**УДК 004.056.55** **DOI:10.30837/rt.2020.4.203.02**

*М.В. ЄСІНА, канд. техн. наук, Б.С. ШАХОВ*

# АНАЛІЗ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ ЦИФРОВОГО ПІДПИСУ PICNIC

## Вступ

Важливою особливістю постквантового періоду у криптографії є суттєва невизначеність щодо вихідних даних для криптоаналізу та протидії в частині можливостей квантових комп'ютерів, їх математичного та програмного забезпечення, а також застосування квантового криптоаналізу до існуючих криптоперетворень та криптопротоколів. В якості основних методів обрано математичні методи цифрового підпису (ЦП), що пройшли суттєвий аналіз та обґрунтування в процесі широких досліджень криптографами та математиками на найвищому рівні. Вони детально описані та пройшли дослідження на першому етапі міжнародного конкурсу NIST США. В процесі другого етапу прийнято ряд рішень стосовно об'єднання деяких кандидатів на постквантовий стандарт ЦП [6].

Для подальших досліджень на 2-му етапі залишили 9 кандидатів: CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow та SPHINCS+. Три з них (Dilithium, FALCON, qTeSLA) засновані на стійкості алгебраїчних решіток (Lattice-based), чотири (GeMSS, LUOV, MQDSS, Rainbow) – на основі багатовимірних перетворень (MQ-перетворення), один (SPHINCS+) – на стійкості геш-функції, один (Picnic) – на стійкості геш-функції та блокових потокових шифрів [4].

За результатами досліджень перспективних постквантових криптографічних алгоритмів типу цифровий підпис протягом 2-го раунду конкурсу NIST США було отримано наступні результати – обрані алгоритми-фіналісти та альтернативні алгоритми. У якості алгоритмів-фіналістів були обрані такі алгоритми ЦП як CRYSTALS-DILITHIUM, FALCON та Rainbow. У якості альтернативних алгоритмів – GeMSS, Picnic та SPHINCS+ [7].

Далі у роботі більш детально розглянемо перспективний постквантовий алгоритм ЦП, що входить до переліку альтернативних алгоритмів – Picnic.

## 1. Опис та параметри алгоритму Picnic

### 1.1. Опис алгоритму

Алгоритм цифрового підпису Picnic – це схема підпису, яка не використовує теоретико-числових або структурованих припущень складності. Зменшення безпеки відносяться до геш-функцій і симетричних блокових шифрів. Підпис Picnic базується на неінтерактивному нульовому доказі знання секретного ключа. Підписується відкритий текст таким чином (через гешування), що тільки власник секретного ключа може вивести доказ і перевірити текст на правильність. Алгоритм ЦП Picnic має невеликий розмір відкритого ключа, але великі підписи. Підпис як і перевірка підписаного тексту досить повільні. Генерація ключів досить ефективна. Довжина підпису залежить від мультиплікативної складності схеми шифрування і від конкретної методики побудови доказу нульового знання (з області безпечних багатоваріантних обчислень). Picnic має модульну схему проектування. Криптографічні примітиви – геш-функції та блоковий шифр – можуть бути створені різними способами. Представлена конструкція використовує LowMC, блоковий шифр з низькою мультиплікативною складністю. LowMC не вивчався так багато, як AES, і, отже, потребує набагато більшого аналізу. Ефект використання AES замість LowMC в Picnic полягає в розширенні довжини підпису на коефіцієнт, який коливається від 6 до 9, в залежності від розміру блоку. Поліпшення обчислень призведе до зменшення підписів. Варто зазначити, що вимоги безпеки для базового блочного шифру менш суворі, ніж загальні вимоги безпеки блокового шифру, тому що тільки одна (випадковий відкритий текст, шифртекст) пара коли-небудь розкривається [2].

### 1.2. Порівняння постквантових алгоритмів

В табл. 1 порівнюються алгоритми, що пройшли до другого етапу міжнародного конкурсу NIST США. Алгоритм ЦП Picnic є унікальним серед представлених кандидатів, бо він єдиний заснований на стійкості геш-функції та блокових потокових шифрів сімейства LowMC [7].

