УДК 681.3.06

І.Д. ГОРБЕНКО[1], А.О. БОЙКО[1], А.М. ГЕРЦОГ[2]

[1]*ЗАТ "Інститут інформаційних технологій", Україна*
[2]*Харківський національний університет радіоелектроніки, Україна*

СТАН СТВОРЕННЯ ТА НАПРЯМИ ДОСЛІДЖЕНЬ І РОЗРОБОК ЗІ СТВОРЕННЯ ПЕРСПЕКТИВНИХ СТАНДАРТІВ ГЕШУВАННЯ

Наводяться результати аналізу властивостей стандартизованих функцій гешування, визначаються їх недоліки, обґрунтовуються вимоги до перспективних функцій гешування та визначаються можливі методи їх побудування. В результаті аналізу сучасних вимог до геш-функцій і стану розробок зі створення перспективних стандартів гешування було зроблено висновки про необхідність заміни діючого в Україні стандарту ГОСТ 34.311–95 новим і про можливість використання геш-функцій, запропонованих в ході конкурсу NIST SHA-3 Competition, в якості основи основи для нового стандарту гешування.

***Ключові слова**: функція гешування, стандартизація, колізійна стійкість, вимоги до геш-функцій.*

Вступ

Функції гешування є одним із основних криптографічних примітивів і широко застосовуються в інформаційно-телекомунікаційних системах для надання користувачам таких базових послуг як цілісність, справжність, неспростовність, конфіденційність тощо [1]. Одним із основних додатків застосування функцій гешування є електронний цифровий підпис. У звязку з широким впровадженням електронного цифрового підпису (ЕЦП) практично в усі інформаційно-телекомунікаційні системи (ІТС) та необхідності обробки інформації в реальному часі до функцій гешування зросли вимоги, перше за все в частині забезпечення криптографічної стійкості та прийнятної складності (швидкості) гешування [2–5]. Детальний аналіз якості таких існуючих функцій гешування як ГОСТ 34.311-95, SHA-1 та SHA-2 підтвердив наявність у них потенційних вразливостей [6–8]. Серед інших напрямів застосувань – побудування функцій вироблення ключів, кодів автентифікації повідомлень, генерування псевдовипадкових послідовностей [3]. Тому вже з 2007 року актуальною стала, по суті міжнародна, проблема створення перспективних функцій гешування з підвищеною швидкодією та криптографічною стійкістю.

Першим суттєвим кроком у вирішенні вказаної проблеми є її обговорення на ряді міжнародних форумів та систематичне проведення міжнародних конкурсів та форумів відносно перспективних функцій гешування. Найбільш значною подією є організація та ефективне проведення міжнародного конкурсу зі створення перспективного стандарту (стандартів) функції гешування, що отримав назву NIST SHA-3 Competition [3]. Організував та проводить цей конкурс Національний інститут стандартів і технологій-NIST США. Як буде показано нижче, проведені перші етапи конкурсу підтвердили можливість побудування криптографічних примітивів типу функція гешування з необхідними властивостями та характеристиками.

Нині в Україні інтенсивно впроваджуються різні системи електронного документообігу. Обов'язковим їх реквізитом є наявність ЕЦП, при виробленні якого в свою чергу використовується функція гешування. З цією метою використовується міждержавний стандарт ГОСТ 34.311-95 [9]. Розглядаються питання гармонізації та використання функцій гешування, що визначені в ISO/IEC 10118, перше за все SHA-2. Але, на наш погляд, фундаментальним вирішенням проблеми появи в Україні ефективного алгоритму або алгоритмів гешування, є врахування як теоретичних так і практичних результатів конкурсу NIST SHA-3 Competition.

Метою цієї статі є обґрунтування та визначення основних вимог до перспективної національної функції гешування, аналіз властивостей таких алгоритмів гешування як ГОСТ 34.311-95, SHA-1 та SHA-2, аналіз стану та результатів виконання перших двох етапів конкурсу NIST SHA-3 та визначення кандидатів на стандарти гешування як на міжнародному так і національному рівнях. Тому актуальність проведення як теоретичних так практичних досліджень та розробок зі створення стандартів на функцію гешування є надзвичайно важливою та необхідною.

1. Вимоги до сучасних функцій гешування

На нинішньому етапі розвитку криптографії визначено основні вимоги до функцій гешування. Вони у достатній мірі викладені в [1–3]. На наш погляд існуючі вимоги необхідно поділити на два ве-

ликих класи – безумовні та умовні. Інколи безумовні вимоги називають мінімальними. І таке визначення є зрозумілим, так як тільки при виконанні цих мінімальних вимог, в принципі певна функція гешування може бути застосована. В подальшому до безумовних вимог будемо відносити такі, що повинні бути виконаними безумовно, інакше, якщо вони не виконуються , то відповідна функція гешування не може бути застосована. До безумовних вимог необхідно віднести криптографічну стійкість та у випадку розгляду функцій гешування також і складність обчислення.

