УДК 004.056:004.424.47

В.А. Лужецький, Ю.В. Баришев

Вінницький національний технічний університет, Вінниця

АПАРАТНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ БАГАТОКАНАЛЬНОГО КЕРОВАНОГО ХЕШУВАННЯ

У даній статті розглянуто апаратну реалізацію багатоканального керованого хешування. Запропоновано підходи до розробки блоків ущільнення та формування вектора керування. Наведено схеми пристроїв, отриманих відповідно до запропонованих підходів, а також оцінки швидкодії та апаратної складності цих пристроїв.

Ключові слова: *хешування, багатоканальність, керованість, оцінки.*

Вступ

Значна частка механізмів автентифікації користувачів та даних використовує методи хешування. Оскільки криптографічні перетворення потребують значних витрат часу для своєї реалізації за допомогою універсальних процесорів, то у низці випадків доречніше використовувати спеціалізовані пристрої, для яких ці криптографічні перетворення будуть природними, а тому виконуватимуться за меншу кількість тактів.

Очевидно, що ще більша швидкість хешування досягається шляхом розпаралелення обчислень [1]. Водночас у галузі комп'ютерної криптографії з появою загальних атак на хеш-функції виникла задача, пов'язана з розпаралеленням обчислень без значної втрати стійкості [2]. Одним з розв'язків цієї задачі є впровадження концепції керованого хешування [2], зокрема реалізації у вигляді спеціалізованого процесора.

Метою даного дослідження є збільшення швидкості керованого хешування за рахунок розпаралелення процесу хешування та його апаратної реалізації. Для досягнення мети необхідно розв'язати такі задачі:

– аналіз відомих розробок, які реалізують керовані криптографічні примітиви;

– розробка блоку ущільнення спеціалізованого процесора;

– розробка блоку формування вектора керування спеціалізованого процесора.

Аналіз відомих підходів до апаратної реалізації керованих криптографічних примітивів

Криптографічні процесори, що реалізують паралельні обчислення, розглядалися у роботі [1], де наводиться структура пристроїв, як для симетричної, так і для асиметричної криптографії. Однак дані пристрої застосовні при шифруванні, а їх використання для реалізації хеш-функцій не забезпечує стійкості до загальних атак. Крім того, такі хеш-функції, як і всі хеш-функції, побудовані на основі шифрів, забезпечують низьку швидкість перетворень, порівняно з хеш-функціями розробленими "з нуля" [3].

Для розробки керованого хешування "з нуля" необхідно попередньо визначити криптографічні примітиви, які будуть використані під час хешування. Дослідження [2] продемонстрували, що з точки зору забезпечення максимальної швидкості реалізації криптографічних примітивів, як апаратно, так і програмно доцільно використати такі операції: циклічний зсув праворуч (>>> u), додавання (+), додавання за модулем 2 ($\oplus$), логічне множення ($\wedge$), логічне додавання $\vee$, інверсії (~).

У роботі [4] запропоновано сукупність керованих примітивів, які використовують дані операції. На рис. 1 наведено схему уніфікованого пристрою для керованих криптографічних перетворень, запропонованого у статті [4].

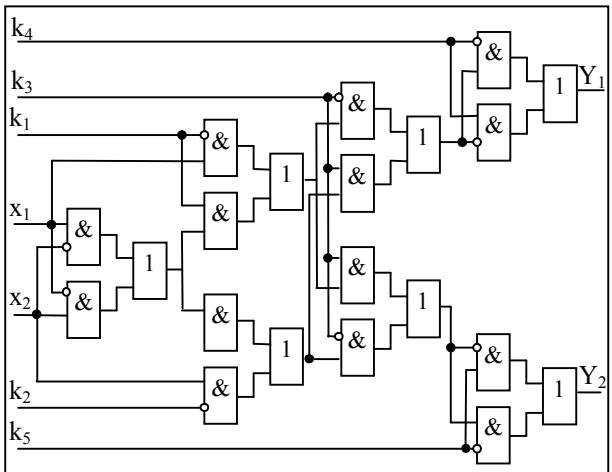

Рис. 1. Схема уніфікованого пристрою

У методах хешування даний пристрій доцільно використати для реалізації функції ущільнення. З рис. 1 видно, що затримка формування вихідного значення за допомогою даного складається із затри-

© В.А. Лужецький, Ю.В. Баришев

мок у 8 елементах. При цьому апаратна складність пристрою становить двадцять один логічний елемент. Такі швидкість роботи спеціалізованого процесора є недостатньою. У роботі [4] втрата швидкості обумовлена універсальністю процесора, а також використанням різних логічних функцій як криптографічних примітивів.

