УДК 681.3.07

МЕТА, СТАН ТА ПОПЕРЕДНІ ПІДСУМКИ ПРОЕКТУ SHA-3

І.Д.ГОРБЕНКО, А.В. БОЙКО, А.М. ГЕРЦОГ

Розглядається мета, аналізується стан та викладаються попередні підсумки виконання проекту NIST США SHA-3 з розроблення перспективних функцій гешування.

The paper considers the aim, analyzes the state and presents the preliminary results of realizing NIST SHA-3 project on the development of perspective hash functions.

ВСТУП

В 2007 році Національним інститутом стандартизації США (NIST) розпочато відкритий конкурс на проект нового федерального стандарту гешування SHA-3 під назвою NIST SHA-3 Competition [1]. Необхідність розробки проекту нового стандарту гешування (попередня назва SHA-3) обумовлена появою повідомлень про побудову алгоритмів атак на функції гешування SHA-1, яка зараз є федеральним стандартом гешування, та MD5, яка є стандартом де-факто. Так атака створення колізій відносно SHA-1 має складність 2^{69} [2], що не так вже й сильно відрізняється від значення складності знаходження колізії через парадокс днів народження. Дійсно, оскільки довжина вихідного блоку функції гешування SHA-1 складає 160 бітів, то складність знаходження колізії через парадокс днів народження оцінюється як 2^{80}. Але і значення складності 2^{69} на даному рівні розвитку обчислювальної техніки є практично недосяжним, а отже атака поки ще є теоретичною, без практичної, поки що, можливості реалізації.

Точна складність атаки на MD5 невідома. Невідомий також і алгоритм атаки. Однак, автори [3] стверджують, що час, необхідний для пошуку колізії на персональному комп'ютері, приблизно дорівнює 8 годинам і така атака практично реалізуєма. В [4] показано, як, користуючись цією атакою, можливо побудувати фальшивий центр сертифікації ключів і здійснити таким чином атаку man-in-the-middle на криптографічні протоколи, які використовують сертифікати, підписані з використанням функції гешування MD5. Найбільш поширеним з таких протоколів є SSL.

Однак тривогу викликає не цей факт. Більш тривожним є той факт, що MD5 і SHA-1 побудовані на спільних принципах і мають схожу архітектуру. Це означає, що атака на MD5 після певних модифікацій потенційно може бути застосована і проти SHA-1, SHA-256 та SHA-512, які мають схожу внутрішню структуру. Щоб запобігти такій ситуації, коли всі стандартизовані функції гешування можуть бути компрометованими NIST США в 2007 році оголосив конкурс на перспективні функції гешування, по суті з перспективою на майбутнє, на 5 років, тобто до 2012 року.

Крім того, з точки зору практики, з'явилось ряд побажань відносно зменшення складності обчислення функцій гешування і як наслідок підвищення швидкодії. Ця вимога особливо актуальна для застосування функції гешування в таких додатках, як електронний цифровий підпис, виробка псевдовипадкових послідовностей тощо.

Метою цієї статті є розгляд положень відносно умов організації та проведення конкурсу, аналіз вимог до кандидатів, визначення основних додатків з точки зору застосування, а також аналіз стану виконання проекту.

Вказаний напрям досліджень є дуже актуальним, так як стандарт гешування Російської федерації ГОСТ Р 34.11 – 94 так і міждержавний стандарт ГОСТ 34.311 – 95 уже не задовольняють вимогам як відносно стійкості, так і швидкодії.

1. УМОВИ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ КОНКУРСУ

До участі в конкурсі допускаються усі бажаючі[1], які надали у встановлені терміни необхідні дані, щодо свого кандидата. По закінченні строку прийому проводиться конференція, яка має на меті обговорення алгоритмів кандидатів та обґрунтування розробниками обраної методики для їх побудови. Оцінювання планується провести в два етапи, на кожному з яких алгоритми кандидати ретельно досліджуються групою експертів NIST. За регламентом обидва етапи є відкритими і кожен бажаючий може прийняти участь у процесі дослідження та обговорення того чи іншого алгоритму. Усім зацікавленим користувачам надається можливість вільного доступу до специфікацій алгоритмів кандидатів та звітів досліджень, що формують групи експертів NIST. Організаторами також підтримуються ініціативи щодо вдосконалення методики досліджень та критеріїв оцінки. Таким чином, основною ідеєю проведення конкурсу є відкрите дослідження та обговорення кандидатів.