З табл. 1 можна зробити висновок, що алгоритм Picnic досить збалансований за усіма критеріями. Але він має набагато більший розмір підпису порівняно з іншими алгоритмами, що у свою чергу безперечно впливає на швидкодію.

Таблиця 1

Порівняння постквантових алгоритмів

| Схема підпису | Безпека | Захищеність від часових атак | Цикли підписання | Цикли перевірки підпису | Розмір відкритого ключа, байти | Розмір підпису, байти |
|---|---|---|---|---|---|---|
| Dilithium | 125 | Так | 789 | 209 | 1472 | 2701 |
| FALCON (NTRU-GVP) | >>128 | Ні | - | - | 1792 | 1200 |
| qTESLA | 98 | Так | 143402 | 19284 | 12582912 | 2444 |
| GeMSS (HmFEv) | 128 | Так | 1497 | 15 | 83100 | 61 |
| LUOV | 128 | Так | 659000 | 290000 | 34100 | 421 |
| MQDSS | 128 | Так | 8510 | 5752 | 72 | 40952 |
| Rainbow | 128 | Так | 68 | 22 | 145500 | 48 |
| Picnic | 128 | Так | 1034 | 194 | 64 | 195458 |
| SPHINCS+ | 128 | Так | 51636 | 1451 | 1056 | 41000 |

### 1.3. Параметри Picnic

У цьому розділі коротко описано кожен з наборів параметрів для Picnic.

В табл. 2 наведені параметри для трьох рівнів безпеки L1, L3 і L5, відповідних безпеці AES-128, AES-192 і AES-256. Для кожного з трьох рівнів безпеки існує два алгоритми підпису, які використовують систему перевірки ZKB, засновану на перетворенні Fiat-Shamir (FS): pincic-L1-FS, picnic-L3-FS і picnic-L5-FS та засновану на перетворенні Unruh (UR): picnic-L1-UR, picnic-L3-UR і picnic-L5-UR. Існує також три набори налаштувань, що використовують перетворення FS і систему підтвердження з алгоритму покращення неінтерактивних нульових знань за допомогою додатків для постквантових підписів [2].

Всі параметри обрані так, що очікується, що вони забезпечать S біт безпеки від класичних атак, і як мінімум S/2 біт безпеки від квантових атак [2].

Параметр u, кількість оспорюваних повторень застосовано лише до наборів параметрів picnic2. Параметр N, кількість сторін в імітації MPC завжди 64 для наборів параметрів picnic2 [2, 3].

Для варіантів FS довжина підпису змінюється в залежності від виклику, тому вказується максимально можливий розмір разом із середнім розміром і стандартним відхиленням, що обчислюється за 100 підписами [3].

Таблиця 2

Параметри рівня безпеки

| Набір параметрів | S | n | s | r | Hash/KDF | $L_h$ | T | u |
|---|---|---|---|---|---|---|---|---|
| picnic-L1-FS | 128 | 128 | 10 | 20 | SHAKE128 | 256 | 219 | - |
| picnic-L1-UR | | | | | | | 219 | - |
| picnic2-L1-FS | | | | | | | 343 | 27 |
| picnic-L3-FS | 192 | 192 | 10 | 30 | SHAKE256 | 384 | 329 | - |
| picnic-L3-UR | | | | | | | 329 | - |
| picnic2-L3-FS | | | | | | | 570 | 39 |
| picnic-L5-FS | 256 | 256 | 10 | 38 | SHAKE256 | 512 | 438 | - |
| picnic-L5-UR | | | | | | | 438 | - |
| picnic2-L5-FS | | | | | | | 803 | 50 |

