№ 2(16) 2023
НАУКА і ТЕХНІКА СЬОГОДНІ
серії: право, економіка, педагогіка, техніка, фізико-математичні науки

УДК 004.0569.55

https://doi.org/10.52058/2786-6025-2023-2(16)-373-380

Гапак Оксана Михайлівна кандидат педагогічних наук доцент кафедри комп'ютерних систем та мереж, доцент, ДВНЗ «Ужгородський національний університет», вул. Університетська,14, м. Ужгород, 88000, https://orcid.org/0000-0003-3448-6670

Гедеон Ганна Олегівна асистент кафедри комп'ютерних систем та мереж, асистент, ДВНЗ «Ужгородський національний університет», вул. Університетська,14, м. Ужгород, 88000, https://orcid.org/0000-0002-5684-6932

АПАРАТНИЙ БЛОК КЕРУВАННЯ ЕФЕКТИВНИМ ВИБОРОМ МОДУЛЮ ХЕШУВАННЯ

Анотація. У статті представлені результати дослідження функцій хешування *CRC*-32 і *Adler*-32 з точки зору безпеки збереження і контролю цілісності даних.

Розроблений апаратний блок керування модулями хешування *CRC*-32 і *Adler*-32 забезпечує вибір найкращого алгоритму для вхідних даних змінної довжини і має два режими роботи (автоматичний і ручний). Його складові: блок вводу даних, блок індикації, блок лічильників, блок керування *CRC*-32 і блок керування *Adler*-32. Ручний режим дозволяє обирати один із запропонованих алгоритмів хешування залежно від потреб користувача, тоді як автоматичний – аналізує вхідну інформацію та обирає найбільш ефективний варіант обрахунку хешу.

У статті наведено алгоритм роботи блоку керування модулями *CRC*-32 і *Adler*-32. Обґрунтовано доцільність використання *CRC*-32 для коротких повідомлень та визначено допустимий розмір вхідних значень, при яких хешування *Adler*-32 має відносно рівномірний розподіл обчислень. Вказано, що обидва алгоритми хешування забезпечують високий рівень виявлення порушень цілісності файлів і при апаратній реалізації, згідно з результатами досліджень, пропонують швидкодію обчислення контрольної суми, яка в сотні разів перевищує можливості програмних додатків.

Зазначено, що з використанням такого блоку керування хеш-функціями можна досягти ефективного обчислення значення хешу, залежно від довжини та вмісту вхідних даних. Апаратний блок є цілісною розробкою, в якій передбачено можливість додавання нових модулів хешування. Роботою модулів хешування керують спеціальні блоки, що базуються на скінченних

автоматах (автоматах Мура). Можливість злому блоку керування вважається мінімальною, адже передбачає процес повного розбору пристрою на складові та розрахунок всіх можливих значень, що надходять від використовуваних компонентів.

Спроектований пристрій хешування, що включає блок керування модулями *CRC*-32 і *Adler*-32, блок хешування *CRC*-32, блок хешування *Adler*-32, блок формування хешу та блок індикації, можна використовувати як мікросхему, що під'єднується до каналу зв'язку, або як окремий засіб для миттєвого відображення контрольної суми для введеного вхідного повідомлення.

Ключові слова: хешування, хеш, хеш-функція, блок керування, пристрій, режим роботи, алгоритм, модуль, апаратний модуль, CRC, Adler.

Hapak Oksana Mykhaylivna Assoc. Prof. Assoc. Prof. at the department of computer system and networks, candidate of pedagogical science, Uzhhorod National University, University St.,14, Uzhhorod, 88000, https://orcid.org/0000-0003-3448-6670

Hedeon Hanna Olehivna Assistant, Uzhhorod National University, University St.,14, Uzhhorod, 88000, https://orcid.org/0000-0002-5684-6932

HARDWARE CONTROL UNIT OF EFFECTIVE OPTION OF HASHING MODULE

Abstract. The article presents the results of the research of CRC-32 and Adler-32 hash functions from the point of view of security of storage and data integrity control.

Developed hardware control unit for CRC-32 hashing modules and Adler-32 provides choice of the best algorithm for the input of variable length and has two modes of operating (automatic and manual). Its components: data input unit, display unit, counter unit, CRC-32 control unit and Adler-32 control unit. Manual mode allows you to choose one of the proposed hashing algorithms depending on the user's needs, while automatic analyzes input information and chooses the most effective option for hash calculation.

