що з врахуванням тотожності (32) дозволяє записати тотожність

$$[L]_I (a^I{}_a - \partial_\sigma b^{I\sigma}{}_a) - \partial_\sigma [L]_I b^{I\sigma}{}_a = 0 . \quad (37)$$

Якщо лагранжіан залежить максимум від перших похідних від полів, тобто при $n = 1$, маємо:

$$J^\sigma_a = -\partial^\sigma_I L a^I{}_a - L h^\sigma_a , \quad (38)$$

$$S^{\mu\sigma}_a = J^{\mu\sigma}_a = -\partial^\sigma_I L b^{I\mu}{}_a , \quad (39)$$

а формула (35) спрощується до

$$J^\mu_a = \partial_I L b^{I\mu}{}_a + \partial^\sigma_I L \partial_\sigma b^{I\mu}{}_a . \quad (40)$$

Висновки. Доведена теорема і спосіб її доведення дають алгоритм побудови струмів калібрувальних зарядів, а також їх суперпотенціалів для широкого кола теорій з калібрувальними симетріями. Найбільш цікаві і практично важливі приклади таких теорій ми плануємо розглянути в подальшому.

ЛІТЕРАТУРА

1. Noether E. Invariant variation problems. *Transport theory and statistical physics*. 1971. № 1(3). P.183-207.
2. Szabados L.B. Quasi-local energy-momentum and angular momentum in GR: A review article. *Living rev. relativity*. 2004. № 4. P.1-140.
3. Самохвалов С.Є. Теоретико-групове підґрунтя голографічного принципу. *Математичне моделювання*, 2010. № 2(23). С.7-11.

Надійшла до редколегії 27.05.2020.

УДК 681.3

DOI 10.31319/2519-2884.36.2020.20

ЛИСЕНКО Г.Л., к.т.н., професор
КУЗЬМЕНКО Л.В., аспірантка

Вінницький національний технічний університет

ХЕШУВАННЯ ДАНИХ ОПТОЕЛЕКТРОННОГО ПРИСТРОЮ НА ОСНОВІ ОПТИЧНО-КЕРОВАНИХ ТРАНСПАРАНТІВ З НАБОРОМ ЛОГІЧНИХ ОПЕРАЦІЙ ДЛЯ РОБОТИ З МАСИВАМИ ДАНИХ

Вступ. Станом на сьогодні в сфері інформаційних технологій має місце збільшення як об'єму даних, що підлягають обробці, так і зростання складності алгоритмів обробки. Для першої складової характерним є збільшення кількості компонентів, що описують дані, їх розмірність та розрядність, взаємозв'язок між різними компонентами даних тощо. Друга складова пов'язана з відображенням більш «тонких» взаємозалежнос-тей між даними та моделлю, яка використовується. Зростання обох складових призводить до значного збільшення часу отримання результату обчислень. Тому важливим є збільшення швидкості оброблення цієї інформації. Одним з ефективних методів, що доз-воляють реалізовувати збільшення швидкості обробки інформації, є метод паралельного одночасного обчислення групи даних, представлених у вигляді матриці. Реалізація цих методів обробки можлива на багатопроцесорних комп'ютерах, масивно-паралельних структурах, конвеєрних пристроях та інших спеціалізованих обчислювачах, які виконують такі функції. Проте вони мають недостатню швидкодію для обробки великорозмірних масивів даних, що пов'язано з обмеженими можливостями електрон-

них обчислюваль-них засобів. Для розв'язання цієї проблеми необхідно розробити паралельні методи та засоби введення, обробки і виведення даних. В даній роботі досліджувалось питання роз-робки швидкодіючих спеціалізованих обчислювачів хеш-функцій для великорозмірних масивів даних на основі оптично-керованих транспарантів. Дана тема є досить актуальною, так як поєднання блочної та паралельно-послідовної обробки даних несе в собі збіль-шення швидкості введення, обчислення та виведення великорозмірних масивів даних з отриманням хешу при використанні оптично-керованих транспарантів [1].