Таблиця 3

Розмір ключа і підпису (в байтах) за рівнем безпеки

| Набір параметрів | Відкритий ключ | Секретний ключ | Підпис (max) | Підпис (avg., std. dev.) |
|---|---|---|---|---|
| picnic-L1-FS | 32 | 16 | 34016 | 32838, 107 |
| picnic-L1-UR | | | 53945 | - |
| picnic2-L1-FS | | | 13786 | 12359, 213 |
| picnic-L3-FS | 48 | 24 | 76764 | 74134, 198 |
| picnic-L3-UR | | | 121837 | - |
| picnic2-L3-FS | | | 29742 | 27173, 315 |
| picnic-L5-FS | 56 | 32 | 132856 | 128176, 315 |
| picnic-L5-UR | | | 209506 | - |
| picnic2-L5-FS | | | 54732 | 46282, 613 |

## 2. Аналіз відомих атак на алгоритм

У цьому розділі аналізується схема підпису Picnic стосовно відомих атак. По-перше, надається спостереження, що коли ми маємо справу з ідеальними примітивами, Corollary 5.4 вже дає нам перевірену прив'язку до безпеки EUF-CMA (Existentially Unforgeable under Chosen Message Attacks – екзистенційна непідроблюваність при атаках на основі (адаптивно) вибраних повідомлень). Оскільки це примітиви, що створені за допомогою конкретних будівельних блоків, ми розглядаємо конкретні атаки на ці будівельні блоки. У цій схемі використовується класичний підхід, щоб перетворити Σ-протоколи в схеми підпису з випадковою моделлю оракула. На основі того факту, що з моменту введення випадкової моделі оракула не було виявлено жодної атаки, яка виникає з припущення, що геш-функція веде себе як випадкова модель оракула (за виключенням деяких штучних контрприкладів), тому можна стверджувати, що найкращими атаками на підпис є атаки, які роблять недійсними твердження, зроблені для базових симетричних примітивів [2].

Всі криптографічні примітиви, за винятком односторонньої функції LowMC, спираються на SHA-3 функцію SHAKE, визнаний і стандартизований примітив, і вона використовується стандартним способом. Що стосується цих примітивів, то вже досягнуто значної згоди в питаннях безпеки завдяки широкому криптоаналізу всередині спільноти. Тому ці будівельні блоки не розглядаються як центральна поверхня атаки. Покращення атак на ці примітиви також призводить до покращення атак на схему підпису [2].

### 2.1. Використання та безпека LowMC

Отже, надалі робиться акцент на атаки на односторонню функцію f. По суті функція f може бути будь-якою односторонньою функцією, але в Picnic використовується сімейство блочних шифрів LowMC, так як саме ці блочні шифри дали в результаті найбільш ефективні підписи. Зокрема, потрібно виходити з того, що використання LowMC, як вказано нижче, дає відповідне сімейство односпрямованих функцій $\{f_u\}_{u \in K_к}$. Ця функція використовується для встановлення відповідних відносин між секретними та відкритими ключами. Зокрема, нехай

$$f_u(x) := E(x, u) \quad (1)$$

і нехай E позначає шифрування LowMC щодо одного блоку u на ключі X. Ключі у використаній схемі підпису генеруються наступним чином. По-перше, вибирають ключ шифрування LowMC – x, а також один блок u рівномірно випадковим чином. Потім відкритий ключ підпису pk, а також секретний ключ підпису sk визначаються наступним чином:

$$pk := (y, u) = (f_u(x), u), \; sk := (pk, x). \quad (2)$$

Вибір кількості раундів у LowMC поставляється з показовим запасом безпеки. Для L1 рівня безпеки з 20 раундами найбільш відома атака на 12 раундів. Для L3 рівня безпеки з 30 раундами найбільш відома атака на 19 раундів. Для L5 рівня безпеки з 38 раундів, найбільш

відома атака – 26 раундів. І навіть ці атаки вимагають від зловмисника знання двох пар відкритого зашифрованого тексту для однієї й тієї самої пари ключів, в той час як у використаній схемі підпису зловмисник бачить лише одну пару вводу-виводу для кожного ключа [2].

