УДК 003.26

# ПОСТРОЕНИЕ ХЭШ-ФУНКЦИИ И СОЗДАНИЕ ЭЦП НА ОСНОВЕ КОМПОЗЦИИ АФФИННОГО И ВЕРОЯТНОСТНОГО МЕТОДОВ ШИФРОВАНИЯ

*Горемыкин А.В., Король Е.В., Шкодина Л.Н.*
*Донецкий национальный университет, Украина*

*В данной статье рассматривается задача обеспечения информационной безопасности электронных документов с помощью электронно-цифровой подписи. Для построения ЭЦП исходный электронный документ обрабатывается с помощью хэш-функции, а ее результат – хэш-код зашифровывается композицией аналитического аффинного и вероятностного шифров. Создан программный комплекс в среде Visual C++ для получения хэш-кода и построения ЭЦП.*

В настоящее время в связи с развитием безбумажных способов передачи информации, хранения данных, электронного перевода денежных средств, проблема виртуального подтверждения аутентичности документов приобрела особую остроту. Развитие любых подобных систем теперь немыслимо без существования электронных подписей под электронными документами. Построение ЭЦП основано на использование хэш-функций [1-6].

В данной работе построена хэш-функция по типу «разработанная с нуля» [3, 6] с использованием операций в классе вычетов [1, 4], а ЭЦП под электронным документом создается при помощи композиции двух шифров: аффинного шифра и шифра основанного на вероятностном шифровании.

**Построение хэш-функции**

Работа хэш-функции начинается с того, что входная последовательность делится на блоки (векторы) по 20 элементов. В случае если длина последнего блока текста меньше 20, то он дополняется заранее обусловленным образом. Далее инициализируется пять целочисленных векторов по 20 элементов: $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ и пять дополнительных векторов с начальными значениями $a_1=A_1$, $a_2=A_2$, $a_3=A_3$, $a_4=A_4$, $a_5=A_5$. Основной цикл, совершаемый над каждым 20-элементным блоком, состоит из последовательного применения различных операторов, состоящих из операций над функциями $g_1(X,Y,Z)$, $g_2(X,Y,Z)$, $g_3(X,Y,Z)$, $g_4(X,Y,Z)$, $g_5(X,Y,Z)$ и векторами $a_1$, $a_2$, $a_3$, $a_4$, $a_5$. В функциях $g_i(X,Y,Z)$, $i=\overline{1,5}$ фактическими аргументами являются переменные массивы $a_i$, $i=\overline{1,5}$ которые в разных операторах при обработке текущего блока исходного текста произвольно выбираются и содержат операции над аргументами, такие как обращения элементов векторов в классе вычетов, сдвиг, сложение векторов и т.д.

$$
\begin{aligned}
&g_1(X,Y,Z)=X^{-1}+Y^2+17+Z;\\
&g_2(X,Y,Z)=X\times Y\times Z+3\cdot Y+Z<<8;\\
&g_3(X,Y,Z)=X\times Y+X\times Y^{-1}+18+Z;\\
&g_4(X,Y,Z)=X<<13+8\cdot X+3+Y+Z\cdot 10+17+Z\times Y;\\
&g_5(X,Y,Z)=X\times Y\times Z-18\cdot X\times Y^{-1}.
\end{aligned}
\qquad (1)
$$

Здесь $X^{-1}$ – вектор, состоящий из элементов, мультипликативных обратных к элементам вектора X в классе вычетов, «×» - по членное перемножение элементов векторов, «·» -умножение вектора на число, $(X<<n)$ – циклический сдвиг элементов X на n позиций. Например, один из операторов над текущим k-м блоком имеет вид:

```
for (i=1; i≤5; ++i)
{
 R=(a2<<(3+i)+gi(a1,a5,a2)+Mk);
 a2=a4+a3;                                   (2)
```

```
 a3=a1<<(4+i)+a5<<5;
 a4=a3-a2<<8+a3;
 a5=a1-a2<<(8+2*i);
 a1=R;
}
```

где $k$ – номер блока, $M_k$ – $k$-й обрабатываемый блок исходного текста.

В формулах (1) и (2) вычисления проводятся в классе вычетов по модулю $n$, где $n$ – длина выбранного алфавита. В качестве исходного алфавита выбран латинский или русский алфавит, дополненный знаками « », «.», «,», «/», «!», «?»,«;». После обработки каждого блока $M_k$ исходные векторы $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ изменяются следующим образом:

$$A_1=A_1+a_1,\ A_2=A_2+a_2,\ A_3=A_3+a_3,\ A_4=A_4+a_4,\ A_5=A_5+a_5.$$

Здесь операции выполняются также по mod $n$. После завершения обработки всего электронного документа получаем хэш-код окончательным суммированием начальных векторов $A_i$ $i=\overline{1,5}$ по формуле:

$$h=(A_1+A_2+A_3+A_4+A_5)\bmod n.$$

Для построения электронной подписи к полученному хэш-коду применяется композиция шифров аффинного и вероятностного шифра построенного на основе RSA-функций.