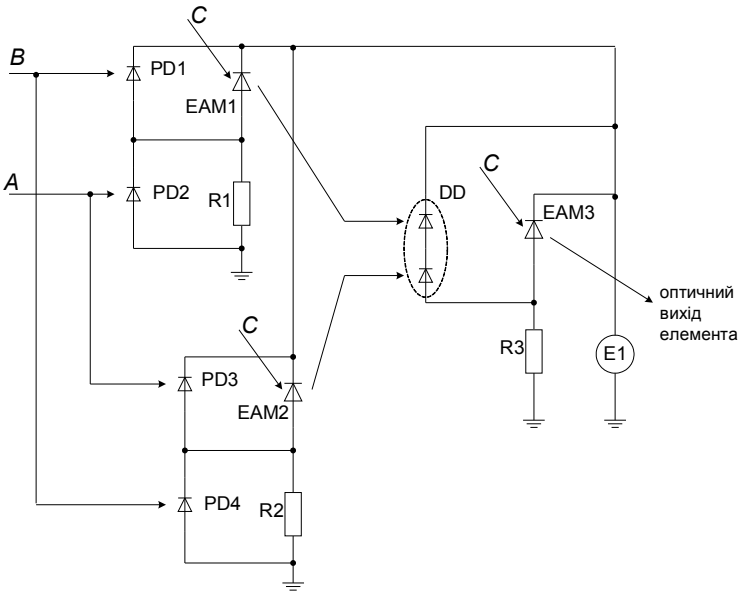
Постановка задачі. Згідно з поставленою задачею, у відповідності з алгоритмом хешування транспаранти будуть виконувати певні логічні операції. В даному випадку будуть виконуватись такі операції, як підсумовування по модулю 2, і функція rot (X, n) – циклічний зсув певного елемента X на задане число позицій n. Такий принцип обробки даних буде мати досить високу швидкодію за рахунок блочно-паралельної оптичної передачі даних, паралелізму процесу виконання логічних операцій над усіма елементами блоку одночасно.

Застосовування оптично-керованих транспарантів у спеціалізованих обчислювачах дозволяє великорозмірні масиви вхідних даних агрегатувати на блоки, паралельно передавати до спеціалізованого обчислювача, отримувати значення хеш-функції блоку, та повертати її назад. В даному випадку хешування даних відбувається згідно з алгоритмом Keccak, який має певну кількість параметрів, що налаштовуються (розмір блоку даних, розмір статку алгоритму, кількість раундів у функції f і інші), з метою забезпечення оптимального співвідношення стійкості та швидкодії.

Результати роботи. ***1. Оптоелектронна схема напівпровідникового транспаранта на базі напівсуматорів та суматорів.*** Як було сказано вище, існує задача створення логічних елементів на базі електрооптичних модуляторів, яка пов'язана з можливістю побудови послідовності таких елементів.

Створені логічні елементи на основі деяких типів SEED-приладів мають певні недоліки, пов'язані з парафазними входами і виходами та різними за рівнями сигналами.

Подолати ці недоліки пропонується, застосувавши схему побудови логічного елемента суми за модулем 2 (рис.1). Врахування перенесення в наступний розряд можливе за допомогою схеми перенесення.



A і B – оптичні входи елемента суми за модулем 2;
C – постійний оптичний сигнал

Рисунок 1 – Оптоелектронна схема логічного елемента суми за модулем 2 [2]

Отже, напівпровідниковий транспарант на основі таких логічних елементів буде виконувати операцію додавання окрім того, що одна комірка транспаранта буде представ-лена схемою з рис.1.

Оптоелектронний повний суматор [1] містить блок суми, інформаційні входи A, B та оптичний вхід C, причому блок суми містить фотодіоди PD1-PD4, електроабсорбційні модулятори EAM1-EAM3, резисто-

ри R1-R3, подвійний діод DD, джерело живлення E1, яке з'єднане з фотодіодами PD1 і PD3, електроабсорбційними модуляторами EAM1-EAM3 та подвійним діодом DD, а також оптичний вихід блока суми.

