ПОСТКВАНТОВЫЕ ЕЛЕКТРОННЫЕ ПОДПИСИ

УДК 003.026:004.056

АНАЛІЗ ПОСТКВАНТОВИХ МЕХАНІЗМІВ ЕЛЕКТРОННИХ ПІДПИСІВ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Н.В.КОВАЛЬОВА, Ю.І. ГОРБЕНКО

Розглядається сутність криптоперетворень, що грунтуються на використанні функцій гешування при побудуванні постквантових електронних підписів. Наводяться результати аналізу криптографічної стійкості вказаних електронних підписів та даються оцінки існуючих алгоритмів та рекомендації з їх застосування.

Ключові слова: алгоритми постквантового електроннного підпису, функції гешування, криптографічна стійкість, застосування у постквантовий період.

ВСТУП

У 2016 році відбулось ряд значущих подій, які уже суттєво вплинули на інтенсивний розвиток постквантової криптографії. До них необхідно віднести заяву у вигляді Інтернет – статті Менезеса (ALFRED J. MENEZES) та Кобліца (NEAL KOBLITZ) [4], організацію та проведення NSA та NIST США VII міжнародної конференції з постквантової криптографії, що проходила у лютому 2016 року у Японії [4]. Надзвичайно важливою подією стало опублікування в США звіту «Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT) [5]», в якому повністю підтверджено можливості успішного квантового криптоаналізу асиметричних криптосистем електронного підпису(ЕП), а також визначені основні проблеми та можливості і етапи їх вирішення. Серед можливих методів та алгоритмів побудови постквантових асиметричних криптосистем ЕП виділяють метод, що грунтується на використанні криптографічних перетворень на функціях гешування (НВ криптографія)[2,3]. Можливість впровадження вказаного методу пов'язано з доведенням криптографічної стійкості, забезпеченням необхідної складності (швидкодії) перетворень та обґрунтуванням і побудуванням загальних параметрів та ключів. Зважаючи на стан досліджень в указаному напряму, на наш погляд, актуальними є задачі відбору та аналізу НВ криптоперетворень по критеріях криптографічної стійкості у постквантовий період, а також дослідження інших властивостей, що визначені NIST США та провідним інститутом ЄС ETSI[4,5].

В [4,5] наведено класи вимог, основними з них є: вимоги з безпеки; техніко-економічні вимоги та техніко-експлуатаційні вимоги. Також запропоновано критерії та показники відбору серед запропонованих кандидатів. Основними серед них є:

- модель безпеки для електронного (цифрового) підпису;
- вимоги до стійкості та додаткові властивості безпеки;
- обгрунтованість та довіра до методів, їх прозорість;
- розміри загальних параметрів та ключів;
- часова та просторова складності тощо.

Також важливим є обгрунтування та виконання відносно об'єктивного порівняльного аналізу ЕП, побудованих на основі різних методів і механізмів.

В ході розв'язання цієї задачі, в першу чергу бажано скористатись методиками, що, наприклад, подані в [5].

Метою цієї статті є відбір кандидатів на постквантові ЕП на основі НВ криптоперетворень, аналіз їх властивостей, обгрунтування та формулювання основних задач аналізу складності побудування параметрів та ключів, а також прямих та зворотних асиметричних криптоперетворень. Зрозуміло, що суттєво актуальною є проблема доведення криптографічної стійкості, на наш погляд вона розв'язуватиметься на світовому рівні ще декілька років.

1. КРИПТОСИСТЕМИ ЕП, ЩО ЗАСНОВАНІ НА ФУНКЦІЯХ ГЕШУВАННЯ

В [2,5] запропоновані криптографічні перетворення типу електронний підпис (ЕП) для постквантового періоду. Вони грунтуються на одноразових схемах ЕП з використанням геш - функцій. Попередній аналіз дозволив виділити серед них ЕП на основі геш - функцій, що відомі як Лампорт - Діффі або Вінтерніц підписи. Особливістю таких підписів є те, що їх криптографічна стійкість грунтується на колізійній стійкості геш-функцій, які застосовуються в механізмах (схемах) ЕП. Оскільки сьогодні уже розроблено та прийнято як геш-функції ряд, по суті постквантових геш-функцій, запропоновані механізми, на наш погляд, є перспективними.

Оскільки підписи Вінтерніц і Лампорт - Діффі не можуть бути використані надійно більш, ніж один раз, вони об'єднані з такими структурами, як бінарні де-

рева так, що замість використання ключа підпису для одноразового використання підпису, ключ може бути використаний для ряду підписів, кількість яких обмежена розміром двійкового дерева. Основною концепцією, що лежить у використанні двійкового дерева зі схемами геш - підпису є те, що кожна позиція на дереві розраховується як геш - конкатенація своїх дочірніх вузлів. Вузли, таким чином, обчислюються послідовно з коренем дерева, що є відкритим ключем глобальної схеми підпису. Листя дерева побудовані з одноразових ключів перевірки підпису.

Ця ідея була введена Merkle в 1979 році, але мала ряд недоліків – великі ключі, великий розмір підпису та достатньо повільний підпис.