Інші вимоги будемо відносити до умовних, їх особливість у тому що вони можуть бути задані при різних значення певних показників та відповідно застосовуватись з різними можливостями. Деякі вимоги можуть бути віднесені як до безумовних так і умовних, наприклад складність обчислення. Таким чином до безумовних будемо відносити ті вимоги, виконання яких відносно функції гешування є обов'язковим.

На основі аналізу основоположних джерел, що стосуються функцій гешування, в тому числі за результатами виконання проекту SHA-3 [2, 3], визначені такі безумовні вимоги. Забезпечення захищеності від усіх відомих та потенційно можливих криптоаналітичних атак, де під захищеністю розуміється той факт, що усі відомі криптоаналітичні атаки носять (мають) експоненційну складність, а критерієм незахищеності – субекспоненційний або поліноміальний характер складності.

Практична захищеність функції гешування від силових та аналітичних атак, яка досягається за рахунок вибору розмірів загальних параметрів та, якщо функція гешування криптографічна, та і ключів. Критерієм практичної захищеності такої функції гешування є вибір таких розмірів загальних параметрів та ключів, при яких складність атак суттєво перевищує існуючу потужність криптоаналітичних систем на рівні технологічно розвинутих держав, в тому числі з урахуванням прогнозу збільшення їх потужності.

Надійність математичної бази, в розумінні відсутності можливостей здійснювати атаки типу „універсальне розкриття" за рахунок недосконалості математичної бази і слабкостей, що можуть бути закладені за рахунок специфічних властивостей загальних параметрів та ключів.

Статистична незалежність результату криптографічного перетворення (виходу), наприклад, геш-значення від вхідного блоку та ключа, якщо функція гешування є криптографічною.

Складність обчислення геш-значення повинна мати лінійну залежність від розміру повідомлення і не перевищує деяких допустимих значень, наприклад Іпр′. Запропоновані наступні умовні вимоги до геш-функцій:

– можливість і умови вільного поширення та застосування міжнародного або національного стандарту функції гешування в Україні з урахуванням нормативно-правових актів України на експорт, імпорт і обмеження на його застосування;

– рівень довіри до алгоритма;

– перспективність застосування алгоритма в якості національного стандарту в Україні;

– часова та просторова складність програмної, програмно-апаратної та апаратної реалізацій;

– можливість та умови застосування алгоритмів з різними параметрами та у різних режимах;

– рівень захищеності при реалізації різних видів загроз при різних умовах здійснення криптоаналітичних атак та відхиленні властивостей загальних параметрів.

Результати аналізу дозволили визначити що вимоги 1 – 4 відносно криптографічної стійкості функцій гешування можуть бути деталізовані та зведені до наступних вимог [1, 2]:

– стійкість до відновлення прообразу;

– стійкість до знаходження другого прообразу;

– стійкість до знаходження колізій (у тому числі на усічених геш-значеннях);

– стійкість до атаки розширення повідомлення (length extension);

– стійкість до атаки знаходження мультиколізій;

– складність розпізнавання вихідної послідовності для генераторів псевдовипадкових послідовностей, що використовують HMAC, побудованих на базі функції гешування зі складністю, меншою ніж знаходження другого прообразу і кількістю запитів до генератора не менше $2^{n/2}$.

До умовних необхідно також віднести такі конкретизовані вимоги [1, 2]:

можливість вироблення геш-значень з різними довжинами ;

високу швидкодію, у змісті не менше ніж вимагається замовником;

можливість реалізації на різних апаратно-програмних платформах тощо.

Сутність вимог відносно стійкості функції гешування до прообразу, другого прообразу та до колізій в необхідній мірі викладена в [1–3, 10], тому розглянемо тільки ті, що залишились нерозглянутими.

Атака розширення повідомлення [3] (length extension) полягає в обчисленні геш-значення $H(M \| M')$ за відомими $H(M), M'$, але без знання M. Для захисту від цієї атаки необхідно використовувати при обчисленні геш-значення довжину повідомлення і прикінцеву функцію перетворення, відмінну від функції стискання.

Атаку знаходження мультиколізій, описано в роботі [11]. Сутність атаки полягає в знаходженні K повідомлень з однаковим геш-значенням зі складністю $O(\log_2(K)\cdot 2^{n/2}$ замість $O(2^{(K-1)n/K})$, як очікується для випадкового оракула. Для захисту від такої атаки необхідно збільшити розмір проміжного геш-значення у порівнянні з вихідним принаймні вдвічі (принцип wide pipe).