### 2.2. Атаки в режимі розрахованому на одного користувача

В режимі, розрахованому на одного користувача, зловмисник завжди бачить тільки одну пару ключів схеми підпису Picnic, тобто одну пару відкритий текст-шифртекст ($f_u(x)$, u) LowMC відносно рівномірно випадкового ключа x і рівномірно випадкового блоку u. Отже, в цьому налаштуванні криптоаналітичні результати для LowMC також безпосередньо застосовані до схеми, що використовується. Слід зазначити, що можна навіть глобально використовувати x для подальшого скорочення розміру ключа pk загальнодоступної версії [2].

### 2.3. Атаки в режимі розрахованому на багато користувачів

Багатокористувацький режим точніше моделює реальність, тому, що є кілька користувачів, кожен з відкритим ключем, і зловмисник вважається успішним, якщо він може атакувати будь-якого з користувачів.

Багатокористувацький EUF-CMA. Приділяється особлива увага сценаріям атак, які стають можливими при переході до налаштування розрахованого на багато користувачів. Тут зловмисник може побачити багато пар ключів підписання, і варто бути обережним у відношенні більш складних атак, які можуть мати місце. Зокрема – на відміну від атак в режимі, розрахованому на одного користувача, – рішення вибрати незалежний і рівномірно випадковий блок u, будучи криптографічною функцією E (·, u) LowMC, на одну пару підписуючих ключів, виявляється важливим. В цих атаках один з n блочних ключів шифрування може бути відновлений за менший час, ніж час відновлення єдиного ключа і атаки стають дуже ефективними для великого n. Інтуїтивно, випадковий блок вибирає унікальну функцію за користувача і роботу для атаки на одного користувача (функцію) не можна використовувати для одночасної атаки на іншого користувача (функцію). Крім того, Банегас і Бернштейн нещодавно показали, що паралельні атаки пошуку співпадінь також можуть бути застосовані для квантових значень параметрів, які також підтримують випадковий вибір u на користувача. Можна вибрати менше значення, яке буде унікальним для кожного користувача (з потенційним зниженням безпеки), щоб зменшити розмір відкритого ключа. Однак, оскільки відкриті ключі в схемах вже малі (максимум 64 байта), в конструкції використовується повний випадковий блок, щоб бути максимально консервативним [2].

### 2.4. Атаки із заміною ключа

Це атаки, де є зловмисник, який має підпис $\sigma_A$ повідомлення M з відкритим ключем $pk_A$, що дозволяє створити відкритий ключ $pk_E$ (з $pk_A$) таким чином, що підпис $\sigma_A$ підтвердиться з ключем $pk_E$ та повідомленням M. Menezes і Smart надають формальну модель для захисту від таких атак, які не покриваються моделлю безпеки EUF-CMA. Безпека від цих типів атак може бути досягнута в цілому. Це було показано в їх роботі, і нижче подана їх теорема.

Теорема 1: Нехай (Gen, Sign, Verify) є EUF-CMA безпечною схемою підпису. Потім, (Gen, Sign', Verify) з Sign': = Sign (sk, pkm) та pk є однозначним кодуванням відкритого ключа є безпечною схемою підпису в налаштуванні, розрахованому на багато користувачів [3].

Вищевикладене призводить, зокрема, до утримання захищеності схем підпису sEUF-CMA. Відкритий ключ додається до повідомлення про підписання, а опис забезпечує однозначне кодування (оскільки відкритий ключ є парою бітових рядків, кодування є тривіальним). Отже, ми маємо наступний наслідок.

Наслідок 1: picnic-FS і picnic-UR забезпечують безпеку в режимі, розрахованому на багато користувачів [5].

### 2.5. Багатоцільові атаки

Дінур і Надлер описують атаки на Picnic версії 1.0 (набори параметрів picnic-FS і picnic-UR). В той час набори параметрів Picnic2 не були визначені, але атака в рівній мірі відноситься до прямого створення примітивів протоколу KKW. Їхні атаки являють собою багатоцільові атаки, де у атакуючого є список значень виду $y_i$: = H ($x_1$),..., $y_S$: = H ($x_S$), а відновлення будь-якого з $x_i$ призводить до успішної атаки. Спеціально значення $x_i$ мають довжину k біт, і зловмисник може відновити ключ k біт. Значення $y_i$ можна взяти з [3]:

- одного підпису (у підписі є близько $2^7$ значень),
- з декількох підписів при умові одного й того самого підписувача,
- з декількох підписів при умові різних підписувачів.

