

ПОРІВНЯННЯ ПЕРСПЕКТИВНИХ ШВИДКОДІЮЧИХ ФУНКЦІЙ ГЕШУВАННЯ

І.Д.ГОРБЕНКО, А.В. БОЙКО, А.М. ГЕРЦОГ

Запропоновані критерії та показники оцінки перспективних функцій гешування, обґрунтовано та розроблено методики їх порівняння та відповідні рекомендації.

The paper suggests criteria and indices of evaluating perspective hash functions, substantiates and develops methods of their comparison and appropriate recommendations.

ВСТУП

Функції гешування є криптографічними примітивами, що широко використовуються практично у всіх електронних цифрових підписах (ЕЦП), в деяких симетричних крипто примітивах, криптографічних механізмах і протоколах автентифікації, встановлення, узгодження, підтвердження ключів тощо. Крім, зрозуміло, криптографічної стійкості до них висуваються жорсткі вимоги відносно складності (швидкості) гешування. Причому, це особливо важливо при виконанні ЕЦП та генеруванні детермінованих випадкових послідовностей (псевдовипадкових послідовностей). Разом із тим до функцій гешування висуваються і інші додаткові вимоги.

Попередній аналіз [1-7] стану вирішення вказаної проблеми дозволив зробити висновок, що відповідного методичного та математичного апарату оцінювання алгоритмів не має. Тому розробка методичного та математичного забезпечень порівняння функцій гешування з метою вибору кращої із запропонованих кандидатів є на даний момент актуальним завданням. Процедура порівняння ускладнюється тим, що розробники пропонують різні підходи до створення функцій гешування [3-6]. Так, наприклад, одним із сучасних підходів до побудування функцій гешування є створення архітектурних моделей, в яких з метою збільшення швидкодії використовуються паралельні обчислення. Далі такі функції гешування називатимемо паралельними. Тобто, зважаючи на сучасні тенденції розвитку, виникає необхідність удосконалення вже існуючої системи критеріїв і показників і формулювання методики порівняння функцій гешування з урахуванням нових пропозицій, щодо особливостей їх архітектури. Також, з розвитком підходів до побудування функцій гешування такий показник, як швидкодія вже не є однозначним. Наприклад, для паралельної функції гешування значення швидкодії на процесорах з декількома обчислювальними ядрами відрізняється від значення швидкодії на процесорах з одним обчислювальним ядром. Також існує суттєва різниця у виконанні паралельних обчислень у програмних, апаратно-програмних та апаратних реалізаціях. Так у програмних реалізаціях потоки виконання по своїй суті є конкуруючими, тому необхідні додаткові витрати на синхронізацію тощо.

Метою цієї статті є осмислення проблемних питань відносно основних вимог, критеріїв та показників оцінки, обґрунтування та розроблення методики оцінки перспективних функцій гешування і, як наслідок, можливості її застосування для оцінювання перспективних функцій гешування міжнародного проекту SHA – 3[2].

Особливістю рішень є аналіз складних вимог відносно складності гешування, переваг, недоліків та проблемних питань паралельних реалізацій функцій гешування, а також забезпечення певної адекватності оцінювання і порівняння функцій гешування. В подальшому також будемо вважати, що функція гешування дозволяє обчислювати вихідне значення з довжиною $hlen$ при довільній довжині l_M вхідного повідомлення M .

1. АНАЛІЗ ВИМОГ ДО ФУНКЦІЙ ГЕШУВАННЯ

Аналіз існуючих вимог [2] показав, що сучасні та перспективні функції гешування повинні неодмінно відповідати таким вимогам:

1) складність знаходження колізії

$$C_{col} \geq 2^{hlen/2}; \quad (1)$$

2) складність відновлення прообразу M

$$C_{preim} \geq 2^{hlen}; \quad (2)$$

3) складність знаходження іншого прообразу

$$C_{sec_preim} \geq 2^{hlen}; \quad (3)$$

4) складність знаходження колізії, усіченої на ht символів

$$C_{tr_col} \geq 2^{(hlen-h)/2}. \quad (4)$$

Наведені вище вимоги, по суті, є безумовними критеріями оцінки функцій гешування.