У роботі [5] наводяться спеціалізовані пристрої для керованих криптографічних перетворень на основі керованих перестановок. Попри високі показники швидкості хешування, які досягалися за допомогою пристроїв, розроблених у роботі [5], програмна реалізація даних методів має низку недоліків, пов'язаних з відсутністю елементів, що реалізують побітову перестановку в універсальних мікропроцесорах. Саме тому методи керованого хешування, запропоновані у [5], істотно залежать від платформ, а тому у низці задач не можуть конкурувати з методами керованого хешування, які розроблялися з урахуванням архітектури універсальних процесорів.

Розробка блока ущільнення

В загальному випадку багатоканальне кероване хешування базується на такій конструкції:

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

де $h_i^{(j)}$ – проміжне хеш-значення, отримане у j-му каналі ($j = \overline{1, q}$) на і-й ітерації ($i = \overline{1, l}$); m_i – і-й блок даних; $f_{v_i^{(j)}}(\cdot)$ – функція ущільнення, що забезпечує сталу довжину вихідного значення; $r_i^{(j)(w)}$ – w-е ($j = \overline{1, z}$) псевдовипадкове число, що використовується у j-му каналі на і-й ітерації; $v_i^{(j)}$ – вектор керування, який визначає параметри перетворення функції ущільнення $f_{v_i^{(j)}}(\cdot)$ у j-му каналі на і-й ітерації; $g(\cdot)$ – функція формування вектора керування.

Для усунення недоліків відомих підходів до розробки керованих криптографічних перетворень, пропонується розглядати перетворення не з точки зору послідовності операцій, які виконуються для двох (трьох) бітів, як це зроблено у роботах [4, 5], а беручи до уваги одразу весь блок даних. Саме тому пропонується виконувати керування кількістю бітів, на яку відбувається циклічний зсув. Якщо взяти за основу логічну функцію, використану у стандарті хешування SHA-2 [6], то отримаємо такий криптографічний примітив для керованого хешування:

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

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

У формулі (2) замість $h_{i-1}^{(j+1)}$ та $h_{i-1}^{(j+2)}$ можуть використовуватись будь-які інші аргументи функції ущільнення, визначені конструкцією (1).

Використовуючи аналогічним чином ще одну функцію зі стандарту SHA-2 [6], отримаємо такий криптографічний примітив:

$$\begin{aligned} h_i^{(j)} = &\left(m_i >>> u_i^{(j)(m1)} \wedge h_{i-1}^{(j+1)} >>> u_i^{(j)(h1)}\right) \oplus \\ &\oplus\left(m_i >>> u_i^{(j)(m1)} \wedge h_{i-1}^{(j+2)} >>> u_i^{(j)(h2)}\right) \oplus \\ &\oplus\left(h_{i-1}^{(j+1)} >>> u_i^{(j)(h1)} \wedge h_{i-1}^{(j+2)} >>> u_i^{(j)(h2)}\right). \end{aligned} \quad (3)$$

На рис. 2 зображено схему пристрою, що реалізує криптографічний примітив (2).

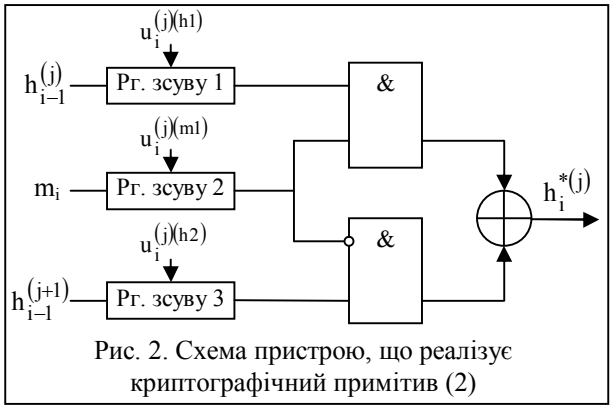

Рис. 2. Схема пристрою, що реалізує криптографічний примітив (2)

