Удосконалення методу малоресурсного гешування HDG

УДК 621.395.7 (043.2)

Віталій Селезньов[1], Володимир Лужецький[2]

Вінницький національний технічний університет,

[1]seleznov.vitalii@gmail.com, [2]v.luzhetskyi@vntu.edu.ua

У сучасному цифровому світі використання малоресурсних пристроїв Інтернету речей (IoT), таких як розумні годинники, браслети, сенсорні вузли, смарт-карти, модулі, що використовують RFID-мітки або NFC технологію (Near Field Communication), тощо стає все більш поширеним. Однак обмежені обчислювальні та енергетичні можливості цих пристроїв створюють нові виклики для забезпечення безпеки та конфіденційності даних, зокрема, у контексті процесу гешування.

Проблема гешування на малоресурсних пристроях полягає в обмеженості їхніх ресурсів, які не завжди дозволяють ефективно виконувати складні алгоритми гешування. Це може призвести до повільної роботи пристроїв, високого споживання енергії та низької продуктивності. Крім того, існує ризик зниження безпеки даних через використання слабких або недостатньо надійних алгоритмів гешування на цих пристроях. У зв'язку з цим виникає потреба в розробці та оптимізації методів гешування, спеціально адаптованих до можливостей малоресурсних пристроїв. Метод малоресурсного гешування HDG (Hash Data Generator) є одним з потенційних рішень для вирішення цієї проблеми, однак він містить суттєвий недолік, а саме обмеження для застосування на даних, мінімальна довжина яких сягає 64 байти.

Метою даної роботи є удосконалення методу малоресурсного гешування HDG для усунення або максимального зменшення обмеження на довжину цільових даних для його застосування.

Метод малоресурсного гешування HDG передбачає використання підходу до гешування «дані – генератор» відповідно до якого проміжні та остаточні значення геш-функції обчислюється шляхом додавання байтів даних у позиціях, що відповідають бітам, які генерується псевдовипадковим чином.

Вхідне повідомлення m подається у вигляді послідовності байт :

$$m = \{m_1, m_2, ..., m_L\} \tag{1}$$

Початкове геш-значення h_0 є сукупністю псевдовипадкових байтів та представлено у вигляді послідовності:

$$h_0 = \{h_{0,0}, h_{0,1}, ..., h_{0,k-1}\}, \tag{2}$$

де $k = l/8$, l – довжина геш-значення в бітах.

Проміжні геш-значення $h_i = \{h_{i,0}, h_{i,1}, ..., h_{i,k-1}\}$ обчислюються на основі попереднього геш-значення h_{i-1} і поточного значення m_i шляхом виконання функції ущільнення f . В якості модифікації оригінального методу HDG пропонується використання $\overline{m_i}$, що є оберненим до оригінального значення даних при обчисленні проміжних геш-значень.

Кроки виконання функції ущільнення для *i*-го байту даних:

1. Для j від 0 до $k-1$ виконувати:
2. Згенерувати псевдовипадковий біт $g_{i,j}$.
3. Виконати обчислення:

$$h_{i,j} = \begin{cases} (h_{i-1,j} + m_i) \bmod 256, & \text{якщо } g_{i,(k-1-j)} = 1 \\ (h_{i-1,j} + \overline{m}_i) \bmod 256, & \text{якщо } g_{i,(k-1-j)} = 0 \end{cases} \quad (3)$$

Схему обрахунку модифікованої функції ущільнення за методом гешування наведено на рис. 1.

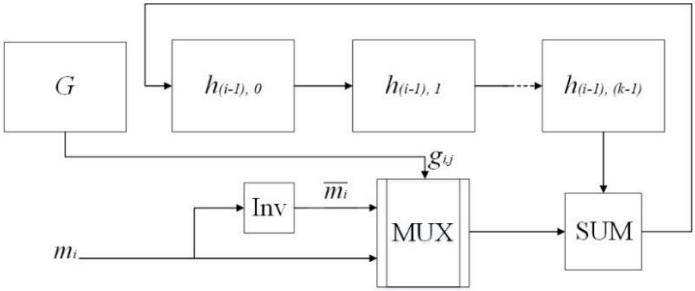

Рис.1. Схема обчислень геш-значень

Виконано статистичне тестування у пакеті NIST STS 800-22 модифікованого методу HDG на основі 200000 геш-значень, що обчисленні на основі повідомлень довжиною L = 8, 10, 11, 12, 16, 64 байт.

Таблиця 1

Результати статистичного тестування модифкованого HDG

L	Номер тесту NIST STS 800-22													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	+	-	+	+	+	-	+	+	+	+	-	+	+	+
9	+	-	+	+	+	+	-	+	+	+	+	+	+	+
10	+	-	+	+	+	+	-	+	+	+	-	+	+	+
11	+	+	+	+	+	+	-	+	+	+	+	+	+	+
12	+	+	+	+	+	+	+	+	+	+	-	+	+	+
13	+	+	+	+	+	+	+	+	+	+	+	+	+	+
14	+	+	+	+	+	+	+	+	+	+	+	+	+	+
15	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Відповідно до таблиці 1 для довжини повідомлення L=8 реалізація модифікованої функції ущільнення забезпечує усі успішні тести, окрім № 2, 6 та 11. Для довжини повідомлень L=12 успішними є усі тести, окрім №11. Починаючи з довжини повідомлень L=13 модифікований метод HDG забезпечує усі успішні тести, що є кращим у порівнянні з оригінальним HDG,

який у свою чергу забезпечує усі успішні тести NIST при значеннях L=64 та більше.

1. Encryption and hash based security in Internet of Things / B. Vinayaga Sundaram та ін. 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), м. Chennai, India, 26–28 берез. 2015 р. 2015. URL: https://doi.org/10.1109/icscn.2015.7219926 (дата звернення: 16.04.2024).
2. DeeR-Hash: A lightweight hash construction for Industry 4.0 / IoT. Journal of Scientific & Industrial Research. 2023. Т. 82, № 01. URL: https://doi.org/10.56042/jsir.v82i1.69938 (дата звернення: 16.04.2024).
3. Селезньов В. І., Лужецький В. А. Метод малоресурсного гешування типу «дані – генератор». Кібербезпека: освіта, наука, техніка. 2023. 2(22). С. 84-95.
4. Селезньов В. І. Програмний засіб для віддаленого статистичного тестування методів малоресурсного гешування за допомогою пакету NIST STS 822 / LIII Науково-технічна конференція факультету інформаційних технологій та комп`ютерної інженерії, ВНТУ, м. Вінниця, 20-22 березня 2024 р.
5. A statistical test suite for random and pseudorandom number generators for cryptographic applications / L. E. Bassham та ін. Gaithersburg, MD : National Institute of Standards and Technology, 2010. URL: https://doi.org/10.6028/nist.sp.800-22r1a (дата звернення: 16.04.2024).
6. NIST 800-22 українською мовою URL: http://www.itsway.kiev.ua/pdf/Articles180106.pdf (дата звернення 13.04.2024).