Оптоелектронний повний суматор працює таким чином. Входи A і B є інформаційними входами, на які інформаційний сигнал може надходити (логічна „1") або не надходити (логічний „0"), а оптичний вхід C є входом, на який постійно надходить оптичний сигнал (логічна „1"), що за рівнями відповідає інформаційним сигналам.

При надходженні одиничних сигналів на інформаційні входи A і B струми, що проходять через фотодіоди PD1, PD2, однакові, і тому напруга на EAM1 не змінюється, і він пропускає постійний оптичний сигнал C, який на нього надходить. Аналогічна ситуація відбувається на фотодіодах PD3, PD4, і оптичний сигнал, що надходить на вхід C, проходить через електроабсорбційний модулятор EAM2. Відповідно, на подвійний діод DD надходять два одиничних сигнали, він відкриється, через що починає збільшуватися струм, який призводить до зменшення напруги на електроабсорбційному модуляторі EAM3. З рис.1 видно, що зменшення напруги на електроабсорбційному модуляторі EAM3 веде до збільшення його коефіцієнта поглинання. Тому він стає непрозорим, і через нього не проходить постійний оптичний сигнал C, на виході блока суми отримуємо нуль.

Також не змінюють напругу на електроабсорбційних модуляторах EAM1 і EAM2 нульові вхідні інформаційні сигнали A і B, призводячи до нульового сигналу на виході блока суми.

Якщо на входах блока суми інформаційні сигнали A=0 та B=1, то напруга на електроабсорбційному модуляторі EAM1 зменшується, призводячи до збільшення його коефіцієнта поглинання випромінювання, і на виході електроабсорбційного модулятора EAM1 присутній нульовий сигнал. Нульовий сигнал, що надходить на вхід A на паралель-ній ділянці схеми блока суми, збільшує напругу на електроабсорбційному модуляторі EAM2, збільшуючи його пропускання, і на його виході існує постійний оптичний одиничний сигнал, який надходить з входу C. Нульовий і одиничний вхідні сигнали на подвійному діоді DD призводять до одиничного сигналу на оптичному виході блока суми.

У випадку, коли на вході блока суми інформаційні сигнали A=1 та B=0, напруга на електроабсорбційному модуляторі EAM2 зменшується, призводячи до збільшення поглинання ним випромінювання, і на виході електроабсорбційного модулятора EAM2 присутній нульовий сигнал. Нульовий сигнал B на паралельній ділянці схеми блока суми збільшує напругу на електроабсорбційному модуляторі EAM1, збільшуючи його пропускання, і на виході електроабсорбційного модулятора EAM1 існує постійний оптичний одиничний сигнал, який надходить із входу C. Одиничний та нульовий вхідні сигнали на подвійному діоді DD призводять до одиничного сигналу на виході блока суми [2].

2. Структура та принцип роботи алгоритму Keccak. Для вирішення вказаних проблем пропонується створити оптоелектронну базу, яка працює із застосування оптично-керованих транспарантів на базі алгоритму хешування даних Keccak і з набором логічних операцій, оскільки напівпровідникові транспаранти мають ряд переваг над іншими приладами або елементами, які виконують такі ж або схожі функції і операції. Серед основних переваг є високий ступінь інтегрованості, можливість зміни своїх оптичних властивостей, тобто поглинання або пропускання в залежності від різних зовнішніх факторів (температура, напруга, величина електричного поля та багато інших), тому саме їх було обрано для створення такої бази. Що стосується алгоритму хешування даних Keccak, оригінальний алгоритм Keccak має безліч параметрів, що налаштовуються з метою забезпечення оптимального співвідношення криптостійкості і швидкодії для певного застосування алгоритму на певній платформі. Регульованими величинами

є: розмір блока даних, розмір стану алгоритму, кількість раундів у функції f та інші. Автори Keccak придумали просту схему типу Sponge-функція або іншими словами – губку [3].

