Міністерство освіти і науки України
Вінницький національний технічний університет

Ю. В. Баришев, В. А. Лужецький

МЕТОДИ ТА ЗАСОБИ ШВИДКОГО БАГАТОКАНАЛЬНОГО ГЕШУВАННЯ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Монографія

Вінниця
ВНТУ
2016

УДК 004.056:004.424.47
ББК 32.972.53
Б-24

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 10 від 31.03.2016 р.)

Рецензенти:

О. Г. Корченко, доктор технічних наук, професор

В. М. Рудницький, доктор технічних наук, професор

Баришев, Ю. В.

Б-24 Методи та засоби швидкого багатоканального гешування даних в комп'ютерних системах : монографія / Ю. В. Баришев, В. А. Лужецький; за заг. ред. В. А. Лужецького – Вінниця : ВНТУ, 2016. – 144 с.

ISBN 978-966-641-672-1

В монографії розглядаються питання організації багатоканального гешування даних підвищеної стійкості до загальних атак на основі мультиколізій. Отримано узагальнену конструкцію багатоканального гешування, на основі якої описано відомі та запропоновано нові конструкції гешування. Розроблено нові методи гешування, що базуються на цих конструкціях. Наведені алгоритми програмних засобів та структури спеціалізованих процесорів для реалізації методів в комп'ютерних системах.

УДК 004.056:004.424.47
ББК 32.972.53

ISBN 978-966-641-672-1

ЗМІСТ

ВСТУП

Функціонування комп'ютерних систем пов'язане з обробкою, передаванням та зберіганням інформації. З цього випливає, що від властивостей інформації значною мірою залежить успіх у досягненні поставлених перед комп'ютерною системою завдань. Зокрема результат використання інформації, обробленої комп'ютерною системою, значно залежить не лише від процесів обробки, а й від вхідних даних, адже недостовірні вхідні дані спричинять появу недостовірних результатів та, як наслідок, прийняття неправильних рішень користувачами цих комп'ютерних систем [1–3]. Сучасні комп'ютерні системи передбачають спільне використання інформації багатьма користувачами, причому часто значна їх частина може змінювати цю інформацію [3–5]. Саме тому важливо забезпечити захищеність інформації від несанкціонованої модифікації, тобто її цілісність [3].

Спільне використання та обробка інформації в комп'ютерних системах багатьма користувачами обумовлює потребу в забезпеченні можливості обміну нею. Оскільки часто користувачі виконують цей обмін за допомогою комп'ютерних мереж, то для успішного функціонування комп'ютерних систем необхідно, щоб користувачі могли впевнитися у автентичності джерела інформації та автентичності самої інформації, яку вони отримують, для запобігання її підміни під час передавання [3–8].

Таким чином, з вищевикладеного випливає, що для функціонування комп'ютерних систем є актуальним забезпечення кібербезпеки, зокрема захист цілісності та автентичності даних, а також автентичності користувачів. Ці задачі наразі розв'язуються за допомогою засобів комп'ютерної криптографії, а саме – криптографічного гешування [1, 2, 5, 8–17]. При цьому відомі засоби комп'ютерної криптографії дозволяють забезпечувати певний рівень стійкості до атак зловмисників [1, 2, 8, 9, 11–13, 16], однак це досягається за рахунок реалізації складних обчислень, які потребують значних часових витрат на своє виконання [1–3, 9, 12, 15].

Важливий внесок у розробку методів та засобів комп'ютерної криптографії взагалі та гешування даних, зокрема, зробили такі вітчизняні та зарубіжні вчені: А. Я. Білецький, І. Д. Горбенко, В. К. Задірака, О. Г. Корченко, О. О. Кузнєцов, В. М. Рудницький,

М. М. Савчук, К. Г. Самофалов, Г. Бертоні, Е. Біхам, П. Гаураварам, І. Дамгаард, О. Данкелман, А. Жу, Дж. Келсі, Т. Коно, Ш. Люкс, Р. Меркль, М. А. Молдовян, О. А. Молдовян, Б. Преніл, Р. Рівест, А. Шамір, Б. Шнайєр, В. В. Ященко та інші [1, 3, 9–12, 18–51].

Реалізація певних процедур, що забезпечують цілісність та автентичність інформації, а також автентичність суб'єктів обміну цією інформацією, передбачає збільшення часових витрат на розв'язання задач, які ставлять перед комп'ютерною системою її користувачі. Внаслідок цього комп'ютерна система стає недружньою до користувача або взагалі не виконує поставлені перед нею задачі, зокрема ускладнюється обмін інформацією в режимі реального часу [1, 8, 9, 13, 17–19]. Класичним способом підвищення швидкості обчислень в комп'ютерних системах є розпаралелення обчислень завдяки створенню декількох обчислювальних каналів. Однак, низка атак [20–22], спрямованих саме проти розпаралеленого гешування, не дозволяє пришвидшити процес гешування без суттєвої втрати стійкості. Саме тому актуальним для комп'ютерних систем є створення криптографічних методів та засобів гешування даних підвищеної швидкості, стійких до таких атак.

У монографії наведено аналіз поняття та сучасний стан розвитку методів криптографічного гешування, а також основні класи атак на них. Для визначення методів підвищення стійкості гешування пропонується узагальнена конструкція гешування. Наведено окремі випадки конструкцій багатоканального гешування, які забезпечують можливість розпаралелення обчислень при збереженні стійкості на рівні нерозпаралеленого гешування. На основі цих конструкцій розроблено методи гешування. Для реалізації й експериментального дослідження яких розроблено структури програмних засобів та спеціалізованих процесорів.

РОЗДІЛ 1
СУЧАСНІ МЕТОДИ ГЕШУВАННЯ

1.1 Основні поняття про гешування даних

Комп'ютерна криптографія є молодою наукою, яка наразі розвивається значними темпами, як й інші науки, що розглядають методи та моделі зберігання, передавання, обробки та захисту інформації. Це обумовлює відсутність у деяких аспектах цієї науки усталених термінів або їх різне тлумачення у літературі внаслідок зміни, розширення, узагальнення, уточнення вже усталених понять відповідно до нових наукових результатів. Саме це спонукає до необхідності уніфікації термінів в межах монографії перед тим, як перейти до її суті. У даній монографії інформація, представлена у вигляді даних, називається *повідомленням* і позначається M [13]. Процес перетворення повідомлення довільної довжини у дані сталої довжини називається *гешуванням* [11, 12, 67]. Дані сталої довжини, отримані внаслідок гешування повідомлення, називають *геш-значенням* цього повідомлення і позначаються h.

Нехай $\boldsymbol{\Theta}$ – множина всіх повідомлень, а $\boldsymbol{\Xi}$ – множина всіх геш-значень повідомлень, тоді відображення $\boldsymbol{\Theta} \to \boldsymbol{\Xi}$ називається *функцією гешування* або *геш-функцією* і позначається $hash(\cdot)$ [1]. Передбачається, що криптографічні геш-функції мають властивість однобічності, що спричинює практичну складність знаходження повідомлення M для відомого геш-значення цього повідомлення $h = hash(\mathrm{M})$ [10], в цьому випадку M називається *першим прообразом* геш-значення h [1, 8, 9, 11, 12]. *Другим прообразом* називається повідомлення M^* $(\mathrm{M}^* \neq \mathrm{M})$ таке, що його геш-значення $h^* = hash(\mathrm{M}^*)$ збігається із геш-значенням h $(h = h^*)$ заданого повідомлення M [1, 9, 8, 11, 12]. Ситуація, коли знайдено два різних повідомлення M та M^*, геш-значення h та h^* яких збігаються, називається *колізією* [1, 9, 11, 12, 37, 20].

На практиці для того, щоб забезпечити гешування повідомлень довільної довжини, вони розбиваються на частини, які називають *блоками даних* і позначають $m_i \ \left(i = \overline{1, l}\right)$, кожен з яких ітеративно обробляється під час гешування [1, 8, 9].

Математична модель визначення наступного проміжного геш-значення при ітеративному гешуванні називається *конструкцією гешування*, а результат обробки *i*-го блоку даних m_i називається *i*-м *проміжним геш-значенням* і позначається h_i [1, 9].

