Vol. 6, No. 1, 2024

DOI: https://doi.org/10.23939/csn2024.01.026

УДК 004.056.5

ДОСЛІДЖЕННЯ ТА ВДОСКОНАЛЕННЯ ОБЧИСЛЮВАЛЬНИХ АЛГОРИТМІВ ДЛЯ РОЗРАХУНКУ ТРИГОНОМЕТРИЧНИХ КОЕФІЦІЄНТІВ АЛГОРИТМУ ХЕШУВАННЯ MD5

А. Я. Горпенюк[1], Н. М. Лужецька[2], М. А. Горпенюк[3]

Національний університет "Львівська політехніка",
[1,3] кафедра захисту інформації,
[2] кафедра безпеки інформаційних технологій
E-mail: andrii.y.horpeniuk@lpnu.ua, nataliia.m.luzhetska@lpnu.ua, mykola.horpeniuk.kb.2023@lpnu.ua

© Горпенюк А. Я., Лужецька Н. М., Горпенюк М. А., 2024

У роботі досліджено проблематику забезпечення автентичності повідомлень. Розглянуто основні вимоги до функцій хешування й особливості та проблеми розроблення обчислювальних алгоритмів для хешування інформації.

Досліджено поширений алгоритм хешування MD5, який є ефективним і швидким алгоритмом хешування повідомлень. Хоча на сьогодні стійкість цього алгоритму недостатня для захисту даних вищих рівнів таємності, алгоритм успішно застосовується для захисту комерційної інформації. У статті детально досліджено основні обчислювальні перетворення алгоритму хешування MD5. Визначено, що особливістю алгоритму хешування MD5 є застосування змінних тригонометричних констант для підвищення надійності алгоритму. Значення цих змінних констант відповідають розгортці функції синуса.

У праці досліджено доцільність застосування для обчислення змінних тригонометричних констант алгоритму хешування MD5 класичних і вдосконалених широкодіапазонних число-імпульсних обчислювальних структур. Показано, що застосування класичних число-імпульсних обчислювальних структур недоцільне через недостатній діапазон відтворення неодмінних тригонометричних функцій. Удосконалені широкодіапазонні число-імпульсні структури забезпечують потрібні функцію перетворення, діапазон і точність. Проте швидкодія таких обчислювачів критично недостатня для обчислення усіх змінних тригонометричних коефіцієнтів алгоритму хешування MD5.

У дослідженні розроблено математичну і програмну модель структури розгортання функції синуса для алгоритму MD5. Математична модель ґрунтується на співвідношеннях для синуса і косинуса суми аргументів, які адаптовані для алгоритму хешування MD5. Застосування розробленої різницевої обчислювальної структури забезпечує економію пам'яті під час реалізації алгоритму.

Ключові слова: криптографія, автентичність повідомлення, функція хешування.

Вступ

У статті розглянуто актуальне питання поліпшення ефективності сучасних систем хешування даних. Сучасні алгоритми хешування даних є потужним інструментом захисту даних від модифікації (тобто гарантують автентичність даних), а також дієвим інструментом радикального одностороннього стискання даних перед застосуванням таких обчислювально складних операцій, як операція цифрового підписування. Сучасні алгоритми хешування можна умовно поділити на дві

великі групи. Алгоритми першої групи ґрунтуються на швидких логічних операціях, характеризуються високою швидкодією і застосовуються переважно для захисту комерційної інформації. До таких алгоритмів, зокрема у цьому дослідженні, належать алгоритм хешування MD5 [1], сімейство алгоритмів SHA тощо. Алгоритми другої групи ґрунтуються на стандартних симетричних блокових шифрах. Вони повільніші за алгоритми хешування першої групи, проте їх обчислювальну стійкість доводити легше, а тому саме такі алгоритми переважно стандартизують і застосовують для захисту інформації, яка становить державну і військову таємницю.

З огляду на стале зростання доступних обчислювальних потужностей, алгоритми хешування постійно вдосконалюються [6, 7]. Водночас добре вивчені і поширені алгоритми хешування, такі як алгоритм хешування MD5, і далі активно застосовуються для захисту від модифікації комерційних даних.

Основними властивостями функції або алгоритму хешування, або інакше вимогами до таких функцій, є такі властивості [2, 3]:

1. Функція хешування може застосовуватися до відкритого тексту будь-якої довжини.
2. Результат функції хешування має мати фіксовану довжину.
3. Хеш-функція h=H(x) має обчислюватися легко (ефективно) для усіх можливих значень x. Водночас спосіб обчислення хешу має бути ефективним як з погляду його апаратної реалізації, так і з погляду програмної реалізації.
4. Якщо відоме хеш-значення h, обчислювально дуже складно визначити x, для якого H(x)=h. Ця особливість хеш-функції вказує на її важкооборотність.
5. Якщо задане вхідне значення x, обчислювально дуже складно визначити таке $y \neq x$, щоб значення хешів x та y збігалися: H(x) = H(y). Це одна з важливих вимог до хеш-функцій. Її називають слабкою стійкістю до колізій.
6. Обчислювально дуже складно визначити такі два різні x та y, хеші яких збігаються: H(x) = H(y). Це ще одна важлива вимога до хеш-функцій – основна вимога. Таку особливість хеш-функції називають сильною стійкістю до колізій.