В атаці Дінура і Надлера значення $x_i$ є параметром, що використовується для кожної сторони, в кожному примірнику MPC [3]. Функція H розширює $x_i$ до випадкової довжини, що використовується під час моделювання протоколу MPC. В Picnic нащадки двох із трьох сторін розкриваються перевіряючому, а в Picnic2 розкриваються n–1 з n. Якщо зловмисник дізнається відсутнього нащадка, він може відновити секретний та загальний ключ підпису. Вихідна довжина залежить від набору параметрів, але завжди перевищує 600 біт. Що робить атаку неочевидною, так це те, що вихід H (випадкової довжини) не розкривається безпосередньо. Вони показують, що це – власність протоколів MPC, яка не робить звичайне поняття безпеки MPC (протокол MPC повинен гарантувати секретність вихідних даних, але припустима деяка хаотичність, якщо через це не виникають проблеми з секретністю даних). Вони також визначають кількість випадкових бітів, які можуть бути відновлені з примірника MPC. У всіх випадках можна ефективно відновити понад k бітів, так що може бути виконано тестування значення кандидата $x_i$. Однак у бітів від кожної цілі є різні позиції у випадковій довжині, що ускладнює ефективність проведення атаки. При типовій багатоцільовий атаці атакуючий обчислює $x_0$, обчислює $y_0$ = H ($x_0$), потім зрівнює $y_0$ з $y_i$, ефективно використовує структури даних (наприклад, геш-таблиця або дерево пошуку). Але тут порівняння довжини кандидата $y_0$ (з усіма відомими бітами) з цільовою довжиною $y_i$ (де для кожної з них відома підмножина бітів), не є очевидним. Дінур і Надлер показують, що його можна зробити таким і точно кількісно оцінити витрати при різних налаштуваннях. У кращому випадку, коли всі підписи створюються одним підписувачем, їх атаки коштують 2k–7/S (інформація теоретично оптимальна). В інших випадках атака коштує дорожче, але все одно нижче очікуваного рівня безпеки [3].

Пом’якшення наслідків. Версія специфікації 2.0 вносить зміни, щоб пом’якшити ці атаки (у всіх наборах параметрів). Зміна додає додаткову інформацію до входу H (названою *salt*), так, щоб нащадок кандидата $x_0$, не міг бути перевірений, витримавши порівняння $y_0$ до всього $y_i$, оскільки кожний $y_0$ вираховується з використанням різної *salt*. *Salt* гарантує, що $y_0$ потрібно буде перерахувати з правильною *salt* перед кожним порівнянням. Для вирішення всіх трьох варіантів атаки *salt* повинна бути унікальною для кожного підпису, для кожного підписувача і для кожного виклику H. Першою зміною є вибір випадкової завади для кожного підпису довжиною 256 біт. Це гарантує, що (з високою ймовірністю) *salt* унікальна. Потім, щоб гарантувати, що *salt* є унікальною у підписах, також включаємо пару лічильників, перше значення відповідає номеру примірника MPC, а друге відповідає номеру виклику H. Спеціалізація вже використовує метод поділу домену, де різні геш-функції створюються для різних цілей наступним чином – $H_i(x)=H(i||x)$. Цей механізм також допомагає гарантувати, що *salt* є унікальною, наприклад при обчисленні дерева нащадків, ми можемо використовувати $H_i$ та при обчисленні дерева Меркле використовувати $H_j$, і не турбуватися про те, що (salt, counters) пари повторюються в обох деревах. Деякі додаткові дані також повинні бути гешовані, але витрати на ЦП в еталонних показниках суттєво не зросли. Ймовірно, це пов’язано з тим, що геш-входи були короткими для початку, і залишаються короткими (меншими, ніж розмір блока геш-функції), навіть з *salt* [3].