З рис. 2 видно, що кожен регістр зсуву має два входи: один – для вхідних даних (m_i або $h_i^{(j)}$ або $r_i^{(j)(w)}$), другий – для керуючого сигналу $u_i^{(j)(x\kappa)}$. Керуючий сигнал визначає кількість бітів, на яку відбувається зсув вхідних даних. З формули (2) випливає, що на перший та третій входи повинні подаватись проміжні хеш-значення $h_i^{(j)}$ або псевдовипадкові числа $r_i^{(j)(w)}$, на другий вхід – блок даних m_i.

Апаратна складність даного блоку становить два логічні елементи, один суматор за модулем 2 та три регістри зсуву.

На рис. 3 зображено схему пристрою, що реалізує криптографічний примітив (3).

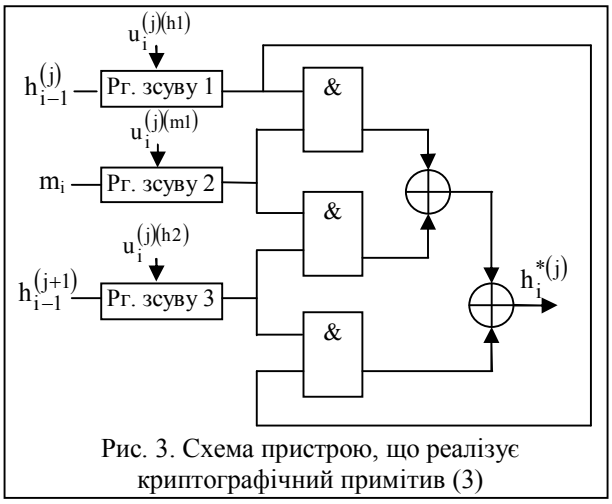

Рис. 3. Схема пристрою, що реалізує криптографічний примітив (3)

Аналогічно пристрою, що реалізує криптографічний примітив (2) пристрій, пристрою, що реалізує криптографічний примітив (3) має шість входів: три – для даних, три – для керуючих сигналів. Однак на відміну від пристрою, що реалізує криптографічний примітив (2), для цього пристрою не принципово які дані подавати на який з входів.

Для одержання вихідного значення за допомогою блоків перетворення виду (3) достатньо чотирьох тактів спеціалізованого процесора. При цьому апаратна складність цих блоків становить три логічні елементи, два суматори за модулем 2 та три регістри зсуву. Для реалізації блоку ущільнення, який має конкретні значення параметрів k, z конструкції хешування (1) пропонується каскадувати блоки перетворення видів (2) та (3). Зокрема на рис. 4 наведено приклад блоку ущільнення для параметрів конструкції $k = 4$, $z = 0$.

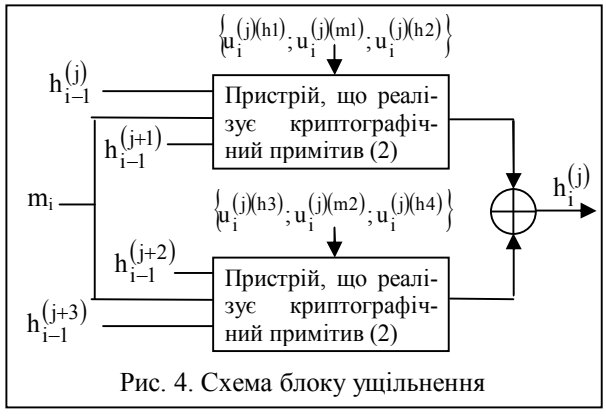

Рис. 4. Схема блоку ущільнення

Блок ущільнення, зображений на рис. 4, формуватиме вихідне значення за чотири такти. При цьому апаратна складність даного блоку становить чотири логічних елементи, три суматори за модулем 2 та шість регістрів зсуву. Аналогічним чином будуються блоки ущільнення для довільних значень параметрів k, z конструкції (1).

