Метод та спеціалізований процесор для байт-орієнтованого гешування даних

В.А. Лужецький
кафедра захисту інформації
Вінницький національний
технічний університет
Вінниця, Україна
lva_zi@mail.ru

Д.В. Кисюк
кафедра обчислювальної техніки
Вінницький національний
технічний університет
Вінниця, Україна
kneimad@gmail.com

А.О.Комаров
кафедра захисту інформації
Вінницький національний
технічний університет
Вінниця, Україна
komand9@gmail.com

Method and specialized processor of byte-oriented data hashing

V. Luzhetskyi
Department of
Information Protection
Vinnytsia National
Technical University
Vinnytsia, Ukraine
lva_zi@mail.ru

D. Kysiuk
Department of
Computer Science
Vinnytsia National
Technical University
Vinnytsia, Ukraine
kneimad@gmail.com

A. Komarov
Department of
Information Protection
Vinnytsia National
Technical University
Vinnytsia, Ukraine
komand9@gmail.com

***Анотація*—Розглядається метод та засіб неітераційного гешування, яке базується на використанні характеристичних ознак даних. Наведено математичні моделі запропонованого методу, а також функціональну та структурну модель спеціалізованого процесора для гешування даних.**

***Abstract* — The method and tool of non-iteration hashing, based on the use of the characteristic features of the data was considered. The mathematical and schematic models and functional and structural model of a specialized processor of this method were presented.**

Ключові слова: гешування, геш-функція, неітеративне гешування, байтова обробка даних, спеціалізований процесор.

Keywords: hashing,hash - function, noniterrate hashing, byte-oriented hashing, specialized processor.

I. Вступ

Зростаючі вимоги, що висуваються до швидкості гешування даних, а також необхідність реалізації пристроями з невеликими обчислювальними можливостями, приводять до необхідності розробки нових методів гешування, а також спеціалізованих пристроїв [1, 2].

Відомі геш-функції передбачають реалізацію у вигляді ітераційних процедур. Проте, у зв'язку з тим, що питання про лавиноподібний ефект з початковим заповненням при великій кількості ітерацій недостатньо досліджений, і, відповідно, використання цих функцій є недостатньо обґрунтованим. Також, до недоліків слід віднести такі їх особливості [3, 6]:

1) різний вплив блоків даних на остаточний результат гешування через нелінійність функцій перетворення (значення першого блоку бере участь у формуванні усіх проміжних геш-значень, а значення останнього блоку враховується лише на останній ітерації);

2) існує потенційна можливість за результатами кожної ітерації знайти колізію, тому зазвичай в даних методах

намагаються ускладнити функцію перетворення на кожній ітерації [4, 5].

Для усунення вказаних недоліків автори пропонують неітераційний метод гешування.

II. Метод Гешування На Основі Характеристчних Ознак Байтової Стуктури Даних

Кожне повідомлення можна охарактеризувати алфавітом, що використовується для побудови цього повідомлення. Як одну із характеристик повідомлення пропонується використовувати кількість входжень до повідомлення кожного окремого елемента алфавіту. Друга характеристика повідомлення – це номери позицій, в яких розташовані конкретні елементи алфавіту. З урахуванням цього пропонується метод, в якому спочатку визначаються такі дві характеристики, а потім – згортка отриманих результатів до коду, що має довжину геш-значення.

Вхідне повідомлення **M** розбивається на послідовність байтів:

$$\mathbf{M} = \{ m_1, m_2, \ldots, m_L \}.$$

Кожен байт розглядається як число n, що відповідає ASCII – коду символу представленого байтом $m_l (l = 1 \div L)$, тобто $n = f(m_l)$.

Повідомлення характеризується кількістю елементів k_n, що мають числовий еквівалент n ($n = 0 \div 255$) та номерами позицій у яких розташовані ці елементи.

На основі цих характеристик утворюється два масиви **K** та **S** :

$$\mathbf{K} = (k_0, k_1, \ldots, k_{255}),$$
$$\mathbf{S} = (s_0, s_1, \ldots, s_{255}),$$

де: k_n – кількість байтів, що мають числовий еквівалент n;

s_n – сума номерів позицій, на яких розташований байт з числовим еквівалентом n;

$$s_n = \left(\sum_{j=1}^{k_n} l_j^n\right) mod\ 2^8.$$

Узагальнена схема процесу гешування наведена на рис. 1.