Під час розроблення сучасних систем хешування двома основними вимогами до них є сильна стійкість до колізій і ефективність (швидкодія). Незалежно від обраного підходу до побудови системи хешування (на основі блокових симетричних шифрів, чи на основі швидких логічних операцій), застосовуються два основні принципи розробки [3]:

- Принцип ітеративної обробки: інформація, що хешується, розбивається на послідовність *n-бітових* блоків. Дані, що хешуються, обробляються послідовно блок за блоком, щоб *одержати n*-бітове значення функції хешування.
- Принцип MD-підсилення: перед хешуванням до повідомлення додається код довжини повідомлення.

Основними режимами застосування хеш-функцій є такі режими [3]:

1. Передаються повідомлення разом із хешем, зашифровані симетричним шифром – крім цілісності, забезпечується також автентичність та конфіденційність.
2. Передається відкрите повідомлення із симетрично зашифрованим хешем – крім цілісності, забезпечується автентичність.
3. Передається відкрите повідомлення із асиметрично (таємним ключем відправника) зашифрованим хешем – забезпечується цілісність, автентичність і цифровий підпис.
4. Передається повідомлення з асиметрично зашифрованим хешем, разом симетрично зашифровані – забезпечуються цілісність, автентичність, підпис і конфіденційність.
5. Відкрито передається повідомлення і хеш повідомлення з приєднаним таємним рядком – забезпечується цілісність і автентичність без шифрування.
6. Передаються симетрично зашифровані повідомлення разом із хешем повідомлення з приєднаним таємним рядком – забезпечуються цілісність, автентичність, конфіденційність.

1. Огляд літературних джерел

MD5 (алгоритм дайджесту повідомлення) [1] – це спосіб обчислення хеш-значення для заданої інформації. Алгоритм розробив Рон Ріверст. MD5 зарекомендувала себе як надійна і швидка хеш-функція.

Вхід алгоритму – повідомлення довільної довжини. На виході отримують коротке повідомлення (дайджест) завдовжки 128 біт. Схему генерування хеш-значення в алгоритмі MD5 [1] подано на рис.1.

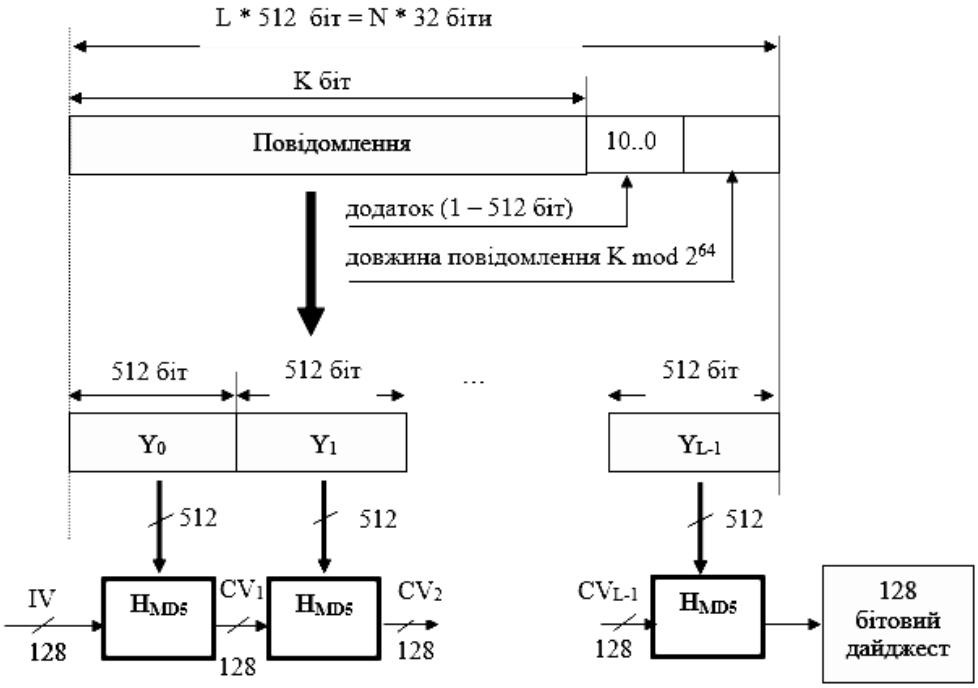

Рис. 1. Схема генерування хеш-значення в алгоритмі MD5

Основними етапами перетворення в цьому алгоритмі є такі етапи [1]:

Етап 1. Розширювальне нормування вхідного повідомлення. Вхідне K – бітове повідомлення доповнюється таким способом, щоб бітова довжина L цього розширеного повідомлення була порівняна з 448 за модулем 512 (тобто конгруентна 448 mod 512). У разі, якщо вхідне повідомлення уже має таку довжину, до нього додається 512 біт. У всіх можливих випадках додаток буде складатися з однієї одиниці і потрібної кількості нулів (100..0). Водночас приписуватися може від 1 до 512 бітів.