Значна перевага ЕП на основі геш – це їх гнучкість, оскільки вони можуть бути використані з будь-якою захищеною функцією гешування, і тому, якщо дефект виявлений в захищеній геш - функції, ЕП просто необхідно перевести на нову – стійку геш - функцію, щоби ЕП був стійким.

Значним недоліком схем, пов'язаних з Merkle , є те, що підписувач повинен стежити, які одноразові ключі підпису вже використовувалися. Це може бути складним у великомасштабних середовищах. Варіанти без збереження стану є предметом поточних досліджень. З точки зору ефективності, наступні ітерації доробки значно поліпшили схеми ЕП на основі геш - функцій, але деякі недоліки залишаються. Так для отримання порівняння бітового рівня безпеки, екземпляр XMSS з AES-128 виробляє підписів в десять разів більше, ніж RSA-2048, тобто ЕП здійснює швидше.

На сьогодні існують декілька схем ЕП, заснованих на геш - функціях, серед яких схема Merkle, XMSS, SPHINCS.

2. ОСНОВНІ СХЕМИ ЕЛЕКТРОННИХ ПІДПИСІВ ПОСТКВАНТОВОГО ПЕРІОДУ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

2.1 Схема підпису Merkle

Найбільша проблема одноразових схем ЕП – це управління ключами [10]. Заміна відкритого ключа є дуже складною. Має бути гарантія того, що відкритий ключ належить передбачуваному партнеру з обміну даними і, що відкритий ключ не був змінений. Таким чином, мають використовуватися кілька відкритих ключів, і відкриті ключі мають бути досить короткими. Але в одноразових схемах для кожного підпису використовується новий відкритий ключ, який є досить великим порівняно з іншими схемами. Для того, щоб зробити одноразові схеми підпису можливими, необхідно застосовувати ефективне управління ключами, що зменшить кількість відкритих ключів та їх розміри. Merkle представив схему підпису Mercle (MSS), в якій один відкритий ключ використовується для підпису багатьох повідомлень.

2.1.1 Генерація ключових даних

Схема ЕП Merkle може використовуватися для підписування обмеженого числа повідомлень з одним відкритим ключем *Kpub*. Число можливих повідомлень має бути ступенем двійки, оскільки можливу кількість повідомлень можна позначити, як $N=2^n$. Перший етап формування відкритого ключа – це створення відкритих ключів X_i і секретних ключів Y_i для 2^n разових підписів. Для кожного відкритого ключа Y_i з $1 \le i \le 2^n$ обчислюється геш - значення $h_i = H(Y_i)$. З цими геш - значеннями будується дерево Merkle (так зване геш - дерево). Вузол дерева позначається $a_{i,j}$, де i – це рівень вузла. Рівень вузла визначається відстанню від вузла до листа. Отже, лист дерева має рівень i = 0 і корінь має рівень i = n. Всі вузли одного рівня нумеруються зліва направо, так що $a_{i,0}$ є крайнім лівим вузлом i-го рівня. У дереві Merkle геш - значення h_i є листям бінарного дерева, таким, що $h_i = a_{0,i}$. Кожен внутрішній вузол дерева є геш - значенням конкатенації двох своїх дочірніх вузлів. Так, $a_{1,0} = H(a_{0,0}||a_{0,1})$ та $a_{2,0} = H(a_{1,0}||a_{1,1})$. Приклад дерева Merkle показано на рис. 1. Таким чином будується дерево із 2^n листям та $2^{n+1} - 1$ вузлами. Корінь дерева $a_{n,0}$ є відкритим ключем схеми підпису Merkle .

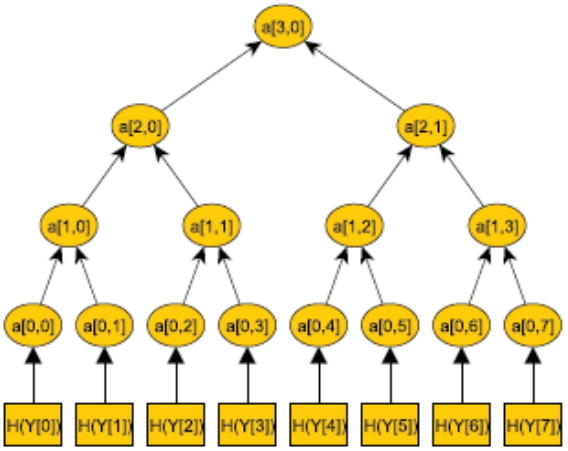

Рис.1. Дерево Merkle з 8 листями

2.1.2 Генерація підпису

Повідомлення M підписується одноразовою схемою підпису, утворюючи в результаті підпис *sig*'. Це робиться за допомогою однієї пари ключів (Xi, Yi,). Відповідний до відкритого однорозового ключа лист геш-дерева Y_i - це $a_{0,i} = H(Yi)$. Шлях геш - дерева є від $a_{0,i}$, до кореня A. Шлях A складається з n + 1 вузлів, A_0, ...A_n, з листям $A_0 = a_{0,i}$, та коренем дерева $A_n=a_{n,0}=$ = *pub*. Для обчислення шляху A потрібен кожний з дочірніх вузлів A_1, ..., A_n. Таким чином відомо, що A_i є дочірнім вузлом A_{i+1}. Для обчислення наступного вузла A_{i+1} шляху A необхідно знати обидва дочірніх вузла A_{i+1}. Тому потрібен вузол – «родич» Ai. Він позначається $auth_i$, такий, що $A_{i+1} = H(A_i||auth_i)$. Отже, для того, щоб обчислити кожен вузол шляху A потрібно n вузлів $auth_0$, ..., $auth_{n-1}$. Далі дані вузлів $auth_0$, ..., $auth_{n-1}$ зберігаютьсяо. Ці вузли з одноразовим підписом *sig*' повідомлення M є ЕП $sig = (sig'||auth2||auth3||...||auth_{n-1})$ схеми Merkle. Приклад

шляху автентифікації проілюстрований на рис. 2.