Перший етап проводиться у вигляді відкритого обговорення, у якому приймають участь усі бажаючі та експерти NIST. До нього допускаються усі кандидати, що були прийняті до участі в конкурсі. На цьому етапі проводиться оцінка криптографічних вразливостей та додаткових переваг і недоліків запропонованих функцій гешування. Право визначати тривалість першого етапу залишає за собою NIST, проте попередньо відведений час становить 12 місяців. В процесі відкритого обговорення, з метою ефективної організації конкурсу, забороняється вносити зміни

до алгоритму гешування. Ближче до закінчення цього етапу для аналізу результатів виконаної роботи проводиться конференція. Мета проведення першого етапу конкурсу – визначити, використовуючи результати відкритого обговорення, п'ять кандидатів, які найбільше відповідають висунутим вимогам.

За попереднім задумом організаторів планувалось, що у другому етапі приймуть участь п'ять алгоритмів, котрі були обрані за результатами відкритого обговорення. Однак після додаткового обговорення у подовженому першому етапі NIST дозволив брати участь чотирнадцяти алгоритмам. На цьому етапі планується провести більш детальний аналіз складових частин та алгоритму кандидату в цілому. Форма проведення майже не відрізняється від попереднього, термін проведення становить 12-15 місяців та може бути змінений організаторами. Проте, до його початку, дозволяється внести незначні зміни в алгоритм для усунення можливих недоліків, після чого модернізовані версії алгоритмів будуть доступні для усіх зацікавлених на офіційній сторінці конкурсу. За результатами роботи другого етапу конкурсу планується провести конференцію та оприлюднити алгоритм, що буде включений до нового стандарту FIPS. Організатори не відкидають можливості того, що висунутим вимогам задовольнятимуть одразу декілька кандидатів. В такому випадку усі алгоритми буде включено до нового стандарту.

2. ВИМОГИ ДО ПЕРСПЕКТИВНИХ АЛГОРИТМІВ ГЕШУВАННЯ ТА ЇХ ЗАСТОСУВАННЯ

Таким чином, метою проведення конкурсу, як вже відзначалося раніше, є розробка алгоритму, який би зміг замінити діючий стандарт функцій гешування. В інформаційному листі NIST наведено ряд вимог до кандидатів на новий стандарт [1]. Основною вимогою є підтримка алгоритмом кандидатом можливості вироблення геш-значень довжиною 224, 256, 384, 512 бітів. Така вимога необхідна для того, щоб не порушити спадкову послідовність відносно SHA-2, і, таким чином, домогтися сумісності нового стандарту гешування з вже існуючими, в яких його планується застосовувати. Можливість функції гешування виробляти значення іншої довжини є перевагою, але за умови, що її визнають учасники відкритого обговорення.

NIST планує використовувати новий алгоритм гешування для надання таких послуг, як:

– вироблення та перевіряння електронного цифрового підпису за FIPS 186-2 та FIPS 186-3, Digital Signature Standard;

– обчислення кодів автентифікації повідомлень за FIPS 198, The Keyed-Hash Message Authentication Code (HMAC);

– встановлення ключів за SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography;

– генерування псевдовипадкових чисел згідно NIST SP 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (DRBGs).

Зважаючи на перелік сервісів, наведених NIST, в яких планується застосовувати функцію гешування, вимоги до її безпеки мають бути жорсткими. Відповідно до цього заявлені вимоги, щодо безпеки функції гешування, наступні:

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

– складність відновлення прообразу не менше 2^n;

– складність знаходження другого прообразу не менше 2^n;

– стійкість до атак length extension;

– стійкість до усічених колізій;

– відсутність атак розпізнавання 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 $2^{n/2}$.

Для геш-функцій з ітеративною структурою застосовується додатковий критерій – розмір внутрішнього стану повинен бути щонайменше вдвічі більший розміру вихідного значення. Причина його введення в атаці на ітеративну структуру, наведену в роботі [5].

Оцінка кандидатів проводиться шляхом відкритого обговорення, у якому беруть участь усі бажаючі та експерти NIST. Критерії, за якими буде оцінюватися функція гешування, наведено в інформаційному листі, проте організатори підтримують внесення пропозицій щодо вдосконалення запропонованих критеріїв та впровадження нових. Усі критерії оцінювання можна умовно поділити на *критерії стійкості, складність обчислення та характеристик застосування*.

Критерії стійкості визначають мінімальний рівень захищеності функції гешування від криптоаналітичних атак. Алгоритм кандидат не повинен мати вразливостей до відомих атак, а також, бажано, і до атак, що можуть бути створені в майбутньому. Критерії формуються згідно вимогам безпеки, які має задовольняти функція гешування.

