рода.

В данной работе будет предложен вариант дальнейшего развития метода, позволяющий устранить выявленные в работах [2, 3] недостатки, а также повысить эффективность вычислений.

Ниже будет приведено описание усовершенствованного метода и показано как введенные изменения в его структуру позволят бороться с выявленными недостатками.

Предлагаемый метод основан на использовании совокупности следующих приемов и правил при построении хеш-функции:

– применение итеративной последовательной схемы Меркла-Дамгарда;

– применение итеративной параллельно-последовательной сжимающей функции

– использование нескольких хаотических отображений в качестве базовых преобразований;

– использование переменных параметров отображений на различных этапах обработки сообщения и его отдельных блоков.

Параметры и константы

Параметрами, константами и входными данными хеш-функции являются:

$q = 0x11$ – константа модификации элементов исходного сообщения в параллельной функции сжатия;

L – требуемая длина результирующего хеша в битах, $1 \le L$, рекомендуется не менее 128 бит;

p – точность организации внутренних вычислений (в битах, кратно 8), $8 \le p$, рекомендуется не менее 16 бит;

d – глубина распараллеливания, $1 \le d$, верхним пределом являются аппаратные возможности вычислительных средств и средняя длина цикла отображений для заданной точности вычислений;

базовые преобразования (виды хаотических отображений) и их инициализирующие значения;

M – исходное сообщение;

N – длина исходного сообщения (в битах).

Базовые преобразования

В качестве базовых преобразований в предлагаемом методе построения хеш-функции могут использоваться любые хаотические отображения, задаваемые рекуррентными уравнениями и имеющие

симметричное распределение (требование симметрии связано с особенностями формирования промежуточных хешей).

В работе [1] использовались представители класса кусочно-линейных отображений: палаточное и кусочно-линейное отображение второго порядка, которое в [1] называют просто кусочно-линейным.

Хотя кусочно-линейные отображения в силу специфики своей структуры имеют ряд недостатков при использовании их в хеш-функции (например, слишком быстро вырождаются на ограниченном множестве состояний), для простоты анализа и сравнения с исходной версией хеш-функции рассмотрим именно их.

Рекуррентное уравнение для палаточного отображения задается в виде:

$$x_{i+1} = \begin{cases} \dfrac{x_i}{a}, & 0 \le x_i < a \\ \dfrac{1-x_i}{1-a}, & a \le x_i \le 1 \end{cases}, \qquad (1)$$

где $x_i \in (0,1)$ – точки траектории;

$a \in (0,1)$ – управляющий параметр отображения (часто называемый ключевым). Рекуррентное уравнение для кусочно-линейного отображения второго порядка задается следующим выражением:

$$x_{i+1} = \begin{cases} x_i/\beta & 0 \le x_i < \beta \\ (x_i-\beta)/(0.5-\beta) & \beta \le x_i < 0.5 \\ (1-x_i-\beta)/(0.5-\beta) & 0.5 \le x_i < 1-\beta \\ (1-x_i)/\beta & 1-\beta \le x_i < 1 \end{cases}, \qquad (2)$$

где $x_i \in (0,1)$ – точки траектории; $\beta \in (0, 0.5)$ – управляющий параметр отображения (ключевой).

Этап инициализации и общая структура хеш-функции

На этапе инициализации исходное сообщение представляется набором векторов (блоков данных) по $L-1$ элементов размером 1 байт в каждом векторе. Последний блок сообщения дополняется до длины кратной $(L-1)\cdot 8$ бит: сначала добавляется 64 бита, в которые записывается длина N исходного сообщения, а потом необходимое количество бит в виде последовательности $(1010\ldots10)_2$. Дополнение делается даже, если длина N исходного сообщения кратна $(L-1)\cdot 8$ битам. В результате получаем набор векторов $\vec{m}'_i$, где $i = 1,2\ldots n'$, а n' – число векторов (блоков) на которые были разбито исходное сообщение. В качестве общей структуры хеш-функции рассмотрим усиленную схему Меркла-Дамгарда [4] (рис. 1).