Часто проміжне геш-значення, отримане після обробки останнього *l*-го блоку даних, є геш-значенням всього повідомлення, тому його ще називають вихідним геш-значенням [1, 9, 12]. У такому випадку, множини проміжних геш-значень $\mathbf{H}$ та геш-значень повідомлень Ξ збігаються. Оскільки гешування передбачає отримання геш-значення фіксованої довжини, тому в цьому процесі використовуються *функції ущільнення*, які забезпечують фіксовану довжину проміжних геш-значень на кожній ітерації [1, 12, 13]. Якщо $\mathbf{D} = \{m_i\}$ – множина блоків даних, то функція ущільнення є відображенням $\mathbf{D} \times \mathbf{H} \to \mathbf{H}$.

Гешування, конструкція якого передбачає зміну параметрів перетворень від ітерації до ітерації у функції ущільнення залежно від значення певної змінної, називається *керованим*, а змінна, відповідно до значення якої визначаються ці параметри – *вектором керування* [12, 30, 31, 54, 65]. Відповідно гешування, конструкція якого передбачає використання лише однієї функції ущільнення протягом всього процесу, називається *некерованим* [12, 65].

Крім того, що на різних ітераціях гешування можуть використовуватись різні функції ущільнення, на одній ітерації може використовуватись декілька функцій ущільнення, що можуть паралельно обчислюватись протягом ітерації, тобто гешування може бути одноканальним та багатоканальним. *Каналом гешування* називається обчислювач, що реалізує функцію ущільнення на кожній з ітерацій [18]. При цьому канал може реалізовувати будь-яку функцію ущільнення з множини функцій ущільнення. При багатоканальному гешуванні проміжні геш-значення формуються залежно від результатів обчислень у кожному каналі або як сукупність цих результатів [1, 18].

1.2 Місце гешування в комп'ютерних системах

Функціонування сучасних комп'ютерних систем неможливе без розв'язання низки задач із захисту інформації. Зокрема це є задачі інформаційної безпеки стосовно захищеності даних (забезпечення конфіденційності, цілісності, вірогідності, юридичної значимості інформації), задачі інформаційної безпеки стосовно працездатності систем (забезпечення захисту від порушення функціонування, несанкціонованого доступу до інформаційних ресурсів, від неправомірних дій користувачів, персоналу) [2, 4, 8, 9, 16, 67–69] Ці задачі розв'язуються за допомогою методів та засобів комп'ютерної криптографії [2, 3, 7, 8, 13, 15, 16, 29, 70–73]. Причому ці методи та засоби повинні бути невід'ємними компонентами самих комп'ютерних систем [6, 12, 15, 69].

Одним із розділів комп'ютерної криптографії є гешування даних та його використання для розв'язання задач захисту інформації [2, 7, 8, 13, 23, 29, 41, 70–73]. Зокрема, методи гешування використовуються для розв'язання задач автентифікації та авторизації користувачів, автентифікації даних, перевірки цілісності даних, створення послідовностей псевдовипадкових чисел та генерування сеансових ключів [1–3, 5, 7, 8, 10–16, 29, 36, 41, 68–72].

Авторизація користувачів комп'ютерної системи та розмежування їх прав доступу до ресурсів системи забезпечує виконання політики інформаційної безпеки. Останнє дозволяє зменшити збої у роботі комп'ютерної системи та забезпечити цілісність, конфіденційність та вірогідність даних, що обробляються в системі [4, 7, 16, 69]. Один з найбільш розповсюджених методів забезпечення авторизації користувачів у комп'ютерних системах базується на використанні ними певного секретного пароля, що належить лише одному користувачу [4, 11, 68].

Здатність комп'ютерних систем опиратися атакам зловмисників на паролі визначається довжиною використовуваних паролів, їх унікальністю, множиною символів, що можуть використовуватися при введенні пароля [1, 2, 4, 7, 16]. Однак, навіть самий довгий пароль, який відповідає всім згаданим вище вимогам, не може забезпечити розмежування прав доступу, якщо зловмисник отримає доступ до бази паролів. Останнє не викликає труднощів у зловмисника, який має певну

кваліфікацію у галузі комп'ютерних систем та інформаційних технологій [1, 2, 4]. Для того, щоб протидіяти цій загрозі, у базі даних замість паролів зберігають їх геш-значення [5, 14, 68]. Завдяки тому, що для відомого геш-значення практично складно знайти повідомлення, якому відповідає це геш-значення, зловмисник, отримавши доступ до бази даних, не може порушити прав доступу [4, 14]. На рис. 1.1 наведено схему авторизації користувачів, що використовує збереження паролів користувачів у вигляді їх геш-значень [5, 68].

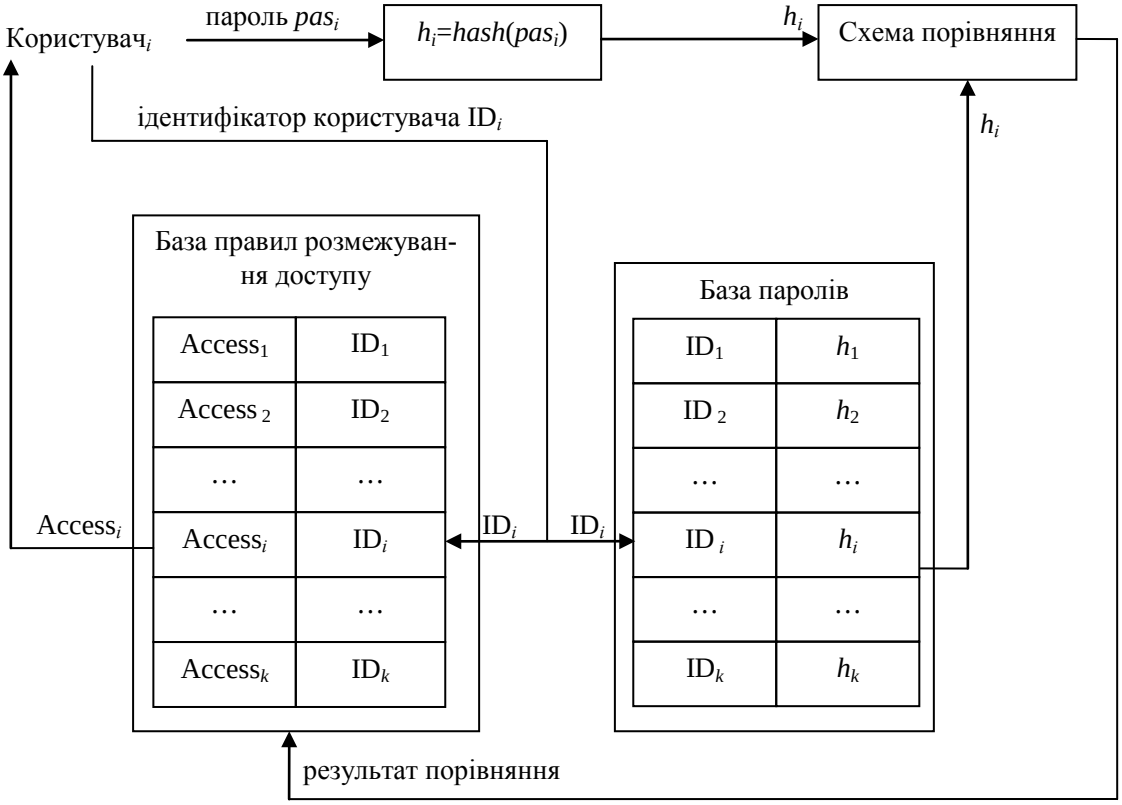

Рисунок 1.1 – Схема авторизації користувача з використанням паролів

Для забезпечення автентичності даних та надання інформації на електронних носіях юридичної значимості використовуються цифрові підписи, елементом яких є геш-значення даних, що підписуються [2, 69, 75]. Останнє пояснюється тим, що основою цифрового підпису є криптографія з відкритим ключем, одним з недоліків якої є тривалий час зашифрування/розшифрування, оскільки вона базується на операціях, які є неприродними для універсальних мікропроцесорів [2, 76]. Відповідно у задачах, коли підписуються великі файли, підписувати безпосередньо данні є недоцільним, натомість замість них підписують

їх геш-значення [2, 5, 7, 14, 15, 68-72]. Крім зменшення часу, що витрачається на підписування, використання геш-значень дозволяє підписувати секретні дані без їх розголошення, оскільки сторона, що підписує, має змогу підписати документ (повідомлення), залишаючи в секреті відомості, що зберігаються у ньому, а натомість можна опублікувати у спеціальній базі даних геш-значення цього документа [14] або взагалі не опубліковувати жодних відомостей щодо документа [5].

