УДК 004.056:004.424.47

УЗАГАЛЬНЕНИЙ МЕТОД ХЕШУВАННЯ БАЙТОВОЇ ФОРМИ ПРЕДСТАВЛЕННЯ ІНФОРМАЦІЇ

В. А. Лужецький, д.т.н., проф.; Д. В. Кисюк, асистент
Вінницький національний технічний університет
kneimad@gmail.com

Одними з найпоширеніших методів захисту інформаційних ресурсів у комп'ютерних системах є методи, що базуються на криптографічних перетвореннях, зокрема на використанні хеш-функцій. Ці функції забезпечують контроль цілісності даних, безпечне збереження паролів та іншої конфіденційної інформації, використовуються у електронному цифровому підписі та різноманітних криптографічних протоколах. Тому, побудові хеш-функцій і алгоритмів, що їх реалізують приділяється значна увага фахівців із захисту інформації. Враховуючи різноманіття прикладних задач, що розв'язуються з їх допомогою, висувається ціла низка вимог до хеш-функцій (висока швидкість хешування, стійкість до атак, максимальне використання особливостей апаратно-програмних засобів).

Оскільки значного поширення набувають мобільні гаджети, у яких найчастіше використовуються 8-розрядні мікропроцесори, тому потрібні хеш-функції, які орієнтуються на байтове представлення даних. У доповіді пропонується новий метод хешування, який передбачає саме таку форму представлення даних.

Відомі на теперішній час методи хешування базуються на ітераційній процедурі обчислення хеш-значення, яка передбачає на кожному кроці ітерації

використання проміжного попереднього хеш-значення і наступного блоку даних, що підлягають хешуванню.

Пропонується принципово новий підхід до хешування даних, який не передбачає ітераційний процес обчислення хеш-значень. Суть методу полягає у тому, що спочатку вхідне повідомлення розбивається на послідовність байтів, далі підраховується кількість байтів однакового змісту, а потім обчислюється хеш-значення з урахуванням цих кількостей та номерів позицій, у яких розташовані ці байти.

Кожен такий байт можна розглядати як число $\boldsymbol{N}(m_l)$ або бітове поле (код) - $\boldsymbol{K}(m_l)$. Кожне повідомлення характеризуватиметься кіль-кістю елементів n_i ($i = 0 \div 255$), що мають однаковий код $\boldsymbol{K}(m_l)$ та деякою функцією $\boldsymbol{S}_j(i)$, аргументами якої є номери позицій байтів з числовим еквівалентом $\boldsymbol{N}(m_l) = i$ ($i = 0 \div 255$). При цьому:

$$S_j^{(i)} = \begin{cases} S_{j-1}^{(i)} * l, \text{якщо } \boldsymbol{N}(m_l) = i \\ S_{j-1}^{(i)}, \text{інакше} \end{cases},$$

де $S_0^{(i)} = 0$; $i = 0 \div 255$; $j = 1 \div n_i$; $*$ – деяка визначена операція. На основі отриманих значень ni та $S_j(i)$ утворюється два масиви $\boldsymbol{N}$ та $\boldsymbol{S}$:

$$\boldsymbol{N} = (n_0, n_1, \dots, n_{255}) \quad \boldsymbol{S} = (s_0, s_1, \dots, s_{255})$$

Пропонуються такі варіанти формування масиву $\boldsymbol{N}$:

a) Звичайна кількість елементів n_i з однаковим кодом:

$$\boldsymbol{N_0} = \left\{N_{n_i}^{(0)}\right\}$$

b) Деяка функція кількості елементів n_i з однаковим кодом:

$$\boldsymbol{N_1} = \left\{N_{n_i}^{(1)}\right\}, \text{де } N_{n_i}^{(1)} = f\left(N_{n_i}^{(0)}\right)$$

c) Добуток значення елементів n_i на кількість таких елементів з однаковим кодом:

$$\boldsymbol{N_2} = \left\{N_{n_i}^{(2)}\right\}, \text{де } N_{n_i}^{(2)} = n_i \cdot N_{n_i}^{(0)}$$

d) Деяка функція значення елементів n_i та кількості таких елементів з однаковим кодом:

$$N_3 = \left\{N_{n_i}^{(3)}\right\}, \text{де } N_{n_i}^{(3)} = g\left(N_{n_i}^{(2)}\right)$$

Залежно від кількості символів у повідомлені можуть виникати різні випадки:

a) якщо кількість символів у повідомлені буде меншою 256, то у масиві N обов'язково матимуть місце елементи зі значенням «0»;

b) якщо кількість символів у повідомлені буде 256, то можлива ситуація, коли всі елементи масиву N будуть містити значення «1», якщо повідомлення міститиме повний набір символів по одному значенню відповідно;

c) якщо кількість символів у повідомлені буде більшою за 256, то масив N може містити елементи зі значенням «0», «1» та більше.

Також на вигляд масивів буде впливати особливість формування тексту повідомлень. Наприклад, мова повідомлення, використання цифр та різноманітних вкладених об'єктів або ж змішані повідомлення, що містять різні види описаних вище елементів. Для підрахунку хеш-коду, необхідно привести довжину масивів ***N*** та ***S*** до потрібного значення. Пропонуються варіанти ущільнення масивів:

a) Ущільнення масивів N та S до розміру хеш-значення та застосування деякої функції до цих двох ущільнених масивів: h = f (C1(N), C2(S));

b) Застосування деякої операції до двох масивів з подальшим ущільненням отриманого результату до довжини хеш-значення.:h = C3(g(N, S))

На основі описаного методу розроблено програмне забезпечення, яке дозволяє отримувати хеш-значення довжиною 256 біт.