У середині цієї «губки» є стан (розміром в 1600 біт для SHA-3), до якого на кожному раунді застосовується одна і та ж функція, що реалізує псевдовипадкову перестановку. Тобто, це по суті блоковий шифр без ключа з розміром блока 1600 біт. І якщо ми заповнимо стан нулями, виконаємо 10 раундів (10 раз застосуємо їх функцію f) в одну сторону, а потім стільки ж у зворотну (з функцією, зворотною f), то знову отримаємо нулі.

Схема складається з двох етапів (рис.2):

1) Absorbing (поглинання). Оригінал тексту M піддається багатораундовим перестановкам f;

2) Squeezing (стискання). Висновок отриманого в результаті перестановок значення Z.

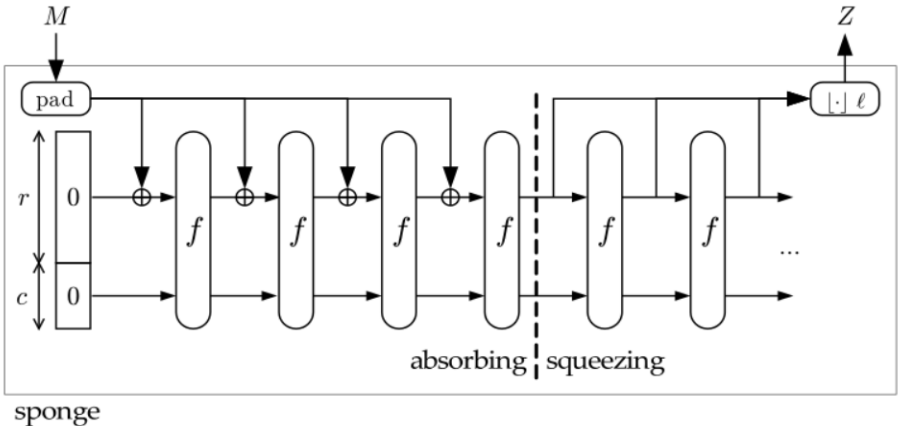


Рисунок 2 – Структурна схема алгоритму хешування Кессак

Отже, як зазначено вище, алгоритм Keccak заснований на конструкції Sponge (губ-ка). Це означає, що для отримання хешу потрібно виконати наступні дії:

1) взяти вихідне повідомлення M і доповнити його до довжини кратної r. У вигляді формули їх можна зобразити таким чином: M = M || 0x01 || 0x00 || .. || 0x00 || 0x80. Пояснити це можливо так: до повідомлення дописується одиничний байт, необхідну кіль-кість нулів і весь цей режим завершує байт із значенням 0x80 [3].

Все описане вище справедливе тільки для випадків, коли додається більше одного байта. Однак в разі, якщо необхідно доповнити всього один байт, то досить додати лише 0x81. Таким чином, код необхідно переписати з урахуванням цього зауваження;

2) потім для кожного блока M_i довжиною r біт виконуємо:

а) додавання за модулем 2 з першими r-бітами набору початкових станів S. Перед використанням цієї функції всі елементи S дорівнюватимуть нулю;

б) N раз застосовуємо до отриманих в результатів даних функцію f. Набором початкових станів S для блока M_{i+1} буде результат останнього раунду блока M_i;

в) після того, як всі блоки M_i будуть мати закінчення, тоді можна взяти підсумковий результат і повернути його в якості хеш-значення.

Хеш-функція Keccak реалізована таким чином, що функцію перестановки f, яка застосовується для кожного блока M_i, користувач може вибирати самостійно з набору визначених функції b = {f-25, f-50, f-100, f- 200, f-400, f-800, f -1600}.

Для того, щоб у реалізації використовувалась функція f-800, необхідно вибрати такі r і c, щоб виконувалося рівність r + c = 800.

Крім того, змінюючи значення r і c, згідно з цим змінюється кількість раундів заданої хеш-функції, оскільки їх кількість обчислюється за формулою n = 12 + 2*l*, де 2^l = (b / 25). Так для b = 1600 кількість раундів дорівнює 24.