Узагальнений вигляд схеми цифрового підпису з використанням гешування наведено на рис. 1.2 [5, 15, 70].

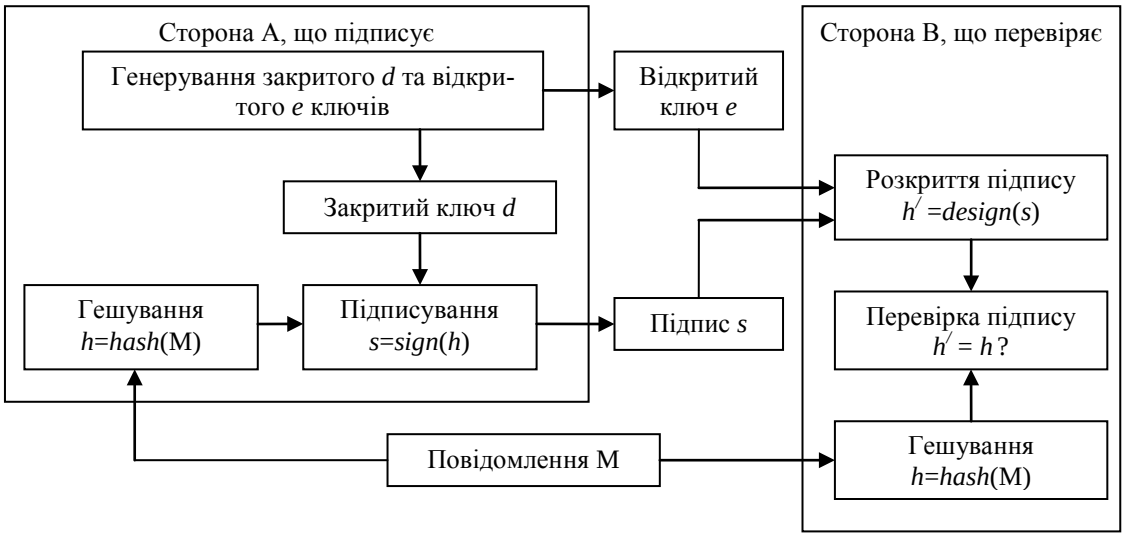

Рисунок 1.2 – Схема цифрового підпису

Під час функціонування комп'ютерних систем виникає задача забезпечення цілісності інформації, що зберігається та циркулює у них. Для розв'язання цієї задачі використовують коди автентичності повідомлення – MAC (message authentication code), та функції їх обчислення такого виду [1, 9, 14, 15, 68-75]:

$$\text{MAC} = f_{\text{MAC}}(k, \text{M}), \tag{1.1}$$

де $f_{\text{MAC}}(\cdot)$ – функція обчислення MAC;

k – ключ.

Один з підходів до генерування MAC базується на використанні ключових геш-функцій. Гешування відбувається для повідомлення, що створене внаслідок конкатенації ключа та повідомлення, що захи-

щається, відповідно воно може бути $k \| \mathrm{M}$, $\mathrm{M} \| k$ або $k \| \mathrm{M} \| k$ [15]. Для підвищення стійкості MAC (1.1) авторами [75] запропоновано спеціальний варіант гешування для обчислення MAC, що отримав назву HMAC, який передбачає гешування такого виду:

$$\mathrm{MAC} = hash\big(k \oplus a \,||\, hash\big(k \oplus b \,||\, M\big)\big), \qquad (1.2)$$

де a, b – константи.

Гешування геш-значення повідомлення в формулі (1.2) ускладнює зловмисникам аналіз впливу конкретних бітів повідомлення на вихідне геш-значення [75].

При передаванні даних комп'ютерною мережею часто виникає задача автентифікації користувачів для того, щоб сторони обміну мали змогу переконатися у тому, що зловмисник не видає себе за іншу авторизовану сторону [4, 16, 69]. Для розв'язання таких задач використовують криптографічні протоколи автентифікації, до яких зокрема відносяться і протоколи, що використовують гешування [7, 14, 68–72]. Такі протоколи автентифікації використовують випадкові (псевдовипадкові) числа або мітки часу для надання унікальності повідомленням, що пересилаються комп'ютерною мережею від одного користувача до іншого [7, 14].

Оскільки внаслідок гешування передбачається отримання геш-значень, що мають рівномірний закон розподілу, його часто використовують для генерування псевдовипадкових послідовностей [7, 12, 14, 70, 74]. Останні широко використовуються в комп'ютерних системах та комп'ютерній криптографії, зокрема у таких задачах: у протоколах автентифікації, у блокових шифрах та геш-функціях для покращення статистичних характеристик їх вихідних значень, для побудови потокових шифрів, як сеансові ключі для створення криптографічно захищених каналів у комп'ютерних мережах тощо [1, 2, 14, 15, 69–71].

Отже в комп'ютерних системах методи гешування використовуються для:

- авторизації користувачів;
- автентифікації користувачів;
- автентифікації даних;
- перевірки цілісності даних;
- генерування псевдовипадкових чисел.

Це свідчить про важливе місце процесу гешування у забезпеченні функціонування сучасних комп'ютерних систем.

1.3 Атаки на геш-функції

У літературі [1, 9, 13] використовують класифікацію атак на геш-функції, що зображена на рис. 1.3.

1.3.1 Атаки «грубої сили»

Атаки «грубої сили» полягають у повному переборі всіх можливих початкових значень векторів ініціалізації та/або варіантів повідомлень для пошуку прообразів та колізій. Розрізняють атаки «грубої сили» на пошук першого та другого прообразів, а також атаки на пошук колізій [9, 11, 12].

Рисунок 1.3 – Класифікація атак на криптографічні геш-функції

Атака «грубої сили» на пошук першого прообразу полягає в тому, що зловмисник, маючи задане вихідне геш-значення h довжиною n біт, гешує всі можливі варіанти вхідних повідомлень доти, доки не отримає таке, що має геш-значення h. Для реалізації такої атаки необхідно здійснити $O\left(2^n\right)$ обчислень геш-функції для знаходження зі значною ймовірністю першого прообразу [1, 9, 11, 12]. У разі реалізації атаки «грубої сили» для пошуку другого прообразу зловмиснику необхідно здійснити $O\left(2^n\right)$ обчислень геш-функції [1, 9, 11, 12].

Складність атаки «грубої сили» на пошук колізії базуються на парадоксі «дня народження», відповідно до якого з ймовірністю 0,5 ця атака реалізується при здійсненні $O\left(2^{n/2}\right)$ обчислень геш-значень [9, 20, 37, 77]. Саме тому цю атаку ще називають атакою «дня народження» [1, 9, 11, 12, 20, 37].

Наведені оцінки складності свідчать про те, що реалізація цих атак вимагає настільки значних витрат часу, що говорять про практичну нездійсненність атак «грубої сили» і вважають геш-функцію стійкою, якщо вона зламується лише за допомогою атак «грубої сили» [1, 9].

Складність реалізації атак «грубої сили» залежить від довжини геш-значень n та не залежить від особливостей реалізації геш-функції. Однак, у низці випадків криптоаналіз дозволяє зменшити кількість обчислень геш-значень для знаходження прообразів та колізій. Очевидно, що атаки, які використовують криптоаналіз, мають сенс, якщо складність їх реалізації є меншою за складність реалізації атак за допомогою атак «грубої сили».

Криптоаналітичні атаки поділяються відповідно до їх застосовності до геш-функцій на загальні атаки та на спеціальні атаки [1, 9].