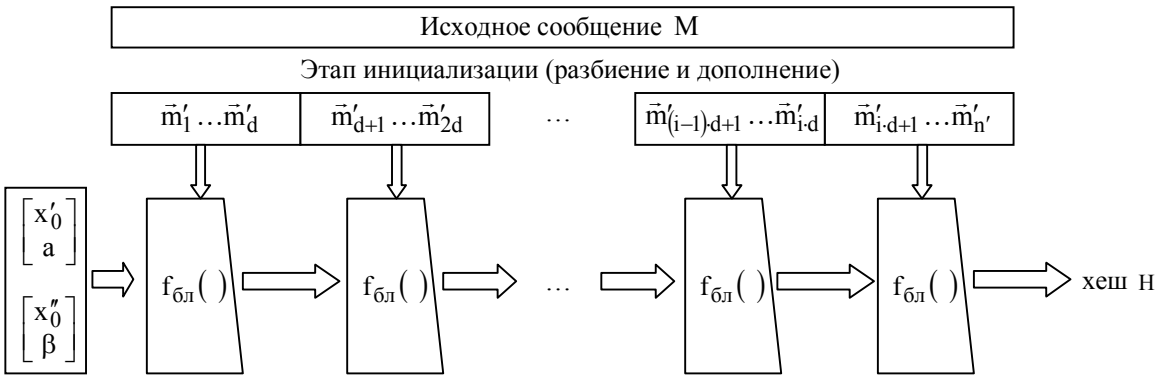

Рис. 1. Общая структура хеш-функции

Отметим, что разбиение и дополнение (препроцессинг) исходного сообщения на вектора $\vec{m}'_i$ может осуществляться последовательно, кратно глубине распараллеливания d по мере обработки сообщения блочной функцией сжатия $f_{бл}(\ )$, т.е. полная предварительная подготовка исходного сообщения не требуется. Следует также отметить, что число векторов d (глубина распараллеливания), обрабатываемых блочной функцией за одну ее итерацию, должно выбираться адаптивно, исходя из соображений стойкости, изложенных в работе [3], и возможностей аппаратных средств (числа поддерживаемых параллельных вычислительных потоков).

В качестве $f_{бл}(\ )$ будет использоваться хеш-функция, предложенная в работе [1], на вход которой будут поступать блоки исходного сообщения (вектора $\vec{m}'_i \ldots \vec{m}'_{i+d}$).

Блочная функция сжатия

$f_{бл}(\ )$ – сложная функция, входом которой являются инициализирующие значения хаотических отображений, вектор $\vec{c}$ (назначение и порядок формирования вектора будут описаны ниже), хеш-значение H, полученное на предыдущей итерации $f_{бл}(\ )$ и очередной блок данных (набор векторов $\vec{m}'_1 \ldots \vec{m}'_d$). Выходом функции являются новые инициализирующие значения для следующей итерации, новый вектор $\vec{c}$ и полученный промежуточный блочный хеш H. Строится на основе параллельной

функции сжатия $f_{пар}(\ )$, инициализирующей $f_{ини}(\ )$ и финализирующей $f_{фин}(\ )$ блочной функции. Структуру $f_{бл}(\ )$ можно представить с помощью рис. 2.

$$f_{ини}\begin{pmatrix}\vec{m}'_1\ldots\vec{m}'_d\\ x'_0,a\\ x''_0,\beta\\ \vec{c},H\end{pmatrix}\Rightarrow\begin{matrix}h_1=f_{пар\,1}(\vec{m}_1)\\ h_2=f_{пар\,2}(\vec{m}_2)\\ \ldots\\ h_d=f_{пар\,d}(\vec{m}_d)\\ x'_0,a\\ x''_0,\beta\\ \vec{c},H\end{matrix}\Rightarrow f_{фин}\begin{pmatrix}\vec{h}\\ x'_0,a\\ x''_0,\beta\\ \vec{c},H\end{pmatrix}\Rightarrow\begin{bmatrix}H\\ x'_0,a\\ x''_0,\beta\\ \vec{c}\end{bmatrix}$$

Рис. 2. Общая структура блочной функции сжатия

$f_{ини}(\ )$ – сложная функция, выходом которой являются набор векторов для работы параллельных функций сжатия, новые инициализирующие значения хаотических отображений и вектор $\vec{c}$ для следующей итерации блочной функции. Число векторов должно соответствовать глубине распараллеливания d, а число элементов в векторе – требуемой длине хеша L. На последней итерации $f_{бл}(\ )$ число векторов может быть меньше d исходя из реальной длины исходного сообщения. Строится на основе базовых хаотических преобразований и специальных математических конструкций.

