БАГАТОКАНАЛЬНЕ КЕРОВАНЕ ХЕШУВАННЯ ДАНИХ

Лужецький В.А., Баришев Ю.В.
Вінницький національний технічний університет
вул. Хмельницьке шосе 95, Вінниця, 21021, тел. +38(0432)598380
E-mail: yuriy.baryshev@gmail.com

Найпоширенішим методом автентифікації даних та користувачів в комп'ютерних системах є криптографічні методи хешування. Однак поява класу загальних атак на хеш-функції віддалила оцінки стійкості відомих хеш-функцій від оцінок стійкості ідеальної хеш-функції. Загальні атаки використовують властивість незмінності параметрів перетворень, які реалізуються у функції ущільнення, від ітерації до ітерації. Зокрема найбільш вразливими до таких атак стали багатоканальні (розпаралелені) хеш-функції та хеш-функції конструкції Меркля-Дамгаарда, які є найпоширенішими та найвідомішими в наш час (зокрема це хеш-функції MD5, SHA-2). Відповідно до цієї конструкції повідомлення $\mathbf{M}$ розглядається як конкатенація блоків даних рівної довжини $\mathbf{M} = m_1 \| m_2 \| ... \| m_l$ (де "||" – операція конкатенації), які ітеративно ущільнюються за таким правилом:

$$h_i = f(h_{i-1}, m_i), \tag{1}$$

де h_i – i-те проміжне хеш-значення;

$f(\cdot)$ – функція ущільнення.

Останнє проміжне хеш-значення h_l, отримане відповідно правилу (1) використовуватиметься як вихідне.

З метою збільшення стійкості хешування до відомих атак пропонуються методи керованого хешування, які передбачають використання конструкцій, параметри перетворення яких змінюються від ітерації до ітерації залежно від значення певного параметра – вектора керування v_i:

$$h_i = f_{v_i}(h_{i-1}, m_i). \tag{2}$$

Керованість методів конструкції (2) дозволяє забезпечити зав'язування каналів багатоканального хешування як за допомогою функції ущільнення, так за допомогою функції формування вектора керування. Останнє дозволяє отримати виграш у швидкості хешування, оскільки тривалість формування вектора керування менша за тривалість функції ущільнення. В загальному випадку конструкція багатоканального хешування представляється так:

$$\begin{cases} h_i^{(1)} = f_{v_i^{(1)}}\left(h_{i-1}^{(1)}, h_{i-1}^{(2)}, \ldots, h_{i-1}^{(k)}, m_i\right) \\ h_i^{(2)} = f_{v_i^{(2)}}\left(h_{i-1}^{(2)}, h_{i-1}^{(3)}, \ldots, h_{i-1}^{(k+1)}, m_i\right) \\ \ldots \\ h_i^{(q)} = f_{v_i^{(q)}}\left(h_{i-1}^{(q)}, h_{i-1}^{(1)}, \ldots, h_{i-1}^{(k-1)}, m_i\right) \\ v_i^{(1)} = g\left(h_{i-1}^{(q)}, h_{i-1}^{(q-1)}, \ldots, h_{i-1}^{(q-\phi+1)}\right) \\ v_i^{(2)} = g\left(h_{i-1}^{(1)}, h_{i-1}^{(q)}, h_{i-1}^{(q-1)}, \ldots, h_{i-1}^{(q-\phi+2)}\right) \\ \ldots \\ v_i^{(q)} = g\left(h_{i-1}^{(q-1)}, h_{i-1}^{(q-2)}, \ldots, h_{i-1}^{(q-\phi)}\right) \end{cases} . \quad (3)$$

де $h_i^{(j)}$ – проміжне хеш-значення, отримане у j-му каналі ($j = \overline{1,q}$) на i-й ітерації.

Вихідне хеш-значення відповідно до (3) визначатиметься як конкатенація вихідних хеш-значень кожного каналу $h_l = h_l^{(1)} \| h_l^{(2)} \| \ldots \| h_l^{(q)}$.

Для реалізації функції ущільнення пропонується використовувати модифіковані шляхом додавання операцій керованого циклічного зсуву та використанням логічного додавання функції, які застосовуються у стандарті хешування SHA-2. Зокрема, пропонується використовувати таку логічну функцію для параметра $k = 2$ конструкції (3):

$$\begin{aligned} h_i^{(j)} = &\left(m_i >>> u_i^{(j)(m1)} \wedge h_{i-1}^{(j)} >>> u_i^{(j)(h1)}\right) \oplus \\ \oplus &\left(m_i >>> u_i^{(j)(m1)} \wedge h_i^{(j+1)} >>> u_i^{(j)(h2)}\right) \oplus \\ \oplus &\left(h_{i-1}^{(j)} >>> u_i^{(j)(h1)} \wedge h_i^{(j+1)} >>> u_i^{(j)(h2)}\right) \end{aligned} , \quad (4)$$

де $u_i^{(j)(x\kappa)}$ – кількість бітів, на яку циклічно зсувається змінна x у κ-й позиції у функції ущільнення на i-й ітерації у j-му каналі.

Крім функції ущільнення (4) пропонується використовувати функцію ущільнення такого виду:

$$\begin{aligned} h_i^{(j)} = &\left(m_i >>> u_i^{(j)(m1)} \wedge h_{i-1}^{(j+1)} >>> u_i^{(j)(h1)}\right) \oplus \\ \oplus &\left(\sim m_i >>> u_i^{(j)(m1)} \wedge h_{i-1}^{(j+2)} >>> u_i^{(j)(h2)}\right) \end{aligned} . \quad (5)$$

Функції ущільнення (4) та (5) легко модифікуються для обробки довільного парного значення параметра k.

Для реалізації даних методів багатоканального керованого хешування було розроблено програмне забезпечення мовою C++. Використовуючи стандарту бібліотеку "time.h" було визначено тривалість обробки даних різної довжини для різних параметрів методів хешування, які наведені у таблиці 1.

Таблиця 1. Оцінки тривалості хешування

Функція ущільненя	Тривалість хешування, cycles			
	10 Кб	100 Кб	1 Мб	10 Мб
$k=2,(4)$	78	781	8219	82328
$k=2,(5)$	141	1516	15297	151813
$k=4,(4)$	94	812	8547	84718
$k=4,(5)$	171	1719	17406	171719

З таблиці 1 видно, що зі збільшенням обсягів даних тривалість хешування зростає лінійно, в той час як більшість відомих методів хешування досягає найкращих показників швидкості при обробці великих масивів даних. Крім того, збільшення параметра k вдвічі спричинює збільшення тривалості хешування приблизно на 20%, що підтверджує тезу про можливість підвищення швидкості багатоканального керованого хешування за рахунок зав'язування каналів за допомогою функції формування вектора керування.