Критерії щодо складності обчислення не мають чітких значень, на відміну від критеріїв стійкості. Організаторами запропоновано два з них: ефективність обчислень, яка характеризує швидкодію алгоритму, та об'єм пам'яті, що необхідний для обчислення геш-значення. Оцінка за цими критеріями має на меті визначення вимог до систем, в яких планується застосовувати алгоритм гешування.

Всього для участі у конкурсі подано 56 заявок. На першому етапі обговорення вибули 10 кандидатів, як такі, що визнані «зламаними». Ще 5 відкликані авторами.

3. АНАЛІЗ ПІДХОДІВ ДО ПОБУДУВАННЯ ПЕРСПЕКТИВНИХ ФУНКЦІЙ ГЕШУВАННЯ

При аналізі специфікацій геш-функцій, поданих на конкурс, можна виділити наступні підходи до їх побудови:

– ітеративна структура Меркли – Дамгарда (МД);

– потокова архітектура;

– обчислення по дереву.

– архітектура Беларе.

Ітеративна структура Меркли – Дамгарда – це класична архітектура, що запропонована в роботах [6, 7].

Потокова архітектура може бути представлена як потоковий шифр, який працює у зворотному напрямі, тобто не генерує гаму, а приймає повідомлення в якості такої гами на вхід. Геш - значенням в цьому випадку стає внутрішній стан лінійного рекурентного регістра. Характеризуються високою швидкодією. На конкурсі представлено всього три кандидати Streamhash, Boole та Waterfall[8-10], які вже визнані зламаними. Тому новим та актуальним завданням є детальне вивчення умов та вимог, яким повинні задовольняти функції перетворення.

Обчислення по дереву, це інший, порівняно новий підхід до побудування функцій гешування, що запропонований в роботі [11]. Особливістю цього підходу є можливість виконання паралельних обчислень, що дозволяє значно підвищити швидкодію алгоритму шляхом ефективного використання сучасних багатоядерних апаратних платформ. Така властивість є перевагою за визначенням NIST і враховується при виборі геш-функції.

Ще один підхід – архітектура Беларе, що запропонована в роботі [12]. Сутність даної архітектури полягає в розділенні операції стискання блоку повідомлення і операції комбінування. Всі операції стискання можуть бути виконані одночасно і паралельно. Операція комбінування представляє собою деяку бінарну операцію, як правило, в групі. Складність операції комбінування у порівнянні з операцією стискання дуже мала, тому функції гешування з такою архітектурою мають найкращі можливості з використання багатоядерних обчислювальних засобів.

В ході конкурсу було представлено нові цікаві підходи до побудування функцій гешування. Більшість нововведень спрямовані на зменшення складності гешування (підвищення швидкодії), вимоги до якої стають дедалі вищими. Деяким кандидатам вдалося перевищити по співвідношенню тактів/байт навіть AES, окремі апаратні реалізації якого досягають швидкодії 17 тактів/байт. В таблиці 1 наведено типи архітектури алгоритмів, які до кінця першого етапу були визнані незламними.

Таблиця 1

Функції гешування та типи їх архітектури

Назва алгоритму	Автор	Тип архітектури
ARIRANG	Jongin Lim	МД
BLAKE	Jean-Philippe Aumasson	МД
Blue Midnight Wish	Svein Johan Knapskog	МД
CHI	Phillip Hawkes	МД
ECHO	Henri Gilbert	МД
FSB	Matthieu Finiasz	МД
Fugue	Charanjit S. Jutla	МД
Grøstl	Lars R. Knudsen	МД
Hamsi	Özgül Küçük	МД
Keccak	The Keccak Team	МД
LANE	Sebastiaan Indesteege	МД
Lesamnta	Hirotaka Yoshida	МД
Luffa	Dai Watanabe	МД
MD6	Ronald L. Rivest	Дерево
SANDstorm	Rich Schroeppel	Дерево
Shabal	Jean-François Misarsky	МД
SHAvite-3	Orr Dunkelman	МД
SIMD	Gaëtan Leurent	МД
Skein	Bruce Schneier	Дерево
SWIFFTX	Daniele Micciancio	МД

Як видно із даних таблиці, абсолютну першість за популярністю у розробників посідає класична ітеративна архітектура Меркле-Дамгарда. Причиною такого широкого використання є бажання розробників скористатися перевагами консервативного підходу до побудови, а саме високим рівнем розвитку механізмів захисту від витончених атак. Проте, суттєвим недоліком такої архітектури є її непристосованість до виконання паралельних обчислень, що не дозволяє підвищувати швидкодію алгоритму без збитків для стійкості.