1.3.2 Загальні атаки

Загальні атаки – це атаки, що використовують властивості конструкцій гешування [9, 19]. Складність реалізації цих атак не залежить від способу реалізації геш-функції, а тому вони можуть бути використані для всіх геш-функцій певної конструкції [9, 19, 78, 79]. Найпоширенішими на сьогоднішній день [14] є геш-функції конструкції Меркля–Дамгаарда та її модифікацій, тому в деякій літературі [9] розглядаються лише атаки на цей вид конструкцій. Водночас геш-функції

конструкцій, відмінних від конструкції Меркля–Дамгаарда, також вразливі до загальних атак [21, 42].

Однією з перших загальних атак стала атака збільшення довжини повідомлення [1], яка обумовлюється ітеративністю процесу гешування та полягає у дописуванні до повідомлення блоків даних доти, доки геш-значення початкового та доповненого повідомлень не збігаються. Причому для здійснення такої атаки необов'язково знати початкове повідомлення, навіть у випадку, коли використовується підсилення Меркля–Дамгаарда, зловмиснику достатньо знати довжину повідомлення для того, щоб до неї згодом додати довжину доповнення до повідомлення [9, 18].

Атаки, що використовують мультиколізії, спрямовані, в першу чергу, на багатоканальні методи гешування – методи, що передбачають паралельне обчислення частин геш-значення та їх об'єднання після завершення гешування [1, 9]. Прикладом такого багатоканального гешування є каскадування, запропоноване Пренілом [1]. Однак атаки з використанням мультиколізій можуть бути застосовані й до одноканального гешування [21, 22].

Першою атакою, що використовувала мультиколізії, стала атака Жу (Joux) [20, 37, 42]. Об'єктом такої атаки став метод каскадування [1]. Розглянемо принцип цієї атаки на прикладі методу каскадування для двох каналів, тобто коли блоки даних паралельно подаються на вхід двох різних функцій ущільнення (або однакових з різним початковим заповненням), а вихідне геш-значення h довжиною n бітів отримується шляхом конкатенації проміжних геш-значень $h^{(1)}$ та $h^{(2)}$, отриманих у першому та другому каналах відповідно, причому їх довжина становить $n/2$ бітів [1]:

$$\begin{cases} h_i^{(1)} = f^{(1)}\left(h_{i-1}^{(1)}, m_i\right); \\ h_i^{(2)} = f^{(2)}\left(h_{i-1}^{(2)}, m_i\right), \end{cases} \tag{1.3}$$

де $f^{(1)}(\cdot), f^{(2)}(\cdot)$ – функції ущільнення, що використовуються в першому та другому каналах відповідно;

$h_i^{(1)}, h_i^{(2)}$ – проміжні геш-значення, отримані в першому та другому каналах відповідно після завершення i-ї ітерації;

m_i – i-й блок даних.

Атака Жу передбачає знаходження мультиколізій в одному з каналів у формулі (1.3) шляхом пошуку для кожного блоку даних m_i іншого блоку m_i^* такого, що виконується така рівність [9, 20, 42]:

$$h_i^{(1)} = f^{(1)}\left(h_{i-1}^{(1)}, m_i\right) = f^{(1)}\left(h_{i-1}^{(1)}, m_i^*\right). \qquad (1.4)$$

За формулою (1.4) для повідомлення довжиною l блоків даних будується 2^l таких колізій за $l \cdot 2^{n/2}$ ітерацій гешування (рис. 1.4) [20, 42].

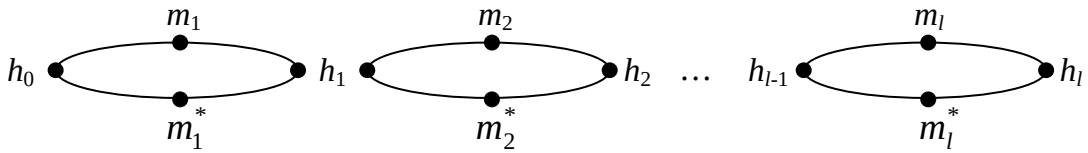

Рисунок 1.4 – Вигляд мультиколізії, отриманої Жу

Цю сукупність колізій Жу назвав мультиколізією [9, 20, 37, 42]. Існує ймовірність, що серед знайдених колізій для першого каналу знайдеться хоча б одна, яка породжує колізію й в іншому каналі [9, 18, 20, 37, 42]. Причому цією ймовірністю не можна знехтувати і вона зростає зі збільшенням l [9, 20, 37, 42].

Після появи атаки Жу, закладені ним підходи були узагальнені та розвинуті в багатьох напрямах [9, 22, 37, 42, 78-80]. Так у роботі [22] набули подальшого розвитку результати Жу для пошуку колізій у блоках даних різної довжини та наведено метод пошуку другого прообразу, у роботі [42] наводиться приклад для побудови мультиколізій для деревоподібних конструкцій гешування та для конструкцій, які передбачають перестановку блоків даних для кожного з каналів гешування.

У роботі [78] розглядають пошук колізій за допомогою знаходження певних аналогів фіксованих точок. Ця атака передбачає послідовне гешування $n/2$ блоків даних та пошук серед них колізії [78]. В подальшому частина ланцюга між блоками даних, що спричиняють колізію, використовується для побудови l колізій, шляхом дописування до повідомлень префікса з l повторень цього ланцюга [78]. Крім того, у роботі [78] запропоновано методику для генерування довільної кількості колізій рівної довжини. Наразі така атака не може

бути реалізована, оскільки для знаходження описаних вище колізій необхідно будувати надзвичайно довгі повідомлення ($O(2^{128})$ блоків даних), а відтак необхідні надзвичайно великі обсяги пам'яті для їх зберігання.

Атака Нострадамуса була запропонована Келсі та Коно (тому вона ще відома, як атака Келсі–Коно) в роботі [21]. Ця атака передбачає попередню підготовку («випасання геш-функцій»), яка включає обчислення різних геш-значень, що групуються у вигляді діамантової структури, приклад якої наведено на рис. 1.5 [21].

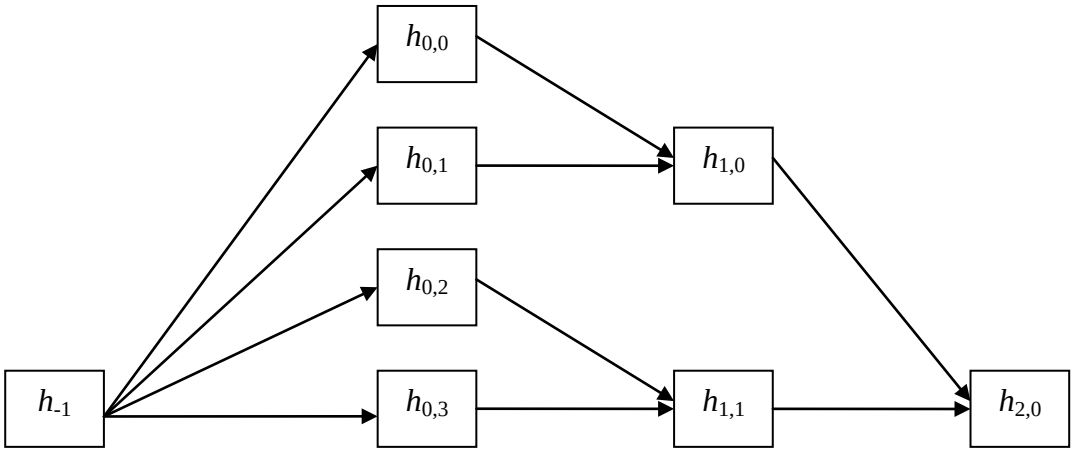

Рисунок 1.5 – Діамантова структура геш-значень

Діамантова структура в подальшому використовується для побудови прообразів. Чим більша діамантова структура, тим менший час витрачається на реалізацію атаки безпосередньо під час обміну даними в комп'ютерній системі. В роботі [21] розглядаються комбінації такої атаки з деякими іншими загальними атаками, зокрема використання «видовжених» діамантових структур, в яких при переході від нульового найширшого рівня до наступних рівнів кількість проміжних геш-значень не зменшується, що збільшує кількість комбінацій вхідних даних, які можуть подаватися на вхід геш-функції.

Атака «зустріч по середині» базується на атаці «дня народження» [1], але замість пошуку повідомлень, що мають однакові вихідні геш-значення, тут шукають повідомлення, які мають однакові проміжні геш-значення, що дозволяє зловмисникам отримувати наперед задане геш-значення.