The article demonstrates the control unit algorithm of CRC-32 and Adler-32 modules. The expediency of using CRC-32 for short messages is justified and the permissible size of input values, at which Adler-32 hashing has a relatively uniform distribution of computations, is determined. It is indicated that both hashing algorithms provide a high detection rate of file integrity violations and during hardware implementation, according to the results of the research, they offer the speed of calculating the checksum, which is in the hundreds times exceeds the capabilities of software applications.

It is noted that with the use of such control unit of hash functions it is possible to achieve an efficient calculation of the hash value, depending on the length and content of input data. The hardware unit is a complete development in which it is possible to add new hashing modules. The work of hashing modules is controlled by special blocks based on finite-state machines (Moore machine). The possibility of hacking of the control unit is considered minimal, because it involves the process of complete disassembly of the device into components and calculation of all possible values which come from the used components.

Designed hashing device, which includes a control unit of CRC-32 and Adler-32 modules, the CRC-32 hashing block, the Adler-32, the hash formation block and the indication block, can be used as a chip that connects to a communication channel, or as a separate means for instant display of the checksum for the entered input message.

Keywords: hashing, hash, hash function, control unit, device, operating mode, algorithm, module, hardware module, CRC, Adler

Постановка проблеми. Забезпечення захисту інформації, в тому числі контролю її цілісності являє собою комплексну науково-технологічну проблему, що призводить до створення та інтеграції математичних алгоритмів, спеціалізованих апаратних і програмних розробок для вирішення питань інформаційної безпеки. Інформаційна безпека полягає у забезпеченні стійкості функціонування системи при випадкових або навмисних впливах, що можуть призвести до порушення цілісності, блокування або несанкціонованого поширення інформації.

Для захисту інформації використовують різні алгоритми шифрування і хешування тощо [1].

Хешування – алгоритм, за допомогою якого масив вхідних даних довільної довжини перетворюється в масив даних наперед визначеної довжини. Сфера застосування хеш-функцій дуже широка. За допомогою хешування можна переконатися в автентичності файлів операційної системи, конфіденційних і цілісності документів, програм. Також хешування використовується для автентифікації транзакцій, повідомлень і цифрових підписів; під час збереження паролів в системі захисту; при пошуку дублікатів в серіях наборів даних тощо [2].

Існує велика кількість алгоритмів хешування, побудованих на простих схемах, а також на складних криптографічних перетвореннях. Велику популярність мають прості алгоритми, і залежно від алгоритму, хеш-функція може бути неефективною або малоефективною для розв'язання конкретної задачі. Так, наприклад, для захисту від ненавмисних спотворень, в тому числі і апаратних помилок, доцільно використовувати швидкі та надійні апаратні

алгоритми, наприклад, *CRC* (англ. *Cyclic redundancy check*) з 32-бітним вихідним кодом і *Adler*-32, розроблений Марком Адлером.

Аналіз останніх досліджень і публікацій. Результати досліджень [3; 4] дозволяють відстежити суттєву різницю між алгоритмами хешування *Adler*-32 і *CRC*32*C* (відрізняється від *CRC-32* поліномом) з точки зору ймовірності невиявлених помилок, обчислених при рівномірному розподілі даних.

Результати представлені в табл. 1, де: *d* – мінімальна відстань на блоці довжини *Block*, *Block* – довжина блоку в бітах, *i/byte* – кількість програмних інструкцій на байт, *Pudb* – імовірність невиявлених групових помилок, *Puds* – імовірність невиявлених одиничних помилок.

Таблиця 1

Вибіркові дані дослідження *RFC* 3385

Алгоритм	*d*	*Block*	*i/byte*	*Pudb*	*Puds*
Adler-32	3	2^19	3	10^-36	10^-35
*CRC*32*C*	3	2^31-1	2.75	10^-41	10^-40

Варто зауважити, що імовірність невиявлених помилок *CRC* залежить від обраного полінома, розподілу помилок і довжини даних. Згідно з результатами *RFC* 3385, *CRC-32* є кращим вибором, в порівнянні з *Adler*-32, як основний механізм виявлення помилок.