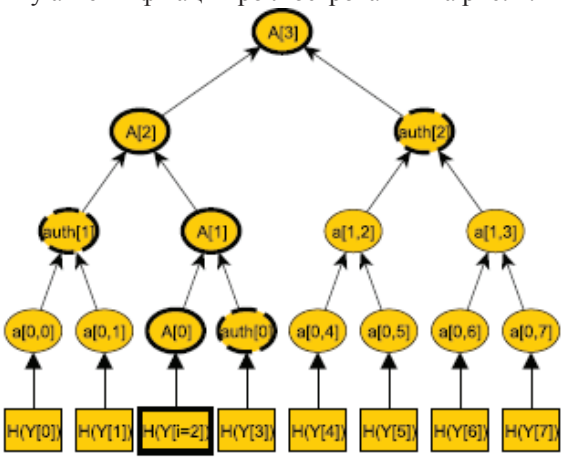

Рис. 2. Дерево Merkle з шляхом A та автентифікацією шляху для i=2

2.1.3 Перевірка підпису

Приймач знає відкритий ключ *Kpub*, повідомлення M, і підпис $sig = (sig'||auth0||auth1||...||authn-1)$. По-перше, приймач перевіряє одноразовий ЕП sig' повідомлення M. Якщо sig' є дійсним ЕП M , то приймач обчислює $A_0 = H(Y_i)$ шляхом гешування відкритого ключа одноразового підпису. Для j =1, .., n−1, вузли A_j шляху A обчислюються так : $A_j = H(a_{j-1}||b_{j-1})$.

Якщо A_n співпадає з відкритим ключем *Kpub* схеми підпису Merkle, то ЕП є дійсний.

2.1.4 Аналіз властивостей ЕП

Велика перевага схеми підпису Merkle в тому, що багато ЕП можуть генеруватися з використанням тільки одного відкритого ключа. Проте, ця перевага приходить зі збільшенням часу обчислень і довжини підпису.

Для того, щоб генерувати відкритий ключ *Kpub*, мають бути генеровані 2^n одноразових ключів підпису. Також має бути обчислений кожен вузол геш - дерева. Дерево складається з $2^{n+1}-1$ вузлів. Для розрахунку вузла необхідна одна геш - операція, так що $2^{n+1}-1$ геш - операцій необхідно для створення відкритого ключа. Очевидно, що розмір такого дерева обмежений. Так обчислення 2^{40} вузлів є дуже дорогим, обчислення 2^{80} вузлів – неможливим. Для створення підпису необхідні вузли $auth_0$, ..., $auth_{n-1}$. Якщо вузли дерева не збережені, вони мають бути генеровані знову для кожного підпису. Створення дерева – є надто складним, так що генерування всього дерева для кожного підпису нездійсненно для великих дерев. Але збереження всіх $2^{n+1}-1$ вузлів призведе до суттєвих вимог до зберігання даних. Отже, необхідна обґрунтована стратегія, щоб обчислювати ЕП швидко без збереження занадто багатої кількості вузлів. Ця проблема називається проблемою обходу дерева Merkle .

Час перевірки ЕП порівняно з часом підпису значно менший. Спочатку має бути перевірений одноразовий підпис. Після цього, обов'язково має бути обчислений шлях $A = A_1, ...,A_n$. Щоб зробити це, необхідно тільки n геш - операцій, по одній для кожного вузла. Підпис схеми Merkle складається з одноразового підпису sig' та n вузлів $auth_0, ..., auth_{n-1}$. Якщо використовується геш-функція 160 біт, розмір підпису буде $|sig| = |sig'| + n * 160$ біт.

2.2 Протокол Лейтона і Мікалі

Протокол ключової угоди (або розподілу) є набором правил зв'язку, за яких два користувачі можуть встановити загальний ключ. Загальний ключ може використовуватися користувачами в майбутньому для забезпечення захищеного зв'язку, наприклад, засобом симетричного шифрування.

На конференції CRYPTO'93 Лейтон і Мікалі запропонували два основні протоколи угоди [6], які були спрямовані на такі сценарії зв'язку, як засновані на чипі Clipper Chip. Потім пропозиції були додатково розширені у вигляді [7]. Перший протокол представлений в роботі [7] є новим і не описаний в [6]. Другий протокол в міститься в [7], по суті він такий же, як перший протокол, що міститься в [6]. Третій протокол також наведений в [7]. В ньому наведено механізм оптимізації другого протоколу [6]. Всі ці протоколи в [7] позначаються LM_{-1}, LM_{-2} і LM_{-3} відповідно.