Фактично розглянуті критерії безпеки функції гешування є булевими змінними, які приймають значення «1» (істина), якщо відповідна умова виконується, та «0» в іншому випадку. При цьому стійкість J -ї функції гешування можливо описати вектором булевих змінних $(w_1^{(j)}, w_2^{(j)}, w_3^{(j)}, w_4^{(j)})$, які формально можуть бути визначені як:

$$\begin{aligned}
 w_1^{(j)} &= \begin{cases} 1, & C_{col} \geq 2^{hlen/2} \\ 0 & \end{cases} \\
 w_2^{(j)} &= \begin{cases} 1, & C_{preim} \geq 2^{hlen} \\ 0 & \end{cases} \\
 w_3^{(j)} &= \begin{cases} 1, & C_{sec_preim} \geq 2^{hlen} \\ 0 & \end{cases} \\
 w_4^{(j)} &= \begin{cases} 1, & C_{tr_col} \geq 2^{(hlen-hl)/2} \\ 0 & \end{cases}
 \end{aligned} \quad (5)$$

Вектор $(w_1^{(j)}, w_2^{(j)}, w_3^{(j)}, w_4^{(j)})$ будемо з деяким уточненням називати інтегральним безумовним критерієм.

Щодо такого показника, як складність (швидкодія) гешування, то на даний момент, на наш погляд, він вже не є однозначним, і тому необхідно визначити і обґрунтувати нові показники та відповідні умовні критерії.

Зважаючи на те, що обчислення функції гешування може здійснюватись паралельно, розглянемо загальні питання паралельних обчислень та вимоги до них.

Згідно закону Амдала, сформульованому в [1], коефіцієнт прискорення K_N при здійсненні N паралельних обчислень визначається за формулою:

$$K_N = \frac{1}{(1-P) + \frac{P}{N}}, \quad (6)$$

де P — частка паралельних обчислень, $1-P$ — відповідно частка послідовних обчислень.

Фізичний зміст цього коефіцієнту прискорення K_N — відношення нормованої швидкодії алгоритму з використанням паралельних обчислень, при його виконанні на N обчислювальних ядрах, до швидкодії цього ж алгоритму при його виконанні на одному ядрі.

Однак, як показав аналіз, визначити частку паралельних обчислень в алгоритмі, базуючись на його специфікації, практично неможливо. В той же час розрахувати K_N коефіцієнт прискорення при використанні відповідної кількості обчислювальних ядер можливо, використовуючи результати вимірювань швидкодії функції гешування на одному та кількох обчислювальних ядрах.

$$K_N = \frac{S_1}{S_N}, \quad (7)$$

де S_1 — швидкодія алгоритму гешування на одному обчислювальному ядрі, а S_N — швидкодія алгоритму гешування на N обчислювальних ядрах.

З формули (7) можна вивести формулу (8), за якою можливо обчислити частку паралельних обчислень, знаючи коефіцієнт прискорення на N обчислювальних ядрах і кількість цих ядер, яка має вигляд

$$P = \frac{(K_N - 1)N}{(N - 1)K_N}, \quad (8)$$

де як і раніше K_N — коефіцієнт прискорення на N обчислювальних ядрах, P — частка паралельних обчислень.

Результати досліджень [1] показали, що нарощування кількості обчислювальних ядер має сенс до певної величини. Причому кожне нове додане обчислювальне ядро дає менший приріст швидкодії, ніж попереднє. Коефіцієнт прискорення $K_{\max}$ при цьому обмежений зверху значенням $\frac{1}{1-P}$. Тобто

$$K_{\max} = \lim_{N \rightarrow \infty} \left(\frac{1}{1-P+P/N} \right) = \frac{1}{1-P}. \quad (9)$$

Однак на практиці уже визнано, що максимальне значення коефіцієнту прискорення є недосяжним, тому з нього в загальному випадку неможливо обчислити максимальну кількість ядер, що можуть ефективно використовуватися алгоритмом. У зв'язку з цим пропонується наступний підхід до вирішення цієї задачі.

