УДК 004.312.26 DOI: 10.30748/soi.2017.150.25

Ю.В. Танасюк, Х.В. Мельничук, С.Е. Остапов

Чернівецький національний університет імені Юрія Федьковича, Чернівці

РОЗРОБКА І ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНИХ ХЕШ-ФУНКЦІЙ НА ОСНОВІ КЛІТИННИХ АВТОМАТІВ

У роботі представлено результати реалізації та дослідження криптографічних хеш-функцій на основі правил перетворення клітинних автоматів. Розроблені функції обробки володіють якісними розсіювальними властивостями і дозволяють отримувати хеш змінної довжини. Покращення статистичних характеристик та лавинного ефекту виявлене для функції перетворення, які поєднують декілька правил обробки клітинних автоматів з операціями побітового зсуву та XOR.

Ключові слова: *криптографічні хеш-функції, клітинні автомати, криптографічна губка, алгоритм Keccak.*

Вступ

Постановка проблеми. Проблема безпеки у сучасних комп'ютерних системах є досить актуальною, і значною мірою вирішується за допомогою криптографічних підходів. Програмні методи захисту інформації – це сукупність алгоритмів і програм, які забезпечують розмежування доступу та запобігають несанкціонованому використанню персональних даних.

Важливою процедурою у криптографії є перевірка автентичності інформації. Зокрема, ідентифікація користувача шляхом порівняння введеного ним паролю з обліковими даними, збереженими у базі даних користувачів; підтвердження достовірності електронного листа за допомогою цифрового підпису за відкритим ключем відправника; обчислення контрольної суми файлу та її порівняння зі значенням, заявленим автором. Для підтвердження цілісності повідомлень у системах ідентифікації цифрової інформації використовують криптографічні хеш-функції. Такі алгоритми хешування як MD5 та SHA-1 використовують при перетворенні даних довільної довжини у хеш фіксованого розміру функції ущільнення, через що неодмінно стикаються з проблемою появи колізій та зовнішніх атак [1]. Заснований на конструкції криптографічної губки стандарт SHA-3 використовує хеш-функції зі змінною довжиною виходу і реалізує механізм псевдовипадкових перемішувань, який унеможливлює простеження залежності результату від вхідних даних і забезпечує стійкість до атак на виявлення колізій [2].

Клітинні автомати (КА) відомі як універсальне обчислювальне середовище, що володіє властивостями самовідтворення, паралельності, однорідності та оновлення за визначеним набором правил. Завдяки здатності генерувати псевдовипадкові послідовності, КА широко застосовуються у криптографії для створення функцій автентифікації повідомлень та хешування [3; 4]. Програмна реалізація таких функцій автентифікації повідомлень дозволяла вдвічі зменшити необхідний час обробки, в порівнянні з традиційними методами, тоді як оптимальне апаратне рішення запропонованого механізму, згідно [4], буде в 10 разів швидшим за будь-яку можливу паралельну реалізацію MD-5.

Мета роботи полягає в розробці криптографічних функцій хешування на основі конструкції губки та правил перетворення клітинних автоматів і дослідження їх розсіювальних властивостей.

Аналіз досліджень. У роботі [4] доповідається про модель ітеративної хеш-функції з використанням структури Меркле-Дамгарда, побудованої на правилах КА, в поєднанні з операціями XOR та побітового зсуву. Для забезпечення криптостійкості розробленої конструкції було запропоновано використовувати комбінацію лінійних та нелінійних правил КА. Лінійні правила (наприклад, «60», «90», «150»), в яких використовується лише операція XOR, забезпечують стійкість до колізій. Тоді як нелінійна група правил з AND-OR логікою (такі як, «22», «30», «86», «135»), використовуються для підтримки однонаправленості та нелінійності хеш-функцій.

У зв'язку з цим викликає інтерес дослідження розсіювальних властивостей хеш-функцій на основі КА, які не використовують функції ущільнення, що можуть спричинити появу колізій, а базуються на псевдовипадкових перетвореннях.