Етап 2. Прив'язка до довжини вхідного повідомлення. Тут до результату попереднього етапу (розширення) дописується довжина K вхідного повідомлення. Ця довжина задається 64 – бітовим числом. У разі, коли довжина вхідного повідомлення більша за 64 біти, приписується число K mod 2^{64}.

Етап 3. Запис початкових ініціювальних даних алгоритму. В MD-5 для зберігання остаточного значення хеш-функції, а також для тимчасового зберігання проміжних значень хеш-функції застосовується 128-бітовий буфер. Такий буфер – це чотири 32-бітові регістри, які позначають символами A, B, C, D. На початку роботи алгоритму MD-5 в буфер записують такі початкові значення в шістнадцятковому коді:

A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476

Етап 4. Реалізація чотирираундового перетворення вхідного 512-бітового повідомлення. Вказаний етап реалізують за допомогою основного модуля перетворення алгоритму хешування MD5. Цей модуль перетворення зазвичай позначають H_{MD5}. Схему перетворень цього етапу подано на рис. 2. Бачимо (рис. 2), що основне перетворення алгоритму H_{MD5} містить чотири однакові раунди перетворення. Кожен раунд має на своєму вході: 512-бітовий блок Yq, а також 128-бітовий вміст буферних регістрів ABCD. Проте кожен із чотирьох раундів обчислює власну логічну функцію: відповідно F, G, H та I. Окрім того, в кожному раунді зокрема використовується своя чверть 64-елементної таблиці констант T[1..64]. Ця таблиця констант T[i] містить значення функції синуса, точніше, цілу частину значень 2^{32} * abs(sin(i)). Водночас значення аргументу і задане в радіанах. Тож усі елементи таблиці T[i] є 32-бітовими числами.

Після виконання усіх чотирьох раундів основного перетворення алгоритму хешування MD5 результат останнього четвертого раунду підсумовується із вхідними даними першого раунду (CVq). Підсумовування виконується за модулем 2^{32}. І лише після цього отримуємо результат CVq+1 основного модуля перетворення алгоритму хешування MD5.

Етап 5. Видача обчисленого хеш-значення алгоритму. За допомогою визначених перетворень Етапу 4 алгоритму хешування MD5 послідовно обробляються усі L 512-бітові блоки, сформовані за результатами реалізації Етапів 1–3. Остаточним результатом хешування алгоритмом MD5 є результат оброблення Етапом 4 перетворення останнього L – го блоку. Саме це 128-бітове коротке повідомленням і є хешем або інакше дайджестом вхідного повідомлення довільної довжини.

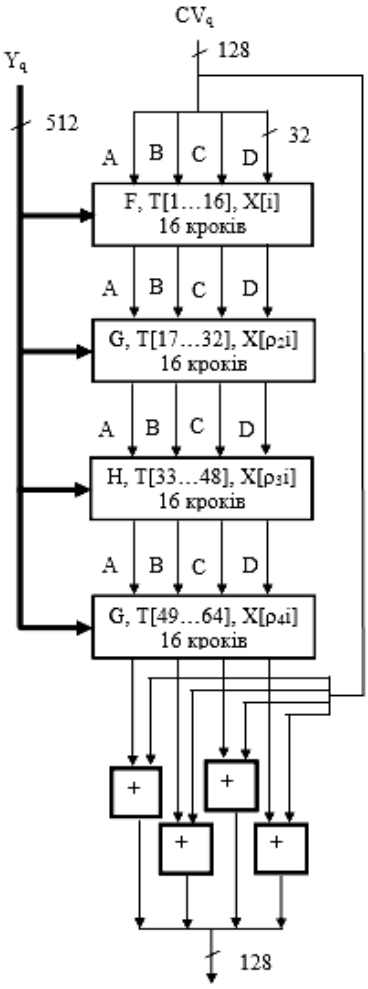

Рис. 2. Структура функції стискання H_{MD5}

2. Постановка завдання

Завдання роботи – дослідити доцільність застосування спеціалізованих різницевих методів обчислення для розрахунку змінних тригонометричних коефіцієнтів алгоритму хешування MD5 (класичних число-імпульсних, а також реверсивних широкодіапазонних число-імпульсних тригонометричних обчислювачів), розроблення вдосконалених різницевих методів обчислення коефіцієнтів алгоритму хешування MD5, рекомендацій щодо їх застосування.

3. Дослідження доцільності застосування класичних число-імпульсних обчислювачів функції синуса для розрахунку коефіцієнтів алгоритму хешування MD5

Як було показано вище, основне перетворення алгоритму MD5, позначене H_{MD5}, містить чотири однакові раунди. Кожен із цих чотирьох раундів обчислює свою задану логічну функцію – F, G, H або I [1]. Кожен з чотирьох раундів має на своєму вході однаковий 512-бітовий результат перетворення переднього блоку Yq. Крім того, на вході кожного раунду різні 128-бітові значення з буферних регістрів ABCD (див. рис. 2).