Стійкість до атаки на розпізнавання псевдовипадкової послідовності для генераторів псевдовипадкових чисел полягає у відсутності кореляцій між блоками вихідних даних, яка може бути знайдена зловмисником з заданою імовірністю.

На даний момент існує велика кількість криптографічних сервісів, які потребують різних ступенів захисту і відповідно різних довжин геш-значень, що передбачено, наприклад, у стандарті SHA-2 [12].

Крім того, важливо мати можливість реалізації геш-функції на різних апаратних платформах, у тому числі з обмеженим об'ємом пам'яті та низькою розрядністю процесора (наприклад смарт-картка). На рівні стандарту важливо, щоб один алгоритм поєднував у собі усі названі вище функціональні можливості. Наявність кількох стандартизованих алгоритмів ускладнює розробку інформаційно-телекомунікаційних систем і приводить до конфліктів між системами різних розробників.

Наведені вище вимоги є такими, що визнаються світовою криптографічною спільнотою як мінімально необхідні, яким має відповідати сучасна геш-функція.

2. Аналіз функції гешування ГОСТ 34.311-95

В Україні широке застосування знайшов міждержавний стандарт ГОСТ 34.311- 95 [9]. За даними досліджень, тому числі нашими, нині діючий в Україні стандарт ГОСТ 34.311-95 має ряд значних недоліків. По суті на сьогодні він не відповідає таким безумовним вимогам, що були названі вище:

– не забезпечує задекларованої стійкості проти знаходження колізій;

– не забезпечує задекларованої стійкості до знаходження другого прообразу;

– не стійкий проти знаходження мультиколізій.

Крім того, він не відповідає таким сучасним умовним вимогам:

– має фіксовану довжину геш-значення 256 бітів;

– має низьку швидкодію.

За останньою характеристикою стандарт ГОСТ 43.311-95 поступається іншим, наприклад SHA-2 [1].

Атака знаходження колізій на ГОСТ 34.311-95 продемонстрована в роботі [6]. Складність цієї атаки оцінюється як 2^{105}, тоді як заявлена в стандарті 2^{128}. Тобто ГОСТ 34.311-95 не забезпечує задекларованої стійкості проти атаки знаходження колізій.

На даний час можна навести щонайменше дві атаки відносно знаходження другого прообразу. Перша з них описана в роботі [6] і має складність 2^{192}, тоді як заявлена в стандарті при довжині геш значення 256 бітів повинна складати 2^{256}.

У якості другої атаки можна вказати атаку знаходження другого прообразу Шнаєра-Келсі на клас функцій гешування з ітеративною структурою [13]. Необхідною умовою її здійснення є рівність розмірів внутрішнього стану і вихідного геш-значення.

Аналіз ГОСТ 34.311-95 показав, що його внутрішній стан складається з двох частин: проміжного геш-значення і проміжної суми блоків. Розмір кожної з частин 256 бітів. Отже загальний розмір внутрішнього стану можна оцінювати в 512 бітів. Але, як буде показано нижче, значення проміжної суми блоків криптоаналітик може легко контролювати. Тому ключовий елемент атаки –побудування так званих повідомлень, що розширюються. Вони представляють собою послідовність блоків $q \| \ldots \| q' \| M_i$, відносно яких виконується умова

$$H(h_{in}, M_i) = H(h_{in}, q \| \ldots \| q' \| M_i), \qquad (1)$$

де h_{in} -початкове проміжне геш-значення, M_i – деякий заданий блок (послідовність блоків) повідомлення, q, q' – «проміжні» блоки, значення яких обираються криптоаналітиком. Щоб загальне значення суми блоків не змінювалось, ці значення необхідно обирати таким чином, щоб їх сума дорівнювала нулю. Наприклад довільним чином обирати значення всіх проміжних блоків крім останнього. Значення останнього необхідно встановлювати таким чином, щоб сума всіх «проміжних» блоків дорівнювала нулю.

В роботі [13] показано, що складність такої атаки може складати приблизно $O(k2^{n/2+1} + 2^{n+1-k})$, де k –кількість побудованих послідовностей вигляду (1).

Для випадку ГОСТ 34.311 вона складає $O(k2^{129} + 2^{257-k})$. Обчисливши нулі першої похідної від функції складності знайдемо, що найменша складність атаки досягається при $k = 128$, тобто довжині повідомлення зі 2^{128} блоками. За цих умов складність можна оцінити як $O(128 \cdot 2^{129} + 2^{257-128})$.