Крім групи загальних атак на геш-функції конструкції Меркля–Дамгаарда, існують загальні атаки й на геш-функції інших конструкцій. Відомі атаки на конструкції гешування з лінійним зв'язком блоків даних, наприклад на конструкції 3C та конструкцію геш-функції ГОСТ [9, 35, 48]. До атак цього типу відносяться атаки Міронова–Нараянана та Гаураварама–Келсі [35]. Атака Міронова–Нараянана досягає цього шляхом подвоєння блоків даних, які спричиняють колізію для проміжних геш-значень, що дозволяє нівелювати вміст підмінених блоків даних у контрольній сумі блоків даних [35]. Атака Гаураварама–Келсі передбачає, що мультиколізії визначаються аналогічно методу, запропонованому Жу, проте крім цього виконується розв'язання системи рівнянь для вибору повідомлень, контрольна сума яких збігається [35]. Використовуючи такий підхід, у роботі [35] наводять приклади реалізації аналогу атаки Нострадамуса для конструкцій з лінійним зв'язком блоків даних.

Таким чином серед загальних атак на геш-функції найбільш небезпечними є атаки, що використовують мультиколізії, оскільки вони дозволяють отримувати показниковий приріст кількості колізій з лінійним зростанням складності реалізації цих атак.

1.3.3 Спеціальні атаки

Спеціальні атаки використовують властивості функцій ущільнення, які є складовими геш-функцій. Переважно ці атаки стають можливими в результаті їх лінійного або диференційного криптоаналізу [34, 81–84].

Лінійний криптоаналіз розроблявся для SP-мереж (мереж підстановки та перестановки) та мав на меті виявити залежність між вхідними та вихідними значеннями [81, 82]. Аналізувалась парність суми різних комбінацій вхідних та вихідних бітів та визначались ті комбінації, які не відповідають рівномірному закону розподілу [81, 82]. Використовуючи найбільші відхилення, зловмисник визначає деякі біти раундових підключів, тим самим зменшуючи кількість варіантів, що необхідно перебирати за допомогою атак «грубої сили» [81, 82].

Перші основи диференціального криптоаналізу були закладені для шифрів і передбачали аналіз різниць, диференціалів, між блоками даних у відкритому варіанті та у зашифрованому [34]. Суть цього аналізу полягає в тому, що для вхідних блоків даних X' та X'' визначають

різницю $\Delta X = X' \oplus X''$ [34, 81-84]. Нехай за результатами раундових перетворень X' та X'' отримано зашифровані Y' та Y'' відповідно. Визначають різницю у шифротекстах, що відповідає різниці відкритих повідомлень $\Delta Y = Y' \oplus Y''$. Аналізуючи набори $\{\Delta X; \Delta Y\}$, визначають ті комбінації, у яких при сталій різниці між блоками даних, ймовірність виникнення певної різниці у зашифрованих аналогах відрізняється від $1/2^n$, де n – довжина вхідних та зашифрованих блоків даних [34, 81, 82, 84].

Специфіка спеціалізованих атак полягає в тому, що зловмисник може починати виконувати аналіз SP-мережі, знаючи лише її структуру і не знаючи вхідного та вихідного текстів. Аналіз відбувається шляхом повного перебору всіх вхідних даних та визначення вихідних значень, що відповідають ним. У зв'язку з тим, що загальні атаки можуть бути застосовані до всіх геш-функцій певної конструкції, відповідно вони є більш небезпечними, ніж спеціальні, оскільки збільшити стійкість до останніх можна, вносячи певні зміни у перетворення, що відбуваються у функції ущільнення або шляхом використання іншої функції ущільнення. Протидіяти ж загальним атакам можливо лише після виявлення недоліків в конструкціях гешування та пропозиції нових конструкцій.

Аналіз розвитку методів криптографічного гешування свідчить про те, що нові конструкції гешування з'являються суттєво рідше, ніж нові геш-функції [9, 44, 45, 65]. Тому виникає задача створення конструкцій гешування, які б забезпечували стійкість до цих атак.

1.4 Конструкції гешування

Проблеми, пов'язані з процесом гешування, почали розглядатися криптографією порівняно нещодавно [71]. Однією з найбільш актуальних серед них є пошук найкращого рішення відповідно критерію швидкість/стійкість [1, 3, 12]. Особливо це стає очевидним, коли брати до уваги, що відомі криптографічні перетворення, які покликані забезпечувати стійкість алгоритму гешування, виконуються протягом тривалого часу з використанням сучасних апаратних засобів [3, 12, 24, 25, 71]. Відповідно вчені почали шукати шляхи підвищення стійкості не лише у перетвореннях, які відбуваються під час гешування над да-

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Preneel B. Analysis and Design of Cryptographic Hash Functions. PhD thesis / Bart Preneel. – Katholieke Universiteit Leuven, 1993. – 338 с. – Режим доступу до дисертаційної роботи: http://homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf.

2. Бернет С. Криптография. Официальное руководство RSA Security. / С. Бернет, С. Пэйн. – М. : Бином-Пресс, 2002. – 384 с.

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

4. Бойцев О. М. Защити свой компьютер на 100 % от вирусов и хакеров / О. М. Бойцев. – СПб. : Питер, 2008. – 288 с.

5. Защита информации в телекоммуникационных системах / Г. Ф. Конахович, В. П. Климчук, С. М. Паук, В. Г. Потапов. – К. : МК-Пресс, 2005. – 288 с.

6. Грайворонський М. В. Безпека інформаційно-комунікаційних систем. / М. В. Грайворонський, О. М. Новіков. – К. : BHV, 2009. – 608 с.

7. Романец Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин ; под ред. В. Ф. Шаньгина. – 2-е изд., перераб. и доп. – М. : Радио и связь, 2001. – 376 с.

8. Dean R. D. Formal Aspects of Mobile Code Security. PhD Dissertation / Richard Drews Dean. – Princeton University, 1999. – 164 с. – Режим доступу до дисертаційної роботи: http://cyphunk.files.wordpress.com/2006/02/ddean-thesis.pdf.

9. Gauravaram P. Cryptographic Hash Functions: Cryptanalysis, Design and Applications. Thesis submitted in accordance with the regulations for Degree of Doctor of Philosophy / Praveen Gauravaram. – 2009. – 298 с. – Режим доступу до дисертаційної роботи: http://eprints.qut.edu.au/16372/1/Praveen_Gauravaram_Thesis.pdf.

10. Введение в криптографию. / Под ред. В. В. Ященко. – М. : МЦНМО, 2000. – 288 с.

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

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

13. Indesteege S. Analysis and Design of Cryptographic Hash Functionhttps://www.cosic.esat.kuleuven.be/publications/thesis-171.pdf, PhD thesis / Sebastiaan Indesteege. – Katholieke Universiteit Leuven, 2010. – 332 с. – Режим доступу до дисертаційної роботи: https://www.cosic.esat.kuleuven.be/publications/ thesis-171.pdf.

14. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Брюс Шнайер. – М. : Триумф, 2002. – 816 с.

15. Фергюсон Н. Практическая криптография. : Пер. с англ. / Нильс Фергюсон, Брюс Шнайер. – М. : Вильямс, 2005. – 424 с.

16. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире / Брюс Шнайер. – СПб. : Питер, 2003. – 368 с.

17. Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA–3) Family. / Department of Commerce. National Institute of Standards and Technology. / Federal Register. – Vol. 72, № 212. – 2007. – С. 62212–62220. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf.

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

19. Biham E. A Framework for Iterative Hash Functions: HAIFA. / Eli Biham, Orr Dunkelman // NIST second cryptographic hash workshop. – 2006. – 9 с. – Режим доступу до статті: http://csrc.nist.gov/ groups/ST/hash/documents/DUNKELMAN_NIST3.pdf.

20. Joux A. Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions / Antoine Joux // Cryptology ePrint Archive. – 2004. – 11 с. – Режим доступу до статті: http://www.iacr.org/cryptodb/ archive/2004/CRYPTO/ 1472/1472.pdf.

21. 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.

22. Kelsey J. Second preimages on n-bit Hash Function for Less than 2n Work / John Kelsey, Bruce Schneier // Cryptology ePrint Archive. – 2004. – 15 с. – Режим доступу до статті: http://eprint.iacr.org/2004/304.pdf.