В ходе инициализации формируются вектора $\vec{x}'$ и $\vec{x}''$ размерностью d и L элементов соответственно. Причем элементы вектора $\vec{x}'$ соответствуют точкам траектории отображения (1), а вектора $\vec{x}''$ – отображения (2). Выбор инициализирующих значений отображений x'_0 и a для (1), а также x''_0 и β для (2) на первой итерации $f_{бл}(\ )$ должен быть ориентирован на получение требуемой минимально допустимой длины цикла при заданной точности вычислений (т.е. длина цикла k должна быть больше глубины распараллеливания d).

С использованием пары векторов $\vec{x}'$ и $\vec{x}''$, а также набора векторов $\vec{m}'_i$ формируются вектор-строка $\vec{c}$ и вектор-столбец $\vec{r}$, элементы которых задаются значениями:

$$c_j=\left(\bigoplus_{i=1}^{d}\left(m_{i,j}\oplus x'_i\right)\right)\oplus c'_j+x''_j;$$

$$r_i=\sum_{j=1}^{L-1}\left(\left(m_{i,j}+x''_j\right)\oplus x'_i\right),$$

где c'_j – элементы вектора $\vec{c}$ полученные на предыдущей итерации $f_{бл}(\ )$; $m_{i,j}$ – элементы векторов $\vec{m}'_i$, $j=1,2,\ldots,L$, $i=1,2,\ldots,d$; $\oplus$ – битовая операция «исключающее или»; «+» и «Σ» – операции сложения по модулю 2^8.

Вектор $\vec{c}$, на первой итерации функции $f_{бл}(\ )$ инициализируется нулями, а число элементов вектора соответствует требуемой длине хеша L и каждый элемент представляется 8 битами. На последней итерации $f_{бл}(\ )$ блок исходного сообщения дополняется вектором $\vec{c}$, который выступает в качестве самостоятельного вектора данных при дальнейшей обработке параллельной функцией сжатия.

На основе векторов $\vec{m}'_i$ и $\vec{r}$ формируется матрица M' размерностью $d\times L$:

$$M'=\begin{bmatrix}m_{1,1} & m_{1,2} & \ldots & m_{1,L} & r_2\\ m_{2,1} & m_{2,2} & \ldots & m_{2,L} & r_3\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ m_{d-1,1} & m_{d-1,2} & \ldots & m_{d-1,L} & r_d\\ m_{d,L} & m_{d,L-1} & \ldots & m_{d,1} & r_1\end{bmatrix}=\begin{bmatrix}\vec{m}_1\\ \vec{m}_2\\ \ldots\\ \vec{m}_{d-1}\\ \vec{m}_d\end{bmatrix}.$$

Вектор-строки $\vec{m}_i$ матрицы M' используются при обработке параллельной функцией сжатия для формирования промежуточных хешей.

$f_{пар}(\ )$ – параллельная функция сжатия, входом которой является подготовленная вектор-строка $\vec{m}_i$. Выходом функции является промежуточный параллельный хеш. Строится на основе базовых хаотических преобразований по итеративной последовательной схеме. Ее ядром является преобразование вида:

$$h_{i,j}=f_{plm}\left(f_{atm}\left(h_{i,j-1},m_{i,j}\right),m_{i,j}\right),$$

которое для каждого параллельного потока $i=1,2,\ldots,d$ итеративно осуществляет преобразование элементов $m_{i,j}$ $(j=1,2,\ldots,L)$ функциями $f_{atm}(\ )$ и $f_{plm}(\ )$.