ЕП LM_{-1} концептуально дуже простий. Проте, протокол не практичний з точки зору кількості секретних ключів, які мають зберігатися користувачем. У LM_{-1} кількість секретних ключів (кожен з k бітів, для кожного окремого користувача) знаходиться в діапазоні від O (B^2 Log N) до O (B^3 Log N), де N – загальне число користувачів і B – максимальне число нечесних користувачів у системі. Як правило, $k \geq 64$. Тепер припустимо, що LM_{-1} застосовується в країні з десятьма мільйонами ($N = 2^{23}$) користувачів, серед яких тисячі ($B \approx 2^{10}$) є зловмисниками. Тоді число секретних ключів, які кожен користувач повинен підтримати, дорівнює щонайменше 2^{24}, що ще гірше, ніж просте рішення, в якому кожен користувач тримає N - 1 секретних ключів.

LM_{-3} у першу чергу є варіантом без пам'яті з протоколу на основі узгодження ключа сервера автентифікації (наприклад, протокол Нідхам-Шредера). Секретний ключ бази даних сервера автентифікації видаляється за допомогою методики, яка в даний час стала класичним способом зниження пам'яті, а саме - використання криптографічно «сильної» псевдовипадкової функції. На практиці криптостійкі псевдовипадкові функції зазвичай реалізуються за допомогою секретного ключа алгоритму шифрування в ході застосування стійкого симетричного блокового шифру.

Протокол LM_{-2} заснований на захищеному VLSI чипі, який містить центральний процесор разом з внутрішньою пам'яттю. Він також припускає існування довіреного агента (або групи агентів з щонайменше одним надійним).

2.3 Схема підпису SPHINCS

Двома основними проблемами, що пов'язані зі схемами підпису на основі геш-функцій, є необхідність підтримувати стан і розмір підписів [8,11]. Цим

можна запобігти тому, щоб рішення, які засновані на геш - функціях, були відповідною заміною для схем ЕП, які використовуються в даний час. SPHINCS вирішує цю проблему шляхом об'єднання підходу Голдрайху з конструкцією традиційного дерева Merkle. Ця вбудована конструкція дерев становить основу SPHINCS.

Повна структура SPHINC складається в цілому з h шарів, поділених над d шарами суб - дерев. Це можна розглядати як гіпердерево двох рівнів абстракцій, де кожен вузол в глобальному дереві є суб-деревом. Кожне з цих суб-дерев складається з h/d шарів самих вузлів. Позначимо дерева, як τ_i , де $i \in \{1,\ldots,d\}$ представляє їх шар у глобальному дереві, і вузли в суб-дереві як $v_{i,j}$, де $j \in \{1,\ldots,h/d\}$ – це їх рівень в суб-дереві.

Дерева τ_i є бінарними геш - деревами, лише незначно змінюючись від вихідного дерева Merkle. Кожен з їх вузлів $v_{i,j}$ для $j \in \{1,\ldots,h/d - 1\}$ містить геш - значення його дочірніх вузлів, в той час як кожен з вузлів листа на шарі h/d містить ключ до OTS. Припустимо, що ми маємо деяку геш-функцію H, з використанням якої обчислюються ці геш-значення. Як і у випадку з деревами Merkle, дайджест в корені дерева може використовуватися для автентифікації всієї структури за допомогою побудови шляхів автентифікації. Всі ці суб-дерева потім з'єднуються один з одним, як у системі Голдрейх. Використовуючи OTS-ключі у вузлах листя $v_{i,h/d}$ дерев τ_i, підписуються кореневі вузли дерев τ_{i+1}; нове суб-дерево прикуте до кожного з вузлів листа. Примітки H містять геш своїх дочірніх вузлів, в той час як вузли OTS включають в себе пару ключів для автентифікації їх дочірніх вузлів.

Ключові пари OTS в листі дерев на нижньому шарі не використовуються для перевірки автентичності більшої кількості однакових суб-дерев [8,11]. Замість цього вони використовуються для перевірки справжності відкритого ключа схеми багаторазового підпису (FTS). FTS застосовується аналогічно до OTS, але може використовуватися кілька разів, перш ніж розкриє занадто багато даних секретного ключа. В ході використання FTS, а не OTS, SPHINCS не вимагає стільки вузлів листя для підтримки того ж рівня безпеки; необхідна максимальна ймовірність вибору повторно одного і того ж вузла може бути набагато вище, не порушуючи систему. Весь шлях на нижньому шарі дерева для підпису повідомлень використовуються ці FTS-ключі.

Рисунок 3 описує основну схему SPHINCS.

