УДК 004.056 : 004.424.47

Комаров А. О.
Науковий керівник: к. т. н., доц. Баришев Ю. В.

МЕТОД ТА ПРОГРАМНИЙ ЗАСІБ РОЗПАРАЛЕЛЕНОГО ГЕШУВАННЯ СТІЙКОГО ДО ЗАГАЛЬНИХ АТАК

Вінницький національний технічний університет

Анотація: *У даній роботі проведено аналіз аналіз сучасного стану розвитку геш-функцій, атак, які грунтуються на знаходженні мультиколізій та методів протидії ним. Для покращення стійкості конструкцій, що передбачають розпаралелення запропоновано конструкції гешування та алгоритми.*
Ключові слова: *конструкція гешування, мультиколізії, атака Жу.*

Abstract: *This work presents an analysis of attacks which are based on multicollision finding and their counteraction methods. Known approaches were analyzed to improve the hash infeasibility against multicollision. The new approach of multicollision resistant hash constructions designing.*
Keywords: *hash construction, multicollisions, Joux attack.*

Для оптимального використання багатоядерних процесорів програмні засоби мають виконувати розпаралелені обчислення. Особливої значущості це набуває для криптографічних обчислень, які передбачають обробку чисел великої розрядності. До таких обчислень зокрема належить гешування. Водночас відома низка загальних атак, що використовують мультиколізії, яка особливо значуща для методів розпаралеленого гешування [1], при цьому стійкість до мультиколізій не залежить від криптографічних примітивів, що використовуються під час гешування, а залежать від використовуваних конструкцій [2-5]. Відповідно постає актуальна задача розробки методів підвищення стійкості гешування до атак на основі мультиколізій, які б дозволяли використовувати можливість розпаралелення обчислень зі збереженням стійкості геш-функції до зламу.

Метою даної роботи є підвищення стійкості методів гешування шляхом зав'язування даних, що гешуються.

Відома низка методів підвищення стійкості до мультиколізій[1, 2, 6-8]. Однак їх аналіз показав, що дані методи лише ускладнюють зловмиснику задачу побудови мультиколізії, при цьому залишаючи саму можливість її побудови.

В роботі [1] показано, що вразливість до цього класу атак криється в конструкціях, тому методи протидії повинні впроваджуватись саме на рівні конструкцій. Для покращення стійкості до мультиколізій пропонується така конструкція:

$$\begin{cases} h_i = f(h_{i-1}, m_i, m_{i-r_i}) \\ r_i = rand(m_i) \end{cases};$$

де $rand(\cdot)$ – функція, що забезпечує рівномірний розподіл вихідних значень r_i на кожній ітерації (якщо $i - r_i < 0$, то обирається блок $i - r_i + l$).

Для покращення стійкості такого методу, з послідовним обчисленням, пропонується визначати номер другого блоку даних залежно від більшої кількості аргументів функції $rand(\cdot)$:

$$\begin{cases} h_i = f(h_{i-1}, m_i, m_{i-r_i}) \\ r_i = rand(m_i, m_{i+1}) \end{cases};$$

Для тестування було згенеровано 5 псевдовипадкових послідовностей довжиною 63992 байт. З додаванням розміру повідомлення і розбиванням на 256 біт, отримано 1000 блоків даних.

Таблиця 1 – Частота вибору блоку у якості аргументу функції ущільнення

	Експе-римент №1	Експе-римент №2	Експе-римент №3	Експе-римент №4	Експе-римент №5
Блоки не були зав'язані	361/1000	385/1000	367/1000	367/1000	364/1000
Зав'язані блоки на 1 ітерації	381/1000	354/1000	478/1000	363/1000	381/1000
Зав'язані блоки на 2 ітераціях	175/1000	172/1000	171/1000	197/1000	173/1000
Зав'язані блоки на 3+ ітераціях	83/1000	89/1000	84/1000	73/1000	82/1000

В таблиці 1 зображено частоту вибору певного блоку на кожні ітерації гешування для кожної з п’яти послідовностей. З таблиці видно, що близько третини блоків не беруть участі у зав’язуванні блоків, близько 40% використовуються на певній ітерації 1 раз, і менше третини блоків використовуються у функції ущільнення більше одного разу.