23. Самофалов К. Г. Організація паралельного обчислення хеш-сигнатур для систем забезпечення цілісності інформації в комп'ютерних мережах / К. Г. Самофалов, О. П. Марковський, Мустафа Акрам Ареф Найеф // Наукові вісті НТУУ «КПІ». – 2002. – № 4. – С.31–38.

24. Патент України на корисну модель № 55211 МПК H 04 L 9/06. Конвеєрний криптографічний обчислювач. / Корченко О. Г., Паціра Є. В., Панасюк А. Л., Гнатюк С. О., Кінзерявий В. М. ; заявник та патентовласник Національний авіаційний університет. – №u201006041 ; заявл. 19.05.10 ; опубл. 10.12.10, Бюл. № 23.

25. Патент України на корисну модель № 55213 МПК H 04 L 9/06. Конвеєрний криптографічний обчислювач. / Корченко О. Г., Паціра Є. В., Панасюк А. Л., Гнатюк С. О., Кінзерявий В. М. ; заявник та патентовласник Національний авіаційний університет. – №u201006044 ; заявл. 19.05.10 ; опубл. 10.12.10, Бюл. №23.

26. Патент України на корисну модель № 18693 МПК G 09 C 1/00. Спосіб ключового хешування теоретично доведеної стійкості / Стасєв Ю. В., Кузнєцов О. О., Євсєєв С. П., Чевардін В. Є., Малахов С. В., Гришко А. В. ; заявник та патентовласник Харківський університет повітряних сил. – №u200605734 ; заявл. 25.05.06 ; опубл. 15.11.06, Бюл. №11.

27. Белецкий А. Я. Модифицированный матричный ассиметричный криптографический алгоритм Диффи–Хеллмана. / А. Я. Белецкий, А. А. Белецкий, Д. А. Стеценко // Штучний інтелект. – 2010. – № 3. – С. 697–705.

28. Белецкий А. Я. Синтез и анализ гарантированно невырожденных шифрующих (0, 1)-матриц высокого порядка / Анатолий Яковлевич Белецкий // Інформаційні технології та комп’ютерна інженерія : Тези доповідей Міжнародної науково-практичної конференції,

м. Вінниця, 19-21 травня 2010 року. – Вінниця : ВНТУ, 2010. – С. 183–184.

29. Кузнецов О. О. Захист інформації в інформаційних системах. Методи традиційної криптографії : навчальний посібник / О. О. Кузнецов, С. П. Євсєєв, О. Г. Король. – Харків : Вид. ХНЕУ, 2010. – 316 с.

30. Рудницький В. М. Модель уніфікованого пристрою криптографічного перетворення інформації / В. М. Рудницький, В. Г. Бабенко // Системи обробки інформації. – 2009. – № 3. – С. 91-95.

31. Рудницький В. М. Синтез математичних моделей пристроїв декодування інформації для криптографічних систем / В. М. Рудницький, В. Г. Бабенко // Системи обробки інформації. – 2010. – № 2. – С. 124–128.

32. Савчук М. Анализ одного метода тестирования случайных последовательностей, основанного на контекстном моделировании / М. Савчук, В. Шарапов // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2008. – № 1 (16). – С. 82–89.

33. Савчук М. Роздільні статистики в схемах розміщення та тести для бінарних послідовностей / Михайло Савчук, Олександр Валецький // ІНТЕРНЕТ-ОСВІТА-НАУКА-2010, сьома міжнародна конференція ІОН-2010, 28 вересня-3 жовтня, 2010 : збірник матеріалів конференції. – Вінниця : ВНТУ, 2010. – С. 389–392.

34. Biham E. Differential Cryptanalysis of DES-like Cryptosystems / Eli Biham, Adi Shamir. – 1990. – 106 с. – Режим доступу до ресурсу: http://sota.gen.nz/crypt_blues/biham91differential.pdf.

35. Gauravaram P. Cryptanalysis of a class of cryptographic hash functions. / Praveen Gauravaram, John Kelsey // Cryptology ePrint Archive. – 2007. – 30 с. – Режим доступу до статті: http://eprint.iacr.org/2007/277.pdf.

36. Fiat A. How to Prove Yourself: Practical Solutions to Identification and Signature problems / Amos Fiat, Adi Shamir. – 1998. – 9 с. – Режим доступу до статті: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.8796&rep=rep1&type=pdf.

37. Joux A. Improved generic algorithms for 3-collisions / Antoine Joux, Stefan Lucks. // Cryptology ePrint Archive. – 2009. – 16 с. – Режим доступу до статті: http://eprint.iacr.org/2009/305.pdf.

38. Fleischmann E. On the Security of Tandem-DM / Ewan Fleischmann, Michael Gorski, Stefan Lucks // Cryptology ePrint Archive. – 2009. – 17 с. – Режим доступу до статті: http://eprint.iacr.org/2009/054.pdf.

39. Ferguson N. Attacks on AURORA-512 and the Double-Mix Merkle-Damgard Transform / Niels Ferguson, Stefan Lucks // Cryptology ePrint Archive. – 2009. – 6 с. – Режим доступу до статті: http://eprint.iacr.org/2009/113.pdf.

40. Weis R. Cryptographic Hash Functions. Resent Results on Cryptanalysis and their Implications on System Security / R. Weis, S. Lucks. – 2006. – 19 с. – Режим доступу до статті: http://www.sane.nl/sane2006/program/final-papers/R10.pdf.

41. Merkle R. C. Secrecy, Authentifications and Public Key Systems. A Dissertation Submitted to the Department of Electrical Engineering and the Committee on Graduate Studies of Stanford University in Particial Fulfillment of the Requirements. / Ralph Charles Merkle. – 1979. – 182 с. – Режим доступу до дисертаційної роботи: http://www.merkle.com/papers/Thesis1979.pdf.

42. Hoch J. J. Breaking the ICE – Finding Multicollisions in Iterative Concatenated and Expanded (ICE) Hash Functions / Jonathan J. Hoch, Adi Shamir. – 2006. – 13. – Режим доступу до ресурсу: http://wisdom.weizmann.ac.il/~yaakovh/papers/hashpaper_submission.pdf.

43. Keccak sponge function family main document. Version 2.0 – September 10, 2009. / G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. – 2009. – 100 с. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Keccak_Round2.zip\Keccak\Supporting_Documentation\Keccak-main-2.0.pdf.

44. Sponge functions / G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. – 2007. – 22 с. – Режим доступу до статті: http://sponge.noekeon.org/SpongeFunctions.pdf.

45. On the Indifferentiability of the Sponge Construction / G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. – 2008. – 17 с. – Ре-

жим доступу до статті: http://sponge.noekeon.org/ SpongeIndifferentiability.pdf.

46. Biham E. The SHAvite-3 Hash Function: Tweaked Version. / Eli Biham, Orr Dunkelman. – 2009. – 41 с. – Режим доступу до джерела: http://csrc.nist.gov/ groups/ST/hash/sha-3/Round2/documents/SHAvite-3_Round2.zip\SHAvite-3\ Supporting_Documentation\Shavite.pdf.

47. The Skein Hash Function Family. Version 1.2 / Niels Ferguson, Stefan Lucks, Bruce Schneier and others. – 2009. – 84 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/ Skein_Round2.zip\Skein\Supporting_Documentation\skein1.2.pdf.

48. Горбенко І. Д. Стан створення та напрями досліджень і розробок зі створення перспективних стандартів гешування / І. Д. Горбенко, А. О. Бойко, А. М. Герцог // Радіоелектронні і комп'ютерні системи. – 2010. – № 5. – С. 67–74.

49. Задирака В. К. Использование арифметики многоразрядных чисел в современных компьютерных технологиях решения задач трансвычислительной сложности / В. К. Задирака, А. Н. Терещенко // Штучний інтелект. – 2010. – № 3. – С. 712–728.

50. Марковский А. П. Об одном подходе к определению сложности случайных и псевдослучайных двоичных последовательностей / А. В. Марковский, Мустафа Акрам Ареф Найеф, А. В. Бойко // Вісник Національного технічного університету України «КПІ». Інформатика, управління та обчислювальна техніка. – 2002. – № 37. – С. 120–129.

