Методи Байт-Орієнтованого Хешування Даних Низькоресурсної Криптографії

Володимир Лужецький
кафедра захисту інформації
Вінницький національний технічний університет
Вінниця, Україна
lva.zi2002@gmail.com

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

Світлана Слободян
кафедра обчислювальної техніки
Вінницький національний технічний університет
Вінниця, Україна
feride.fe@list.ru

Олександр Ювковецький
кафедра обчислювальної техніки
Вінницький національний технічний університет
Вінниця, Україна
alex.yuvkovetskyi@gmail.com

Methods of byte-oriented hashing of lightweight cryptography

Volodymyr Luzhetskyi
Department of Information Protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
lva.zi2002@gmail.com

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

Svitlana Slobodyan
Department of Information Protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
feride.fe@list.ru

Olexander Yuvkovetskyi
Department of Information Protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
alex.yuvkovetskyi@gmail.com

***Анотація*—Запропоновано принципово новий байт-орієнтований підхід до побудови хеш-функцій для низькоресурсних систем з використанням характеристичних ознак даних. Наведено схеми обчислень і структурні схеми спеціалізованих процесорів для хешування та оцінки їх апаратної складності.**

***Abstract*—A fundamentally new byte-oriented approach to constructing hash functions for lightweight systems using the characteristic features of data is proposed. Schemes of calculations and structural schemes of specialized processors for hashing and estimation of their hardware complexity were presented.**

Ключові слова—хеш-функція, байт-орієнтоване хешування, низькоресурсне хешування, спеціалізований процесор, апаратна складність.

Keywords—hash-function, byte-oriented hashing, lightweight hashing, specialized processor, hardware complexity .

I. Вступ

Основною особливістю сучасних IT- та Інтернет-технологій є значне поширення різноманітних інтелектуальних пристроїв, зокрема гаджетів, що мають доступ до Інтернет. Стрімкий розвиток таких технологій формує ряд актуальних задач пов'язаних з їх інформаційною безпекою. Також, разом з широким переліком традиційних інтернет-пристроїв, таких як персональні комп'ютери та ноутбуки, смартфони, розумні годинники і браслети, почали з'являтися побутові пристрої, термінали, безпілотні літаки, транспортні зчитувачі і навіть автомобілі з автопілотом та навіть давачі, що мають доступ до певної мережі (зазвичай Інтернет) [1,2].

Такі технології отримали назву Internet of Things (IoT) або Інтернет речей, що є бездротовою мережею, яка самоконфігурується, з різноманітних засобів, до складу яких входять давачі та сенсори. Також, останнім часом, набули поширення модулі, що використовують RFID-мітки (Radio Frequency Identification) або NFC технологію (Near Field Communication). Крім того, їх регулярно використовують численні програми, пов'язані з обробкою персональних даних, біометричних даних, відомостей фінансового та військового характеру і т. п. [1,2].

Саме тому особливо актуальною постає задача ефективної реалізації алгоритмів захисту інформації, левову частку яких складають криптографічні методи захисту інформації. Беручи до уваги особливості побудови та функціонування таких пристроїв, найкращим рішенням для них є вид криптографії, що носить назву «низькоресурсна криптографія» (LightWeight Cryptography, LWC) [3-7]. Вона потрібна для тих сфер застосування, де є жорсткі вимоги чи обмеження на ціну, габарити, ресурси системи (такі як пам'ять, обчислювальна потужність, джерело живлення і т. п.) при проектуванні та промисловому виготовленні подібних пристроїв. Більшість вимог, які висуваються до алгоритмів, що застосовуються у низькоресурсних засобах, обумовленні міжнародним стандартом ISO/IEC FDIS 29192 – Information technology – Security techniques – Lightweight cryptography.

Одним з декількох основних напрямків розвитку низькоресурсної криптографії є більш ефективна реалізація вже відомих алгоритмів шифрування, у більшості випадків з незначною їх модифікацією. При цьому передбачається, що рівень безпеки такого шифрування знизиться до рівня, який є достатнім для більшості його застосувань.