**Аналитический аффинный шифр**

Рассмотрим вначале первый шифр этой композиции – аффинный. Этот шифр относится к классу шифров, основанных на аналитических преобразованиях шифруемых данных. Они основаны на подстановке не отдельных символов, а $k$–грамм (шифр Хилла) или биграмм (шифр Плэйфера). Общая идея таких шифров состоит в том, что рабочим алфавитом открытых и закрытых текстов является кольцо остатков $Z_n$, где $n$ – количество символов алфавита естественного языка (речь идет о европейских языках). Все вычисления производятся во множестве $Z_n$ и в остатках от деления по модулю $n$. Если $k$ – порядок шифра, то при генерировании ключей выбирают обратимую матрицу, размером $k \times k$ с элементами из кольца $Z_n$ и столбец над тем же кольцом размера $k \times 1$. Пусть это будет матрица $A$ и столбец $S$.

Открытый текст разбивается на k-граммы и описывается через символы кольца $Z_n$. Обозначив k-грамму открытого текста через $X$, а соответствующий шифротекст через $Y$, процедуру шифрования можно описать так:

$$E(X) = (A \cdot X + S) \bmod n = Y \qquad (3)$$

Результатом шифрования является набор образов вида Y.

Алгоритм дешифрования выглядит следующим образом:

$$D(Y) = (A_1 \cdot Y + S_1) \bmod n \equiv X, \qquad (4)$$

где $A_1 = A^{-1}$, а $S_1 = -A_1 \cdot S$.

Открытый текст представляет из себя набор прообразов вида $X$.

Для того, чтобы алгоритм шифрования $E(X)$, задаваемый формулой (3) имел обратный алгоритм дешифрования $D(Y)$ по формуле (4) необходимо, чтобы матрица $A$ имела обратную матрицу $A_1=A^{-1}$. Это условие выполняется тогда и только тогда, когда $НОД(det(A),n)=1$.

В случае, когда $S = 0$, то мы имеем линейный шифр k-ого порядка.

Ключевое пространство для линейного шифра k-го порядка совпадает с множеством обратимых матриц k-го порядка над $Z_n$. Это количество оценивается значением обобщенной функции Эйлера:

$$\varphi_k(n) = n^{k^2} \prod_{i=1}^{k} \prod_{j=1}^{s} \left(1 - \frac{1}{p_j^i}\right), \qquad (5)$$

где $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_s^{\alpha_s}$.

При атаке на аффинный шифр с шифротекстом для $k = 1$ и $k = 2$ применим частотный метод

(простой или с учетом частот биграмм). В случае атаки с известным открытым текстом можно существенно уменьшить количество вариантов перебора. Когда $k > 2$, аффинный шифр не поддается частотному анализу. Оценивая величину (5) для $k \geq 5$ отметим, что прямая атака оказывается бесперспективной.

**Вероятностные алгоритмы шифрования**

В качестве второго алгоритма в композиции шифров для построения ЭЦП используется вероятностный шифр на основе криптосистемы RSA. Система шифрования RSA основана на односторонней функции. Пусть $n$ и $e$ натуральные числа. Функция $f$ реализующая схему RSA, устроена следующим образом

$$f : x \to x^e \bmod n. \tag{6}$$

Для расшифровки сообщения $y = f(x)$ достаточно решить сравнение

$$x^e = y \bmod n. \tag{7}$$

При некоторых условиях на n и $e$ это сравнение имеет единственное решение X.

Число n – основной модуль криптосистемы RSA выбирается как произведение двух больших простых чисел p и q. В этом случае функция Эйлера [1,4]

$$\varphi(n) = (p - 1)(q - 1).$$

Если показатель степени $e$ в формуле (7) взаимно прост с функцией $\varphi(n)$, то формула (7) имеет единственное решение. Для того чтобы найти его, определим целое число $d$, удовлетворяющее условиям

$$de \equiv 1 \bmod \varphi(n),\ 1 \leq d < \varphi(n). \tag{8}$$

Такое число существует, поскольку $НОД(e, \varphi(n)) = 1$, и притом единственное.

Из формулы (8) вытекает, что число $d$ является мультипликативным обратным к $e$ по модулю $\varphi(n)$ и находится из обобщенного алгоритма Евклида. Классическая теорема Эйлера, утверждает, что для каждого числа $x$, взаимно простого с $n$, выполняется сравнение $x^{\varphi(n)} \equiv 1 \bmod n$ и, следовательно.