2.3.1 Генерація ключових даних

Генерація ключів для SPHINCS є досить дешевою операцією завдяки структурі Голдрайху. Виділяються випадкові значення $SK_1 \in \{0, 1\}^n$ і $SK_2 \in \{0, 1\}^n$. Перше з них використовується для генерації ключів, але зручно мати довгострокове випадкове число,

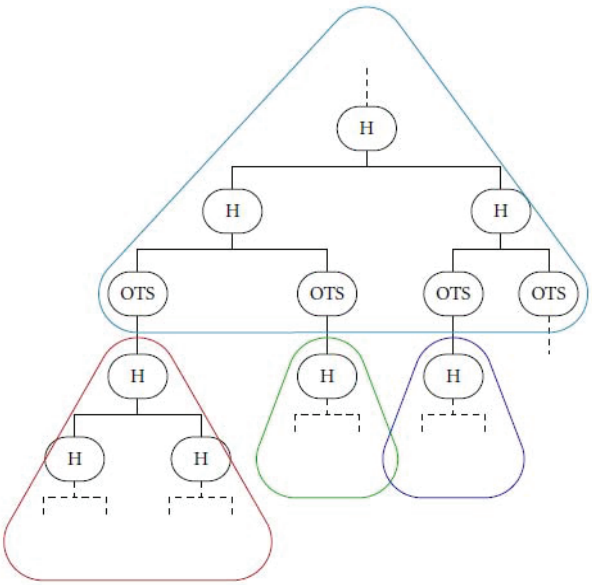

Рис. 3. Зв'язування суб-дерев разом

також доступне для підписання. Крім того, генерується безліч Q випадкових бітових масок, також у межах $\{0, 1\}^n$, які використовуватимуться в різних місцях. Ці маски використовуються у всіх геш - деревах, також як і в OTS, і FTS – на даний момент. Отже, $SK = (SK_1, SK_2; Q)$.

Для того, щоб генерувати відкритий ключ, необхідно сформувати принаймні одне дерево – τ_i -дерево у верхній частині конструкції. Це вимагає генерації OTS-ключів вздовж нижнього шару цього дерева. Важливо, що ці ключі мають бути генеровані детермінованим чином; використовуючи адрес і SK_1 як вхідні дані в деякій односторонній функції можна отримати випадкові значення (seed) для цього ключа. Потім бінарне геш - дерево може бути побудовано на відкритих ключах ключової пари OTS, а також кореневий вузол цього дерева є частиною відкритого ключа SPHINCS (PK_1). Оскільки бітові маски необхідні для перевірки, то вони також мають бути включені з відкритим ключем: $PK = (PK_1; Q)$.

Слід відзначити, що, в той час як SK_1, SK_2 і PK_1 мають розмір n бітів, Q – значно більше. Бітові маски, таким чином, складають найбільшу частину ключів. В SPHINCS-256 конфігурації, наприклад, PK_1 містить тільки 32 байти, в той час, як Q відповідає за 2* 32* 16 = 1024 байт. У загальному випадку, кількістю бітових масок визначається частина схеми, яка вимагає найбільшого числа FTS, OTS або геш - дерев.

2.3.2 Електронний підпис

Як правило, в схемах ЕП з відкритим ключем спочатку обчислюється геш-значення повідомлення, яке має бути підписане, а потім підписують отримане геш - значення. Це гарантує, що вхідна константа буде досить невеликої довжини. У схемі без збереження стану (для схеми Голдрейх), потім обирається випадкова пара ключів в нижньому шарі дерева, щоб підписати геш-значення. Але у схеми SPHINCS, однак,

пара ключів вибирається на основі самого геш-значення повідомлення. Для того, щоб запобігти атакам на конкретні ключові пари, має бути включений деякий випадковий або невідомий фактор – це ключ SK_2. Спочатку потрібно обчислити значення R за допомогою псевдовипадкової функції, яка приймає SK_2 і повідомлення як вхідні дані, а потім використовує частину цього повідомлення, випадкової величини R, щоб вибрати пару ключів FTS. Крім того, інша частина R використовується для обчислення випадкового геш-значення D повідомлення. Цей дайджест належить підписуванню. Як практичний результат всього цього є вибір пари ключів FTS, що є повністю детермінованим щодо секретного ключа SK_2 і повідомлення M.

Після вибору конкретної пари ключів FTS, вона має бути сформована на основі випадкових значень, отриманих від його місця розташування і SK_1, а потім використовуватися для підпису D для отримання підпису σ_{FTS}. Разом з повідомленням, отриманим вище і індексом idx обраної пари ключів, цей підпис формує першу частину загального підпису SPHINCS. Позначимо цей ЕП SPHINCS як Σ. Потім генерується пара ключів OTS для батьківського вузла $v_{d,h/d}$ у відповідному суб - дереві у τ_d (знову використовуючи його позицію і SK_1), вона і використовується, щоб підписати відкритий ключ FTS. Позначимо вироблений підпис, як $\sigma_{OTS,d}$. Цей підпис також додається до Σ. Відкритий ключ цього OTS має пройти перевірку справжності, тому обчислюються всі вузли вздовж його шляху автентифікації по всьому дереву у τ_i і включаються в Σ . Позначимо вузли шляху автентифікації в обраному дереві на шарі d як $Auth_d$. При досягненні кореня дерева генерується ключова пара OTS, яка належить до батьківського вузла у $v_{d-1,h/d}$ і використовується для підписання. Ця процедура триває весь шлях до кореня одного дерева в τ_i ,який включений в РК. На шляху всі підписи OTS і вузли поряд зі шляхом автентифікації мають бути додані до Σ.