Значение функции $f_{atm}(\ )$ определяется координатой точки траектории отображения (1); z – порядковый номер точки, т.е. $f_{atm}\left(h_{i,j},m_{i,j}\right)=x_z$. Значение управляющего параметра отображения на каждой итерации функции сжатия определяется как

$a_{i,j} = (i/n + j/L)/2$, а начальная точка траектории как $x_0 = h_{i,j-1}$, где $h_{i,j-1}$ – выход предыдущей итерации $f(\ )$ или $h_{i,0} = m_{i,128}/256$ для ее первой итерации.

Значение функции $f_{plm}(\ )$ определяется координатой точки траектории отображения (2), z – порядковый номер точки, т.е. $f_{plm}(h_{i,j}, m_{i,j}) = x_z$. Значение управляющего параметра отображения на каждой итерации функции сжатия определяется как $\beta_{i,j} = a_{i,j}/2$, а начальное значение отображения – как выход функции $f_{atm}(h_{i,j}, m_{i,j})$, т.е последняя точка траектории отображения (1).

Таким образом, $f_{пар}(\ )$ последовательно перебирая элементы вектора $\vec{m}_i$ формирует наборы точек $h_{i,j}$. Предлагается осуществлять два прохода $f_{пар}(\ )$ по элементам векторов $\vec{m}_i$ в прямом и обратном порядке. Порядковые номера отбираемых точек траектории предлагается вычислять для функции $f_{atm}(\ )$ как $z_j = \lfloor (j/L)\cdot((m_{i,j} \vee q) \wedge 0x0F) \rfloor$ и $z_j = \lfloor (j/L)\cdot(((m_{i,j} \vee q) \wedge 0xF0)/0x0F) \rfloor$ при прямом и обратном проходе соответственно. Для функции $f_{plm}(\ )$ при прямом проходе $z_j = \lfloor (1-j/L)\cdot((m_{i,j} \vee q) \wedge 0x0F) \rfloor$ и при обратном $z_j = \lfloor (1-j/L)\cdot(((m_{i,j} \vee q) \wedge 0xF0)/0x0F) \rfloor$. Здесь операции «$\wedge$» – битовое «И», а «$\vee$» – битовое «ИЛИ» Промежуточные параллельные хеши h_i строятся путем округления промежуточных значений $h_{i,j}$, полученных при обратном проходе функции сжатия $f_{пар}(\ )$ после каждой итерации. При этом из полученных последовательностей (цифр 0 и 1) формируется битовый хеш соответствующей длины.

Обобщенная структура функции сжатия $f_{пар}(\ )$ приведена на рис. 3.