Якщо розташовувати суматори за модулем 2 у вигляді бінарного дерева та використовуються лише блоки перетворення виду (2) тривалість формування вихідного значення визначається так:

$$t_{\text{reductoin}(2)} = 3 + \log_2\left(\frac{k+z}{2}\right) = 2 + \log_2(k+z). \quad (4)$$

Аналогічно тривалість формування вихідного значення блоком ущільнення при використанні в його структурі блоків перетворення виду (3) буде на один такт більшою.

Необхідно передбачити структуру блоку формування вектора керування, який визначатиме значення керуючих сигналів $u_i^{(j)(x\kappa)}$.

Розробка блоку формування вектора керування

Для того, щоб вектор керування впливав однаково на кожне значення $u_i^{(j)(x\kappa)}$, пропонується формувати його шляхом конкатенації значень $u_i^{(j)(x\kappa)}$. Зокрема для блоку ущільнення, зображеного на рис. 4, вектор керування має такий вигляд:

$$v_i^{(j)} = u_i^{(j)(h1)} \,||\, u_i^{(j)(m1)} \,||\, u_i^{(j)(h2)} \,||\, u_i^{(j)(h3)} \,||\, u_i^{(j)(m2)} \,||\, u_i^{(j)(h4)}, \quad (6)$$

де "||" – позначення конкатенації.

Довжина вектора керування (6) становить:

$$n_v = 3 \cdot \frac{k+z}{2} \cdot \log_2\left(\frac{n}{q}\right) = 6 \cdot \log_2\left(\frac{n}{q}\right), \quad (7)$$

де n – довжина вихідного хеш-значення.

У цьому випадку вектор керування формуватиметься шляхом простого розгалуження провідників (аналогічно до запропонованого у роботі [5]). Такі припущення накладають істотні обмеження на параметри конструкцій хешування. Саме тому пропонується формувати вектор керування шляхом додавання за модулем 2 аргументів функції формування вектора керування (див. формулу (1)) доти, доки довжина результату додавання не дорівнюватиме n_v. Схему блоку формування вектора керування для випадку, коли $n_v = a \cdot n / q$ зображено на рис. 5.

Тривалість формування вектора керування за допомогою пристрою, зображеного на рис. 5, становитиме $(1 + \varphi / a)$ тактів при апаратні складності n_v тригерів та n_v суматорів.

У застосуваннях, де швидкість хешування важливіша за апаратні витрати, пропонується використовувати каскади суматорів, тоді оцінка тривалості визначатиметься за такою формулою:

$$t_{vector} = \log_2\left(\frac{n\cdot\varphi}{q\cdot n_v}\right)+1. \qquad (8)$$

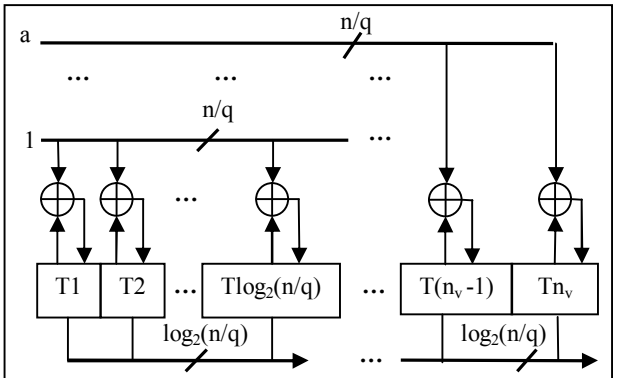

Рис. 5. Схема блоку формування вектора керування

Таким чином, хешування, що має такі параметри $q = 16, k = \varphi = 8, z = 2, n = 256$ (з формули (7) випливає $n_v = 3\cdot 5\cdot 4 = 60$), виконуватиметься за допомогою спеціалізованого процесора за вісім тактів.

Шляхом поєднання блоків ущільнення та формування вектора керування, отримуємо загальну схему спеціалізованого процесора, що реалізує j-й канал керованого хешування (рис. 6).