3. Реалізація алгоритму Кессак. Процес хешування складається з двох багатораундових етапів: введення (Absorbing, вбирання) і висновок (Squeezing, вижимання) [4].

Етап введення інформації:

- стан S ініціалізується нулями. Повідомлення M доповнюється до довжини кратної r, а потім розбивається на блоки довжини r;
- перший блок – повідомлення M підсумовується за допомогою операції суми по модулю 2 (операція XOR) з S1; результати операції XOR і S2 передаються на вхід функції f;
- другий блок – повідомлення M, сума по модулю 2 з першими r бітами виходу функції f; результат операції XOR і останні з біт результату дії функції f знову передаються на вхід функції F другого раунду;
- етап триває до тих пір, поки не будуть оброблені всі блоки повідомлення M.

Особливістю етапу є те, що на кожному раунді блок повідомлення підсумовується по модулю 2 тільки з частиною стану, а функція f відтворює перетворення всього стану і робить його залежним від усього повідомлення M.

Етапи формування.

Крок θ. Для всіх i і k, таких що $0 \le i < 5$, $0 \le k < w$,

$C(i, k) = A[i, 0, k] \oplus A[i, 1, k] \oplus A[i, 2, k] \oplus A[i, 3, k] \oplus A[i, 4, k]$

x=0

$C[1] = A[1,0] \oplus A[1,1] \oplus A[1,2] \oplus A[1,3] \oplus A[1,4]$

$C[1] = A[1,0] \oplus A[1,1] \oplus A[1,2] \oplus A[1,3] \oplus A[1,4]$

$C[2] = A[2,0] \oplus A[2,1] \oplus A[2,2] \oplus A[2,3] \oplus A[2,4]$

$C[3] = A[3,0] \oplus A[3,1] \oplus A[3,2] \oplus A[3,3] \oplus A[3,4]$

$C[4] = A[4,0] \oplus A[4,1] \oplus A[4,2] \oplus A[4,3] \oplus A[4,4]$

$D(i, k) = C[(i-1) \bmod 5, k] \oplus C[(i + 1) \bmod 5, (k-1) \bmod w]$

$D[0] = C[-1] \oplus rot(C[1],1)$

$D[1] = C[0] \oplus rot(C[2],1)$ $D[2] = C[1] \oplus rot(C[3],1)$

$D[3] = C[2] \oplus rot(C[4],1)$ $D[4] = C[3] \oplus rot(C[5],1)$

Для всіх (i,j,k), таких що $0 \le i < 5, 0 \le j < 5, 0 \le k < w$,

$A'[i,j,k] = [i,j,k] \oplus D[i, k]$

$A[0,y] = A[0,y] \oplus D[0]$

$A[1,y] = A[1,y] \oplus D[1]$ $A[2,y] = A[2,y] \oplus D[2]$

$A[3,y] = A[3,y] \oplus D[3]$ $A[4,y] = A[4,y] \oplus D[4]$

Крок -ρ&π.

Для всіх k, таких що $0 \le k < w$, $A'[0,0,k] = A'[0,0,k]$.

Нехай на початку (i,j) = (1,0). Для t від 0 до23:

1. Для всіх k, таких що $0 \le k < w$,

$$A'[i,j,k] = A\left[i, j\left(k - \frac{(t+1)(t+2)}{2}\right)\right] \bmod w$$

2. (i, j) = (j, (2i+3j)mod 5

Для всіх (i,j,k), таких що $0 \le i < 5, 0 \le j < 5, 0 \le k < w$,

A'[i,j,k] = A[(i+3j)mod 5,i,k]

B[0,0] = rot(A[0,0], r[0,0]) B[1,3] = rot(A[0,1], r[0,1])

B[2,6] = rot(A[0,2], r[0,2]) B[3,9] = rot(A[0,3], r[0,3])