Для обчислення максимально досяжного прискорення пропонується використовувати замість $K_{\max}$ деяке $K'_{\max}$, яке близьке до $K_{\max}$, але менше нього. Наскільки $K'_{\max}$ близьке до $K_{\max}$, визначимо, використовуючи значення коефіцієнту наближення r , який близький до одиниці, але менше неї

$$K'_{\max} = rK_{\max}. \quad (10)$$

Рекомендоване для практичних розрахунків значення $r = 0,95$. Вибір цього значення обумовлено тим фактом, що збільшення кількості обчислювальних ядер після досягнення $K'_{\max}$ практично не впливає на швидкодію. Тому саме це значення доцільно використовувати для обчислення інших показників.

2. ВИБІР КРИТЕРІЇВ ТА ПОКАЗНИКІВ ОЦІНКИ ФУНКЦІЙ ГЕШУВАННЯ

Усі критерії, які можливо застосувати до геш-функцій, поділимо на два класи: безумовні (функції гешування, що оцінюються, повинні обов'язково відповідати цим критеріям) та умовні (якими можна поступитися заради досягнення відповідності безумовним критеріям та/або іншим умовним критеріям).

До безумовних критеріїв відносяться всі критерії, що пов'язані зі стійкістю. Це, по суті, часткові безумовні критерії (1) — (4). Згідно (5) інтегральний критерій формулюється наступним чином: геш-функція вважається прийнятною, якщо відповідний частковий показник має значення «істина», тобто геш-функція є стійкою проти атак типу (1) — (4).

В якості умовних показників, які однозначно характеризують швидкість роботи геш-функції, пропонується максимально досяжна швидкодія і

максимальна кількість обчислювальних ядер, які можуть бути ефективно задіяні в паралельних обчисленнях.

Під максимально досяжною швидкодією розуміється швидкодія алгоритму обчислення функції гешування, яка може бути принципово досягнута для оптимізованої реалізації у випадку використання паралельних обчислень і відповідно використання максимальної кількості обчислювальних ядер, які можуть бути ефективно задіяні в обчисленні геш-значення.

Відповідний критерій формулюємо наступним чином: в парі функцій гешування перевага надається тій функції гешування, яка має більшу максимально досяжну швидкодію.

Під максимальною кількістю обчислювальних ядер, які можуть бути ефективно використані алгоритмом, розуміється така кількість обчислювальних ядер, перевищення якої не принесе значимого приросту швидкодії при використанні паралельних обчислень в алгоритмі.

Умовний критерій для заданих умов формулюється наступним чином: в парі функцій гешування, що використовують паралельні обчислення і мають однакову максимально досяжну швидкодію, перевага надається тій функції гешування, що має менше значення максимальної кількості обчислювальних ядер. Обґрунтування цього критерію в тому, що паралельні обчислення використовуються для підвищення швидкодії, а отже за однакової швидкодії функція гешування, що використовує меншу кількість обчислювальних ядер, матиме більш дешеву і просту реалізацію.

Основні співвідношення для обчислення максимально досяжної швидкодії вже було наведено вище. Підставивши формулу (9) у (10), отримаємо уточнену формулу для обчислення максимально досяжного коефіцієнту прискорення.

$$K'_{\max} = r \frac{1}{1-P}, \quad (11)$$

де як і раніше r — коефіцієнт наближення, P — частка паралельних обчислень.

З (11) та (9) випливає, що максимально досяжна швидкодія може бути обчислена як

$$S_{\max} = S_1 r \frac{1}{1-P}, \quad (12)$$

де S_1 — швидкодія алгоритму гешування на одному обчислювальному ядрі функції гешування.

З формули (8) отримуємо вираз для визначення максимального значення коефіцієнту прискорення

$$N_{\max} = \frac{P}{\frac{1}{K'_{\max}} - 1 + P}, \quad (13)$$

де $K'_{\max}$ — значення максимального коефіцієнту прискорення, розраховане за формулою (11).

Слід зауважити, що внаслідок присутності залежностей між даними в алгоритмах, максимальна кількість обчислювальних ядер і максимальна досяжна швидкодія можуть в деяких випадках відрізнятись від обчисленої.