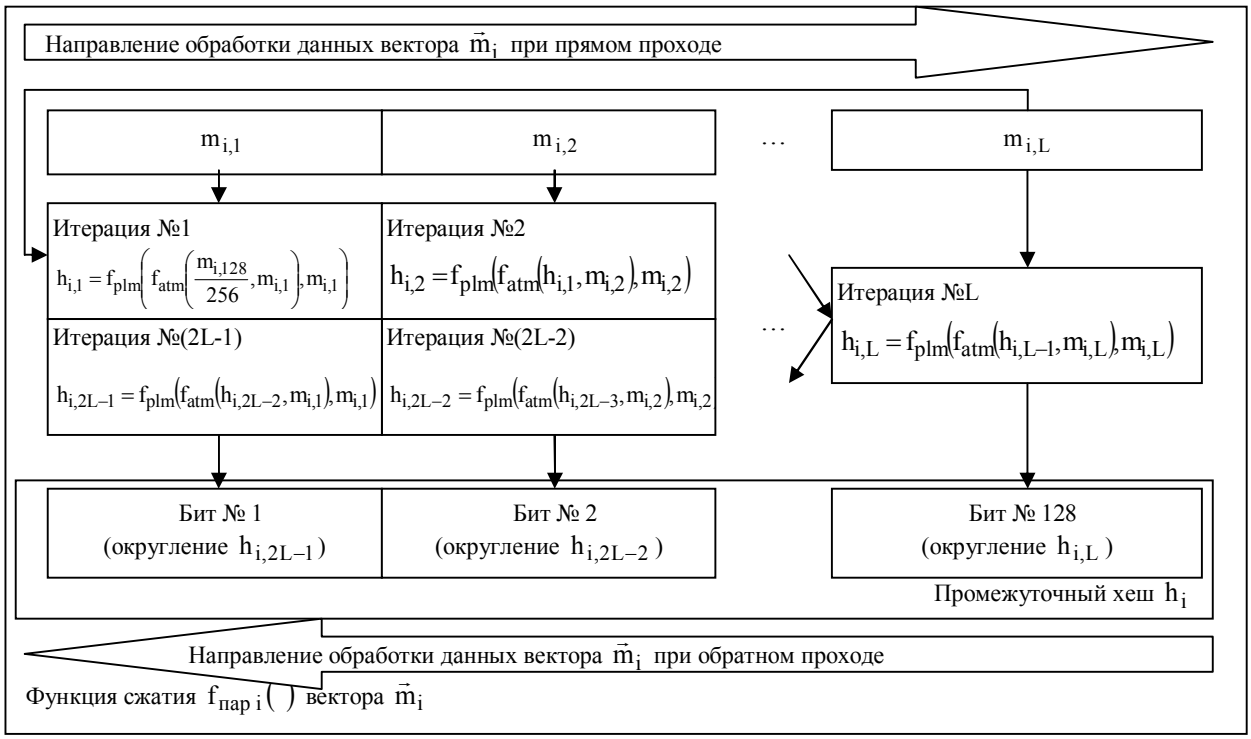

Рис. 3. Обобщенная структура параллельной функции сжатия

$f_{фин}(\ )$ – финализирующая блочная функция, входом которой являются выходы параллельных функций сжатия (промежуточные параллельные хеши h_i), хеш-значение H' предыдущей итерации $f_{бл}(\ )$, вектор $\vec{c}$. Выходом функции являются новые значения инициализирующих значений, вектор $\vec{c}$, новый промежуточный блочный хеш H. На последней итерации $f_{бл}(\ )$ выходом $f_{фин}(\ )$ будет только хеш H, который и будет хеш-значением всего сообщения.

Новый промежуточный блочный хеш строится путем объединения всех хешей h_i, а также хеш-значения H' предыдущей итерации $f_{бл}(\ )$, операцией «исключающее или» в хеш-значение всего блока:

$$H = \left(\bigoplus_{i=1}^{n} h_i \right) \oplus H' .$$

Вектор $\vec{c}$, сформированный в функции $f_{ини}(\ )$ передается на выход без изменений. Новые инициализирующие значения x'_0 и a отображения (1), а также x''_0 и β отображения (2) для следующей итерации $f_{бл}(\ )$ выбираются следующим образом. Отбираются последние элементы векторов $\vec{x}'$ и $\vec{x}''$, сформированных функцией $f_{ини}(\ )$, и исходные значения управляющих параметров a и β. Далее для усиления хеш-функции предлагается корректировать отобранные инициализирующие значения, комбинируя их, например, со значением промежуточного хеша H. Порядок объединения должен выбираться исходя из свойств и характеристик конкретных базовых (хаотических) преобразований, с целью обеспечения наиболее выгодных режимов работы отображений. В рассматриваемом варианте использования представителей класса кусочно-линейных отображений можно их комбинировать, разбивая промежуточный хеш H на блоки кратные точности вычислений p и объединяя все блоки и исходные инициализирующие значения бинарной операцией «исключающее или».

Общая характеристика предложенного метода

Как видно из описания усовершенствованного метода построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений, новая структура метода в целом сохранила основные особенности исходного метода, предложенного в работе [1]. В частности:

– использование итеративной параллельно-последовательной сжимающей функции

– использование нескольких хаотических отображений в качестве базовых преобразований;

– использование переменных параметров отображений на различных этапах обработки сообщения и его отдельных блоков.