У цілому підпис SPHINCS Σ тепер містить випадкове R, індекс обраної пари ключів FTS, підпис σ_{FTS} та d пар підписів OTS і вузлів вздовж шляху автентифікації ($\sigma_{OTS,i}$, $Auth_i$). Все разом: Σ = (idx, R , σ_{FTS}, ($\sigma_{OTS,1}$, $Auth_1$),..., ($\sigma_{OTS,d}$, $Auth_d$).

2.3.3 Перевірка підпису

Процедура для перевірки ЕП повідомлення M дуже схожа на сам ЕП. Після обчислення D (з використанням R і M), підпис FTS є верифікованим. Як було згадано вище, функції перевірки OTS і FTS, що використовуються в SPHINCS, виводить відкритий ключ. Підписи OTS на цьому відкритому ключі тепер можуть бути верифіковані, в результаті чого отримується відповідний відкритий ключа OTS. Оскільки шлях автентифікації також наводиться в Σ, то тепер може бути обчислений кореневий вузол дерева τ_D. Подібно до того, як створювався підпис, далі продовжуємо вгору по дереву вздовж шляхів автентифікації з перевіркою підпису на кореневих вузлах кожного суб-дерева. З рештою, верифікація переходить до кореневого вузла одного дерева в τ_i. Цей кореневий вузол має бути рівний PK_1, включеного в РК. Якщо це так, то ЕП дійсний.

2.4 Схема підпису XMSS

Розширена схема підпису Merkle (XMSS - The eXtended Merkle Signature Scheme) була введена в 2011 році і стала проектом в 2015 році [12].

Основна конструкція (Рисунок 4) [12] виглядає як дерево Merkle, за виключенням деяких речей. Дерево XMSS має маску для операції XOR дочірніх вузлів перед тим, як вони гешуються у вузлі їх «батьків». Це інша маска для кожного вузла.

Друга особливість полягає в тому, що лист дерева XMSS не є гешем одноразового відкритого ключа підпису. Корінь іншого дерева називається L-деревом. L-дерево має ту саму ідею масок, застосованих до його вузлів геш-значень, на відмінну від основних

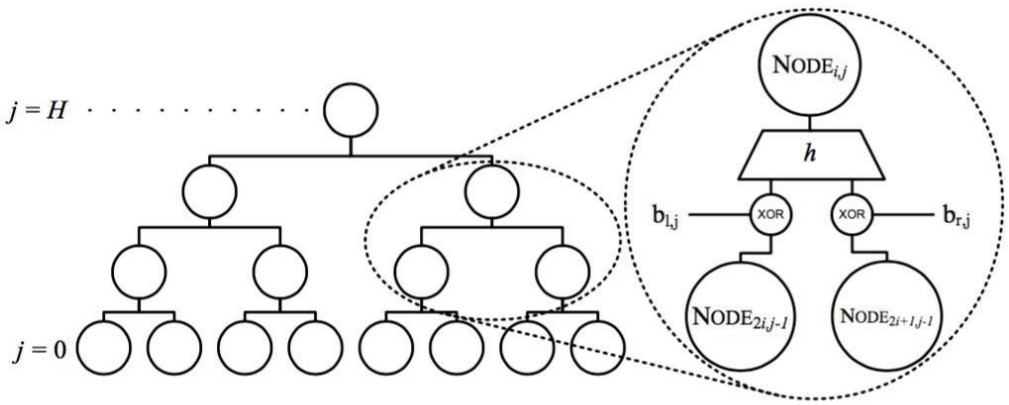

Рис. 4. Конструкція дерева XMSS

XMSS дерев, але спільну для всіх L-дерев. Всередині листя будь-якого L-дерева зберігаються елементи WOTS + відкритого ключа. Дерево не використовується для зберігання відкритого ключа WOTS, але використовується для гешування його таким чином, що можна довести, що геш-функція другого прообразу досить стійка (замість стійкості до колізії). Крім того, основний відкритий ключ складається з кореневого вузла дерева XMSS, а також бітових масок, що використовуються в дереві XMSS і L-дереві.

XMSS є більш актуальною схемою, що знаходиться в процесі стандартизації. Вона заснована на Деревах Merkle з такими поліпшеннями:

- більша ефективність при обході дерева, тобто обчислення шляху вузлів, пов'язаних з заданою одноразовою схемою підпису до кореня дерева і загального відкритого ключа.
- зменшені розміри приватного ключа та краща секретність за рахунок використання генератора псевдовипадкових чисел для створення ключів одноразового підпису.

3.БЕЗПЕКА СХЕМ ПІДПИСУ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Традиційно, безпеку НВ схем підпису було пов'язано зі стійкістю до колізій використовуваної геш-функції. В останні роки кілька робіт зосереджені на основі безпеки на більш м'яких припущеннях, таких як стійкість другого прообразу. Є дві основні причини слідування цієї тенденції. З одного боку, атаки проти стійких до колізій SHA1 і MD5 мотивували дослідників розробляти схеми підпису, стійкі до колізій [9]. З іншого боку, стійкість до колізій породжує атаки в той час, як стійкості (другого) прообразу немає. Отже, щоб досягти рівня безпеки λ бітів, необхідна геш - функція з $n = 2\lambda$ біт геш-значень якщо потрібна стійкість до колізії, хоча стійкість прообразу (або другого прообразу) лише для $n = \lambda$ дайджестів. При чому скорочення вихідного розміру геш-функції, що використовується, одразу вдвічі скорочує підписи і розміри ключів схем ЕП, що засновані на геш-функціях.