В кожному з чотирьох раундів для розрахунків використовується одна чверть 64-елементної таблиці тригонометричних констант T[1..64]. Така таблиця містить нормовані за визначеними правилами значення функції синуса. Конкретніше, це цілі значення чисел 2^{32} * abs(sin(i)), де значення аргументу і задане в радіанах. Аргумент і змінюється від одного до 64 радіан. Отже, в таблиці T[i] містяться цілі 32-бітові числа.

Отже, під час обчислення змінних тригонометричних констант алгоритму хешування MD5 табулюється тригонометрична функція синуса. Іншими словами, виконується розгортання функції синуса. Необхідні в алгоритмі MD5 значення функції синуса (для значень аргументу від одного до 64 радіан) потрібно обчислювати, або вибрати з наперед обчисленої таблиці,. Найчастіше реалізується саме табличний варіант. Проте нерідко вимоги до обсягів пам'яті пристрою хешування за алгоритмом MD5 є дуже жорсткими. Наприклад, коли передбачається виконання алгоритму хешування MD5 на інтелектуальних карточках. У подібних випадках треба відмовитися від табличного представлення змінних коефіцієнтів алгоритму, а реалізувати натомість оптимальне біжуче обчислення змінних тригонометричних коефіцієнтів.

У процесі виконання досліджень було проаналізовано доцільність застосування класичного число-імпульсного обчислювача функції синуса [5] для обчислення тригонометричних коефіцієнтів функції хешування MD5. Розглянемо послідовність синтезу такого число-імпульсного обчислювача функції синуса [5].

Отже, потрібно обчислити функцію:

$$y = \sin x. \tag{1}$$

1. Диференціюємо функцію (1):

$$dy = \cos x \cdot dx. \tag{2}$$

2. Розкладаємо (2) в систему диференційних рівнянь Клода Шеннона, яка породжує потрібну нам функціональну залежність (1). Також вводимо допоміжні змінні для позначення допоміжних функцій:

$$z = \cos x, \tag{3}$$

$$dz = -\sin x \cdot dx = -y \cdot dx, \tag{4}$$

$$\begin{cases} dy = z \cdot dx \\ dz = -y \cdot dx \end{cases}. \tag{5}$$

Отже, ми отримали систему породжувальних диференціальних рівнянь функції (1).

Із співвідношення (5) можна зробити висновок, що для різницевого обчислення заданої функції синуса (1), потрібно застосувати два однакові інтегратори, охопивши їх різнознаковими перехресними зворотними зв'язками.

Зауважимо тут, що для наближеної реалізації обох рівнянь системи (5) можна замість ідеальних інтеграторів застосувати так звані цифрові інтегратори. Цифрові інтегратори називають інакше число-імпульсних помножувачами і будують як правило на нагромаджуючих суматорах [5]. Якщо врахувати особливості роботи число-імпульсних помножувачів на нагромаджуючих суматорах, система породжуючих рівнянь (5) трансформується в таку систему:

$$\begin{cases} \Delta Y = Z \cdot \Delta X / 2^n \\ \Delta Z = -Y \cdot \Delta X / 2^n \end{cases}, \tag{6}$$

де n – розрядність суматорів число – імпульсних помножувачів. Структуру [5], що реалізує систему (6), подано на рис. 3.

За допомогою такої структури, забезпечивши початкові умови $Y_0 = 0$; $Z_0 = 2^n$, наближено обчислимо:

$$Y = 2^n \cdot \sin\left(X / 2^n\right), \tag{7}$$

$$Z = 2^n \cdot \cos\left(X / 2^n\right). \tag{8}$$

Іншими словами, якщо виконати попереднє масштабування: $X = 2^n \cdot x; Y = 2^n \cdot y; Z = 2^n \cdot z$, схема на рис. 3 зможе наближено обчислювати функцію (1). Щодо точності обчислення, наявні результати імітаційного моделювання подібного обчислювача свідчать про його доволі високу точність. Наприклад, абсолютна похибка перетворення такого обчислювача лежить в таких межах:

$$-1 < \Delta_n Y, \Delta_n Z < 1, \text{ або:} \tag{9}$$

$$-1/2^n < \Delta_n y, \Delta_n z < 1/2^n. \tag{10}$$

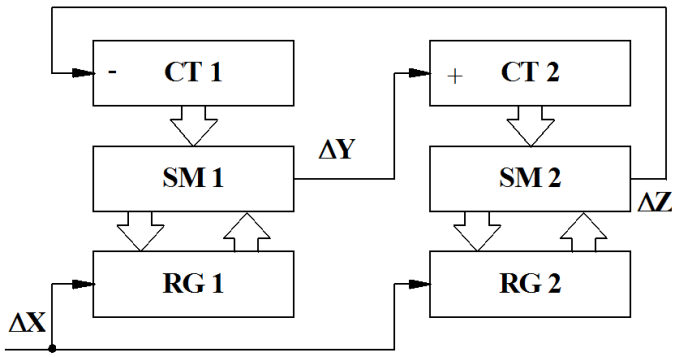

Рис. 3. Структура число-імпульсного синусного обчислювача

Водночас відтворення функції (1) структурою на рис. 3 можливе тільки в діапазоні вхідної змінної:

$$0 \le x < \pi/2 . \tag{11}$$

Оцінка (11) діапазону перетворення класичного число-імпульсного обчислювача синуса означає, що такий обчислювач не може застосовуватися для обчислення змінних тригонометричних коефіцієнтів алгоритму хешування MD5. Адже під час обчислення функції синуса в алгоритмі хешування MD5 аргумент треба змінювати від одного до 64 радіан.