3. МЕТОДИКИ ПОРІВНЯННЯ ФУНКЦІЙ ГЕШУВАННЯ ПО ШВИДКОДІЇ

Аналітичний огляд літературних джерел дозволив виділити щонайменше три підходи до реалізації паралельних обчислень функцій гешування:

1) використання класичної архітектури Меркле — Дамгарда (Merkle-Damgard) [3] з паралельним виконанням частини операцій всередині ітерації;

2) використання архітектури Міхіра Белларе [4] з незалежним перетворенням вхідних блоків повідомлення і комбінуванням результатів цього перетворення;

3) гешування по дереву [5].

Ітеративна архітектура Меркле-Дамгарда є класичною архітектурою реалізації функції гешування. В ній вхідне повідомлення M розбивається на блоки певної довжини $M = M_0 \parallel \dots \parallel M_i \parallel \dots \parallel M_n$, після чого відбувається ітеративне обчислення за наступними формулами [3]:

$$\begin{aligned} h_0 &= f(M_0, IV), \\ h_i &= f(M_i, h_{i-1}), \end{aligned} \quad (14)$$

де IV — початковий вектор (ключ) гешування.

Після обчислення h_n для останнього блока повідомлення M_n (спосіб доповнення до повного блоку не має значення) обчислюється значення певної функції гешування.

Отримане в результаті значення і є тим геш-значенням, яке необхідно обчислити.

Як видно з розгляду, наведена вище архітектура є послідовною, тому програмні реалізації функцій гешування з такою архітектурою нездатні ефективно використовувати обчислювальні потужності сучасних багатоядерних процесорів.

Розглянемо класичну архітектуру Меркле-Дамгарда [3,6] з паралельним виконанням частини операцій всередині ітерації. В [6] пропонується використовувати паралельні обчислення всередині ітерації. Проте в такому разі обчислення потребують або спеціальних інструкцій процесора (таких як MMX, SSE та інші), або чіткої синхронізації між потоками в програмних реалізаціях. Вказане вимагає деяких додаткових потужностей, внаслідок чого реальний вигравш при здійсненні паралельних обчислень геш-значень завжди менше того, що очікується. Крім того, для даного алгоритму, як правило, наявні залежності даних, які необхідно враховувати при обчисленнях. Залежність даних проявляється в тому, що незважаючи на велику частку паралельних обчислень в алгоритмі, кількість обчислювальних ядер, що можуть бути ефективно використані для паралельних обчислень, менша, ніж обчислена за законом Ам-

дала і визначається особливостями внутрішньої структури ітерації алгоритму.

З урахуванням вказаного вище, пропонується наступна методика 1 обчислення пари показників $\{S_{\max}, N_{\max}\}$ для архітектури Меркле-Дамгарда.

1. Виміряти швидкодію S_1, S_2 для алгоритмів з архітектурою Меркле-Дамгарда (Merkle-Damgard), використовуючи два обчислювальні ядра. Значення кількості обчислювальних ядер для вимірювань продиктовані тим фактом, що до початку вимірювання значення максимальної кількості обчислювальних ядер невідоме, а перевищення цього значення значно погіршить точність оцінювання. При цьому, для функцій гешування, у яких не використовуються паралельні обчислення, значення S_1, S_2 практично співпадуть, а значення $K'_{\max}$ та $N_{\max}$ дорівнюватимуть одиниці.

2. Обчислити коефіцієнт прискорення K_2 за формулою (7).

3. За формулою (8) обчислити частку паралельних обчислень в алгоритмі.

4. Шляхом експертного аналізу специфікації алгоритму визначити наявність залежностей даних і максимальну кількість обчислювальних ядер, що можуть бути ефективно використані алгоритмом.

5. За формулами (11) та (13) обчислити максимальну кількість обчислювальних ядер, які можуть ефективно використовуватись для обчислення значення функції гешування.

6. Порівняти значення, що отримані на кроках 4 та 5, і для подальших розрахунків використовувати менше з них.