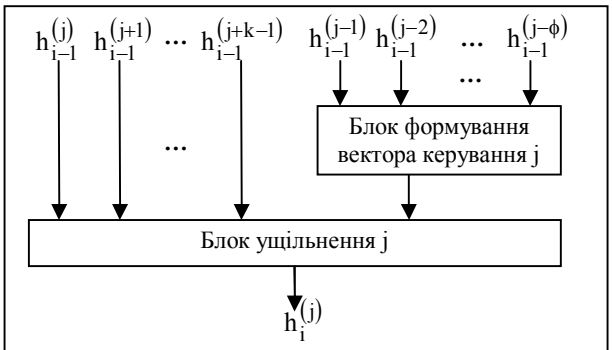

Рис. 6. Схема пристрою, що реалізує j-й канал хешування

Спеціалізований процесор, що реалізує багатоканальне кероване хешування, складається з q пристроїв, з'єднаних відповідним чином, кожен з яких реалізує j-й канал хешування і має структуру, що наведена на рис. 6.

Висновки

З виконаного аналізу відомих пристроїв, що реалізують багатоканальне та кероване хешування виливає, що для збільшення швидкості доцільно виконувати керування регістровими операціями, а не побітовими. Серед запропонованих пристроїв, що реалізують криптографічні примітиви доцільніше використовувати блоки перетворення виду (2), оскільки вони мають кращі показники як швидкості, так і апаратної складності. Підходи, використані при побудові спеціалізованого процесора у даній статті дозволяють реалізувати методи багатоканального керованого хешування конструкції (1) з довільними параметрами k, z, φ. Одержані оцінки тривалості хешування свідчать про те, що у низці випадків (залежно від конкретних значень параметрів k, z, φ) запропоновані підходи дозволяють будувати спеціалізовані процесори багатоканального керованого хешування, які швидші за відомі.

Список літератури

1. Корченко А.Г. Построение систем защиты информации на нечетких множествах. Теория и практические решения / Александр Григорьевич Корченко. – К.: "МК-Пресс", 2006. – 320 с.

2. Лужецький В.А. Криптографічні примітиви для реалізації керованого хешування / В.А. Лужецький, Ю.В. Баришев // Вісник ВПІ. – 2011. – №1. – С. 108-111.

3. Петров А.А. Компьютерная безопасность. Криптографические методы защиты / А.А. Петров. – М.: ДМК, 2000. – 448 с.

4. Рудницький В.М. Модель уніфікованого пристрою криптографічного перетворення інформації / В.М. Рудницький, В. Г. Бабенко // Системи обробки інформації: зб. наук. пр. – Х.: ХУПС, 2009. – Вип. 3. – С. 91-95.

5. Молдовян Н.А. Криптография: от примитивов к синтезу алгоритмов / Н.А. Молдовян, А.А. Молдовян, М.А. Еремеев. – СПб.: БХВ-Петербург, 2004. – 448 с.

6. Secure Hash Standard: Federal Information Processing Publication Standard Publication 180-3 [Електронний ресурс]. – Gaithersburg, 2008. – 27 с. – Режим доступу до стандарту: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.

Надійшла до редколегії 19.04.2011

Рецензент: д-р техн. наук, проф. В.О. Хорошко, Державний університет інформаційно-комунікаційних технологій, Київ.

АППАРАТНЫЕ СРЕДСТВА ДЛЯ РЕАЛИЗАЦИИ МНОГОКАНАЛЬНОГО УПРАВЛЯЕМОГО ХЕШИРОВАНИЯ

В. А. Лужецкий, Ю. В. Барышев

В данной статье рассмотрено аппаратную реализацию многоканального управляемого хеширования. Предложено подходы к разработке блоков сжатия и формирования вектора управления. Приведено схемы устройств, полученных согласно предложенных подходов, а также оценки скорости и аппаратной сложности этих устройств.

Ключевые слова: *хеширование, многоканальность, управляемость, оценки.*

HARDWARE MEANS FOR MULTIPIPE DRIVEN HASHING IMPLEMENTATION

V. A. Luzhetsky, Y.V. Baryshev

The hardware implementation of multipipe driven hashing is considered in the article. Approaches of reduction block and blocks of driving vector forming development are proposed. Devises charts received according to the approaches and estimations of these devises rapidity and hardware complexity are presented.

Keywords: *hashing, multipipe, driving, estimations.*