**Висновки**

1. У ході семінару другого етапу NIST США рекомендував до подальших досліджень 9 криптографічних примітивів типу ЦП. Порівняння постквантових алгоритмів можна дослідити за табл. 1.

2. За результатами досліджень перспективних постквантових криптографічних алгоритмів типу цифровий підпис протягом 2-го раунду конкурсу NIST США було отримано наступні результати – обрані алгоритми-фіналісти та альтернативні алгоритми. У якості алгоритмів-фіналістів обрані такі алгоритми ЦП як CRYSTALS-DILITHIUM, FALCON та Rainbow. У якості альтернативних алгоритмів – GeMSS, Picnic та SPHINCS+.

3. На конкурс NIST США щодо стандарту ЕП було подано механізм ЕП, що ґрунтується на стійкості геш-функції та блокових потокових шифрів (Picnic). Наразі він досліджується на третьому етапі конкурсу NIST США у якості альтернативного алгоритму.

4. Побудова ЕП в алгоритмі Picnic заснована на перетворенні Fiat-Shamir (FS) та на перетворенні Unruh (UR).

5. Оцінка параметрів алгоритму говорить про те, що сам алгоритм досить збалансований за усіма критеріями, але має великий розмір підпису. Спроби зменшення розміру підпису призводять до зменшення безпеки алгоритму.

6. Після аналізу відомих атак на алгоритм можна зробити висновок, що алгоритм є безпечним і його можна рекомендувати для використання.

7. Великою перевагою алгоритму є великий простір для покращення та змін, які постійно проводяться, та коли буде представлена кінцева версія алгоритму, його можна затверджувати для застосування.

**Список літератури:**

1. Daniel Kales Efficient FPGA Implementations of LowMC and Picnic / Daniel Kales, Sebastian Ramacher, Christian Rechberger, Roman Walch, Mario Werner. Режим доступу: https://eprint.iacr.org/2019/1368.pdf.

2. The Picnic Signature Scheme. Design Document / Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Greg Zaverucha //. November 29, 2017. Version 1.0. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-1-submissions.

3. The Picnic Signature Scheme Design Document / Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, Greg Zaverucha. March 30, 2019. Version 2.0. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions.

4. NIST submissions. Picnic. Picnic-FS. Picnic-UR. Non-interactive Proof of Knowledge. Електронний ресурс. Режим доступу: https://pqc-wiki.fau.edu/w/Special:DatabaseHome.

5. Itai Dinur The Picnic Post-Quantum Signature Scheme and its Security Analysis. Режим доступу: https://www.cs.technion.ac.il/~biham/Workshops/Cryptoday/2018/Slides/cryptoday-2018-itai-dinur-picnic.pdf.

6. Горбенко І. Д. Методи, методика та результати порівняльного аналізу кандидатів на постквантовий стандарт електронного підпису / І. Д. Горбенко, О. Г. Качко, М. В. Єсіна, В. А. Пономар // XX Ювілейна Міжнар. наук.-практ. конф. "Безпека інформації в інформаційно-телекомунікаційних системах", 22-24 травня, 2018, м. Буча. С. 96-97.

7. Gorbenko I. Electronic signature mechanisms. The Current State, the Existing Contradictions and Prospects of Practical Use for the Post-Quantum Period / I. Gorbenko, A. Kuznetsov, Yu. Gorbenko, S. Kavun, O. Kachko, M. Yesina // ASC Academic Publishing Minden, Nevada, USA, 2017. 165 p.

*Надійшла до редколегії 29.10.2020*

*Відомості про авторів:*

**Єсіна Марина Віталіївна** – канд. техн. наук, старший викладач кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, Україна; e-mail: rinayes20@gmail.com, ORCID: https://orcid.org/0000-0002-1252-7606

**Шахов Богдан Сергійович** – студент кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, Україна; e-mail: bogdanshahov2000@gmail.com