Усе це обумовлює актуальність проведення досліджень у напрямку створення засобів, які задовольняють LWC. Типовими обмеженнями для низькоресурсної криптографії є складність апаратної реалізації, час, що витрачається на виконання програми, споживана енергія.

Серед криптографічних перетворень особливе місце займають ключові хеш-функції, відомі як коди перевірки справжності повідомлення (Message Authentication Code - MAC) або імітовставка, та безключові хеш-функції або коди виявлення модифікації інформації (Modification Detection Code - MDC).

Відомі стандарти хеш-функцій для своєї реалізації потребують понад 10000 умовних логічних елементів (GE – gate equivalent), тоді як гранична складність не повинна перевищувати 2000 GE [5]. Певною мірою вимоги низькоресурсної криптографії забезпечують хеш-функції сімейства QUARK [5], однак продовжуються пошуки підходів до хешування, які б забезпечили ще кращі показники апаратної складності та часу хешування.

II. Методи хешування

Авторами пропонується новий підхід до побудови LW хеш-функцій з байт-орієнтованою обробкою даних, суть якого полягає в такому. Вхідне повідомлення M розглядається як послідовність байтів $M=\{ m_1, m_2, ..., m_L\}$. Хеш-функції – це функція, яка певним чином пов'язує ASCII–коди байтів m_l та номери позицій l (l = 1, 2, …) цих байтів у повідомленні [8-11].

Передбачається хешування даних довільної довжини та отримання хеш-значення різного розміру (128, 192, 256 біт).

Метод 1.

Хеш-значення формується з ASCII–кодів байтів з урахуванням номерів позицій l. Таку хеш-функцію будемо скорочено позначати SNP.

Виходячи із заданої довжини хеш-значення l_h визначається кількість байт, що будуть використовуватися для обчислень:

$$g = l_h / 8 . \qquad (1)$$

Наприклад, при довжині хеш-значення 128 біт, це хеш-значення представляється у вигляді набору з 16 елементів, кожен з яких є байтом (128=8×16), відповідно для довжини 192 біт набір складається з 24 елементів (192=8×24), а для 256 – з 32 елементів (256=8×32).

При цьому, номер позиції байта, використовуваний у подальшому, визначається за формулою:

$$q = l \bmod 2^g .$$

Початкове хеш-значення h_0 є сукупністю псевдовипадкових чисел і має вигляд:

$$h_0 = \{h_{0,0}, h_{0,1},..., h_{0,(g-1)}\} .$$

Номер позиції q байта у повідомленні розглядається як двійковий код:

$$q = \sum_{i=0}^{g-1} a_i \cdot 2^i .$$

Проміжні хеш-значення $h_l = \{h_{l,0}, h_{l,1},..., h_{l,(g-1)}\}$ обчислюється на основі попереднього хеш-значення h_{l-1}, двійкового представлення номера позиції q та ASCII–кода байта n_l:

$$h_{l,j} = h_{(l-1),(j-1)} \oplus (n_l \cdot a_{j-1}),$$
$$h_{l,0} = h_{l,(g-1)} \oplus (n_l \cdot a_{g-1}),$$

де $j = 1,2,...,(g-1)$.

Схему обчислень хеш-значеннь за даним методом наведено на рис.1.

Для хешування одного байту даних потрібно виконати g операцій додавання за модулем 2 двох кодів байтів і $(g+1)$ операцій запису коду в регістр.