Попередній аналіз отриманого значення дозволяє зробити висновок, що на сучасному рівні розвитку техніки максимально доступна криптоаналітику

довжина повідомлень складає приблизно 2^{32} байтів, що складає 2^{27} блоків. При цих параметрах складність атаки оцінюється як $O(27 \cdot 2^{129} + 2^{230}) \approx O(2^{230})$. Наведені дані дозволяють зробити висновок, що на даний момент атака не може бути реалізована практично. В той же час необхідно констатувати що вона набагато простіша, ніж описана в роботі [6].

Атака створення мультиколізій була запропонована в [11]. Необхідні умови її реалізації ті ж, що і для атаки Ннаєра-Келсі. Отже, ГОСТ 34.311- 95 вразливий проти вищезгаданої атаки.

Для доведення наведемо алгоритм знаходження проміжної колізії.

Алгоритм 1. Знаходження проміжної колізії зі збереженням суми.

Вхід:

h_i –вхідне проміжне геш-значення.

Вихід:

$M_i \| M_{i+1}$, $M_i + t \| M_{i+1} - t$ – пара послідовностей блоків повідомлень, для яких виконується $H(H(h_{in}, M_i), M_{i+1}) = H(H(h_{in}, M_i + t), M_{i+1} - t)$, а t – деяке довільне значення.

Алгоритм знаходження зводиться до виконання наступних дій.

1. Обрати випадково $M_i \| M_{i+1}$

2. Завести масив $htemp[2^{256/2}]$

3. Для t від 1 до $2^{256/2}$

3.1 $htemp[t] = H(H(h_{in}, M_i + t), M_{i+1} - t)$

4. Знайти такі m, n, що $htemp[m] = htemp[n]$

5.Повернути $M_i \| M_{i+1}$,

$M_i + n - m \| M_{i+1} - n + m$

Забезпечивши таким чином незмінність значення суми блоків, ми можемо використати алгоритм побудування мультиколізій, використовуючи метод, запропонований Joux [11] .

Також необхідно вказати, що нині діючий стандарт ГОСТ-34.311-95 підтримує вироблення геш-значення тільки фіксованої довжини – 256 біт. В той же час міжнародні стандарти ЕЦП, які діють або проходять процес гармонізації в Україні, використовують еліптичні криві з порядком базової точки від 2163 до 2509 [14,15]. Відповідну довжину повинні мати і геш-значення, які використовуються в відповідній ЕЦП. Тому можна зробити висновок, що використання геш-значень з довжиною 256 біт обмежує зверху стійкість систем з ЕЦП.

Також, як показали дослідження [1], стандарт ГОСТ 34.311-95 має низьку швидкодію, що стає критичним для значного числа інформаційно-телекомунікаційних систем. На наш погляд, недостатню швидкодію можна пояснити властивостями блокового симетричного шифру ГОСТ 28147-89, що використовується при обчисленні геш-значень.

Наприклад, за результатами вимірювань швидкодії ГОСТ 34.311-95 з асемблерними вставками, наведеними в [1], український стандарт вдвічі повільніший за неоптимізовану програмну версію SHA-2-256 [1]. Крім того ГОСТ 34.311-95 завдяки ітеративній архітектурі не допускає ефективне використання паралельних обчислень. Тому, для України дуже важливим є завдання розроблення та впровадження нового криптографічного примітиву типу геш-функції.

Вказана задача може бути вирішена або засобом розроблення нового стандарту і прийняття його в якості національного стандарту, або гармонізації стандарту, що буде або будуть прийняті в ході виконання проекту SHA-3 Competition.

3. Аналіз сучасного стану створення перспективних функцій гешування

Найбільш загальним критерієм, за яким класифікуються (поділяються) існуючі функції гешування, є тип їхньої архітектури виконання обчислень. В якості основних необхідно назвати такі [16–21]:

- архітектура Меркле-Дамгарда;
- деревовидна архітектура;
- архітектура Беларе.

Архітектура Меркле-Дамгарда є класичною, добре вивченою і користується найбільшою популярністю серед розробників [16-19]. Але, як встановлено на нинішній час, такий підхід не дозволяє ефективно реалізувати паралельні обчислення [3].

Деревовидна архітектура запропонована в [20]. Вона допускає використання паралельних обчислень, і, як наслідок, дозволяє підвищити швидкодію. При реалізації деревовидної архітектури складність гешування повідомлення з n блоків може складати $O(\log n)$, але за умови можливості використання достатньої кількості обчислювальних ядер. Серед деревовидних архітектур найбільше поширення знайшла архітектура двійкового дерева фіксованої «висоти» [3, 20].

В якості недоліків деревовидних архітектур необхідно відмітити такі :

- неефективність використання обчислювальних ядер, кількість яких відмінна від степені 2;
- порівняно великий обсяг пам'яті, яка потрібна для зберігання результатів проміжних обчислень, що є критичним при гешуванні на платформах з ма-

лим обсягом пам'яті (наприклад мікроконтролерах, смарт-картах);

– необхідність узгодження параметрів двійкового дерева вимагає додаткових витрат та може стати причиною несумісності систем різних розробників.