Загалом на конкурс було представлено 6 алгоритмів, які підтримують паралельні обчислення. В табл. 2 наведено перелік паралельних функцій гешування та типи їхньої архітектури.

Таблиця 2

Паралельні алгоритми, що було представлено на конкурс

Назва алгоритму	Автор	Тип архітектури
ECOH *	Daniel R. L. Brown	Белларе
EnRUPT *	Sean O'Neil	Дерево
ESSENCE *	Jason Worth Martin	Дерево
MD6	Ronald L. Rivest	Дерево
SANDstorm	Rich Schroeppel	Дерево
Skein	Bruce Schneier	Дерево

* – зламані алгоритми (на момент закінчення 1-го етапу).

Швидкість функцій гешування незалежно від принципів побудови суттєво відрізняється. В табл. 3 представлені дані відібрані із специфікацій алгоритмів, які ілюструють показники швидкодії алгоритмів, що претендували на участь у другому етапі.

Таблиця 3

Заявлена авторами швидкодія функцій гешування

Назва алгоритму	Автор	Тип архітектури	Швидкодія (тактів / байт)[1]	
			32 біта	64 біта
Arirang	Jongin Lim	МД		
224			21,6	16,1
256			21,5	16,1
384			65,1	12,8
512			65,2	12,9
Blake	Jean-Philippe Aumasson	МД		
224			27,4	18,4
256			27,4	18,4
384			61,3	13,8
512			61,3	13,8
Blue Midnight Wish	Svein Johan Knapskog	МД		
224			29,28	26,28
256			9,01	8,10
384			13,06	4,29
512			13,14	4,27
CHI	Phillip Hawkes	МД		
224			51	26
256			51	26
384			80	18
512			80	18
Echo	Henri Gilbert	МД		
224			–	–
256			–	–
384			–	–
512			–	–
FSB	Matthieu Finiasz	МД		
224			257	–
256			297	–
384			324	–
512			423	–
Fugue	Charanjit S. Jutla	МД		
224				
256				
384				
512				
Grøstl	Lars R. Knudsen	МД		
224			77,9	27,2
256			77,9	27,2
384			123,4	45,5
512			123,4	45,5

Продовження табл. 3

Назва алгоритму	Автор	Тип архітектури	Швидкодія (тактів / байт)[1]	
			32 біта	64 біта
Hamsi	Özgül Küçük	МД		
224			–	–
256			–	–
384			–	–
512			–	–
Keccak	The Keccak Team	МД		
224			–	–
256			–	–
384			–	–
512			–	–
LANE	Sebastiaan Indesteege	МД		
224			40,46	152,24
256			40,46	152,24
384			26,17	145,31
512			26,17	145,31
Lesamnta	Hirotaka Yoshida	МД		
224			68,9	78,4
256			68,9	78,4
384			97,7	65,4
512			97,7	65,4
Luffa	Dai Watanabe	МД		
224			–	–
256			–	–
384			–	–
512			–	–
MD6	Ronald L. Rivest	Дерево		
224			63	26
256			68	28
384			87	36
512			106	44
SANDstorm	Rich Schroeppel	Усічене дерево		
224			–	–
256			–	–
384			–	–
512			–	–
Shabal	Jean-François Misarsky	МД		
224			–	–
256			–	–
384			–	–
512			–	–
SHAvite-3	Orr Dunkelman	МД		
224			35,3	26,7
256			35,3	26,7
384			58,4	38,2
512			58,4	38,2

Закінчення табл. 3

Назва алгоритму	Автор	Тип архітектури	Швидкодія (тактів / байт)[1]	
			32 біта	64 біта
SIMD	Gaëtan Leurent	МД		
224			90	63
256			90	63
384			118	85
512			118	85
Skein	Bruce Schneier	Дерево		
224			38,1	9,8
256			38,1	9,8
384			44,3	7,3
512			44,3	7,3
Swifftx	Daniele Micciancio	МД		
224			–	–
256			–	–
384			–	–
512			–	–

[1] – на платформі заяв NIST: Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 32-bit (x86) and 64-bit (x64) Edition, наведені дані взяті із специфікацій алгоритмів, наданих авторами.