51. Мустафа Акрам Ареф Найеф. Статистическая оценка нелинейности и лавинного эффекта булевых функций / Мустафа Акрам Ареф Найеф // Вісник Національного технічного університету України «КПІ». Інформатика, управління та обчислювальна техніка. – 2002. – № 38.– С. 106–113.

52. Лужецький В. А. Підходи до побудови швидких алгоритмів хешування / В. А. Лужецький, Ю. В. Баришев // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – №. 2 (19). – 2009. – С. 57-65.

53. Лужецький В. А. Криптографічні примітиви для реалізації керованого хешування / В. А. Лужецький, Ю. В. Баришев // Вісник Вінницького політехнічного інституту.– 2011. – № 1. – С. 108–111.

54. Лужецький В. А. Методи формування векторів керування для керованого багатоканального хешування даних. / В. А. Лужецький, Ю. В. Баришев, О. В. Оводенко. // Наукові праці Донецького національного технічного університету. Серія: «Обчислювальна техніка та автоматизація». – 2011. – №21 (183). – С. 103–108. – Режим доступу до статті: http://www.nbuv.gov.ua/portal/natural/Npdntu_ota/2011_21/article_21_15.pdf.

55. Лужецький В. А. Апаратні засоби для реалізації багатоканального керованого хешування. / В. А. Лужецький, Ю. В. Баришев. // Системи обробки інформації.– 2011. – №3. – С. 130–133.

56. Лужецький В. А. Методи та засоби паралельного керованого хешування / В. А. Лужецький, Ю. В. Баришев // Наукові праці ВНТУ. – 2011. – № 2. – 5 с. – Режим доступу до статті: http://www.nbuv.gov.ua /e-journals/VNTU/2011_2/2011-2.files/uk/11valpch_ua.pdf.

57. Лужецький В. А. Узагальнена модель стійкого паралельного хешування / В. А. Лужецький, Ю. В. Баришев // Проблеми й перспективи розвитку ІТ-індустрії в Україні : матеріали І Міжнародної науково-технічної конференції. – Харків : ХНЕУ, 2009. – С. 166–167.

58. Лужецький В. А. Багатоканальне кероване хешування даних./ В. А. Лужецький, Ю. В. Баришев // Обробка сигналів і негауссівських процесів : праці ІІІ Міжнародної науково-практичної конференції, присвяченої пам’яті професора Ю. П. Кунченка. – Черкаси : ЧДТУ, 2011. – С. 204–206.

59. Патент України на корисну модель № 41313 МПК G 09 C 1/00. Спосіб паралельного ключового хешування теоретично доведеної стійкості / Лужецький В. А., Баришев Ю. В., Дмитришин О. В. ; заявник та патентовласник Вінницький національний технічний університет. – №u200900489 ; заявл. 23.01.09 ; опубл. 12.05.09, Бюл. № 9.

60. Патент України на корисну модель № 48279 МПК G 09 C 1/00. Спосіб паралельного ключового хешування / Лужецький В. А., Баришев Ю. В. ; заявник та патентовласник Вінницький національний технічний університет. – №u200909901 ; заявл. 28.09.09 ; опубл. 10.03.10, Бюл. № 5.

61. Патент України на корисну модель № 54813 МПК G 09 C 1/00. Спосіб паралельного ключового хешування / Лужецький В. А.,

Баришев Ю. В. ; заявник та патентовласник Вінницький національний технічний університет. – №u201006156 ; заявл. 21.05.10; опубл. 25.11.10, Бюл. № 22.

62. Баришев Ю. Алгоритм паралельного хешування даних / Ю. Баришев // Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування : тези IV Міжнародної науково-технічної конференції. Частина 1. – Вінниця : ВНТУ, 2009. – С. 9

63. Баришев Ю. В. Методи побудови швидких алгоритмів хешування. / Ю. В. Баришев // Методи та засоби кодування, захисту й ущільнення інформації. Тези доповідей другої Міжнародної науково-практичної конференції. – Вінниця : ВНТУ, 2009. – С. 138–139.

64. Баришев Ю. В. Методи та програмні засоби керованого багатоканального хешування. / Ю. В. Баришев // Методи та засоби кодування, захисту й ущільнення інформації : тези доповідей III Міжнародної науково-практичної конференції. – Вінниця : ВНТУ, 2011. – С. 100–101.

65. Баришев Ю. В. Підхід до хешування, що стійке до аналізу зловмисника. / Ю. В. Баришев // Системи обробки інформації – 2010. – № 3(84). – С. 99–100.

66. Баришев Ю. Структура спеціалізованого криптографічного процесора для керованого хешування / Юрій Баришев // Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування : матеріали V Міжнародної науково-технічної конференції. – Вінниця : ВНТУ, 2011. – С. 169–170.

67. Андерсон Д. А. Дискретная математика и комбинаторика / Джеймс А. Андерсон : пер. с англ. М. М. Беловой. – М. : Вильямс, 2004. – 960 с.

68. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам : учебное пособие для вузов. / А. А. Афанасьев, Л. Т. Веденьев, А. А. Воронцов и др. ; под ред. А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. – М. : Горячая линия-Телеком, 2009. – 552 с.

69. Информационная безопасность открытых систем : учебник для вузов : в 2-х томах. Том 2 : Средства защиты в сетях / С. В. Запеч-

ников, Н. Г. Милославская, А. И. Толстой, Д. В. Ушаков. – М. : Горячая линия-Телеком, 2008. – 558 с.

70. Рябко Б. Я. Криптографические методы защиты информации : учебное пособие для вузов. / Б. Я. Рябко, А. Н. Фионов. – М. : Горячая линия-Телеком, 2005. – 229 с.

71. Смарт Н. Криптография / Н. Смарт. – М. : Техносфера, 2005. – 528 с.

72. Щербаков Л. Ю. Прикладная криптография. Использование и синтез криптографических интерфейсов / Л. Ю. Щербаков, А. В. Домашев. – М. : Русская Редакция, 2003. – 416 с.

73. Богуш В. М. Інформаційна безпека держави. / В. М. Богуш, О. К. Юдін. – К. : МК-Прес, 2005. – 432 с.

74. Фомичев В. Ф. Дискретная математика и криптология : курс лекций / В. Ф. Фомичев : под общ. ред. д-ра физ.-мат. н. Н. Д. Подуфалова. – М. : Диалог-МИФИ, 2003. – 400 с.

75. Bellare M. Keying Hash Functions for Message Authentication / M. Bellare, R. Canetti, H. Krawczyk. – 1996. – 19 с. – Режим доступу до статті: http://cseweb.ucsd.edu/users/mihir/papers/kmd5.pdf.

76. Магда Ю. С. Ассемблер для процессоров Intel Pentium. / Ю. С. Магда. – СПб. : Питер, 2006. – 410 с.

77. Ендовицкий П. Точная асимптотическая оценка размера группы в обобщении парадокса дней рождения / Павел Ендовицкий. // ІНТЕРНЕТ-ОСВІТА-НАУКА-2010 : Збірник матеріалів VII міжнародної конференції. – Вінниця : ВНТУ, 2010. – С. 400–403.

78. Halunen K. An Automata-Theoretic Interpretation of Iterated Hash Functions - Application to Multicollisions / K. Halunen, J. Kortelainen, T. Kortelainen // Cryptology ePrint Archive. – 2009. – 13 с. – Режим доступу до статті: http://eprint.iacr.org/2009/456.pdf.

79. Klima V. Generic collision attacks on narrow-pipe hash functions faster than birthday paradox, applicable to MDx, SHA-1, SHA-2, and SHA-3 narrow-pipe candidates / Vlastimil Klima, Danilo Gligoroski // Cryptology ePrint Archive. – 2010. – 4 с. – Режим доступу до статті: http://eprint.iacr.org/2010/430.pdf.

80. Nandi M. Multicollision Attacks on Some Generalized Sequential Hash Functions / M. Nandi, D. R. Stinson. – 2006. – 15 с. – Режим до-

ступу до статті: http://citeseerx.ist.psu.edu/viewdoc/download?doi =10.1.1.104.5370&rep=rep1&type=pdf.