7. На основі формул (6) та (7) обчислити максимально досягну швидкодію $S_{\max}$.

$$S_{\max} = \frac{S_1}{(1-P) + \frac{P}{N_{\max}}} \quad (15)$$

Таким чином, запропонована методика дозволяє отримати пару значень показників $\{S_{\max}, N_{\max}\}$ для функції гешування з архітектурою Меркле-Дамгарда і відповідно паралельними обчисленнями всередині ітерації.

Архітектура функції гешування, запропонована Міхіром Белларе у роботі [4], повністю орієнтована на використання паралельних обчислень. Схема цієї архітектури показана на рис. 1.

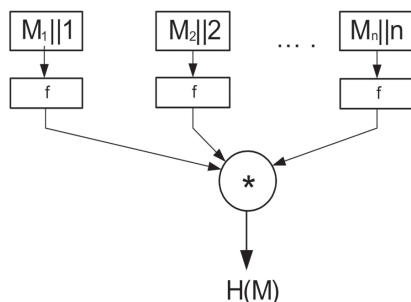


Рис. 1. Архітектура функції гешування Міхіра Белларе

Геш-функція з архітектурою Міхіра Белларе має два основних компонента – функція стискання f та операція комбінування $*$. Як видно з рис. 1, кожний блок даних обробляється незалежно, тому існує можливість здійснення їх паралельної обробки. Якщо комбінуючи, операція є комутативною, то результати обробки блоків даних функцією f можуть бути подані на її вхід навіть не в тому порядку, в якому слідують вхідні блоки повідомлення. В такій архітектурі відсутні залежності даних, тому для обчислення значень показників функцій гешування з такою архітектурою можна використати наступну методику.

Методика 2 обчислення пари показників $\{S_{\max}, N_{\max}\}$ для архітектури Міхіра Белларе.

1. Виміряти швидкодію алгоритму S_1 та S_2 для архітектури Міхіра Белларе, використовуючи два ядра.

2. Обчислити коефіцієнт прискорення K_2 за формулою (7).

3. За формулою (8) обчислити частку паралельних обчислень в алгоритмі.

4. За формулою (15) обчислити максимально досягну швидкодію $S_{\max}$.

5. За формулами (11) та (13) обчислити максимальну кількість обчислювальних ядер, що можуть ефективно використовуватись для обчислення геш-значення.

В результаті виконання обчислень для функцій гешування згідно методики 2 отримуємо пару значень показників $\{S_{\max}, N_{\max}\}$.

Паралельні обчислення геш-значення можна також реалізувати за допомогою архітектури «дерево». Цей підхід було запропоновано в роботі [5]. Основою реалізації паралельних обчислень в ньому є приведення структури алгоритму гешування до структури двійкового дерева порядку t (рис. 2).

Кількість обчислювальних ядер у дереві порядку t визначається виразом 2^t . Між собою обчислювальні ядра з'єднані таким чином, що дані з виходу обчислювальних ядер P_{2i} та P_{2i+1} подаються на вхід обчислювального ядра P_i .

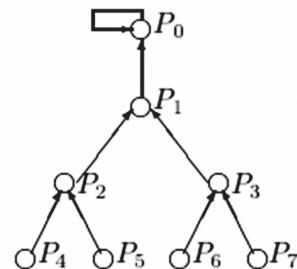


Рис. 2. Двійкове дерево порядку $t = 3$

Кожне обчислювальне ядро реалізує колізійно стійку функцію стискання f , яка приймає на вхід бітовий рядок довжиною n і повертає бітовий рядок довжиною m , причому $n \geq 2m$. У роботі [5] представлено інкрементальний алгоритм, що реалізує двійкове дерево порядку t на 2^t обчислювальних ядрах, де $0 \leq t' \leq t-1$ з відповідним

зростанням часу роботи алгоритму. Тобто якщо реалізувати цей алгоритм для дерева порядку t на 2^{t-1} обчислювальних ядрах, то час виконання збільшиться у два рази порівняно з реалізацією на 2^t обчислювальних ядрах. Для $t=0$ ця архітектура перетворюється в послідовну архітектуру Меркле-Дамгарда. Тому для алгоритму обчислення геш-значення по дереву максимальне число обчислювальних ядер $N_{\max}$, що можуть бути ефективно використані для обчислення, дорівнює 2^t , де t – порядок дерева.