4. Дослідження доцільності застосування реверсивних широкодіапазонних число-імпульсних обчислювачів функції синуса для розрахунку коефіцієнтів алгоритму хешування MD5

Забезпечити необхідне для алгоритму MD5 значення діапазону перетворення дає можливість структура широкодіапазонного число-імпульсного обчислювача синуса [5]. Схему такого обчислювача подано на рис. 4.

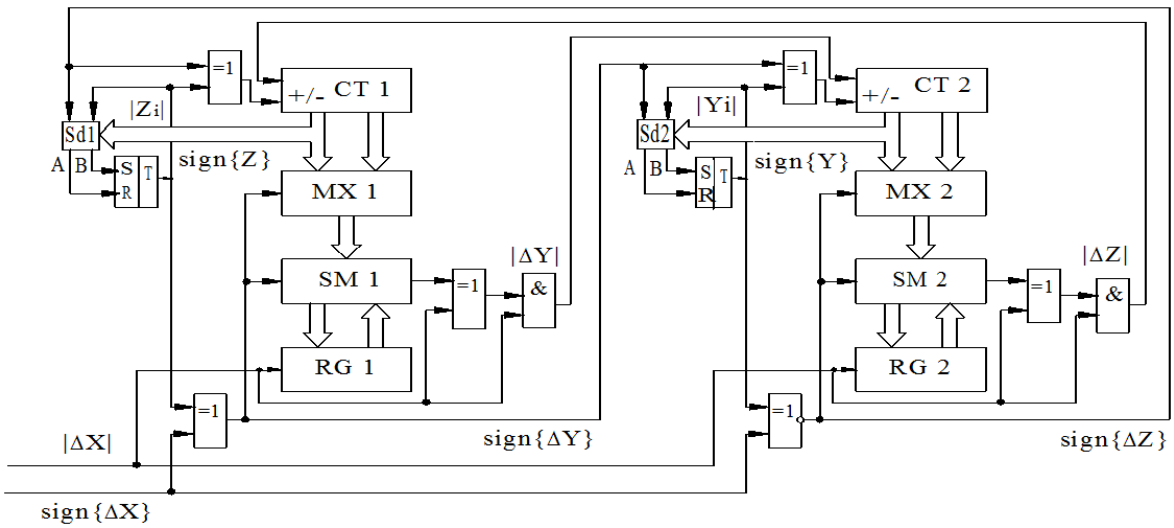

Рис. 4. Структура широкодіапазонного число-імпульсного синусного перетворювача

Схему побудовано із застосуванням реверсивних число-імпульсних помножувачів, розглянутих у роботі [4]. Як показано в [5], схема на рис. 4, як і класичний число-імпульсний обчислювач синуса (рис. 3), забезпечує відтворення функції (7) з похибкою (9). Проте на відміну від класичного обчислювача, який працює в діапазоні (11), широкодіапазонний обчислювач (рис. 4) здатний працювати в необмеженому діапазоні зміни аргументу (12):

$$-\infty < x < \infty . \tag{12}$$

На перший погляд, структура широкодіапазонного число-імпульсного обчислювача функції синуса (рис. 4) може застосовуватися для обчислення змінних тригонометричних коефіцієнтів алгоритму хешування MD5. Адже нас влаштовує і функція перетворення (7), і діапазон перетворення (12). Проте якщо детальніше проаналізувати функцію перетворення (7) і взяти до уваги той факт, що під час обчислення змінних тригонометричних коефіцієнтів алгоритму хешування MD5 аргумент треба змінювати від одного до 64 радіан з кроком один радіан, доведеться визнати, що структура на рис. 4 також не придатна для обчислення коефіцієнтів алгоритму хешування MD5.

Справді, відповідно до співвідношення (7) приросту аргументу X в один радіан відповідає 2^n одиничних приростів вхідного число-імпульсного коду структури. Це означає, що для оброблення приросту аргументу в один радіан через число-імпульсні структури (рис. 3, рис. 4) треба пропустити 2^n одиничних приростів вхідного число-імпульсного коду. В пристосуванні до алгоритму хешування MD5, в якому треба отримати 32-бітовий результат, через ці структури потрібно на кожен радіан пропускати 2^{32} одиничних приростів. За таких умов очевидно, що навіть на найвищих доступних тактових частотах такі обчислювачі будуть мати критично низьку швидкодію.

