УДК 681.301

**Є. О. Ількевич**
**В. А. Лужецький**

# АЛГОРИТМ «ЛЕГКОЇ» ГЕШ-ФУНКЦІЇ

Вінницький національний технічний університет

***Анотація***

*Розглянуто застосування малоресурсної криптографії, зокрема, геш-функцій, в забезпеченні конфіденційності та перевірки цілісності інформації в IoT-інноваціях. Запропонований алгоритм "легкої" геш-функції дозволяє його реалізацію з мінімальною апаратною складністю та максимально можливою стійкістю. Описано структурну схему пристрою для гешування.*

**Ключові слова:** "легка" геш-функція, малоресурсна криптографія, електронні засоби, апаратна складність.

***Abstract***

*The use of low-resource cryptography, in particular, hash functions, in ensuring confidentiality and checking the integrity of information in IoT innovations is considered. The proposed algorithm of "easy" hash function allows its implementation with minimal hardware complexity and maximum possible stability. The structural scheme of operation of the hashing algorithm is described.*

**Keywords**: "easy" hash function, low-resourced cryptography, electronic means, hardware complexity.

## Вступ

На сьогоднішній день дуже поширеними стали різні електронні засоби та IoT-інновації, зокрема, "розумний" дім чи автомобіль, а також ряд інших сервісів міжмашинного зв'язку наступного покоління [1]. В забезпеченні конфіденційності та перевірки цілісності інформації в подібних електронних засобах вирішальну роль відіграє криптографія. Саме криптографічні функції гешування є одними з найважливіших примітивів, які використовують для створення криптографічних засобів захисту інформації. Їх призначення широко розповсюджене для таких цілей як підтвердження цілісності даних в електронному цифровому підписі та цифрових сертифікатах, електронних валютах, різних протоколах автентифікації користувачів та повідомлень, комп'ютерних системах контролю цілісності та виявлення втручань тощо [2].

У вбудованих системах також широко застосовуються геш-функції. Наприклад системи, що використовують RFID-мітки для автоматичної ідентифікації об'єктів. Вони можуть розпізнавати як живі істоти так і неживі предмети, наприклад, транспортні засоби, контейнери, одяг і багато іншого [3]. І це лише одне з багатьох застосувань, зокрема, таких, як смарт-карти, сенсорні мережі, USB-ключі, інтелектуальні карти, OTP-токени, системи охоронно-пожежної сигналізації та контролю доступу, системи промислово-побутової автоматизації та моніторингу [2].

З розвитком різних вбудованих систем з'явилась і необхідність захисту інформації в них, що стало причиною інтенсивних досліджень способів ефективної реалізації криптографічних алгоритмів, за умови обмеженості ресурсів, яку накладають ці системи. Такими ресурсами є: споживана потужність, продуктивність процесорного ядра, розмір пам'яті. Згодом можуть з'являтися і інші обмеження. Все це залежить від конкретних умов застосування електронного засобу.

Метою роботи є дослідження та реалізація нового алгоритму "легкої" геш-функції для мінімізації ресурсів, використовуваних для забезпечення допустимого рівня криптостійкості і швидкості роботи.

## Результати дослідження

У більшості вбудованих систем, в яких співвідношення ціни та витрат є критичними, обчислювальна потужність сконцентрована у недорогих центральних процесорах. З огляду на це, у роботі запропоновано нову функцію гешування "Символ-генератор", що була розроблена для забезпечення мінімальної апаратної складності, залишаючись при цьому досить стійкою.

Даний алгоритм утворює вихідний геш розміром 128 або 256 біт з вхідного повідомлення P будь-якої довжини $L$. Повідомлення P розглядається як послідовність байтів:

$$\text{P} = s_0, s_1, \dots, s_{L-1} .$$

Проміжне та остаточне геш-значення представляються у вигляді масиву з 16-ти 8-бітних елементів:

$$H_i = h_{i,0}, h_{i,1}, \dots, h_{i,15} .$$

Усі елементи $H_0$ є нульовими кодами.

В основі даного алгоритму лежить додавання ASCII-кодів байтів повідомлення за модулем 256 до елементів масиву та циклічний зсув цих елементів вліво. В яких саме позиціях буде відбуватись додавання визначається 16-ти бітною послідовністю $G$, що генерується регістром зсуву з лінійним зворотним зв'язком. Ця послідовність має таке представлення:

$$G_i = g_{i,0}, g_{i,1}, \dots, g_{i,15} .$$

Якщо $g_{i,j}$ = 1, то відбувається додавання до $j$-го елементу масиву. Отже з урахуванням циклічного зсуву, формування елементів проміжного геш-значення $h_{i+1,j}$ описується формулами:

$$h_{i+1,j} = \left( h_{i,j+1} + g_{i,j+1} * s_i \right) mod\ 256 \text{, де } j = 1, \dots, 15 ;$$

і

$$h_{i+1,15} = \left( h_{i,0} + g_{i,0} * s_i \right) mod\ 256 .$$

Для реалізації цього алгоритму пропонується структурна схема пристрою, що наведена на рисунку 1.