Значення показника максимально досяжної швидкодії для такого алгоритму можна виміряти безпосередньо на $N_{\max}$ обчислювальних ядрах, якщо доступна така конфігурація ЕОМ, або обчислити з результатів вимірювання швидкодії реалізації інкрементального алгоритму на меншому числі обчислювальних ядер з відповідним перерахунком до $N_{\max}$.

Аналітичний вираз для визначення коефіцієнту прискорення такого алгоритма паралельних обчислень, за умови, що він виконується на $N_{\max}$ обчислювальних ядрах, представлено у роботі [5]. Коефіцієнт прискорення K для дерева визначається як

$$K = \frac{\Psi}{R}, \quad (16)$$

де Ψ – кількість звернень до функції стискування, R – загальна кількість операцій, що виконуються паралельно. Число звертань для дерева можна визначити, використовуючи вираз [5]

$$\Psi = (q+2)2^t + 2 \left\lceil \frac{r}{2n-2m} \right\rceil - 1, \quad (17)$$

де n – довжина блоку в бітах на вході функції стискування, q – кількість повних блоків повідомлення, які подаються на вхід дерева, t – порядок дерева, r – довжина в бітах останнього неповного блоку, m – довжина виходу функції стискування f в бітах. Значення R обчислюється як

$$R = q + t + 2, \quad (18)$$

де q – кількість повних блоків повідомлення, які подаються на вхід дерева, t – порядок дерева, а величина K

$$K = \frac{(q+2)2^t + 2 \left\lceil \frac{r}{2n-2m} \right\rceil - 1}{q + t + 2}. \quad (19)$$

Як видно з (17), зі зростанням q коефіцієнт прискорення наближається до 2^t , дійсно

$$\lim_{q \rightarrow \infty} \frac{(q+2)2^t + 2 \left\lceil \frac{r}{2n-2m} \right\rceil - 1}{q + t + 2} = 2^t. \quad (14)$$

Для практичних обчислень в [5] пропонується використовувати наближену формулу

$$K \approx 2^t \left(\frac{1}{1 + t/(q+2)} \right), \quad (20)$$

де q – кількість блоків повідомлення, які подаються на вхід дерева, t – порядок двійкового дерева.

Аналіз виразу (20) показує, що високий порядок двійкового дерева забезпечить збільшення швидкості обчислень тільки для «довгих» повідомлень. Причому, як і у випадку з архітектурою Міхіра Белларе, максимальне значення коефіцієнту прискорення 2^t є недосяжним. Тому для дерева при визначенні коефіцієнту максимального прискорення $K_{\max}$ необхідно вважати, що повідомлення є достатньо довгим, але коли коефіцієнт прискорення досягає $2^t r$.

Число повних блоків q повідомлення, що подаються на вхід дерева, яке необхідно для досягнення максимального прискорення, визначається з наведеного наближення для визначення коефіцієнта швидкодії (20).

Крім того, оскільки теоретичної межі прискорення досягнути неможливо, то для практичного визначення коефіцієнту максимальної швидкодії пропонується використовувати коефіцієнт наближення $r=0,95$, тому

$$K'_{\max} = 0,95 \cdot 2^t, \quad (21)$$

де t – порядок двійкового дерева, а $K'_{\max}$ – максимальний коефіцієнт прискорення.

Підставляючи замість K у (20) значення $K'_{\max}$ з (21), визначимо кількість блоків повідомлення q , необхідних для досягнення максимальної швидкодії:

$$q = \frac{t}{\left(\frac{1}{0,95} - 1 \right)} - 2, \quad (22)$$

де як і раніше q – кількість повних блоків повідомлення, t – порядок двійкового дерева.