Як зазначається у [2], алгоритм хешування SHA-3 (Keccak) побудований на функції губки (Sponge), яка є узагальненим поняттям криптографічної хеш-функції зі змінним виходом і може виконувати всі квазі-симетричні криптографічні функції, від генерування псевдовипадкових послідовностей

до перевірки достовірності шифрування [2]. Єдиний і достатній критерій стійкості конструкції губки до всіх можливих атак – її подібність до випадкового оракула (Random Oracle). Випадковий оракул – це ідеалізована функція, що описує роботу машини з практично нескінченним об'ємом пам'яті, яка на будь-який запит може видати ідеальне випадкове число і запам'ятати пару «запит-відповідь». Дана конструкція схожа на функцію хешування, з тією відмінністю, що зв'язок між вхідним повідомленням і результатом хешування неможливо вирахувати. Автори Keccak використали псевдовипадкові перестановки, які з високою точністю імітують поведінку випадкового оракула, а отже запобігають виникненню колізій [2].

Схематично структуру губки зображено на рис. 1. Губка має внутрішній стан S – масив двійкових значень фіксованого розміру b (бітів). На думку авторів хеш-функції, для захисту від колізій цей масив повинен бути вдвічі більшим за розмір кінцевого хешу. Стандартом SHA-3 передбачено використання внутрішнього стану довжиною 1600 бітів [2].

Масив внутрішнього стану губки складається з двох частин – r і c. Значення r називається бітовою швидкістю, саме ця частина масиву об'єднується з фрагментами вхідного повідомлення та використовується для одержання значення хешу з визначеною швидкістю. Величину c називають потужністю, c=b-r [3]. Вона безпосередньо не пов'язана з вхідним повідомленням і відповідає за рівень захисту хеш-функції. Автори Keccak довели, що стійкість конструкції губки подібна до випадкового оракула з розміром дайджесту c/2. Зокрема, для одержання хешу з визначеною математичною стійкістю, значення потужності c повинно бути вдвічі більшим за довжину хешу Z.

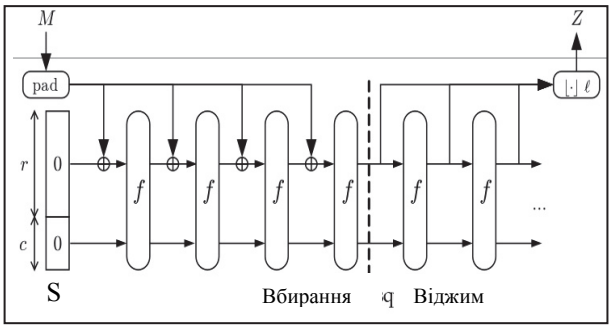

Рис. 1. Конструкція криптографічної губки [2]

Таким чином, параметри алгоритму Keccak можна налаштовувати з метою забезпечення оптимального співвідношення криптостійкості та швидкодії для застосування алгоритму на визначеній платформі (табл. 1). Слід зазначити, що даний алгоритм дозволяє одержати значення хешу як фіксованого зі змінною довжиною виходу (224, 256, 384 і 512 бітів), так і довільного розміру. Рівень захисту алгоритму хешування визначається як N=Z/2 і вказує, що для здійснення найбільш ефективних атак необхідно виконати 2^N кроків [1].

Таблиця 1

Параметри алгоритму Keccak
для хешу різної довжини

Довжина хешу, Z (біти)	Бітова швидкість, r (біти)	Потужність, c (біти)	Рівень захисту, N
224	1152	448	112
256	1088	512	128
384	832	768	192
512	576	1024	256

До внутрішнього стану губки на кожному етапі обробки застосовується одна і та сама функція f, що реалізовує псевдовипадкову перестановку за допомогою побітових операцій циклічного зсуву, XOR, AND, NOT (рис. 1). В оригінальному алгоритмі Keccak стан губки подається як масив 64-бітових слів розміром 5×5, а кількість раундів застосування функцій перетворення визначається за формулою: $n=12+2\log_2 k$, де k=(b/25), і для b=1600 дорівнює 24.

Схема алгоритму роботи хеш-функції Keccak на основі конструкції губки зображена на рис. 2.

На початковій стадії ініціалізації формується блок даних розміру b, визначаються параметри хеш-функції, вхідне повідомлення M, якщо потрібно, доповнюється до довжини, кратної r, та розбивається на рівні частини цієї ж довжини.

Доповнення відбувається за принципом: до повідомлення дописується одиничний байт 0x01, необхідна кількість нулів і до всієї послідовності додається байт із значенням 0x80. Якщо необхідно додати всього один байт, рядок доповнюється значенням 0x81 [2].

