УДК 003.26

МЕТОДИ ПОБУДОВИ ШВИДКИХ АЛГОРИТМІВ ХЕШУВАННЯ

Ю. В. Баришев, магістр з інформаційної безпеки, аспірант
Вінницький національний технічний університет
numen@svitonline.com

Відомо, що для хеш-функцій з теоретично доведеною стійкістю, їх стійкість або опір колізіям прямо пропорційний довжині значення. В той же час, чим більша розрядність хеш-значення, тим більше обчислень функції необхідно виконати для його визначення, а тому ці обчислення вимагають достатньо багато часу. Збільшення розрядності мікропроцесорів дозволяє виконувати такі обчислення швидше. Однак, з іншого боку, воно і дозволяє зловмиснику швидше "зламати" хеш-функцію, відповідно доводиться знову збільшувати розрядність хеш-значення. Таким чином, покращення апаратури не обумовлює, в загальному випадку, збільшення швидкості обчислення хеш-значення. Виходячи з цих міркувань, можна зробити висновок, що актуальною є задача побудови хеш-алгоритмів, які б дозволяли підвищити швидкість обчислень без втрати стійкості хеш-функції.

Розглядається хеш-функція теоретично доведеної стійкості, яка базується на операції піднесення до степеня за модулем простого числа. Для обчислення занчень таких хеш-функцій вхідні дані розбиваються на блоки $X = \{x_i\}$ ($i = 1 \div n$) і в кожному з раундів обчислюється значення функції:

$$y_i = f(x_i, y_{i-1}),\ y_0 = const\ . \qquad (1)$$

Відомо, що операція піднесення до степеня чисел великої розрядності вимагає достатньо багато машинного часу. Якщо ж суттєво зменшити розрядність числа, то піднесення до степеня буде виконуватись швидко, але це дозволить швидше "зламати" хе-функцію. Для того, щоб запобігти цьому, пропонується заміщувати одне рівняння з великою розрядністю системою рівнянь, а хеш-значення на кожному раунді отримувати шляхом конкатенації хеш-значень кожного *j*-го рівняння $y_i = \{y_{ij}\}$ ($j = 1 \div k$). Тоді вхідні дані X будуть розбиватись не n, а на $n \cdot k$ частин, а процес хешування можна розпаралелити, обчислюючи на кожному процесорі рівняння:

$$y_{ij} = f_j\left(x_i, y_{(i-1)j}\right). \qquad (2)$$

Проте вихідне хеш-значення, отримане шляхом конкатенації на останньому раунді всіх хеш-значень рівнянь $y_{(n\cdot k)1} \| y_{(n\cdot k)2} \| \ldots \| y_{(n\cdot k)k}$ буде не стійким до атаки Жоукса (Joux) та похідних від неї атак на хеш-функції. Тому пропонується з метою запобігання цих атак виконувати зав'язку на кожному *i*-му раунді кожного *j*-го хеш-значення, що отримується за допомогою *j*-го рівняння, та всіх хеш-значеннь, отриманих на попередньому $(i-1)$-му кроці. Таким чином, рівняння (2) буде заміщене таким рівнянням:

$$y_{ij} = f_j\left(x_i, y_{(i-1)1}, y_{(i-1)2}, \ldots, y_{(i-1)k}\right). \qquad (3)$$

А хеш-значення $y_{(n\cdot k)1} \| y_{(n\cdot k)2} \| \ldots \| y_{(n\cdot k)k}$ буде стійким до атак, похідних від атаки Жоукса, одночасно дозволяючи виконати розпаралелення обчислень.