B[4,12] = rot(A[0,4], r[0,4]) B[0,2] = rot(A[1,0], r[1,0])

B[1,5] = rot(A[1,1], r[1,1]) B[2,8] = rot(A[1,2], r[1,2])

B[3,11] = rot(A[1,3], r[1,3]) B[4,14] = rot(A[1,4], r[1,4])

B[0,4] = rot(A[2,0], r[2,0]) B[1,7] = rot(A[2,1], r[2,1])

B[2,10] = rot(A[2,2], r[2,2]) B[3,13] = rot(A[2,3], r[2,3])

B[4,16] = rot(A[2,4], r[2,4]) B[0,6] = rot(A[3,0], r[3,0])

B[1,9] = rot(A[3,1], r[3,1]) B[2,12] = rot(A[3,2], r[3,2])

B[3,15] = rot(A[3,3], r[3,3]) B[4,18] = rot(A[3,4], r[3,4])

B[0,8] = rot(A[4,0], r[4,0]) B[1,11] = rot(A[4,1], r[4,1])

B[2,14] = rot(A[4,2], r[4,2]) B[3,17] = rot(A[4,3], r[4,3])

B[4,20] = rot(A[4,4], r[4,4])

Етап поглинання можна представити у вигляді такої функції:

```
Keccak-f[b](A)
{ forall i in 0…nr-1
A = Round[b](A, RC[i])
 return A }
```

Тут b – значення обраної функції (за замовчуванням 1600). А функція Round () – псевдовипадкова перестановка, що застосовується на кожному раунді. Кількість раундів nr обчислюється із значень r і c.

Операції виконуються на кожному раунді, вони мають вигляд наступної функції:

```
Round[b](A,RC)
{ θ step
 for(int x=0; x<5; x++)
C[x] = A[x,0] xor A[x,1] xor A[x,2] xor A[x,3] xor A[x,4];
for(int x=0; x<5; x++)
D[x] = C[x-1] xor rot(C[x+1],1);
for(int x=0; x<5; x++)
A[x,y] = A[x,y] xor D[x];
ρ and π steps
for(int x=0; x<5; x++)
for(int y=0; y<5; y++)
B[y,2*x+3*y] = rot(A[x,y], r[x,y]);
χ step
for(int x=0; x<5; x++)
```

```
for(int y=0; y<5; y++)
A[x,y] = B[x,y] xor ((not B[x+1,y]) and B[x+2,y]);
ι step A[0,0] = A[0,0] xor RC
 return A}
```

Вона складається з 4 кроків, на кожному з яких над вхідними даними проводиться низка логічних операцій.

Тут функція rot (X, n) позначає циклічний зсув елемента X на n позицій.

Масив r є визначеним набором значень, в якому вказується, на скільки необхідно зрушувати байти на кожному раунді. Значення всіх елементів даного масиву наведено в табл.1 [4].

Таблиця 1 – Округлення констант RC [i] (the round constant)

RC[0]	0x0000000000000001	RC[12]	0x000000008000808B
RC[1]	0x0000000000008082	RC[13]	0x800000000000008B
RC[2]	0x800000000000808A	RC[14]	0x8000000000008089
RC[3]	0x8000000080008000	RC[15]	0x8000000000008003
RC[4]	0x000000000000808B	RC[16]	0x8000000000008002
RC[5]	0x0000000080000001	RC[17]	0x8000000000000080
RC[6]	0x8000000080008081	RC[18]	0x000000000000800A
RC[7]	0x8000000000008009	RC[19]	0x800000008000000A
RC[8]	0x000000000000008A	RC[20]	0x8000000080008081
RC[9]	0x0000000000000088	RC[21]	0x8000000000008080
RC[10]	0x0000000080008009	RC[22]	0x0000000080000001
RC[11]	0x000000008000000A	RC[23]	0x8000000080008008

Масив RC – це набір констант, які теж є зумовленими.