$$y^d \equiv x^{de} \equiv x \bmod n. \tag{9}$$

Таким образом, решение формулы (7) может быть найдено в виде

$$x \equiv y^d \bmod n. \tag{10}$$

Используя бинарный метод возведения в степень функция (6), принятая в системе RSA, вычисляется достаточно быстро. Обратная к $f(x)$ функция $x \equiv y^d \pmod n$ вычисляется по тем же правилам, что и $f(x)$, лишь с заменой показателя степени $e$ на $d$. Для вычисления функции (6) достаточно знать лишь числа $e$ и $n$. Именно они составляют открытый ключ для шифрования. А вот для вычисления обратной функции требуется знать число $d$, оно и является «секретом». Казалось бы, ничего не стоит, зная число $n$. разложить его на простые сомножители, вычислить затем с помощью известных правил значение $\varphi(n)$ и, наконец, с помощью (8) определить нужное число $d$. Все шаги этого вычисления могут быть реализованы достаточно быстро, за исключением первого. Именно разложение числа $n$ на простые множители и составляет наиболее трудоемкую часть вычислений. В теории чисел несмотря на многолетнюю её историю и на очень интенсивные поиски в течение последних 20 лет, эффективный алгоритм разложения натуральных чисел на множители так и не найден. Простой перебор всех простых сомножителей занимает огромное количество времени. Известны и более эффективные способы разложения целых чисел на множители, чем простой перебор простых делителей, но и они работают очень медленно.

Итак, лицо, заинтересованное в организации шифрованной переписки с помощью схемы RSA, выбирает два достаточно больших простых числа $p$ и $q$. Перемножая их, оно находит число $n = pq$. Затем вычисляется функция Эйлера $\varphi(n)$ и с помощью соотношения (8) – число $d$. Числа $e$ и $n$ публикуются, число $d$ остается секретным, множители $p$ и $q$ также являются секретными. Теперь любой может отправлять зашифрованные с помощью (6) сообщения организатору этой системы, а

организатор легко сможет расшифровывать их с помощью формулы (10).

Вероятностное шифрование. Понятие вероятностного шифрования было изобретено Шафи Голдвассером (ShafiGoldwasser) и Сильвией Микали. Вероятностное шифрование позволяет избежать даже частичной утечки информации об оригинальном сообщении. При использовании криптографии с открытыми ключами криптоаналитик иногда может узнать кое-что о некоторых битах. Таким способом можно извлечь не много информации, но потенциальная возможность криптоаналитика расшифровать случайные сообщения открытым ключом может создать определенные проблемы. Каждый раз, дешифруя сообщение, криптоаналитик может извлечь немного информации. Никто не знает, насколько значительна эта информация. При вероятностном шифровании остается скрытой и такая информация.

Цель вероятностного метода состоит в том, чтобы ни вычисления, проводимые над шифротекстом, ни проверка любых других открытых текстов не смогли дать криптоаналитику никакой информации о соответствующем открытом тексте. При вероятностном шифровании алгоритм шифрования является вероятностным, а не детерминированным. Результат вероятностного шифрования всегда будет больше открытого текста.

Вероятностное шифрование на основе RSA-функций. Процедура генерирования ключей производится так же, как и в самой RSA.

Открытый ключ: $e, n$, где $n = pq$, $НОД(e, \varphi(n)) = 1$.

Секретный ключ: $d$ такое число, что $ed \equiv 1 \bmod \varphi(n)$.

Шифрование. Открытое сообщение $M$ записывается в двоичной форме

$$M = m_1 m_2 ... m_l, \quad \text{где } m_i \in \{0,1\},$$

и преобразуется в шифротекст $C = c_1 c_2 ... c_l$, где $c_i \in Z_n$, для каждого $i = 1,2,\ldots,l$ с использованием следующих вероятностных процедур:

1) если $m_i = 0$, то в $Z_n$ выбирается случайное четное число $x_i$, а если $m_i = 1$, то выбирается случайное нечетное число $x_i$;

2) вычисляется $c_i \equiv x_i^e \ (mod\ n)$.

Дешифрование. По шифротексту $C = c_1 c_2 ... c_l$ открытый текст $M = m_1 m_2 ... m_l$ получают по правилу:

$$m_i = \begin{cases} 0, & если\ c_i^d \bmod n \quad четное, \\ 1, & если\ c_i^d \bmod n \quad нечетное, \end{cases} \quad i = 1,2,\ldots,l$$

Доказано, что надежность рассматриваемой системы равносильна допущению, что RSA может быть вскрыта полиномиальным алгоритмом

Для построения хэш-кода и электронно-цифровой подписи под исходным документом создан программный комплекс в среде Visual C++.

Проведенные в работе численные исследования показывают, что изменение хотя бы одного символа в исходном тексте, приводит к полнейшему изменению результата построенного хэш-кода и ЭЦП под пересылаемым документом.

**Литература**


[1] Коблиц Н. Курс теории чисел и криптографии/Н. Коблиц – М. ТВП, 2001.–259 с.

[2] Мао В. Современная криптография: Теория и практика/В. Мао. – М.: Вильямс, 2005. – 763 с.

[3] Петров А.А. Компьютерная безопасность. Криптографические методы защиты / Петров А.А. – М.: ДМК, 2000. – 445 с.

[4] Столингс В. Криптография и защита сетей / В. Столингс. – М.: Вильямс, 2001. – 669 с.

[5] Тилборг Ван Х.К.А. Основы криптологии. Профессиональное руководство и интерактивный учебник/В. Тилборг. – М.: Мир, 2006. – 471 с.

[6] Шнайер Б. Прикладная криптография: протоколы, алгоритмы, исходные тексты на языке Си/ б. Шнайер. – М.: Триумф, 2002. – 816 с