Конструкція губки працює у двох режимах: «вбирання» (absorbing) та «віджим» (squeezing).

На етапі «вбирання» фрагмент вхідного повідомлення довжиною r об'єднується з r-частиною стану S за допомогою операції XOR. Після цього до всього стану застосовується функція перемішування f, яка реалізує псевдовипадкову перестановку протягом фіксованої кількості раундів.

Після обробки всього повідомлення починається етап «віджиму», на якому результуючий хеш одержується шляхом подальшого застосування функції перетворення f до стану, виокремлення на кожному етапі фрагмента визначеної довжини з частини r і конкатенації з бітовим рядком, одержаним на попередніх етапах, до тих пір, поки не буде досягнуто необхідної довжини дайджесту [2].

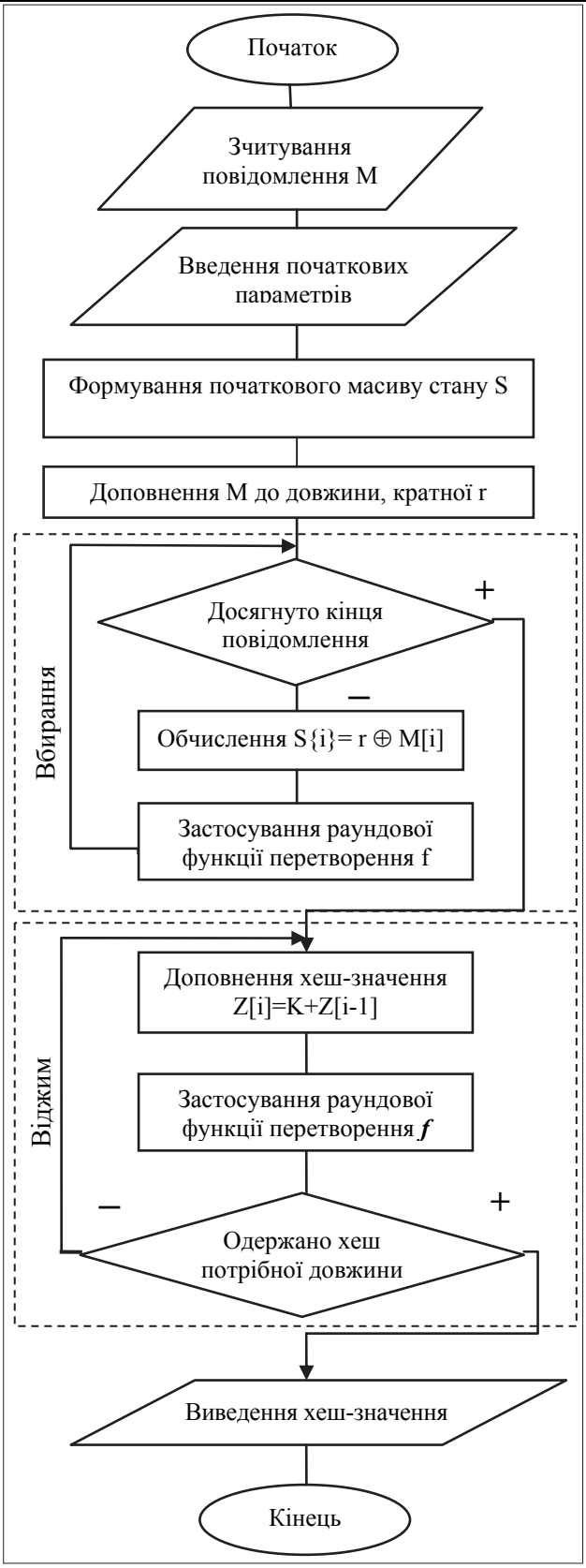

Рис. 2. Схема алгоритму хешування на основі криптографічної губки

Основний матеріал

У даній роботі розробка хеш-функції на основі алгоритму Keccak виконувалася засобами мови C++. Внутрішній стан криптографічної губки S реалізовано у вигляді лінійного клітинного автомата – одновимірного масиву двійкових значень, бітового рядка довжиною 1600 бітів, значення параметрів r і c, якого визначалися за довжиною хешу (табл. 1).