Таблиця 2 – Заміщення обертання (the rotation offsets)

	x = 3	x = 4	x = 0	x = 1	x = 2
y = 2	25	39	3	10	43
y = 1	55	20	36	44	6
y = 0	28	27	0	1	62
y = 4	56	14	18	2	61
y = 3	21	8	41	45	15

Сама ж функція Keccak має наступний вигляд:

```
Keccak[r,c](M) {Initialization and padding
for(int x=0; x<5; x++)
for(int y=0; y<5; y++)
S[x,y] = 0; P = M || 0x01 || 0x00 || … || 0x00;
P = P xor (0x00 || … || 0x00 || 0x80);
 //Absorbing phase
forall block Pi in P
for(int x=0; x<5; x++)
for(int y=0; y<5; y++)
S[x,y] = S[x,y] xor Pi[x+5*y];
S = Keccak-f[r+c](S);
//Squeezing phase
Z = empty string;
Do {for(int x=0; x<5; x++)
for(int y=0; y<5; y++)
```

if((x+5y)<r/w) Z = Z || S[x,y];
S = Keccak-f[r+c](S) } while output is requested
return Z; }

На етапі Absorbig виконується обчислення хеш значення, а на етапі стискання висновку результатів обчислення виконуються до тих пір, поки не буде досягнута необхідна довжина хешу [5].

Висновки. У даній статті запропоновано оптоелектронний пристрій з набором логічних операцій. Запропоновано застосовувати алгоритм хешування даних Кессак із змінною довжиною виходу 224, 256, 384 і 512 біт. Сам запропонований пристрій складається з двох шарів: перший шар містить керований оптичний транспарант на основі логічних елементів по модулю 2, другий шар включає в себе функцію rot (X,n), яка означає циклічний здвиг елемента X на n позицій. Описано принцип роботи алгоритму хешування Кессак, принцип роботи і особливості оптично-керованих транспарантів. Показано переваги і недоліки транспарантів і самого алгоритму хешування даних.

Також показано, як в алгоритмі Кессак працюють функції логічних елементів. В даній статті показано у вигляді коду операції, які виконуються на кожному раунді і яку представляють функцію. Обчислення в алгоритмі відбувається у два етапи (як зазначалось вище): на етапі поглинання виконується обчислення хеш значення, а на етапі стискання відбувається підсумовування результатів. Даний результат буде виконуватись до тих пір, поки не буде досягнута необхідна довжина хешу.

ЛІТЕРАТУРА

1. Лисенко Г.Л., Мялківська І.В. Оптимізація спеціалізованих обчислювальних систем для виконання складних матричних операцій на основі оптичних транспарантів [електронний ресурс]. *Контроль і управління в складних системах (КУСС-2008)*: зб. тез доп. IX Міжнар. конф., 21-23 жовтня 2008р. Вінниця, 2008. http://www.vstu.vinnica.ua/ mccs2008/ukr/abstracts_UA.html.
2. Лисенко Г.Л.,. Мялківська І.В, Дюдюн О.В. Оптоелектронний пристрій на основі транс-парантів з повним набором логічних операцій для роботи з матрицями. *Оптико-електронні інформаційно-енергетичні технології*, 2009р. № 1(17). С.71-76.
3. Micah B. Yairi. An optically controlled optoelectronic switch: from theory to 50 gigahertz burst-logic demonstration. A dissertation submitted to the department of applied physics and the committee on graduate studies of Stanford University in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2001. 210p.
4. Захаров С.М., Федоров В.Б., Цвєтков В.В. Оптоэлектронные интегральные схемы с применением полупроводниковых вертикально излучающих лазеров. *Квантовая электроника*, 1999. №3, 28. С.189-206.
5. Lysenko G.L., Kuzmenko L.V., Kisała P., Klimek J. & Kalimoldayev M. (2019, November). The use of optically controlled transparent and blockchain technology for the processing of large-scale data arrays. In *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019* (Vol. 11176, p. 111760G). International Society for Optics and Photonics.
6. Morris J. Dworkin SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. DOI:10.6028/nist.fips.202.

Надійшла до редколегії 25.06.2020.