Архітектура Беларе, що запропонована в [21], дозволяє використовувати паралельні обчислення та, на відміну від деревовидної архітектури, більш гнучко керувати процесом обчислення. Також при її реалізації без узгодження параметрів можна використовувати всі наявні обчислювальні ядра. Крім того, коефіцієнт прискорення може збільшуватись пропорційно кількості обчислювальних ядер. На рис. 1 наведено схематичне зображення функції гешування архітектури Белларе:

Архітектура Белларе вимагає менше памяті і дозволяє більш ефективно реалізувати гешування на платформах з обмеженими апаратними ресурсами. Тому така архітектура має перевагу при створення нових функцій гешування з підвищеною швидкодією. Однак стійкість таких функцій гешування ще не доведена, більш того, наявні обґрунтовані підозри про їх слабкість. Тому рекомендувати таку архітектуру в якості перспективної поки що поспішно. Як показав міжнародний конкурс NIST SHA-3 Competition, на жаль, алгоритми з архітектурою, здатною виконувати паралельні обчислення, поки що не забезпечують належного рівня стійкості, крім того більшість з них була зламана [10].

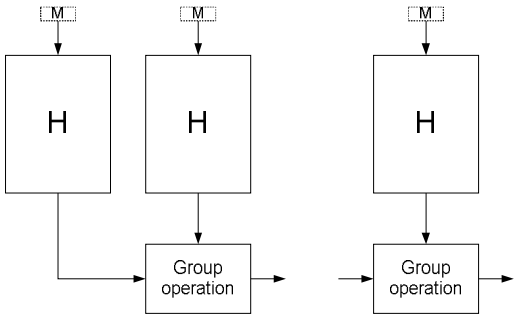

Рис. 1. Архітектура функції гешування Белларе (варіант)

Аналіз ходу виконання проекту NIST SHA-3 Competition показав, що станом на 18.01.2009 залишилось 10 учасників, які продовжують боротьбу у другому турі [10]. Також найбільш перспективними є алгоритми Blue Midnight Wish [23], Shabal [24] та Skein [22]. Перші два побудовані за архітектурою Меркле-Дамгарда та мають найкращі показники швидкодії [25]. Алгоритм Skein підтримує паралельні обчислення по дереву і разом з тим має класичну архітектуру функції стискання [22].

На даний момент ні на один з трьох названих алгоритмів гешування не знайдено жодної атаки, складність якої була б менше ніж складність атаки «груба сила» [10]. Проведемо аналіз названих функцій гешування більш детально. Функція гешування Blue Midnight Wish (BMW) [23] позиціонується розробниками як алгоритм, що побудований за класичними та перевіреними часом принципами.

Класична архітектура Меркле-Дамгарда, що реалізована з дотриманням в ній принципу wide pipe, коли розмір проміжного геш-значення відносно кінцевого більший принаймні вдвічі, забезпечує стійкість алгоритму проти атак length extension, мультиколізій [23] та атаки на знаходження другого прообразу [13]. Допустимими довжини геш-значень є 224, 256, 384 та 512 бітів. Фактично, розробниками запропоновано два алгоритми-BMW-256 та BMW-512. Геш-значення з довжинами довжиною 224 та 384 отримуються шляхом усікання геш-значень алгоритмів BMW-256 та BMW-512. Проте для вироблення геш-значення кожної з довжин передбачено окреме значення вектора ініціалізації та розміра слів, якими оперує алгоритм. Розмір слова складає 32 біта для алгоритмів з довжинами геш-значень 224 та 256 біт та 64 біта для алгоритмів BMW-384 та BMW-512. Загальний об'єм пам'яті, що необхідний для обчислення геш-значення, складає 264 байт для BMW-256 та 528 байт для BMW-512 .

Архітектура алгоритму послідовна, тому не дозволяє ефективно виконувати паралельні обчислення. В той же час, за результатами оцінки швидкодії кандидатів SHA-3 Competition, алгоритми BMW-256 та BMW-512 показали найкращі результати [5].

Функція гешування Shabal [24], що представлена на конкурс SHA-3 Competition, грунтується на використанні класичних підходів. В функції гешування Shabal [24] реалізовано схему Меркле-Дамгарда з підтримкою принципу wide pipe, що робить її захищеною від атак описаних в роботах [11, 13]. Ця функція гешування підтримує вироблення геш-значень з довжинами 192, 224, 256, 384, 512 бітів. Послідовний тип архітектури функції гешування Shabal не дозволяє ефективно реалізувати паралельні обчислення. За результатами оцінки швидкодії серед учасників другого туру SHA-3 Competition Shabal показав другий результат [5].