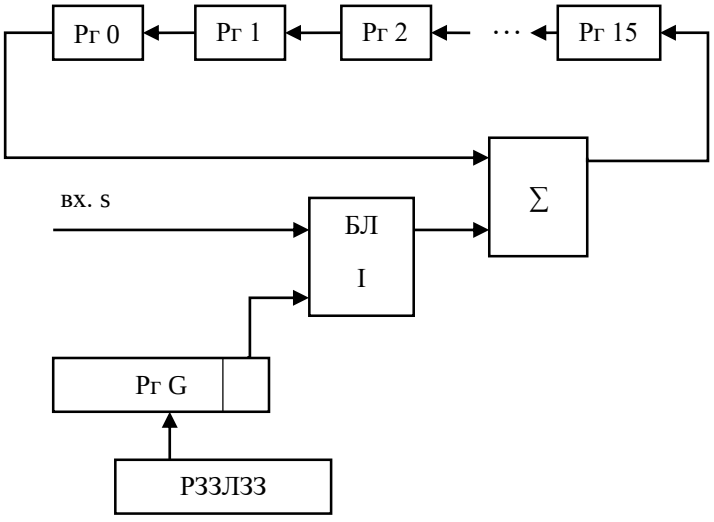


Рисунок 1 – Структурна схема пристрою

Регістри з Рг0 по Рг15 призначені для зберігання елементів геш-значень $h_{i,j}$. Генерування послідовності $G$ забезпечує регістр зсуву з лінійним зворотнім звязком РЗЗЛЗЗ. Регістр $G$ зберігає елементи послідовності $G$. Блок елементів "I" БЛ I реалізує операцію множення $g_{i,j+1} * s_i$. Суматор $\sum$ реалізує додавання за модулем 256.

Складність пристрою $S$ обчислюється за формулою:

$$S = 128\ \text{Тг} + 8\ \text{См} + 8\ \& + 16\ \text{Тг} + 16\ \text{Тг} + 5 \oplus ,$$

де Тг – тригер складністю 5,33 $GE$, См – однорозрядний суматор складністю 14,33 $GE$, & - елемент "І" складністю 1,33 $GE$, $\oplus$ - "Виключне АБО" складністю 2,67 $GE$.

З цієї формули випливає, що складність пристрою у разі геш-значення довжиною 128 біт становить 991 $GE$, а для отримання геш-значення довжиною 256 біт потрібен пристрій складністю 1844 $GE$. В

порівнянні з U-QUARK у якій найбільш “легка” реалізація має складність 1379 $GE$ при довжині геш-значення 128 біт, чи DM-PRESENT-128, складність якої становить 1886 $GE$, дана геш-функція значно перевершує їх результати [4].

**Висновки**

Отже, представлений алгоритм “легкої” геш-функції має досить просту реалізацію, що доводить його превагу над іншими алгоритмами подібних геш-функцій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Прудников В.А. Тенденции развития малоресурсной криптографии. URL:http://ucom.ru/doc/na.2018.12.02.046.pdf..
2. Я. І. Грабовський, Я. Р. Совин, І. Я. Тишик. Порівняння реалізацій нових алгоритмів гешування SHA-3 та гост Р 34.11-2012 для 8/32-бітових мікроконтролерних архітектур.
3. Технология RFID, метки, ридеры и ее применение. URL:https://realtrac.com/ru/company/blog/princip_raboty_tehnologii_rfid_i_ee_primenenie.
4. Жуков А.В. Легковесная криптография. Ч. 1 // Вопросы кибербезопасности. 2015. №1. М. 2015. С. 26-43.



***Ількевич Євгеній Олегович*** — студент групи 1бс-16б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: Evgeniy07109817@gmail.com

***Лужецький Володимир Андрійович*** — д. т. н., професор, завідувач кафедри захисту інформації, Вінницький національний технічний університет, Вінниця

***Ilkevich Yevheniy O.*** — Department of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email : Evgeniy07109817@gmail.com

***Luzhetskiy Vladimir A.*** — Doctor of Technical Science, Professor, Head of Information Security Department, Vinnytsia National Technical University, Vinnytsia