5. Розроблення і дослідження різницевого обчислювача змінних тригонометричних коефіцієнтів алгоритму хешування MD5

З огляду на те, що в алгоритмі хешування MD5 застосовуються змінні значення тригонометричної константи, якою є нормоване значення функції синуса, обчислене для значень аргументу від одного до 64 радіан з кроком один радіан, відомі число-імпульсні обчислювачі функції синуса (рис. 3, рис. 4) не можуть застосовуватися для обчислення таких констант. Адже для того, щоб обробити один-єдиний приріст аргументу в один радіан і отримати 32-бітовий результат, через такий обчислювач, відповідно до (7), потрібно пропустити 2^{32} одиничних приростів аргументу. Це неприпустимо багато, навіть за наявності сучасної високошвидкісної елементної бази. Тому в роботі розроблено і досліджено спеціалізований різницевий обчислювач змінних тригонометричних коефіцієнтів алгоритму хешування MD5, який працює з приростом аргументу, що дорівнює одному радіану.

Для побудови практично придатного різницевого методу розрахунку тригонометричних констант алгоритму хешування MD5 запропоновано спертися на відомі формули для синуса та косинуса суми двох аргументів (13), (14):

$$\sin(\alpha+\beta)=\sin\alpha\cdot\cos\beta+\cos\alpha\cdot\sin\beta\,, \tag{13}$$

$$\cos(\alpha+\beta)=\cos\alpha\cdot\cos\beta-\sin\alpha\cdot\sin\beta\,. \tag{14}$$

Далі, зважаючи на те, що аргумент функції синуса в алгоритмі хешування MD5 змінюється від одного до 64 радіан з кроком один радіан, отже, приріст аргументу є фіксований і завжди дорівнює одному радіану, формули (13), (14) можна так модифікувати:

$$\sin(\alpha+1)=\sin\alpha\cdot\cos1+\cos\alpha\cdot\sin1\,, \tag{15}$$

$$\cos(\alpha+1)=\cos\alpha\cdot\cos1-\sin\alpha\cdot\sin1\,. \tag{16}$$

Аргумент α тут змінюється від нуля до 63. Спираючись на співвідношення (15), (16), в роботі розроблено алгоритм і програму рекурентного уточнення змінних тригонометричних коефіцієнтів алгоритму MD5. Структуру алгоритму такого обчислювача подано на рис. 5.

Фактично, на рис. 5 – алгоритм підпрограми Рекурентного Обчислення Коефіцієнтів MD5 відповідно до формул (15), (16). Входом алгоритму є попередні значення Y та Z, або інакше Y_{i-1} та Z_{i-1} ($Y_{i-1}=(2^{32}$ * abs(sin(i-1))mod 2^{32}, $Z_{i-1}=(2^{32}$ * abs(cos(i-1))mod 2^{32}). Тобто початкові значення Y та Z – 32-бітові беззнакові значення синуса (32-бітовий нуль) та косинуса (32-бітова одиниця) нуля відповідно. Під час обчислень, відповідно до формул (15), (16), застосовуються також дві константи: CCOS1 та CSIN1 – відповідно, 32-бітове значення результату обчислення 2^{32} * abs(cos(1))mod 2^{32} та 32-бітове значення результату обчислення 2^{32} * abs(sin(1))mod 2^{32}.

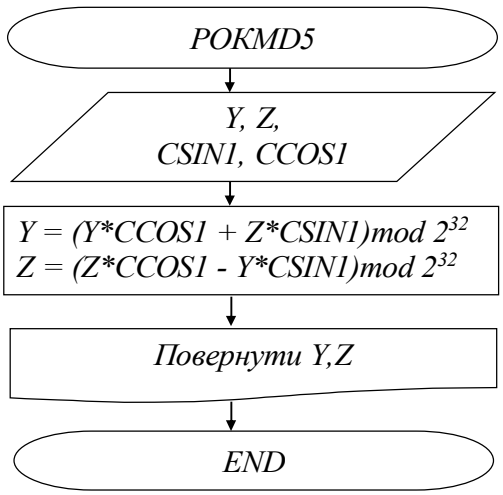

Рис. 5. Структура алгоритму рекурентного обчислення рухомого коефіцієнту алгоритму MD5

Як випливає із співвідношень (15), (16), а також із структури розробленого алгоритму уточнення тригонометричних коефіцієнтів алгоритму хешування MD5 (рис. 5), для обчислення одного тригонометричного коефіцієнту алгоритму хешування MD5 треба виконати всього чотири множення 32-бітових чисел на 32-бітові константи, а також два додавання 32-бітових чисел.

За допомогою розроблених математичної (15, 16) та програмної (рис. 5) моделей різницевого алгоритмічного обчислювача змінних коефіцієнтів алгоритму хешування MD5 було обчислено всі 64 значення цих коефіцієнтів. Отримані результати обчислення збігаються з наведеними в літературі константними значеннями [1].