Дослідження швидкодії алгоритмів *Adler*-32 і *CRC-32* (апаратна реалізація у пакеті *NI Multisim*), що включило серію тестів для згенерованих випадковим чином вхідних даних (коди доступу довжиною від 40 до 64 біт), показало, що апаратний модуль *Adler*-32 виконує обчислення контрольної суми в 1,481 рази швидше, ніж модуль *CRC*-32 [5].

Проте, враховуючи результати *RFC* 3309 [4], в яких описано доцільність використання *CRC32C* замість *Adler*-32 для *SCTP* (англ. S*tream Control Transmission Protocol* – «протокол передачі з керуванням потоком») для повідомлень, розміром до 128-байт (максимальне значення *A* для 128-байтних вхідних даних дорівнює 32 641, що менше ніж 65 521 – число, з яким виконується операція *mod* в процесі обрахунку контрольної суми), перевага у швидкодії *Adler*-32 доречна виключно для вхідних даних при яких *A* буде перевищувати 65 521 (найбільше просте число, менше за 2^16).

Скориставшись дослідженням авторів Алексеєва В.Д. та Матвеєва Д.І. [2], зауважимо, що для хешування кодів, номерів телефонів, номерів кредитних карток, інших складових платіжної інформації, таких як *PAN*, *CVC*, *PIN*-код, є неефективним, проте, апаратний модуль повинен забезпечувати хешування

будь-яких інших даних хеш-функцією, яка буде давати найшвидший результат.

Також, відповідно до результатів публікації [6], алгоритми хешування *CRC*-32 і *Adler*-32 забезпечують високий рівень виявлення відмінностей між двома файлами; ідентифікація подібних файлів неможлива, адже при незначних змінах вмісту файлу контрольна сума суттєво відрізняється. Для контролю цілісності файлів передбачено ручний режим вибору алгоритму хешування.

Об'єктом нашого дослідження є апаратні модулі хешування на базі алгоритмів загального призначення *CRC*-32 і *Adler*-32. Предметом дослідження є визначення ефективності модулю хешування залежно від вхідних даних.

Мета статті є дослідження апаратних модулів хешування відносно доцільності їх вибору з огляду на безпеку збереження і контролю цілісності даних. А також реалізація блоку керування модулями *CRC*-32 і *Adler*-32 для оптимального їх вибору залежно від масиву вхідних даних.

Виклад основного матеріалу. Блок керування модулями *CRC*-32 і *Adler*-32.

Апаратний блок керування модулями хешування *CRC*-32 і *Adler*-32 реалізований в пакеті *NI Multisim* і складається з блоку вводу даних, блоку індикації, блоку лічильників, блоку керування *CRC*-32 і блоку керування *Adler*-32 (*A1*, див. рис. 1). Спроектований пристрій є цілісною розробкою.

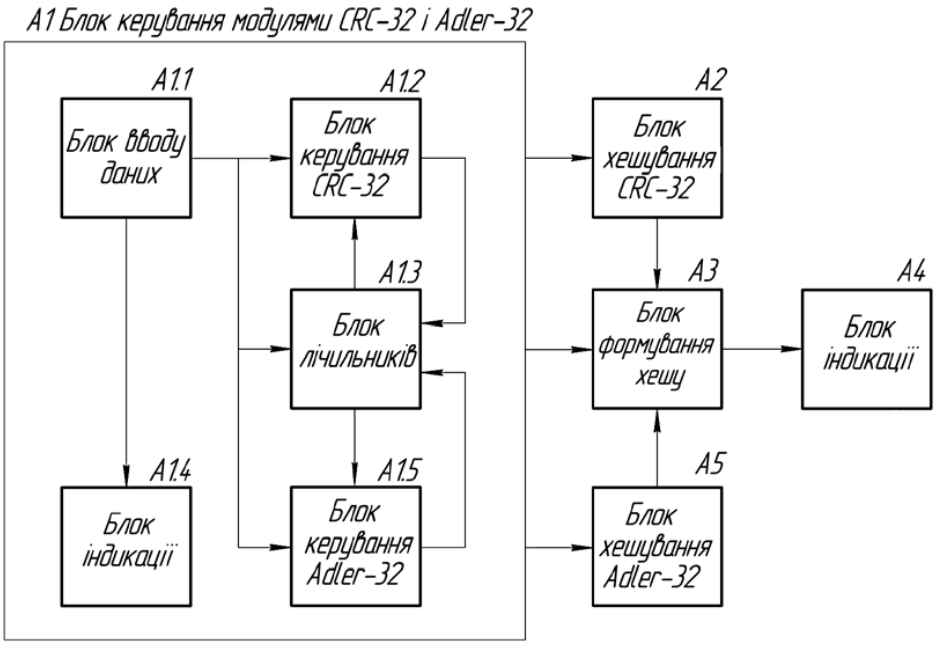

Рис. 1. *Структурна схема апаратного модулю хешування з деталізованим блоком керування*