На наш погляд функція гешування Skein є однією з найбільш перспективних та, можливо, цікавих. Функція гешування Skein, на відміну від інших, характеризується великою кількістю налаштувань і режимів роботи.

Розробниками заявлено, що Skein [22] допускає довільну довжину вихідних значень, тобто геш-значень. Але найбільш важливим є те, що алгоритм може функціонувати як в послідовному, так і в паралельному деревовидному режимах. В останньому можливі налаштування висоти дерева, розміру листів дерева, та степені вкладеності (арності) дерева. Авто-

ри стверджують, що швидкодія Skein складає декілька тактів на байт. Такі показники є результатом оптимізації алгоритма під 64-бітну архітектуру.

В той же час відносно функції гешування Skein є деякі проблемні питання, до яких необхідно перше за все віднести такі:

– в сертифікатах відкритих ключів необхідно передбачувати поля для параметрів геш-функції, для чого необхідно змінювати структуру сертифіката і відповідне програмне забезпечення;

– неповні реалізації функції гешування, тобто які не підтримують усіх можливих значень параметрів, наприклад довжини геш-значення, можуть стати причиною несумісності систем різних розробників.

Однак слід відмітити, що розмір внутрішнього стану Skein може приймати також і значення 256, 512, 1024 бітів. У цілому ж, розмір вихідного геш-значення функції гешування, з урахуванням вимоги захищеності від атаки мультиколізій, не може перевищувати 512 бітів [11].

На рис. 2 показано схему вироблення вихідного геш-значення Skein.

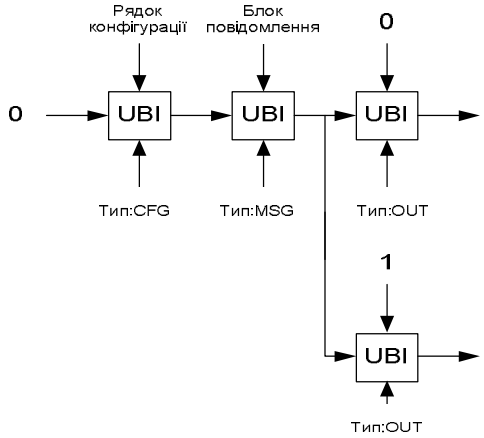

Рис. 2. Схема формування вихідного значення Skein

4. Перспективи використання в Україні

Як вже відзначалося раніше, і було показано у розділі 2 на даний момент існує потреба у перегляді діючого стандарту ГОСТ 34.311-95. Вимоги до нового національного стандарту гешування мають бути сформовані з огляду на сучасні світові тенденції у галузі криптографічного захисту інформації.

Використовуючи досвід проведених та перспективних міжнародних конкурсів на нові криптографічні стандарти (NIST SHA-3 Competition, NESSIE) необхідно визначити мінімальний перелік вимог, якому має відповідати геш-функція. Перелік вимог умовно поділяється на дві частини: вимоги до стійкості та вимоги до функціональності алгоритму. Вимоги до стійкості мають відповідати простому правилу – не повинно існувати жодної атаки на алгоритм складністю менше ніж атака «груба сила». Наступні критерії є ключовими для забезпечення захищеності, про що свідчать матеріали міжнародних конкурсів на нові криптографічні стандарти [2, 3,10]:

– складність знаходження колізії $2^{n/2}$;

– складність відновлення прообразу 2^n;

– складність знаходження другого прообразу не менше 2^n;

– стійкість до атак length extension;

– стійкість до усічених колізій;

– стійкість до атак мультиколлізій;

– відсутність атак розпізнавання для генераторів псевдовипадкових послідовностей, що використовують HMAC, побудованих на базі геш-функції зі складністю, меншою ніж знаходження другого прообразу і кількістю запитів до генератора не менше $2^{n/2}$;

– надійність математичної бази.

Вимоги до функціональності алгоритму гешування формуються зважаючи на ймовірну галузь застосування.

Перш за все необхідно забезпечити сумісність нового стандарту з вже діючими на території України стандартами ЕЦП, а також передбачити можливість суміщення з рядом міжнародних стандартів та перспективних стандартів, що можуть бути прийняти в межах України.

Функціональність нового стандарту гешування має відповідати наступному переліку вимог:

– довжина виробленого геш-значення;

– максимальна швидкодія;

– максимальна кількість процесорів, що може бути ефективно використана для паралельних обчислень;

– мінімальні вимоги до обчислювальних ресурсів;

– можливість реалізації алгоритма на різноманітних програмних, програмно-апаратних та апаратних платформах;

– простота архітектури алгоритма.