Для реалізації операцій псевдовипадкового перемішування, як раундові функції перетворення f внутрішнього стану губки у наших дослідженнях використовувалися такі правила обробки клітинних автоматів, як:

«30»:

$$a[i]' = a[i-1] \oplus (a[i] \vee a[i+1]),$$

«86»:

$$a[i]' = (a[i-1] \vee a[i]) \oplus a[i+1],$$

«150»:

$$a[i]' = a[i-1] \oplus a[i] \oplus a[i+1],$$

де a[i] – стан поточної комірки, a[i-1] – стан попередньої комірки, що розташована ліворуч (молодший біт), a[i+1] – стан наступної комірки, розташованої праворуч від поточної комірки (старший біт), a[i]' – оновлений стан комірки після застосування правила; $\vee$ та $\oplus$ позначають операції побітового АБО та виключного АБО (XOR), відповідно [5].

Наші дослідження передбачали оцінку розсіювальних властивостей розробленої хеш-функції при застосуванні на кожному етапі окремого правила, послідовності декількох правил, а також перетворень, поєднаних з операціями побітового циклічного зсуву та XOR. Як зазначалося вище, при застосуванні правила наступний стан комірки визначається її поточним станом та станами її найближчих сусідів і всі клітини переходять до нового стану одночасно. Отже, на кожному раунді обробки для перетворення внутрішнього стану губки, організованого у вигляді бітового рядка, правила необхідно застосовувати послідовно до кожної комірки. Також слід окремо виконувати обробку крайніх комірок: бітовий масив замикається і для першого біту його попереднім сусідом є останній біт масиву і навпаки. Такі перетворення передбачають застосування обраного правила 1600 разів, що негативно позначається на швидкості обробки.

Для вдосконалення процесу перетворень у даній програмній розробці реалізована паралельна обробка вектору стану губки за правилами клітинних автоматів. Зокрема, правила застосовуються не до кожної клітини по черзі, а до всього стану одночасно.

Схематично запропонований принцип обробки за правилом «30» можна зобразити наступним чином.

На кожному етапі обробки створювалися два вектори внутрішнього стану губки S: циклічно зсунутий вправо масив, що ототожнював усі попередні клітинки:

$$A[1600]= S>>>1,$$

і циклічно зсунутий вліво масив, що зберігає значення усіх наступних клітинок:

$$B[1600]=S<<<1.$$

Отже, оновлений вектор стану губки S’ при застосуванні правила «30» для таких бітових конструкцій одержується як:

$$S'=A \oplus (S \vee B).$$

Як свідчать результати досліджень, 10^6 ітерацій обробки стану губки довжиною 1600 бітів за правилом «30» на комп’ютері з процесором Intel Pentium N3700 (4 ядра, 2.4 ГГц) та 4 ГБ ОЗП, при послідовному підході виконувалися протягом 5 хв і 304 сек, тоді як паралельна обробка займала всього 5 сек. Тобто запропонований нами підхід дозволив скоротити час перетворень приблизно в 60 разів.

Розсіювальні властивості розробленої хеш-функції досліджувалися за допомогою 16 статистичних тестів, що входять до пакету NIST STS. Дана методика працює з послідовностями довжиною 10^8 бітів і дозволяє перевірити випадковість згенерованих двійкових даних. Статистичний пакет розбиває послідовність на 100 рівних частин довжиною 10^6 бітів, кожна з яких підлягає тестуванню. Перевірка виконується декілька разів із різними параметрами, тому загальна кількість тестів становить 189, а мінімальне значення проходження тестів повинно знаходитись на рівні 96 % [6].

Дослідження проводилися на двійкових послідовностях, згенерованих створеною криптографічною хеш-функцією на основі КА з такими параметрами (біти): розмір стану b=1600, довжина хешу Z=512, r=576, c=1024.

Стан губки довжиною 1600 бітів підлягав попередній обробці: r-частина заповнювалася двійковими одиницями, а c – нулями, після чого до всього стану 150 разів застосовувалося правило «86». На кожному етапі одержання хешу визначена функція перемішування, у вигляді одного правила, чи комбінації декількох правил, застосовувалася 25 разів.