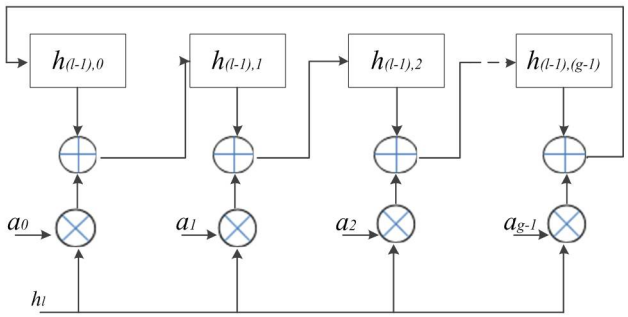

Рис. 1. Схема обчислень хеш-значеннь за методом 1

Метод 2.

Хеш-значення формується з псевдовипадкових чисел, що відповідають номерам позицій *l* байтів, з урахуванням ASCII–кодів байтів. Таку хеш-функцію будемо скорочено позначати PRNS.

Розрядність двійкового коду псевдовипадкових чисел визначається за формулою (1).

Початкове, проміжні та остаточне хеш-значення складаються з 8 елементів довжиною *g* біт кожен.

Нехай початкове хеш-значення має вигляд:

$$h_0 = \{h_{0,0}, h_{0,1}, ..., h_{0,7}\}$$

Тут кожна складова може приймати будь-які значення. У разі, коли ці значення відомі, маємо справу з безключовою хеш-функцією, а коли початкове хеш-значення є секретним, то це буде ключова хеш-функція.

Проміжні хеш-значення $h_l = \{h_{l,0}, h_{l,1}, ..., h_{l,7}\}$ обчислюються на основі попереднього хеш-значення, ASCII-коду байту $\{a_{l.0}, a_{l.1} ... a_{l.7}\}$ і чергового псевдовипадкового числа s_l з циклічним зсувом складових хеш-значення:

$$h_{l,j} = h_{(l-1),(j-1)} \oplus (s_l \cdot a_{j-1}),$$
$$h_{l,0} = h_{l,7} \oplus (s_l \cdot a_7),$$

де $j = 1,2,\ldots 7$.

Для генерування послідовності псевдовипадкових чисел пропонується використовувати регістр зсуву з лінійним зворотнім зв'язком (LFSR). Як псевдовипадкове число розглядається стан цього регістру.

Вибір генератора на основі LFSR обґрунтовується такими його властивостями [12].

1. Псевдовипадкові числа належать інтервалу $[1, 2^n - 1]$.
2. В межах періоду генератора числа не повторюються.
3. Регістри зсуву в конфігурації Галуа забезпечують паралельну обробку за один такт кожного з двійкових розрядів, зі швидкістю, що порівняна зі швидкістю перемикання транзисторів типових мікросхем.
4. Складність апаратної реалізації може дорівнювати $(4 \cdot n + 6)$ GE.

Схему обчислень хеш-значеннь за даним методом наведено на рис.2.

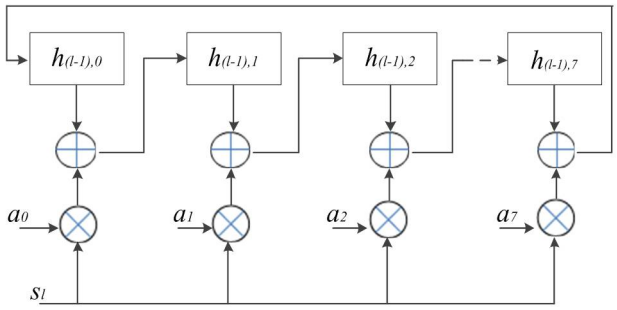

Рис. 2. Схема обчислень хеш-значеннь за методом 2

Для хешування одного байту даних необхідно виконати 8 операцій додавання за модулем 2 двох *g*-розрядних кодів, 9 операцій запису коду в регістр і одну операцію для формування псевдовипадкового числа.

III. Спеціалізовані процесори для хешування даних

Для отримання хеш-коду використовується спеціалізований процесор МН (message hashing), який певним чином підключається до центрального процесора і обмінюється з ним потоками даних, виконуючи певні функції. Передбачається, що інформація, для якої формується хеш-значення знаходиться в оперативній пам'яті, тому зчитування цих даних покладається на центральний процесор, а обчислення хеш-значення на МН-процесор.