6. Результати дослідження

За результатами дослідження доцільності застосування класичних число-імпульсних обчислювачів функції синуса для розрахунку коефіцієнтів алгоритму хешування MD5 встановлено, що подібні обчислювачі не можуть застосовуватися для обчислення змінних тригонометричних коефіцієнтів алгоритму хешування MD5. Адже під час обчислення функції синуса в алгоритмі хешування MD5 аргумент треба змінювати від одного до 64 радіан, а класичні число-імпульсні обчислювачі функції синуса працюють в діапазоні (11). Іншими словами, такі обчислювачі працюють в одному квадранті: допускають зміну значень аргументу від нуля до $\pi/2$.

За результатами дослідження доцільності застосування реверсивних широкодіапазонних число-імпульсних обчислювачів функції синуса для розрахунку коефіцієнтів алгоритму хешування MD5 встановлено, що такі обчислювачі можуть працювати в безмежному діапазоні (12) зміни значень аргументу. Водночас, якщо детальніше проаналізувати функцію перетворення (7) і взяти до уваги той факт, що під час обчислення змінних тригонометричних коефіцієнтів алгоритму хешування MD5 аргумент треба змінювати від одного до 64 радіан з кроком один радіан, доведеться визнати, що структура на рис. 4 також не придатна для обчислення коефіцієнтів алгоритму хешування MD5. Адже відповідно до співвідношення (7) приросту аргументу X в один радіан відповідає 2^n одиничних приростів вхідного число-імпульсного коду структури. Це означає, що для оброблення приросту аргументу в один радіан через число-імпульсну структуру (рис. 4) треба пропустити 2^n одиничних приростів вхідного число-імпульсного коду. В пристосуванні до алгоритму хешування MD5, в якому треба отримати 32-бітовий результат, через таку структуру потрібно на кожен радіан пропускати 2^{32} одиничних приростів. За таких умов очевидно, що навіть на найвищих доступних тактових частотах такі обчислювачі будуть мати критично низьку швидкодію.

Аналізуючи результати дослідження розроблених різницевих обчислювачів тригонометричних коефіцієнтів алгоритму хешування MD5, отримаємо значно кращі результати. Як випливає із співвідношень (15), (16), а також із структури розробленого алгоритму уточнення тригонометричних коефіцієнтів алгоритму хешування MD5 (рис. 5), для обчислення одного тригонометричного коефіцієнту алгоритму хешування MD5 потрібно виконати всього чотири множення 32-бітових чисел на 32-бітові константи, а також два додавання 32-бітових чисел.

За допомогою розроблених математичної (15, 16) та програмної (рис. 5) моделей різницевого алгоритмічного обчислювача змінних коефіцієнтів алгоритму хешування MD5 було обчислено всі 64 значення цих коефіцієнтів. Отримані результати обчислення збігаються з наведеними в літературі константними значеннями [1].

Висновки

Досліджені в статті алгоритми хешування є дієвим інструментом забезпечення автентичності даних. Сучасні алгоритми хешування можна поділити на дві великі групи. Алгоритми першої групи будують на основі швидких (зазвичай логічних) обчислювальних операцій. Такі алгоритми характеризуються високою швидкодією і як наслідок – високою обчислювальною ефективністю. Водночас стійкості алгоритмів хешування першої групи часто недостатньо для захисту особливо важливої інформації. Тому такі алгоритми застосовують переважно для захисту комерційної інформації.

Алгоритми хешування другої групи будують на базі стандартних симетричних блокових шифрів. Такі алгоритми характеризуються гарантованою обчислювальною стійкістю, а тому можуть застосовуватися для захисту особливо важливої інформації. Однак швидкодія таких алгоритмів хешування істотно нижча за швидкодію алгоритмів першої групи.

Поширений алгоритм хешування MD5, який досліджувався в роботі, належить до алгоритмів першої групи. Відповідно основною перевагою цього алгоритму є його швидкодія. Вважається, що стійкості алгоритму сьогодні недостатньо для захисту інформації особливої ваги, тому застосовують його для захисту комерційної інформації.

Особливістю алгоритму хешування MD5 є застосування змінних тригонометричних коефіцієнтів на різних раундах алгоритму. Ці коефіцієнти можуть обчислюватися таблично. Тобто в пристрої хешування потрібно передбачити пам’ять для зберігання 64-ох 32-бітових констант. Для економії пам’яті, за наявності в пристрої хешування потрібних обчислювальних засобів, можна на кожному раунді обчислювати черговий коефіцієнт.