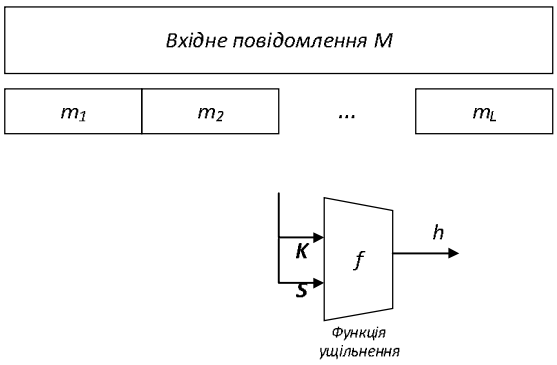

Рис. 1. Узагальнена схема процесу гешування

Для підрахунку геш-коду виконуються такі операції над елементами масивів **K** та **S**.

1. Додаються побайтно елементи масивів **K** та ***S***, які мають розмір 8 байт, причому елементи масиву **K** беруться з початку, а елементи масиву **S** беруться з кінця.
2. Над першою половиною байтів результату виконується циклічний зсув праворуч на 4 біти.
3. До перших 4 байтів результату додаються за модулем два останні 4 байти, які переставлені в зворотному порядку.
4. Перші 2 байти результату циклічно зсуваються на 4 біти.
5. Додаються за модулем два 1 і 3, 2 і 4 байти.
6. Перший байт циклічно зсувається на 4 біти.
7. Додаються за модулем 2 перший і другий байти.

Результатом виконання цих операцій є послідовність **H*** з 256 байт.

Далі масив $\mathbf{H}^*$ представляється як 8 32-байтних чисел $\mathbf{H}^*=(H_1^*, H_2^*, H_3^*, H_4^*, H_5^*, H_6^*, H_7^*, H_8^*)$. Виконується нагромаджувальне множення елементів масиву **H***:

$$D_1 = H_1^* \boxtimes H_2^*,\ D_2 = D_1 \boxtimes H_3^*,\ D_3 = D_2 \boxtimes H_4^*,\ \ldots\ D_7 = D_6 \boxtimes H_8^*.$$

Тут символ $\boxtimes$ означає звичайне множення двох 32-байтних співмножників, результатом якого є 64-байтний код, з подальшим додаванням за модулем 2^{256} старших і молодших 32 байтів.

Результат гешування – $h = D_7$.

III. Спеціалізований процесор

IV. Спеціалізований процесор для гешування даних названо МН-процесор (message hashing). МН-процесор певним чином підключений до центрального процесора комп'ютера і вони обмінюються потоками даних, виконуючи функції, перелік яких наведено в табл. 1.

ТАБЛИЦЯ І. РОЗПОДІЛ ВИКОНУВАНИХ ФУНКЦІЙ МІЖ ПРОЦЕСОРАМИ

Виконувані функції	
Центральний процесор	***Спеціалізований процесор***
1. Зчитування файлу або даних та формування послідовності **M**. 2. Отримання послідовності **H**.	1. Обчислення геш-значення за запропонованим алгоритмом.

Оскільки файли, для яких буде обчислюватись геш-значення, зберігаються в пам'яті комп'ютера, то передбачається, що центральний процесор буде виконувати зчитування файлу та формування послідовності M. На МН-процесор покладаються обчислення геш-значення.

Схему оброблення потоку даних у МН-процесорі наведено на рис. 2. Появлення даних на вхідній шині МН-процесора приводить до ініціювання процесу «ВХІДНИЙ ДИСПЕТЧЕР» (DI), який спрямовує потік даних (**M**) до процесу ініціалізації (PI). Далі потік даних спрямовується до процесу «ЗГОРТКА» (PC). Результат процесу згортки ініціює процес «МНОЖЕННЯ» (PM). Завершення процесу PM приводить до дії процес «ВИХІДНИЙ ДИСПЕТЧЕР» (DO), який збирає отримані результати і організовує з них потік даних(**H**) до центрального процесора.