Структурну схему МН-процесора, що реалізує хешування за методом 1, наведено на рис. 3. До складу пристрою входять *g* однобайтних регістрів для формування хеш-значення, лічильник розрядністю g, для формування коду номера позиції *q* поточного байту, регістр зсуву для зберігання коду числа *q*, блок з 8 суматорів за модулем 2 і блок з 8 логічних елементів І.

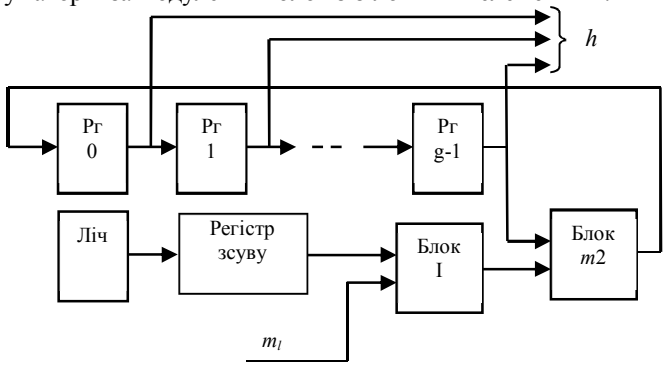

Рис. 3. Узагальнена структурна схема МН-процесора, що реалізує хеш-функцію SNP

Складність пристрою в умовних одиницях GE, при такій розрядності його складових, обчислюється за формулою:

$$S = 60 \cdot g + 24$$

Структурну схему МН-процесора, що реалізує хешування за методом 2, наведено на рис. 4. До складу пристрою входять вісім *g*-розрядних регістрів Рг0…Рг7, в

яких зберігаються проміжні хеш-значення, *g*-розрядний регістр зсуву з лінійним зворотнім зв'язком РЗЗЛЗЗ, який забезпечує формування послідовності псевдовипадкових чисел, 8-розрядний регістр зсуву, який забезпечує перетворення паралельного 8-розрядного ASCII–коду у послідовність бітів. Блок I складається із *g* логічних елементів І. Блок m2 складається із *g* суматорів за модулем 2.

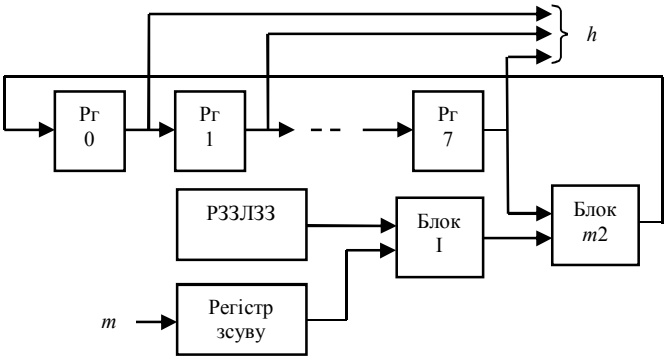

Рис. 4. Узагальнена структурна схема МН-процесора, що реалізує хеш-функцію PRNS

Складність такого спеціалізованого процесора в умовних одиницях GE обчислюється за формулою:

$$S = 57 \cdot g + 48 .$$

У таб. 1 наведено характеристики хеш-функцій сімейства QUARK та запропонованих авторами хеш-функції PRNS та SNP. Тут С/байт – позначає кількість операцій для обчислення хеш-значення для одного байту даних.