Змінні тригонометричні коефіцієнти алгоритму хешування MD5 – це значення цілої частини функції $2^{32} * \text{abs}(\sin(i))$, де значення аргументу і задається в радіанах та змінюється від 1 до 64 радіана. Оскільки йдеться про табулювання функції, а сама функція обчислювально достатньо складна, в роботі було досліджено різницеві методи обчислення такої функції. Насамперед було досліджено можливість і доцільність застосування число-імпульсних тригонометричних обчислювачів [5]. За результатами проведених досліджень виявлено, що класичний число-імпульсний обчислювач функції синуса не може застосовуватися для обчислення коефіцієнтів алгоритму хешування MD5 через обмежений діапазон перетворення. Натомість широкодіапазонний реверсивний число-імпульсний обчислювач функції синуса має необмежений діапазон перетворення. Проте за результатами досліджень було виявлено, що такий обчислювач також недоцільно застосовувати для обчислення коефіцієнтів алгоритму хешування MD5. Причиною є те, що число-імпульсний обчислювач синуса розгортає загальний приріст аргументу (один радіан в MD5) в послідовність з 2^{32} одиничних імпульсів. Тобто для оброблення одного радіана через число-імпульсну структуру треба “пропустити” 2^{32} імпульсів число-імпульсного коду. А це цілковито не прийнятно з погляду швидкодії навіть за найвищих показників тактової частоти обчислювача.

Для забезпечення можливості алгоритмічного обчислення змінних коефіцієнтів алгоритму хешування MD5 в роботі розроблено математичну і програмну модель структури розгортання

функції синуса. Математична модель ґрунтується на співвідношеннях для синуса і косинуса суми аргументів, які адаптовані для алгоритму хешування MD5. Програмна модель потребує для обчислення одного тригонометричного коефіцієнту всього чотири множення і два додавання 32-бітних чисел. Застосування розробленої різницевої обчислювальної структури дає змогу економити пам'ять під час реалізації алгоритму на засобах з обмеженими ресурсами пам'яті.

Список літератури

1. Rivest R. The MD5 Message-Digest Algorithm. Technical Report Internet. RFC-1321, IETF, 1992. Available at: https://www.ietf.org/rfc/rfc1321.txt (Accessed: 26 February 2024).

2. Schneier B. One-Way Hash Functions, Dr. Dobb's fournal, vol. 16, No. 9, Sep. 1991, pp. 148–151. Available at:https://doi.org/10.1002/9781119183471.ch18 (Accessed: 26 February 2024).

3. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C. John Wiley and Sons, New York, second edition, 1998, DOI:10.1002/9781119183471.

4. Gorpeniuk A. Fast algorithms and computing means of cryptological functions, International Scientific Journal of Computing. October 2005, Vol. 4, Issue 2, pp. 69–76. DOI: https://doi.org/10.47839/ijc.4.2.339.

5. Horpenyuk A., Dudykevych V., Luzhetska N. (2009). Conveyor sine-cosine pulse- number functional converter, Automation, measurement and control, Lviv Polytechics, Num.639, pp.94-101. (in Ukrainian). Available at: https://vlp.com.ua/files/13_4.pdf (Accessed: 26 February 2024).

6. Yang Y., Bi J., Chen X., Yuan Z., Zhou Y. and Shi W. (2018). Simple hash function using discrete-time quantum walks. Quantum Information Processing. 17:8. (1–19). Online publication date: 1-Aug-2018. Available at: https://doi.org/10.1007/s11128-018-1954-2 (Accessed: 26 February 2024).

7. Faragallah O. (2018). Secure Audio Cryptosystem Using Hashed Image LSB watermarking and Encryption. Wireless Personal Communications: An International Journal. 98:2. (2009–2023). Online publication date: 1-Jan-2018. Available at: https://doi.org/10.1007/s11277-017-4960-2 (Accessed: 26 February 2024).

RESEARCH AND IMPROVEMENT OF COMPUTING ALGORITHMS FOR CALCULATING THE TRIGONOMETRICAL COEFFICIENTS OF THE HASHING ALGORITHM MD5

A. Horpenyuk [1], N. Luzhetska [2], M. Horpenyuk [3]

Lviv Polytechnic National University,
[1,3] Information Security Department, [2] Department of Information Technology Security

© Horpenyuk A., Luzhetska N., Horpenyuk M., 2024

The paper examines the problems of ensuring the authenticity of messages, as well as analyzes the modern requirements for hash functions and the problems of designing algorithms for calculating hash functions.

The common MD5 hashing algorithm was investigated. These days, its level of security is considered insufficient for protecting high-level data confidentiality. However, it is an effective and fast algorithm for hashing messages and is successfully used to protect commercial information. The paper examines the main computational transformations of the MD5 hashing algorithm. It is shown that variable constants are used in the MD5 algorithm to improve stability. A sweep of the sine function is used to calculate these variable constants.

The paper examines the feasibility of using number-pulse computing structures for the calculation of variable trigonometric constants of the MD5 hashing algorithm. It is shown that the use of classical number-pulse computing structures is impractical due to the insufficient range of reproduction of the necessary trigonometric functions. Advanced wide-band digital-pulse structures provide the necessary conversion function, range and accuracy. However, the speed of such calculators is critically insufficient to calculate all the trigonometric coefficients of the MD5 hashing algorithm.

The paper developed a mathematical and software model of the structure of the sine function expansion for the MD5 algorithm. The mathematical model is based on the relations for the sine and cosine of the sum of the arguments, which are adapted for the MD5 hashing algorithm. The use of the developed differential computing structure allows saving memory when implementing the algorithm on devices with limited memory resources.

Keywords: cryptography, message authenticity, hash function.