Блок керування модулями *CRC*-32 і *Adler*-32 має два режими роботи: ручний (перший) і автоматичний (другий). Ручний режим дозволяє обирати один із запропонованих алгоритмів хешування (*CRC*-32 або *Adler*-32) залежно від потреб користувача, тоді як автоматичний – аналізує вхідну інформацію та обирає найбільш ефективний варіант обрахунку хешу.

У другому режимі роботи пріоритет має хеш-функція *CRC*-32 для числових даних та вхідних повідомлень розміром до 2000 символів. *Adler*-32, в свою чергу, обирається для текстових даних, розміром від 2000-байт (цифри, літери кирилиці й латиниці, розділові знаки, пробіли, додаткові символи). Такий вибір зумовлено особливістю функції *Adler*-32, що не забезпечує рівномірний розподіл обчислень для коротких наборів інформації.

Під час розрахунку допустимих розмірів вхідних повідомлень для алгоритму хешування *Adler*-32 враховано відносну частоту використання літер англійської мови[5] без урахування розділових знаків і пробілів. Для великих букв значення частини *A* перевищило 65 521 при використанні 867 символів (*A* – 65 565), тоді як для маленьких – 610 символів (*A* – 65 629). Отже, орієнтовний розмір повідомлень становить від 650 або від 900 англійських літер нижнього або верхнього регістрів відповідно. Враховуючи, що апаратний модуль хешування повинен забезпечувати ефективний обрахунок хешу для різних наборів символів, визначена допустима кількість є неоптимальною. Символи !, ", #, $, %, &, ', (,), *, +, -, ., / в кодуванні *ASCII* представлені значеннями від *21h* до *2Fh*, тому при хешуванні алгоритмом *Adler*-32 900-байтного повідомлення, що складається виключно з цих символів, максимальне значення *A* складатиме 42 301, а мінімальне – 29 701, що значно менше ніж 65 521. Так, для перевищення числа 65 521 необхідно використати близько 2000 символів.

Алгоритм роботи блоку керування модулями *CRC*-32 і *Adler*-32:

Крок 1. Блок вводу даних (*A1.1*) отримує вхідне повідомлення для якого необхідно обрахувати хеш та обраний режим роботи пристрою. При ручному виборі алгоритму хешування сигнал дозволу на виконання передається блоку керування *CRC*-32 або *Adler*-32 (*A1.2* або *A1.5*) відповідно. При виборі автоматичного режиму роботи виконується перехід до кроку 2. Обраний режим демонструється на блоці індикації (*A1.4*).

Крок 2. У блоці лічильників (*A1.3*) визначається розмір вхідного повідомлення. При посимвольному зсуві даних у регістрі, що знаходиться в блоці *A1.1*, виконується інкремент відповідного лічильника. У випадку, якщо лічильник досяг значення *b011111010000* або регістр зсуву порожній, один з блоків керування (*A1.2* або *A1.5*) отримує сигнал дозволу на виконання від компаратора. Компаратор є складовою блоку *A1.3*.

Крок 3. Блок керування алгоритмом хешування (*A1.2* або *A1.5*) передає керуючі сигнали в блок хешування (*A2* або *A5*) та блок лічильників (*A1.3*).

Після того, як всі дані були оброблені, блок *A1.1* передає сигнал дозволу на формування хеш-суми в блок формування хешу (*A3*). Результат демонструється блоком індикації (*A4*).

Реалізація самих модулів хешування *CRC*-32 і *Adler*-32 детально описана в дослідженнях [3].

Таким чином, з використанням такого блоку керування хеш-функціями ми можемо досягти ефективного обчислення значення хешу, залежно від довжини та вмісту вхідних даних. Розроблений пристрій хешування можна використовувати як мікросхему, що під'єднується до каналу зв'язку, або як окремий засіб для миттєвого відображення контрольної суми для введеного вхідного повідомлення.