Новий національний стандарт, який має бути розроблено на заміну ГОСТ 34.311-95, планується використовувати у широкому колі криптографічних додатків, які висувають до нього ряд специфічних вимог технічного характеру. Галузь застосування нового алгоритму охоплює:

– використання виробленого геш-значення у якості інструмента контролю цілісності інформації при передачі, зберіганні та розповсюдженні. Найбільш відомими прикладами є протоколи встановлення та розповсюдження ключів, механізми надання послуги неспростовності, асиметричні шифри, систе-

ми електронного цифрового підпису з додатком або відновленням повідомлення та ін.;

– вироблення кодів автентифікації повідомлень за технологією HMAC;

– використання геш-функції у якості генератора псевдовипадкових послідовностей.

Наведений перелік є нормальною міжнародною практикою використання геш-функцій як криптографічних алгоритмів для вирішення задач криптографічного захисту інформації.

Висновки

У зв'язку з виявленням атак на ГОСТ 34.311-95 стає актуальною задача розробки нового стандарту гешування для України.

Створення нового стандарту–дорогий і складний процес. Тому використання міжнародного досвіду, накопиченого в ході конкурсу NIST SHA-3 Competition є перспективним напрямком.

За вимогами організаторів конкурсу претенденти на звання нового стандарту мають за своїми показниками безпеки та швидкодії перевершувати провідні існуючі стандарти. Основною рисою конкурсу є відкритість для участі в обговоренні та дослідженні алгоритмів учасників, а також вільний доступ до результатів цих досліджень. Таким чином, кожний алгоритм проходить апробацію з боку міжнародної криптографічної спільноти і тим самим зменшується ймовірність існування прихованих вразливостей.

Всі три описаних алгоритми є лідерами за швидкодією, тому можуть бути успішно використані в якості основи для національного стандарту України. Для вибору найкращого з них необхідно провести їх детальне дослідження і порівняння.

У зв'язку з використанням різних режимів роботи Skein, їх порівняння потребує подальших досліджень.

Література

1. Горбенко І.Д. Захист інформації в інформаційно-телекомунікаційних системах ч. 1 / І.Д. Горбенко, Т.О. Гріненко // Криптографічний захист інформації. – Харків: ХНУРЕ, 2004. – 368 с.

2. NESSIE Cryptographic call primitives / NESSIE – official website. – Режим доступу: https://www.cosic.esat.kuleuven.be/nessie/ – 20.10.2010. – Загол. з екрану.

3. Announcing request for Candidate algorithm nominations for a new cryptographic hash algorithm nominations for a new cryptographic hash algorithm (SHA-3) family / NIST Computer security resourcecenter. – Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/index.html – 20.01.2010. – Загол. з екрану.

4. Classification of the SHA-3 Candidates / Cryptology ePrint Archive. – Режим доступу: http://eprint.iacr.org/2008/511 – 20.01.2010. – Загол. з екрану.

5. Горбенко І.Д. Мета, стан та попередні підсумки проекту SHA-3 / І. Д. Горбенко, А.О. Бойко, А.М. Герцог // Прикладная радиоэлектроника. – 2009. – Т. 8. – № 3. – С. 315–320

6. Preimage Attacks on 41-Step SHA-256 and 46-Step SHA-512 / Cryptology ePrint Archive. – Режим доступу: http://eprint.iacr.org/2009/479.pdf – 20.10.2010. – Загол. з екрану.

7. Xiaoyun W. Finding collisions in the full SHA-1 / W. Xiaoyun, L. Y. Yiqun, Yu. Hongbo // In Victor Shoup, editor, Advances in Cryptology – CRYPTO '05. – 2005. – V. 3621. – P. 17–36.

8. ГОСТ 34.311–95. Інформаційна технологія. Криптографічний захист інформації. Функція хешування. – Введ. 1995-01-01. – К: Держспоживстандарт, 1994. – 23 с.

9. Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition / NIST Computer Security Resource Center. – Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/ documents/sha3_NISTIR7620.pdf – 20.10.2010. – Загол. з екрану.

10. Joux Multicollisions in Iterated Hash Functions Application to Cascaded Constructions / Joux // Crypto 04. – 2004. – V. 3152. – P. 306–316.

11. FIPS Pub 180-3 Secure Hash Standard (SHS) / NIST Computer Security Resource Center. – Режим доступу: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf - 20.10.2010. – Загол. з екрану.

12. Second Preimages on n-bit Hash Functions for Much Less than 2n Work / Bruce Schneier's security blog. – Режим доступу: http://www.schneier.com/paper-preimages.pdf. – 20.10.2010. – Загол. з екрану.

13. ДСТУ 4145-2002 Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка. Введ. 2002-01-01. К.: Держ. спожив. стандарт, 2001. – 26 с.