Функція перетворення, позначена як Serial_30_86_150, передбачала послідовне використання правил «30», «86» і «150». Функція Mix використовує операції перемішування за таким принципом:

На парному кроці: rule_30 (S);
S = S>>>37.
На непарному кроці: rule_86(S);
Y = S<<<11;
S = S XOR Y;
rule_150(S).

На рис. 3; 4 зображено статистичні портрети криптографічних хеш-функцій на основі клітинних автоматів при використанні різних правил перетворення. Наведені результати для послідовностей, одержаних з використанням правил «30» та «86» (рис. 3, а, б, відповідно) добре співвідносяться з даними, одержаними в [7] для генераторів псевдовипадкових чисел на основі тих самих правил, але при їх послідовній реалізації.

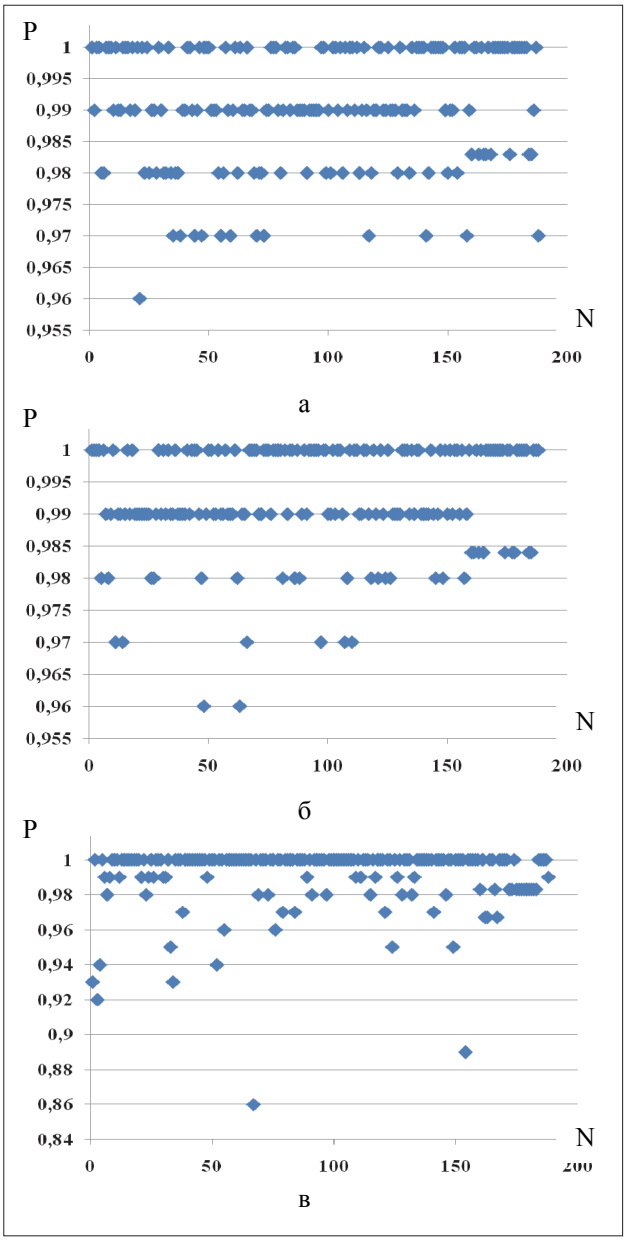

Рис. 3. Статистичний портрет криптографічних хеш-функцій з використанням перетворень на основі правил клітинних автоматів : а – «30»; б – «86»; в – «150». Тут P позначає пропорцію тестових послідовностей, які пройшли тест, а N – номер тесту