Метод опирается на свойства хаотических отображений, описывается достаточно простой математической моделью и позволяет распараллелить вычисления. При этом усовершенствованный метод построен на основе хорошо изученной, надежной и апробированной итеративной последовательной схемы Меркла-Дамгарда. Также введены ряд дополнительных мер направленных на повышение производительности и стойкости метода. Ниже будут рассмотрены особенности усовершенствованного метода по сравнению с исходной версией, предложенной в работе [1], и показано как предпринятые меры по развитию метода позволяют повысить эффективность вычислений и устранить уязвимости, выявленные в работах [2, 3].

Эффективность вычислений усовершенствованного метода

Несмотря на распараллеливание вычислений исходная версия метода, предложенного в работе [1], достаточно требовательна к вычислительным ресурсам. Так в параллельной функции сжатия на каждой ее итерации рассчитывалось $z = \lfloor (j/L) \cdot m_{i,j} \rfloor$ и $z = \lfloor (1 - j/L) \cdot m_{i,j} \rfloor$ точек отображений (1) и (2) соответственно. В предположении, что элементы исходного сообщения статистически равновероятны, то один проход функции сжатия требовал не менее $128 \cdot 2.25 \cdot 128$ простых арифметических операций (таких как вычитание и деление, без учета дополнительных затрат вычислительных ресурсов на условные переходы), причем в данном выражении 128 – среднестатистическое значение элемента вектора $\vec{m}_i$, 2.25 – среднее количество операций для вычисления одной итерации хаотического отображения, 128 – фиксированное число элементов в векторе $\vec{m}_i$. Т.е. один проход функции сжатия требовал около 36864 арифметических операций. Ввод второго прохода увеличивал бы это значение вдвое до 73728 операций, а при увеличении длины хеша, число операций увеличивалось бы пропорционально. Для сравнения алгоритм SHA-512 блок данных длиной 128 байт (1024 бита) обрабатывает в среднем за 2080 арифметических и битовых операций, формируя при этом более длинный хеш в 512 бит. Таким образом, не смотря на распараллеливание вычислений, задача понижения сложности вычислений в функции сжатия оставалась по-прежнему актуальной;

В усовершенствованном методе были предприняты меры по уменьшению длительности рассчитываемых траекторий хаотических отображений на каждой итерации функции сжатия $f_{пар}(\ )$. Учитывалось, что с одной стороны высокие характеристики хаотических отображений позволяют добиться достижения лавинного эффекта уже на малых отрезках траектории. С другой - вследствие особенностей реализации хаотических отображений на конечном множестве состояний на длинных участках наблюдения траектории начинают демонстрировать циклическое поведение. Таким образом, предложенное решение значительно не повлияет на стойкость параллельной функции сжатия (особенно при введении второго ее прохода), но позволит значительно сократить сложность ее вычисления. Реализовано данное предложение в рассмотренном выше усовершенствованном методе отбором значения младшего октета элемента вектора $\vec{m}_i$ при прямом проходе и старшего октета при обратном для рас-

чета длины траекторий хаотических отображений. Таким образом, на каждом проходе функции сжатия будут использоваться уникальные элементы векторов $\vec{m}_i$, а общая сложность вычисления значения функции сжатия уменьшится в среднем до 4608 операций при двух проходах для длины хеша $L = 128$. Такая мера позволит в целом повысить вычислительную эффективность рассматриваемого метода построения хеш-функции. Хотя сложность функции сжатия все еще остается более высокой, чем, например, у алгоритма SHA-512, но за счет распараллеливания вычислений (например, при организации вычислений на четырехядерной/четырехпроцессорной платформе) затрачиваемое время на вычисление хеша для достаточно длинных сообщений у них будет соизмеримо. При дальнейшем увеличении количества параллельных вычислительных потоков рассматриваемая хеш-функция будет более быстрой.