Дане судження є лише половиною правди, бо воно тримається на неявному припущенні, що геш-функція використовується тільки один раз. Очевидно, що для багатьох криптографічних конструкцій це не так. Розглянемо це на прикладі стійкості прообразу (one-wayness). Для багатьох криптографічних конструкцій при противник зможе взнати величину значень функції, і відбудеться компрометація, як тільки він знайде прообраз для одного з них. Більш конкретно, припустимо, що геш - функція з n бітовими виходами використовується d раз у криптографічній конструкції. Якщо успішно інвертувати геш - функцію на будь-якому з виходів, то це негативно вплине на безпечність схеми. В цьому випадку складність атаки знижується до $O(2^n / d)$ замість $O(2^n)$. Інтуїтивно зрозуміло, що це відбувається тому, що кожне вхідне значення, яке противник намагається дістати, має ймовірність бути правильним $d / 2^n$ замість $1 / 2^n$. Це справедливо, якщо геш-функцію розглядати як випадкову функцію.

4.УДОСКОНАЛЕНІ СХЕМИ ПІДПИСУ

4.1 XMSS-T

У 2016 році запропоновано схему підпису XMSS-T [9], яка не є уразливою для багатьох цільових атак. З цією метою були запропоновані нові поняття (концепції) стійкості для прообразу, другого прообразу і стійкості до колізій. Була проаналізована загальна безпека геш - функцій в зв'язку з цими новими властивостями проти класичних і квантових атак. Це дозволило отримати верхні і нижні оцінки складності загальних атак.

Більш конкретно, перший тип понять моделює концепцію (з однією функцією, багатоцільову), яка неявно використовується в недавніх схемах підпису, стійких до колізій, таких, як XMSS, XMSSMT і SPHINCS [8,9,11,12]. У цій концепції, противник А отримує цільові значення р і випадкову функцію з сімейства геш - функцій. Потім А намагається знайти прообраз (або другий прообраз, відповідно) для одного з цільових значень в рамках даної геш - функції. Доведено, що порівняно зі стандартною стійкістю (до другого прообразу), складність запиту загальних атак знижується на коефіцієнт р для класичних і на $\sqrt{p}$ для квантових. Потім була введена інша концепція з багатьма функціями (багатоцільова) зі стійкістю до прообразу і стійкості другого прообразу. Для такої концепції, А дає кілька пар функцій і цільові значення, отримані незалежно один від одного в випадковому порядку. Тепер мета А - знайти прообраз (або другий прообраз, відповідно) для одного з цільових значень за відповідною функцією. Доведено, що в цьому випадку складність запиту загальних атак така ж, як і для стандартних (з однією функцією, тобто одноцільовою) понять. З огляду на те, що багатофункціональні і багатоцільові поняття настільки ж важкі, як стандартні уявлення про стійкість прообразу і другого прообразу, побудована нова схему підпису з безпекою на основі цих нових концепцій. Оскільки основна конструкція наслідує це з XMSS, то нову схему позначають XMSS-T, що вказує на XMSS з посиленням заходів безпеки. У той час як XMSS втрачає багато в бітовій безпеці за кількома параметрами, включаючи загальну висоту дерева, XMSS-T втрачає тільки два біти, незалежно від будь-яких параметрів. Відмінністю між XMSSMT і XMSS-T є різні геш - дерева і така одноразова схема підпису, що безпека може бути заснована на багатофункціональних багатоцільових властивостях. Основна зміна полягає в тому, що для кожного виклику геш - функції в геш - дереві або геш-ланцюзі використовуються різні ключі геш - функцій і різні бітові маски. XMSS-T-схема зі збереженням

стану, тому вона може бути не придатна в деяких практичних випадках використання. Але позитивним є те, що можна легко зробити аналогічні зміни в схемах підпису SPHINCS без збереження стану. Грубо кажучи, це зводиться до заміни геш – дерев, що використовуються, а також одноразових підписів.

4.2 SPHINCS-256

Параметри схеми SPHINCS-256 [8,11] обрані з двома цілями: довгострокова безпека 2^{128} від зловмисників, що мають доступ до квантових комп'ютерів, та достатній компроміс між швидкістю і розміром підпису. Перша мета визначається параметром безпеки n=256, який в свою чергу визначив назву SPHINCS-256. Оптимізуючи інші параметри, потрібно прийняти рішення про відносну важливість швидкості і розміру підпису. Після пошуку у великому просторі параметрів були вибрані такі [ ]: *m = 512, h = 60, d = 12, w= 16, t = 216;*

k = 32, l = 67, x = 6, a = 64.

Для ряду додатків такий вибір досить простий і досить швидкий. Звичайно, можна також визначити різні реалізації SPHINCS, змінюючи інші параметри на користь тієї чи іншої вимоги, наприклад складності чи розміру підпису.

В SPHINCS-256 як параметри використовуються такі значення:

n = 256; m = 512; h = 60; d = 12; w = 16; t = 2^{16}; k=32.