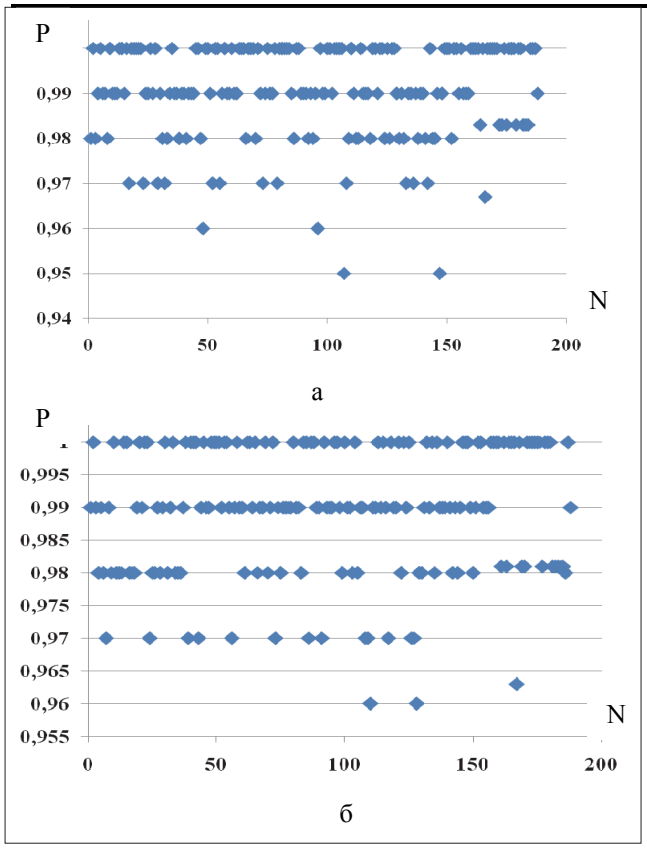

Рис. 4. Статистичний портрет криптографічних хеш-функцій на основі клітинних автоматів з використанням функції перетворення: а – Serial_30_86_150; б – Mix

Таблиця 2

Узагальнені результати статистичного тестування для різних функцій перемішування із застосуванням правил клітинних автоматів

Рівень проходження тестів, %	Rule_30	Rule_86	Rule_150	Serial_30_86_150	Mix
	Кількість тестів, які пройшли тестування (%)				
100	81 (43%)	89 (47,1%)	129 (68,3%)	80 (42,5%)	70 (37%)
99	58 (31%)	65 (34,4%)	16 (8,5%)	57 (30,5%)	60 (32%)
98	36 (19%)	26 (14%)	23 (12,2%)	34 (18%)	43 (23%)
97	12 (6,5%)	6 (3,2%)	8 (4,2%)	13 (7%)	13 (7%)
96	1 (0,5%)	2 (1%)	2 (1%)	2 (1%)	2 (1%)
95	-	-	3 (1,6%)	2 (1%)	-
< 95	-	-	6 (3,2%)	-	-

Як свідчать одержані статистичні дані, щонайменше 94–96 % усіх послідовностей успішно пройшли перевірку за всіма тестами NIST STS. Це означає, що двійкові дані, згенеровані створеними хеш-функціями на базі як одного правила (рис. 3) так і комбінації правил (рис. 4) перетворення КА, за своїм характером наближаються до псевдовипадкових. Узагальнені результати статистичного тестування наведено у табл. 2.

Застосування розроблених функцій перетворення для одержання хеш-образу виявили залежність розсіювального ефекту від довжини хешу, типу функції та кількості раундів обробки. Для усіх запропонованих хеш-функцій дайджест вхідного повідомлення повністю оновлювався при зміні довжини хешу (224, 256, 384, 512 бітів) вже при 25 раундах обробки. Проте, для функції перемішування на основі одного або декількох правил КА вплив лавинного ефекту обмежувався довжиною вхідного повідомлення при фіксованій довжині блоку. А саме, довільні зміни вхідного повідомлення лише частково впливали на результат хешу, помітного оновлення якого вдавалося досягти при застосуванні функцій перетворення протягом 100 і більше раундів. В той час, як поєднання декількох правил обробки КА з операціями побітового зсуву та XOR, позитивним чином впливає на прояв лавинного ефекту вже при 50 ітераціях. Слід зауважити, що при послідовній та паралельній реалізації функцій перемішування одержуються ідентичні хеш-значення. Швидкодія запропонованої хеш-функції при формуванні хешу 512 бітів і довжині блоку 576 байтів на комп'ютері з вищезазначеними параметрами приблизно становить 130 кбайт/с. Нижче наведено результуючі значення хешу при використанні функції Mix протягом 50 раундів для таких тестових векторів:

1) The quick brown fox jumps over the lazy dog

0c0ef31612ad1de84d8df2e9aa5b0357f86adedb86ca0e660beaf755dfc37a3c4a7e36019130d61070cf0e84e08c3a7de7e9c851706d88016eb9301a8718188c

2) The quick brown fox jumps over the lazy dog**.** (додано крапку в кінці)

f50651198b8f1de8b5c406ba91c9a31130e4027948da3d3427464c4c8e2f77f093b503021d339cf6baba3cc0a4078389e1b3f88e5d7aaec2329d273092293666

3) the quick brown fox jumps over the lazy dog. (з маленької літери)

ece72bcb4ad1dacb6e438db6199220df03bb49ccf3c14569f2536c09410c841a034e36029ca9d361d3045f1b5954d29d782730653784b9639b930685857206c3

Висновки

Підсумовуючи усе наведене вище, можна зробити такі висновки.