Стойкость к коллизиям усовершенствованного метода

В работе [2], было показано, что исходная версия метода являлась уязвимой к возникновению близких коллизий и к атакам на восстановление второго прообраза с использованием алгоритма поиска близких коллизий. Причина уязвимости крылась в недостаточном связывании элементов исходного сообщения в массиве M, несовершенстве структуры функции сжатия, которая вычисляла биты промежуточных хешей последовательно, и несовершенстве последовательно-параллельной схемы хеш-функции, в которой промежуточные хеши объединяются простой операцией «исключающее или». Причем последовательное вычисление битов промежуточных хешей являлось основным фактором успеха данной атаки. Очевидно, что «хорошая» функция сжатия должна обладать всеми свойствами хеш-функции, в том числе свойством лавинного эффекта, когда изменение любого бита исходного сообщения приводит к изменению около половины бит промежуточного хеша. В работе [1] предлагалось решать эту задачу в основном на подготовительном этапе (формирования матрицы M), что, как было показано в работе [2], являлось недостаточным. Таким образом, наиболее очевидным и оправданным способом устранения данной уязвимости является коррекция параллельной функции сжатия с целью усиления лавинного эффекта, обеспечения зависимости всех бит промежуточного хеша от любого элемента обрабатываемого вектора $\vec{m}_i$. Простейшим способом добиться требуемого эффекта (не меняя значительно структуру функции сжатия) является использование двух проходов функции сжатия над элементами вектора $\vec{m}_i$ (в частности, прямого и зеркального обратного), причем значения промежуточного хеша должны вычисляться по второму проходу функции. Такое решение, примененное в предложенном усовершенствованном методе построения хеш-функций, а так же меры по более тесному связыванию элементов исходного сообщения и применение итеративной последовательной схемы Меркла-Дамгарда, позволили устранить уязвимости к атакам на поиск близких коллизий и значительно усилить лавинный эффект в функции сжатия.

Также в работе [3], было показано, что исходная версия метода являлась уязвимой к атакам, использующим особенности реализации хаотических систем на конечном множестве состояний в цифровых вычислительных системах, и доказано, что она не является устойчивой к возникновению коллизий первого и второго рода. Причина уязвимости крылась в недостаточном связывании элементов исходного сообщения в массиве M, несовершенстве последовательно-параллельной схемы хеш-функции, в которой промежуточные хеши объединяются простой операцией «исключающее или», особенностями реализации хаотических отображений на конечном множестве состояний. Причем особенности реализации хаотических отображений на конечном множестве состояний (цикличность траекторий отображений) являлись основным фактором успеха рассмотренных в работе [3] атак, а сами уязвимости проявлялись при достаточной длине исходного сообщения. Для борьбы с обнаруженными уязвимостями в предложенном усовершенствованном методе были предприняты следующие меры:

– повышение точности промежуточных вычислений, что позволяет расширить пространство внутренних состояний (как следствие увеличить среднюю длину цикла отображений) и повысить стойкость функции. Причем следует отметить, что точность вычислений выбирается адаптивно (в виде параметра хеш-функции), исходя из необходимого баланса стойкости и вычислительных возможностей аппаратных средств при практической реализации хеш-функции;

– коррекция общей структуры метода, путем более совершенной комбинации параллельной и последовательной итеративной схем, в частности наложения исходной схемы на структуру Меркла-Дамгарда, что позволило в значительной степени устранить структурные уязвимости метода;

– коррекция и внесения ограничений на параметры функции в целом, способ формирования матрицы M, направленные на устранение уязвимостей вызванных особенностями реализации хаотических отображений на конечном множестве состояний;

– вариативность глубины распараллеливания вычислений (в виде параметра хеш-функции), исходя из необходимого баланса стойкости и вычислительных возможностей аппаратных средств при практической реализации хеш-функции.

Кроме того следует отметить, что в ходе анализа в работах [2, 3] не рассматривались более частные ситуации, в которых исходное сообщение представлено, например, длинными последовательностями нулей. В этом случае все итерации функции сжатия будут «холостыми» и продублируют исходное значение, т.е. промежуточный хеш будет также представлен последовательностью из одних нулей или единиц (в зависимости от начального значения). Конкретные атаки, использующие этот факт, не рассматривались, но очевидно, что такая предсказуемость результатов вычислений является неприемлемой для надежной хеш-функции.