Наприкінці першого етапу серед кандидатів залишилась 21 функція гешування, на яку ще не було знайдено жодних атак [13]. Детально з аналізом по кандидатам можна ознайомитись в роботі [14].

Однак експерти NIST визначили 14 кандидатів, що можуть бути представлені у другому етапі змагання, за умови внесення певних змін, що посилюють стійкість, в специфікації. Всупереч очікуванням серед претендентів на місце нового стандарту не виявилось алгоритму MD6, який показав необхідний рівень стійкості, високу гнучкість та здатність до паралельного обчислення геш-значень. Вочевидь це зумовлено тим, що його функція стискання побудована за тими самими принципами, що і MD-х сімейства, на яке було знайдено ряд атак.

На даний момент актуальним завданням є проведення більш детального аналізу алгоритмів учасників другого етапу як за критеріями захищеності, так і за додатковими можливостями. З цією метою на основі представлених на конкурс вихідних програмних кодів було реалізовано стенд для вимірювання швидкодії. Отримані результати представлені в табл. 4.

Наведені дані ілюструють загальні можливості функцій гешування щодо швидкості виконання обчислень.

ВИСНОВКИ

На даний момент складно надати попередні прогнози стосовно будь-якого з алгоритмів, зважаючи на те, що перед початком другого етапу до алгоритмів можуть бути внесені зміни з метою їх удосконалення. Проте слід зауважити, що серед учасників лишився один алгоритм, архітектура якого підтримує паралельні обчислення. На нашу думку майбутнє геш-функцій саме за паралельними алгоритмами, зважаючи на сучасні тенденції розвитку апаратних частин обчислювальної техніки.

Таблиця 4

Геш-функції, що пройшли до другого етапу

Алгоритм	Швидкодія (тактів/байт)[1]
Blake	30.8633
BlueMindightWish	18.082
CubeHash	3657.95
Echo	72.1367
Fugue	122.855
Grostl	106.777
Hamsi	246.703
JH	89.1289
Keccak	73.4688
Luffa	38.6367
Shabal	19.7969
Shavite	59.5078
SIMD	149.445
Skein	46.2891

[1] – Тестова платформа: Core 2 Duo 2.66 GHz, RAM 2.0 Gb, OS Win XP Prof. (32 bit). Compiler – MVS 2005.

На момент здачі роботи до друку було знайдено атаки на алгоритми CubeHash та JH. Складність цих атак оцінена як така, що лише в рази менше атак грубою силою. Таким чином, кількість учасників другого етапу може скоротитися до 12. Крім того, як видно із таблиці 4 CubeHash має велику складність. Відповідно до вимірювань, представлених на [15], CubeHash має значні стрибки в складності обчислення для різних параметрів. Це може свідчити про залежність швидкодії від параметрів кеш-пам'яті процесора і можливо про слабкість до атак підрахунку промахів при зверненні до кеш-пам'яті, описаних в [16].

Література.

[1] Federal Register Notice published on November 2, 2007, http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf

[2] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu Finding Collisions in the Full SHA-1.

[3] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD.

[4] Marc Stevens, Alex Sotirov, Jake Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik and Benne de Weger Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate.

[5] John Kelsey and Bruce Schneier Second Preimages on n-bit Hash Functions for Much Less than 2^n Work

[6] Damgard. A design principle for hash functions. Crypto 89, LNCS 435, pp. 416–427.

[7] Merkle, R.C.: One Way Hash Functions and DES.

[8] Michal Trojnara StreamHash Algorithm Specifications and Supporting Documentation http://ehash.iaik.tugraz.at/uploads/0/09/Streamhash.pdf

[9] Gregory G. Rose Design and Primitive Specification for Boole http://seer-grog.net/BoolePaper.pdf

[10] Bob Hattersley NIST SHA-3 Competition Waterfall Hash Algorithm Specification and Analysis http://ehash.iaik.tugraz.at/uploads/1/19/Waterfall_Specification_1.0.pdf

[11] Palash Sarkar, Paul J. Shellenberg A Parallelizable Design Principle for Cryptographic Hash Functions

[12] Mihir Bellare, Daniele Micciancio A New Paradigm for Collision-free Hashing: Incrementality at Reduced Cost

[13] http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo

[14] Ewan Fleischmann1, Christian Forler, and Michael Gorski Classification of the SHA-3 Candidates

[15] http://bench.cr.yp.to/results-hash.html

[16] http://www.daemonology.net/papers/cachemissing.pdf

Надійшла до редколегії 18.09.09

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: геш-функції, побудування крипто примітивів з підвищеною швидкодією.

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