1. Запропоновано функції перетворення стану у структурі «криптографічної губки», що реалізову-

ють псевдовипадкові перетворення на основі одновимірних клітинних автоматів з використанням правил «30», «86», «150». Показано, що вони володіють якісними розсіювальними характеристиками.

2. Застосування у функції перетворення декількох правил обробки, поєднаних з побітовими операціями циклічного зсуву та XOR, призводить до збільшення частки успішно пройдених статистичних тестів (від 96% і вище) та скорочення кількості раундів обробки до 50 ітерацій для досягнення стійкого лавинного ефекту.

3. Тестування лавинного ефекту на стандартних вхідних векторах показало повну зміну результуючого хеш-образу при додаванні або зміні одного вхідного символу.

Список літератури

1. Paar Ch. Understanding cryptography / Ch. Paar, J. Pelzl. – Springer-Verlag Berlin Heidelberg, 2010. – 372 p.

2. Bertoni G. [Electronic resource]. – The Keccak sponge function family. – Access mode : http://keccak.noekeon.org/.

3. Тоффоли Т. Машины клеточных автоматов / Т. Тоффоли, Н. Марголус. – М.: Мир, 1991. – 284 с.

4. Jeon J.-Ch. Analysis of hash functions and cellular automata based schemes / J.-Ch. Jeon // International Journal of Security and Applications, 2013. – Vol. 7, No. 3. – P. 303-316.

5. The wolfram atlas of simple programs. Elementary Cellular Automata. [Electronic resource]. – Access mode: http://atlas.wolfram.com/01/01/.

6. Потій О.В. Метод статистичного тестування NIST STS та математичне обґрунтування тестів / О.В. Потій, А.В. Лєншин, Ю.А. Ізбенко // Технічний звіт ІІТ-001-2004. – Інститут інформаційних технологій. – 2004. – 62 с.

7. Val O.D. Development and investigation of the key stream generators on the base of cellular automata. / O.D. Val, V.V. Zhikharevich, R.I. Ovchar, S.E. Ostapov // Progressive Informatics Technologies, Radio Electronics, Computer Science, Control. – 2015. – No. 3. – P. 58-63.

Надійшла до редколегії 3.05.2017

Рецензент: д-р техн. наук доц. М.А. Павленко, Харківський національний університет Повітряних Сил ім. І. Кожедуба, Харків.

РАЗРАБОТКА И ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ НА ОСНОВЕ КЛЕТОЧНЫХ АВТОМАТОВ

Ю.В. Танасюк, К.В. Мельничук, С.Э. Остапов

В работе представлены результаты реализации и исследования криптографических хэш-функций на основе правил преобразования клеточных автоматов. Разработанные функции обработки владеют качественными рассеивающими свойствами и позволяют получить хэш переменной длины. Улучшение статистических характеристик и лавинного эффекта было получено для функций преобразования при совместном использовании нескольких правил обработки клеточных автоматов и операций побитового сдвига и XOR.

Ключевые слова: *хэш-функции, клеточные автоматы, криптографическая губка, алгоритм Keccak.*

DEVELOPMENT AND RESEARCH OF CRYPTOGRAPHIC HASH FUNCTIONS ON THE BASIS OF CELLULAR AUTOMAT

Yu. Tanasyuk, Kh. Melnychuk, S. Ostapov

The paper presents results of implementation and investigation of cryptographic hash functions on the basis of cellular automata processing rules. The developed permutation functions possess satisfactory scattering properties and enable one to generate variable-length message digest. Improvement of statistical characteristics and avalanche effect has been achieved for processing functions, which combine several transformation rules of cellular automata with bitwise operations of cyclic shift and XOR.

Keywords: *cryptographic hash functions, cellular automata, cryptographic sponge, Keccak algorithm.*