81. Heys H. M. A Tutorial on Linear and Differential Cryptanalysis / Howard M. Heys. – 2002. – 33 с. – Режим доступу до статті: http://www2.itu.edu.tr/~orssi/dersler/cryptography/ldc_tutorial.pdf.

82. Jakobsen B. T. Linear and Differential Cryptanalysis / B. T. Jakobsen, M. Abyar, P. S. Nordholt. – 2006. – 22 с. – Режим доступу до статті: http://www.daimi.au.dk/~ivan/LinDifAnalyse.pdf.

83. McDonald C. Differential Path for SHA-1 with complexity O(2^{52}) / C. McDonald, P. Hawkes, J Pieprzyk // Cryptology ePrint Archive. – 2009. – 14 с. – Режим доступу до статті: http://eprint.iacr.org/2009/ 259.pdf.

84. Albrecht M. Algebraic Techniques in Differential Cryptanalysis / Martin Albrecht, Carlos Cid // Cryptology ePrint Archive. – 2008. – 17 с. – Режим доступу до статті: http://eprint.iacr.org/2008/177.pdf.

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

86. NISTIR 7620. Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition / Andrew Regenscheid, Ray Perlner, Shu-jen Chang та інші. – 2010. – 21 с. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/ sha3_NISTIR7620.pdf.

87. Hirose S. Some Plausible Constructions of Double-Block-Length Hash Functions / Shoichi Hirose. – 2006. – 13 с. – Режим доступу до статті: http://www.iacr.org/archive/fse2006/40470213/40470213.pdf.

88. Shabal, a Submission to NIST’s Cryptographic Hash Algorithm Competition / Initiated by the Saphir project. – 2008. – 300 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/ Round2/documents/Shabal_Round2.zip\ShabalSupporting_Documentation\ description.pdf.

89. SHA-3 proposal BLAKE / Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan. – 2008. – 76 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-

3/Round2/documents/ LAKE_Round2.zip\blake_aumasson\Supporting_ Documentation\blake.pdf.

90. Provably Secure FFT Hashing / V. Lyubashevsky, D. Micciancio, C. Peikert, A. Rosen. – 2006. – 12 с. – Режим доступу до статті: http://www.eecs.harvard.edu/~alon/PAPERS/lattices/description.pdf.

91. Lyubashevsky V. Towards Practical Lattice-Based Cryptography. A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy / Vadim Lyubashevsky. – 2008. – 75 с. – Режим доступу до дисертаційної роботи: http://www.eecs.harvard.edu/~alon/PAPERS/lattices/description.pdf.

92. Бассар Ж. Современная криптология / Ж. Бассар. – М. : ПОЛИМЕД, 1999. – 176 с.

93. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. – М. : МЦНМО, 2003. – 328 с.

94. Xu Z. Dynamic SHA / Zijie Xu // Cryptology ePrint Archive. – 2007. – 34 с. – Режим доступу до статті: – http://eprint.iacr.org/2007/476.pdf.

95. Cryptanalysis of Dynamic SHA(2) / J-P. Aumasson, O. Dunkelman, S. Indesteege and B. Preneel // COSIC publications, 2009. – 18 с. – Режим доступу до ресурсу: https://www.cosic.esat.kuleuven.be/publications/article-1277.pdf.

96. Kucuk O. The Hash Function Hamsi. / Ozgul Kucuk. – 2009. – 19 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Hamsi_Round2.zip\Hamsi\Supporting_ Documentation\ Hamsi_Spec_2ndRound.pdf.

97. SHA-3 Proposal: ECHO / Ryad Benadjila, Olivier Billet, Henri Gilbert [та ін.]. – 2008. – 47 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/ECHO_ Round2.zip\ECHO\Supporting_Documentation\echo_description.pdf.

98. Cryptographic Hash Function Blue Midnight Wish / Danilo Gligoroski, Vlastimil Klima, Svein Johan Knapskog [та ін.] – 2009. – 71 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Blue_Midnight_Wish_Round2.zip\Blue_Midnight_ Wish\Supporting_Documentation\BlueMidnightWishDocumentation.pdf.

99. Groestl – a SHA-3 candidate. / Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz [та ін.]. – 2008. – 32 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Grostl_Round2.zip\Grostl\Supporting_Documentation\Groestl.pdf.

100. Wu H. The Hash Function JH. / Hongjun Wu. – 2009. – 42 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/JH_Round2.zip\JH\SupportingDocumentation\jh20090915.pdf.

101. Leurent G. SIMD Is a Message Digest. / G. Leurent, C. Bouillaguet, P.-A. Fouque. – 2009. – 270 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/SIMD_Round2.zip\SIMD\Supporting_Documentation\SIMD.pdf.

102. Bernstein D. J. Cube Hash Specification (2.B.1) / Daniel J. Bernstein. – 2009. – 5 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/CubeHash_Round2. zip\CubeHash\Supporting_Documentation\spec.pdf.

103. Halevi S. The Hash Function «Fugue» / Shai Halevi, William E. Hall, Charanjit S. Jutla. – 2009. – 96 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Fugue_Round2.zip\Fugue\ Supporting_Documentation\fugue.pdf.

104. De Canniere C. Hash function Luffa. Version 2.0.1 / Christophe De Canniere, Hisayoshi Sato, Dai Watanabe. – 2009. – 26 с. – Режим доступу до джерела: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Luffa_Round2_Update.zip\Luffa_v2_update_October2009\Luffa_v2_Specification_20091002.pdf.

105. Бардачов Ю. М. Дискретна математика. / Ю. М. Бардачов, Н. А. Соколова, В. Є. Ходаков ; за ред. В. Є. Ходакова. – К. : Вища шк., 2002. – 287 с.

106. Логачев О. А. Булевы функции в теории кодирования и криптологии. / О. А. Логачев, А. А., Сальников, В. В. Ященко. – М. : МЦНМО, 2004. – 470 с.

107. Либерти Дж. Освой самостоятельно C++ за 21 день, 4-е издание / Джесс Либерти. – М. : Вильямс, 2003. – 832 с.

108. Шеферд Дж. Программирование на Microsoft Visual C++ .NET. Мастер-класс / Джордж Шеферд. – 2-е изд. – М. : Русская Редакция ; СПб. : Питер, 2005. – 928 с.

109. Description of Known Answer Test (KAT) and Monte Carlo Test (MCT) for SHA-3 Candidate Algorithm Submissions. Revision 3: February 20, 2008. – 6 с. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/documents/SHA3-KATMCT1.pdf.

110. ANSI C Cryptographic API Profile for SHA-3 Candidate Algorithm Submissions. Revision 5: February 11, 2008. – 4 с. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/documents/SHA3-C-API.pdf.

111. Test files and Source Code for Conducting KAT and MCT / NIST. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/sha-3/documents/KAT1.zip

112. Томашевський В. М. Моделювання систем. / Валентин Миколайович Томашевський. – К. : BHV, 2005. – 352 с.

113. Розанов Ю. А. Теория вероятностей, случайные процессы и математическая статистика : учебник для вузов. / Ю. А. Розанов. – 2-е изд., доп. – М. : Наука. Гл. ред. физ.-мат. лит., 1989. – 320 с.

114. Бибило П. Н. Основы языка VHDL / Петр Николаевич Бибило. – 3-е изд. доп. – М. : ЛКИ, 2007. – 328 с.

Наукове видання

Баришев Юрій Володимирович
Лужецький Володимир Андрійович

МЕТОДИ ТА ЗАСОБИ ШВИДКОГО БАГАТОКАНАЛЬНОГО ГЕШУВАННЯ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Монографія

Редактор С. Малішевська

За заг. ред. В. А. Лужецького

Оригінал-макет підготовлено Ю. В. Баришевим

Підписано до друку 21.07.2016 р.
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різографічний. Ум. др. арк. 8,32.
Наклад 300 (1-й запуск 1–75) пр. Зам № В2016-20

Вінницький національний технічний університет,
КІВЦ ВНТУ,
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-85-32.
publish.vntu.edu.ua; *email*: kivc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано ФОП Барановська Т. П.
21021, м. Вінниця, вул. Порика, 7.
Свідоцтво суб'єкта видавничої справи
серія ДК № 4377 від 31.07.2012 р.