ТАБЛИЦЯ І. ХАРАКТЕРИСТИКИ ХЕШ-ФУНКЦІЙ

Хеш-функція	**Довжина хеш-значення, *біт***	**Складність, *GE***	**С/байт**
D-QUARK	160	1702	547
T-QUARK	224	2296	33
U-QUARK	128	1379	33
PRNS	128	960	18
PRNS	192	1416	18
PRNS	256	1872	18
SNP	128	984	33
SNP	192	1464	49
SNP	256	1944	65

IV. Висновки

Запропоновані методи хешування забезпечують апаратну реалізацію, складність якої не перевищує 2000 GE, що відповідає вимогам міжнародного стандарту ISO/IEC FDIS 29192, і ця складність у 1,44 рази менша за складність реалізації найкращих відомих алгоритмів хешування.

Для запропонованої хеш-функції PRNS кількість операцій, що потрібна для хешування одного байту даних, не залежить від довжини хеш-значення і в 1,8 рази менша порівняно з відомими хеш-функціями сімейства QUARK.

Результати дослідження запропонованих методів хешування з використанням тестів NIST показали, що вони забезпечують виконання вимог щодо криптографічної стійкості хеш-функції.

Література References

[1] Bogdanov A., Leander G., Paar C., Posch A., Robshaw M., Seurin Y. Hash Functions and RFID Tags: Mind the Gap -10. In ter na tio nal Work shop on Cryp to gra phic Hard ware and Em bed ded Sys tems - CHES 2008, Wa shington, USA. Au gust 10 - 13, 2008. Pages 283-299.

[2] Yoshida H., Watanabe D., Okeya K., Kitahara J., Wu J., Kucuk O., and Preneel B.. MAME: A Compression Function With Reduced Hardware Requirements. In P. Paillier and I. Verbauwhede, editors, Cryptographic Hardware and Embedded Systems — CHES 2007, volume 4727 of Lecture Notes in Computer Science, pages 148–165. Springer-Verlag, 2007.

[3] Manifavas C., Hatzivasilis G., Fysarakis K., Rantos K. Lightweight Cryptography for Embedded Systems - A Comparative Analysis, SETOP'2013.

[4] Panasenko S., Smagin S. Lightweight Cryptography: Underlying Principles and Approaches. International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011, pp. 516-520.

[5] Жуков А. Е. Легковесная криптография / А. Жуков // – Москва: Вопросы кибербезопасности №1(9), 2015.

[6] Bogdanov A., Knudsen L., Leander G., Paar C., Poschmann A., Robshaw M., Seurin Y., Vikkelsoe C. Present - An Ultra-Lightweight Block Cipher. In Proceedings of Workshop on Cryptographic Hardware and Embedded Systems — CHES 2007, Lecture Notes in Computer Science v. 4727, pp. 450-466, 2007.

[7] Preneel B. Perspectives on Lightweight Cryptography," Inscript 2010, Shanghai, China, 20-24 October 2010.

[8] 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.

[9] Лужецький В. А. Метод хешування даних на основі їх характеристичних ознак / В. А. Лужецький, Д. В. Кисюк // II міжнародна науково-практична конференція (Закарпатська область, Міжгірський район, село Верхнє Студене, туристичний комплекс «Едельвейс». 24-27 лютого 2016 р.). – К.: Видавництво Європейського університету, 2016. – 97-100 с.

[10] Лужецький В. А. Метод байт-орієнтованого хешування / В. А. Лужецький, Д. В. Кисюк, Л. А. Савицька // V Міжнародна науково-практична конференція «Методи та засоби кодування, захисту й ущільнення інформації». 19-21 квітня 2016 року, Вінниця – Немирів, ВНТУ. – 37-39 с.

[11] Лужецький В. А. Узагальнений метод хешування байтової форми представлення інформації / В. А. Лужецький, Д. В. Кисюк // Тези доповідей Четвертої Міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія» м. Вінниця, 28-30 травня 2014 року. – Вінниця: ВНТУ, 2014., – ⁻275 с.

[12] Watanabe D., Okamoto K., Kaneko T. A Hardware-Oriented Light Weight Pseudo-Random Number Generator Enocoro-128v2. The 2010 Symposium on Cryptography and Information Security, SCIS 2010, 3D1-3, 2010.