Поэтому в усовершенствованном методе был введен механизм коррекции исходных данных для устранения «холостых» итераций функции сжатия путем сложения (бинарная операция «ИЛИ») элементов $m_{i,j}$ вектора $\vec{m}_i$ с константой $q = 0x11$. При этом гарантированно на каждой итерации функции сжатия (как при прямом, так и при обратном проходе) будет вычисляться хотя бы одна точка траектории хаотического отображения.

Следует также отметить, что большая гибкость и вариативность параметров метода (например, возможность выбора различных хаотических отображений в его реализациях, возможность задания требуемой длины хеша и т.д.) закладывают необходимый потенциал для дальнейшей его модернизации и развития.

Выводы

В работе [1] был предложен оригинальный и перспективный механизм (метод) построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений. Однако, в работах [2, 3] было показано, что соответствующая хеш-функция не является стойкой к близким коллизиям, коллизиям первого и второго рода.

При этом было указано, что, несмотря на выявленные недостатки, сам метод представляет интерес для дальнейшего изучения и развития.

Поэтому в данной работе был предложен вариант его усовершенствования и устранения выявленных недостатков, повышения вычислительной эффективности. В частности, можно утверждать, что усовершенствованный метод является стойким к выявленным в работах [2, 3] уязвимостям.

В дальнейших исследованиях будет проведен анализ усовершенствованного метода на наличие потенциальных уязвимостей с помощью методов статистического анализа.

Список литературы

1. Yantao Li Parallel Hash function construction based on chaotic maps with changeable parameters [Text] / Li Yantao , Di Xiao, Shaojiang Deng, Qi Han, Gang Zhou // Neural Computing and Applications – 2011. – Vol. 20, № 8/ – P. 1305-1312.

2. Антонов А.В. Анализ уязвимостей структуры хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений [Текст] / А.В. Антонов // Системи управління, навігації та зв'язку. – 2012. – № 2(22). – С. 157-162.

3. Антонов, А.В. Анализ уязвимостей реализации в цифровых вычислительных системах хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений [Текст] / А.В. Антонов // Збірник наукових праць ХУПС. – Х.: ХУПС, 2012. – Вип. 4(33). – С. 123-129.

4. Merkle, R.C. A Certified Digital Signature. In Advances in Cryptology [Text] / R.C. Merkle // CRYPTO '89 Proceedings, Lecture Notes in Computer Science – G. Brassard, ed, Springer-Verlag. – 1989. – V. 435. – P. 218-238.

Поступила в редколлегию 16.05.2012

Рецензент: д-р техн. наук, проф. П.Ю. Костенко, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

РОЗВИТОК МЕТОДУ ПОБУДОВИ ҐЕШ-ФУНКЦІЇ НА ОСНОВІ ХАОТИЧНИХ ВІДОБРАЖЕНЬ ЗІ ЗМІННИМИ ПАРАМЕТРАМИ Й ПАРАЛЕЛЬНОЇ ОРГАНІЗАЦІЇ ОБЧИСЛЕНЬ

А.В. Антонов

Пропонується варіант посилення стійкості геш-функції, побудованої на основі хаотичних відображень зі змінними параметрами й паралельної організації обчислень. Показана його стійкість до атак, що використовують алгоритм пошуку близьких колізій та особливості реалізації хаотичних систем на кінцевій множині станів в цифрових обчислювальних системах.

Ключові слова: *геш-функція, хаотичне відображення, розпаралелювання, колізія.*

AN IMPROVED METHOD OF CONSTRUCTING OF HASH FUNCTION BASED ON CHAOTIC MAPS WITH CHANGEABLE PARAMETERS AND PARALLEL CALCULATIONS

A.V. Antonov

The variant of improved of hash function based on chaotic maps with changeable parameters and parallel calculations were proposed. It appeared to be resistant to attacks that use near-collisions search algorithm and implementation features of chaotic systems on a finite set of states in a digital computer systems.

Keywords: *hash function, the chaotic map, paralleling, collision.*