Тому, беручи до уваги порушників, які мають доступ до потужних квантових комп'ютерів, можна зробити такий висновок. Якщо припустити, що кращі атаки проти геш-функції, що використовується, є загальними атаками, то $H_{к,t}$ забезпечує безпеку вище 2^{128} щодо стійких підмножин, а F і H забезпечують 2^{128} захист від атаки до знаходження прообразу, другого прообразу, а також атак невиявлення F. Аналогічним чином, значення PRFs і PRGS забезпечують безпеку 2^{128}. Таким чином, SPHINCS-256 при вказаних припущеннях забезпечує захист від постквантових порушників 2^{128}.

ВИСНОВКИ

НВ схеми підпису вважаються найбільш перспективною постквантовою альтернативою існуючим схемам, таким як RSA і ECDSA, уразливим для квантових атак. Це особливо актуально, тому що безпека криптографічних геш - функцій добре вивчена. Крім того, є точні докази, що встановлюють зв'язок між складністю порушення схеми та складністю порушення властивостей безпеки геш-функцій, які використовуються в схемах. Це дозволяє точну оцінку безпеки конкретних наборів параметрів.

Важливим є те, що на сьогодні в Україні розроблені та стандартизовані по суті постквантові алгоритми гешування – геш-функції ДСТУ 7564:2014. «Інформаційні технології. Криптографічний захист інформації. Функція гешування».

Література

[1] *Горбенко І.Д., Горбенко Ю.І.* «Прикладна криптологія». Теорія. Практика. Застосування. Монографія. Видання 2-ге, перероблене й виправлене. Харків. Видавництво «ФОРТ». 2012. – 878с.

[2] *Горбенко Ю.І.* Методи побудування та аналізу, стандартизація та застосування криптографічних систем: Монографія / За загальною редакцією Професора Горбенко Івана Дмитровича. – Харків: Форт, 2015. – 959 с.

[3] *Горбенко Ю. І.* Аналіз можливостей квантових комп'ютерів та квантових обчислень для криптоаналізу сучасних криптосистем / Ю. І Горбенко, Р. С. Ганзя. // Східно-європейський журнал передових технологій. – 2014, № 1/9 (67). – С. 8–15.

[4] A RIDDLE WRAPPED IN AN ENIGMA. NEAL KOBLITZ AND ALFRED J. MENEZES Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A

[5] *Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone.* Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT). https://www.google.com.ua/search?

[6] *T. Leighton and S. Micali*, New approaches to secret-key exchange, Presented at Crypto'93.

[7] *T. Leighton and S. Micali,* Secret-key agreement without public-key cryptography (extended abstract), in: Advances in Cryptology - CRYPTO'93, Lecture Notes in Computer Science 773 (Springer, Berlin, 1994). 456-479.

[8] *J.Rijneveld*. Implementing SPHINCS with restricted memory.2015

[9] *A.Hulsing, J.Rijneveld, F.Song.* Mitigating multi-target attacks in hash-based signatures.

[10] *G.Becker*. Merkle signature schemes, Mercle Trees and their cryptanalysis.2008.

[11] *D.Bernstein, D.Hopwood, A.Hulsing, T.Lange, R.Niederhagen, L.Papachristodoulou, M.Shneider, P.Schwabe, Z.Wilcox-O'Hearn*.SPHINCS: practical stateless hash-based signatures.2015.

[12] *J.Buchmann, E.Dahmen, A.Hulsing*. XMSS – A practical forward secure signature scheme based on minimal security assumptions.2011

Ковальова Наталія Вікторівна, студентка кафедри БІКС ХНУ ім.В.Н.Каразіна. Область наукових інтересів: дослідження міжнародних стандартів, аналіз стійкості цифрових підписів, постквантова криптографія.

Горбенко Юрій Іванович, канд. техн. наук, технічний директор проектів АТ «ІІТ», лауреат державної премії. Наукові інтереси: проектування та застосування криптографічних комплексів, систем та засобів, інфраструктур відкритого ключа.

УДК 003.026:004.056

Анализ постквантовых механизмов цифровой под.-писи на основе хеш-функций / Н.В.Ковалёва, Ю.И.. Горбенко // Прикладная радиоэлектроника: науч.-техн. журнал - 2016. – Том 15. №3. – С. 195 – 202.

Рассматривается суть криптопреобразований, которые основываются на использовании хеш-функций при построении постквантових электронных подписей. Приведены результаты анализа криптографической стойкости указанных электронных подписей и оценки существующих алгоритмов, а также рекомендации по их применению.

Ключевые слова: алгоритмы постквантовой электронной цифровой подписи, функции хеширования, криптографическая стойкость, применение в постквантовый период.

Ил.: 04. Библиогр.:12 назв

UDC 003.026:004.056

Analysis of postquantum digital signature schemes based on hash functions / N.V.Kovaleva, Yu.I. Gorbenko // Applied Radio Electronics: Sci. Journ. – 2016. Vol. 15. №3. – P. 195 – 202.

The paper considers the essence of cryptotransformations based on hash functions for building postquantum digital signature schemes. The results of analyzing the cryptographic strength of these electronic signatures and evaluation of existing algorithms and recommendations for their use are given.

Keywords: postquantum digital signature algorithms, hash-functions, cryptographic strength, use in postquantum period.

Fig.: 04. Ref.: 12 items.