Практично для розрахунків показників функцій гешування з подібною архітектурою можливо використати співвідношення

$$N_{\max} = 2^t,$$

$$S_{\max} = K'_{\max} S_1. \quad (23)$$

4. ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ

Раніше вказувалось, що кожна функція гешування повинна задовольняти всім безумовним критеріям. Тому, як запропоновано в [7], узагальнений безумовний критерій W пропонується записувати за допомогою булевої операції кон'юнкції

$$W = w_1 \wedge w_2 \wedge w_3 \wedge w_4. \quad (24)$$

В той же час наявність двох умовних критеріїв виду (23) обумовлює складність процедури неформального порівняння та вибору. Їх неможливо

об'єднати між собою і з узагальненим безумовним критерієм за допомогою булевих операцій. Для цього випадку пропонується використати деяку узагальнюючу цільову функцію $D(W, S_{\max}, N_{\max})$, максимальне значення якої відповідатиме «найкращій» функції ґешування. Наведемо основні властивості цієї функції.

1. Для всіх можливих значень $S_{\max}, N_{\max}$ виконується умова

$$D(0, u_1, u_2) = 0. \quad (25)$$

2. Для всіх можливих значень $N_{\max}$ та $\Delta S > 0$ виконується нерівність

$$D(1, S_{\max} + \Delta S, N_{\max}) > D(1, S_{\max}, N_{\max}). \quad (26)$$

3. Для всіх можливих значень $S_{\max}$ та $\Delta N > 0$ виконується нерівність

$$D(1, S_{\max}, N_{\max} + \Delta N) < D(1, S_{\max}, N_{\max}). \quad (27)$$

Для побудування такої цільової функції можна скористатися методами підтримки прийняття рішень, наприклад методом аналізу ієрархій. Співвідношення (25), (26) та (27) можуть бути використані для перевірки адекватності побудованої цільової функції D .

ВИСНОВКИ

Запропоновані критерії оцінки швидкодії алгоритму ґешування дозволяють дати більш повну та адекватну оцінку переваг однієї функції ґешування над іншою, та здатні враховувати зміни в підходах до побудування функцій ґешування, перше за все зважаючи на сучасні тенденції розвитку комп'ютерної техніки.

Методики порівняння функцій ґешування розроблені з урахуванням архітектурних особливостей основних на даний момент підходів, що реалізують паралельні обчислення ґеш-значень.

Результати порівняння функцій ґешування за розглянутим набором критеріїв можуть бути легко формалізовані, що дозволить застосувати методи оптимізації для підтримки прийняття рішень. Такий підхід максимально спрощує та дає можливість автоматизувати процес вибору функції ґешування.

Література.

- [1] Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities", AFIPS Conference Proceedings, (30), pp. 483-485, 1967.
- [2] NIST Computer security resource center. Announcing request for Candidate algorithm nominations for a new cryptographic hash algorithm nominations for a new cryptographic hash algorithm (SHA-3) family.
- [3] I. Damgard. A Design Principle for Hash Functions. In Advances in Cryptology - CRYPTO '89 Proceedings, Lecture Notes in Computer Science Vol. 435, G. Brassard, ed, Springer-Verlag, 1989, pp. 416-427.
- [4] Bellare, M. A new paradigm for collision-free hashing: incrementally at reduced cost.
- [5] Sarkar, P. A parallelizable design principle for cryptographic hash functions.
- [6] Schnorr, C.P. Parallel FFT-Hashing.
- [7] Бондаренко М.Ф., Денисенко Б.И., Горбенко Ю.И. Методика сравнения алгоритмов ЭЦП в группе точек эллиптической кривой. Прикладная радиоэлектроника, Том 7, 2008, №3, с.256–263.

Надійшла до редколегії 21.09.2009



Горбенко Іван Дмитрович, доктор технічних наук, професор, завідувач кафедри безпеки інформаційних технологій ХНУРЕ, головний конструктор ЗАТ «Інститут інформаційних технологій». Область наукових інтересів: проектування та розробка систем та засобів криптографічного захисту інформації.



Бойко Артем Олександрович, аспірант кафедри безпеки інформаційних технологій ХНУРЕ. Область наукових інтересів: ґеш-функції, побудування крипто примітивів з підвищеною швидкістю.



Герцог Андрій Миколайович, стажер-дослідник кафедри безпеки інформаційних технологій ХНУРЕ. Область наукових інтересів: дослідження перспективних алгоритмів ґешування.