Даний експеримент був проведений для реалізації, у якій згенерований номер блоку даних залежить від значення двох блоків на кожній ітерації (m_i та m_{i+1}). Як наведено в табл. 2, в даному випадку кожен блок даних став аргументом функції ущільнення як мінімум один раз (не враховуючи те, що він є аргументом m_i на i-ій ітерації).

Таблиця 2 – Частота вибору блоку у якості аргументу функції ущільнення

	Експе-римент №1	Експе-римент №2	Експе-римент №3	Експе-римент №4	Експе-римент №5
Блоки не були зав’язані	-	-	-	-	-
Зав’язані блоки на 1 ітерації	1000/1000	1000/1000	1000/1000	1000/1000	1000/1000
Зав’язані блоки на 2 ітераціях	375/1000	368/1000	371/1000	389/1000	380/1000
Зав’язані блоки на 3+ ітераціях	258/1000	264/1000	255/1000	262/1000	261/1000

Порівнявши дані таблиць можна зробити висновок, що зав’язаність блоків в цілому(кількість використань блоку на інших ітераціях) збільшилась більше ніж в 2 рази.

З аналізу загальних атак, що використовують мультиколізії, та відомих конструкцій гешування випливає, що наразі невідомі методи, які б унеможливлювали побудову мультиколізій. Для цього було запропоновано низку конструкцій гешування повідомлення, які передбачають використання аргументу у функції ущільнення, що визначається за певним псевдовипадковим законом. Аналіз цього підходу дозволив формально та експериментально обґрунтувати підвищення стійкості, зокрема до атак Жу та Келсі-Коно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лужецький В. А. Конструкції хешування стійкі до мультиколізій / В. А. Лужецький, Ю. В. Баришев // Наукові праці ВНТУ. – 2010. – №1. – С 1-8

2. Joux A. Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions / Antoine Joux // Lecture Notes in Computer Science. – 2004. – № 3152. – С. 306-316

3. Kelsey J. Herding hash functions and the Nostradamus attack / John Kelsey, Tadayoshi Kohno. – 2005. – 18 с. – Режим доступу до статті: http://archives.scovetta.com/pub/ crypto/Nostradamus%20Attack.pdf

4. Lucks S. Design Principles for Iterated Hash Functions / S. Lucks // Cryptology ePrint Archive. – 2004. – 22 с. – Режим доступу до ресурсу: http://eprint.iacr.org/2004/253.pdf

5. Hoch J. Breaking the ICE - Finding Multicollisions in Iterated Concatenated and Expanded (ICE) / Jonathan J. Hoch, Adi Shamir. – 2006. – 16 с. – Режим доступу до статті: http://link.springer.com/chapter/10.1007%2F11799313_12

6. Gauravaram P. Cryptographic Hash Functions: Cryptanalysis, Design and Applications. / Praveen Gauravaram. – 2009. – 298 с. – Режим доступу до ресурсу: http://eprints.qut.edu.au/16372/1/Praveen_Gauravaram _Thesis.pdf

7. Preneel B. Analysis and Design of Cryptographic Hash Functions: PhD thesis / Bart Preneel. – Leuven: Katholieke Universiteit Leuven, 1993. – 323 с. – [Електронний ресурс]. – Режим доступу до ресурсу: http://homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf

8. Bertoni G. The Keccak sponge function family. / G. Bertoni, J.Daemen, M. Peeters, G. V. Assche. – [Електронний ресурс]. – Режим доступу: URL http://keccak.noekeon.org/specs_summary.html

Комаров Андрій Олегович факультет інформаційних технологій так комп’ютерної інженерії, студент групи ЗІ-15м, Вінницький національний технічний університет, Вінниця, komand9@gmail.com.

Баришев Юрій Володимирович кандидат технічних наук, доцент кафедри захисту інформації, Вінницький національний технічний університет, Вінниця, yuriy.baryshev@gmail.com.

Andrii Komarov faculty of Information Technologies and Computer Engineering, student group ZI-15m, Vinnytsya National Technical University, Vinnytsya, komand9@gmail.com.

Yurii Baryshev ph.d associated professor of information protection chair, Vinnytsya National Technical University, Vinnytsya, yuriy.baryshev@gmail.com.