14. ISO/IEC 15946-4 Information technology — Security techniques — Cryptographic techniques based on elliptic curves – Part 4: Digital signatures giving message recovery. Введ. 2004-01-01. ISO/IEC, 2003. – 28 p.

15. Damgard I. AdDesign principle for hash functions / I. Damgard // Crypto 89. – 1989. – V. 435. – P. 416– 427

16. Design principles for iterated hash functions Candidates / Cryptology ePrint Archive. – Режим доступу: http://eprint.iacr.org/2004/253 – 20.01.2010. – Загол. з екрану.

17. Lucks S. Afailure-friendly design principle for hash functions / S. Lucks // In Proceeding of ASIACRYPT 2005. – 2005. – V. 3788. – P. 474–494.

18. C. Malinaud J.-S. Coron, Y. Dodisand P.Puniya. Merkle–Damgard revisited: How to construct a hash function / C. Malinaud, J.-S. Coron,

Y. Dodisand, P.Puniya // In Proceeding of CRYPTO2005. – 2005. – V. 3621. – P. 430–440.

19. A parallelizable design principe for cryptographic hash functions Lecture Notes in Computer Science / Cryptology ePrint Archive. – Режим доступу: http://eprint.iacr.org/2002/031 – 20.01.2010. – Загол. з екрану.

20. A new paradigm for collision-free hashing: incrementaly at reduced cost [Електроний ресурс] / Cryptology ePrint Archive. – Режим доступу: http://eprint.iacr.org/1997/001.ps – 2010.

21. The Skein Hash Function Family [Електроний ресурс] / NIST Computer security resourcecenter. – Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Skein_Round2.zip. – 2010.

22. Cryptographic Hash Function BLUE MIDNIGHT WISH Submission to NIST (Round 2) [Електроний ресурс] / NIST Computer security resourcecenter. – Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Blue_Midnight_Wish_Round2.zip – 20.01.2010. – Загол. з екрану.

23. Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition [Електроний ресурс] / NIST Computer security resourcecenter. – Режим доступу: http://csrc.nist.gov/ groups/ST/hash/sha-3/Round2/documents/Shabal_Round2.zip – 2010.

24. Matyas S.M. Generating strong one-way functions with cryptographic algorithms / S.M. Matyas, C.H. Meyer, and J. Oseas // IBM Technical Disclosure Bulletin. – 1985. – V. 27. – No. 10A. – P. 5658–5659.

Поступила в редакцію 19.01.2010

Рецензент: д-р техн. наук, заст. головного конструктора О. В. Потій, ЗАТ «Інститут інформаційних технологій», Харків.

СОСТОЯНИЕ СОЗДАНИЯ И НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ И РАЗРАБОТОК ПО ПО СОЗДАНИЮ ПЕРСПЕКТИВНІХ СТАНДАРТОВ ХЕШИРОВАНИЯ

І. Д. Горбенко, А. А. Бойко, А. Н. Герцог

Приводятся результаты анализа свойств стандартизированных функций хеширования, отмечаются их недостатки, обосновываются требования к перспективным функциям хеширования и определяются возможные методы их построения. . В результате аналіза современных тренований к хеш-функциям и состояния разработок по созданию перспективних стандартов хеширования сделаны выводы о необходимости замены действующего в Украине стандарта ГОСТ 34.311-95 новым и о возможности использования хеш-функций, предложенных в ходе конкурса NIST SHA-3 Competition, в качестве основы для нового стандарта хеширования.

Ключевые слова: функция хеширования, стандартизация, коллизионная стойкость, требования к хеш-функции.

STATE-OF-ART AND RESEARCHES ON PERSPECTIVE HASHING STANDARD DEVELOPMENT

I. D. Gorbenko, A. O. Boyko, A. M. Gertsog

The state-of-the-art review of hash functions research and development is presented. Disadvantages of already standardized hash functions are showed. Some methods of perspective hash function development are proposed. Analysis of modern requirements to hash functions and perspective hash functions standard development state shows the necessity to change the current Ukrainian standard GOST 34.311-95. Ability to use algorithms presented at NIST SHA-3 Competition as base for national standard is considered.

Key words: hash function, standardization, collision strength, requirements to hash functions .

Горбенко Іван Дмитрович–доктор техн. наук, професор, генеральний конструктор ЗАТ "Інститут інформаційних технологій", завідувач кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна, e-mail: GorbenkoI@iit.kharkov.ua.

Бойко Артем Олександрович–інженер-програміст ЗАТ "Інститут інформаційних технологій" аспірант кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна, e-mail: artboyko2006@gmail.com.

Герцог Андрій Миколайович–аспірант кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна e-mail: kuklowod@gmail.com.