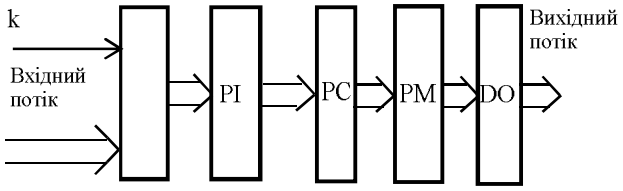

Рис. 2. Функціональна модель МН-процесора

Функціональній моделі МН-процесора, відповідає структурна модель, яку наведено на (рис.3).

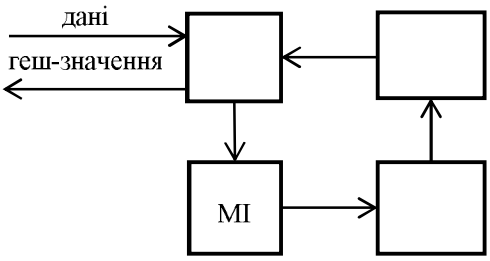

Рис. 3. Структура МН-процесора

До складу МН-процесора входять:

- модуль системних операцій (МСО), що здійснює зв'язок з центральним процесором і реалізує процеси DI і DO;
- модуль ініціалізації (MI), що реалізує процес PI;
- модуль згортки (МЗ), що реалізує процес PC;
- модуль множення (ММ), що реалізує процес PM.

Модуль ініціалізації містить дві пам'яті обсягом 256x8 байтів кожна, які використовуються для зберігання масивів **K** та **S.** Лічильники Ліч1 і Ліч2, що забезпечують послідовне зчитування байтів даних від молодшого байту до старшого та навпаки, відповідно. Лічильник Ліч3 використовується для формування номеру позиції, в якій знаходиться конкретний байт даних. Суматор СМ1 використовується для збільшення кількості певного байту в повідомленні на 1. Суматор СМ2 використовується для додавання номеру позиції.

Модуль згортки містить пам'ять обсягом 256 байтів, що використовується для зберігання блоку даних $\mathbf{H}^*$, 8 однобайтних суматорів, 10 однобайтних регістрів і лічильник.

Модуль множення складається з пристрою множення 256 бітних кодів та суматора за модулем 2^{256}.

ВИСНОВКИ

Запропонований метод хешування не передбачає ітеративної процедури обчислення геш-значення і,як наслідок, до нього не можуть бути застосовані відомі атаки на ітеративні геш-функції.

Використання характеристичних ознак вхідних даних та їх побайтна обробка значно прискорюють процес обчислення геш-значення, а також зменшують апаратні витрати на реалізацію такого методу.

ЛІТЕРАТУРА REFERENCES

[1] Лужецький В. А. Новий підхід до побудови криптографічних хеш-функцій / В. А. Лужецький, Д. В. Кисюк // «Інформаційні технології та комп'ютерна інженерія»; матеріали статей п'ятої міжнародної науково-практичної конференції, м. Івано-Франківськ, 27-29 травня 2015 року. – Івано-Франківськ: Супрун В. П., 2015 р. – с. 206-208..

[2] Лужецький В. А. Узагальнений метод хешування байтової форми представлення інформації / В. А. Лужецький, Д. В. Кисюк // IV міжнародна науково-практична конференція «Інформаційні технології та комп'ютерна інженерія». – Вінниця: ВНТУ, 2014., - 275с.

[3] Aumasson J. P. SHA-3 proposal BLAKE / Henzen L., Meier W., Phan R. - 2010.

[4] Bos J. W. Performance analysis of the SHA-3 candidates on exotic multi-core architectures / J. W. Bos, D. Stefan. - 2010.

[5] Neves S. ChaCha implementation. - 2009. – Режим доступу до статті: http://eden.dei.uc.pt/sneves/chacha/chacha.html.

[6] Knezevic M. Fair and consistent hardware evaluation of fourteen round two SHA-3 candidates / M. Knezevic, K. Kobayashi, J. Ikegami, S. Matsuo, A. Satoh, U. Kocabas, J. Fan April 2011.