Висновки. У статті розглянуто використання алгоритмів хешування *CRC*-32 і *Adler*-32 для різних вхідних даних. Аналіз досліджень показав, що хеш-функція *Adler*-32 неефективна для коротких вхідних повідомлень, тоді як *CRC*-32 забезпечує рівномірний розподіл обчислень, але поступається *Adler*-32 у швидкодії. Обидва алгоритми хешування забезпечують високий рівень виявлення порушень цілісності файлів.

Реалізований апаратний блок керування модулями *CRC*-32 і *Adler*-32 забезпечує вибір оптимального алгоритму хешування залежно від вхідних даних і має два режими роботи (автоматичний і ручний). Будова блоку керування передбачає можливість додавання нових модулів хешування, для яких зарезервовані бітові комбінації. Апаратна реалізація такого блоку забезпечує цілість і захищеність пристрою хешування.

Результати роботи можуть бути використані при апаратній реалізації контрольної суми Флетчера, що лежить в основі алгоритму *Adler*-32, та дослідженні особливостей апаратного моделювання алгоритмів хешування на прикладі циклічного надлишкового коду.

Література:

1. Вишня В. Б. Основи інформаційної безпеки : навч. посіб. / В.Б. Вишня, О.С. Гавриш, Е.В. Рижков. – Дніпро : ДДУВС, 2020. – 128 с.

2. Алексєєв В.Д. Хешування інформації може бути неефективним або небезпечним./ В.Д. Алексєєв, Д.І. Матвєєв // *ΛΟΓΟΣ. ONLINE*. – 2019. – №4. URL: Режим доступу: https://ojs.ukrlogos.in.ua/index.php/2663-4139/article/view/529.

3. Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC)/ Sheinwald D., Satran J., Thaler P., Cavanna V.// Checksum Considerations. *Network Working Group Request for Comments*. – 2002. – RFC 3385. P. 1-23.

4. Stream Control Transmission Protocol (SCTP)/ Stone J., Stewart R., Otis D. // Checksum Change. *Network Working Group Request for Comments.* – 2002. RFC 3309. – P. 1-17.

5. Гапак О.М. Апаратна реалізація модулів хешування на базі алгоритмів CRC-32 і Adler-32 / О.М. Гапак, Г.О. Гедеон // Науковий вісник Ужгородського університету. Серія: Математика і інформатика. – 2021. – Том 39 – № 2. – С.145-151.

6. Glaser A. The Use of Cyclic Redundancy Check Checksums for Source (CRC-32) and Adler-32 Code Verification. / A. Glaser *Drug // Information journal: DIJ/Drug Information Association.* – 2003. – Vol 37. – P. 147-154.

References:

1. Vyshnia, V. B., Gavrish, O. S., & Rizhkov, E.V. (2020). *Osnovy informatsiinoi bezpeky* [Basics of information security]. Dnipro: DDUVS [in Ukrainian].

2. Aleksieiev, V. D., & Matvieiev, D. I. (2019). Kheshuvannia informatsii mozhe buty neefektyvnym abo nebezpechnym [When hashing information may be ineffective or dangerous]. *ΛΟΓΟΣ.ONLINE*, 4. Retrieved from https://ojs.ukrlogos.in.ua/index.php/2663-4139/article/view/529 [in Ukrainian].

3. Sheinwald, D., & Satran, J. (2002). Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC). Checksum Considerations. *Network Working Group Request for Comments,* RFC 3385, 1-23. doi: 10.17487/RFC3385.

4. Stone, J., Stewart, R., & Otis, D. (2002). Stream Control Transmission Protocol (SCTP) Checksum Change. *Network Working Group Request for Comments,* RFC 3309, 1-17. doi: 10.17487/RFC3309.

5. Hedeon, H. O., & Hapak, O. M. (2021). Aparatna realizatsiia moduliv kheshuvannia na bazi alhorytmiv CRC-32 i Adler-32 [Hardware implementation of hashing modules based on algorithms CRC-32 and Adler-32]. *Sci. Bull. of Uzhhorod Univ. Ser. of Math. and Inf.*, 39, 145-151. doi: 10.24144/2616-7700.2021.39(2).1 45-151 [in Ukrainian].

6. Glaser, A. (2003). The Use of Cyclic Redundancy Check (CRC-32) and Adler-32 Checksums for Source Code Verification. *Drug information journal: DIJ/Drug Information Association*, 37, 147-154. doi: 10.1177/009286150303700203.