УДК 004.056.5 **DOI:10.30837/rt.2024.2.217.06**

Ю.І. ГОРБЕНКО, канд. техн. наук, Є.В. ОСТРЯНСЬКА

ОЦІНКА ТА ПОРІВНЯННЯ КРИПТОПЕРЕТВОРЕНЬ ТИПУ ЕП НА ОСНОВІ КРИПТОГРАФІЇ НА РЕШІТКАХ КОНКУРСУ NIST США «DIGITAL SIGNATURE SCHEMES»

Вступ

За останнє десятиліття постквантова криптографія досягла переломного моменту; інституційні органи та зацікавлені сторони ініціювали стандартизацію та розгортання, і різноманітні проєкти досягли достатньо високого рівня прогресу та, навіть розгортання та впровадження.

Це підтверджується нещодавньою стандартизацією NIST у 2020 р. геш-підписів XMSS і LMS, а також з цієї причини Національний інститут стандартів і технологій (NIST) проводить конкурс та пропонує перехід до квантово-стійкої криптографії. Протягом трьох раундів було запропоновано багато алгоритмів для шифрування з відкритим ключем, механізмів інкапсуляції ключів і електронного підпису. Для схем електронного підпису було три критерії оцінки: 1) безпека (властивість нульового знання, надійність безпеки в ROM/QROM, спрощення атак бічними каналами, складність основної проблеми), 2) складність і продуктивність та 3) алгоритм і характеристики реалізації на програмнотехнічних засобах.

У липні 2022 р., наприкінці 3-го раунду, щодо постквантових цифрових підписів було запропоновано три кандидати на стандартизацію NIST: один підпис на основі MLWE (Crystals-Dilithium), один підпис на основі NTRU (Falcon) і один підпис на основі гешу (Sphincs+). Хоча профілі ефективності та безпека «чорної скриньки» цих схем добре зрозумілі, стійкість до атак із бічних каналів залишається слабким місцем для всіх них.

Під час атаки бічними каналами криптоаналітик може дізнатися інформацію про фізичне виконання алгоритму, наприклад час його роботи або його вплив на енергоспоживання, а також електромагнітне або акустичне випромінювання пристрою, на якому він запущений. Потім ці допоміжні знання можна використати для відновлення конфіденційної інформації, наприклад криптографічних ключів. Було запропоновано кілька бічних атак проти схем, які NIST розглядає для стандартизації, таких як Dilithium, Falcon або SPHINCS і XMSS. Наведений вище список аж ніяк не є вичерпним, і загалом криптографічні алгоритми вимагають реалізації контрзаходів, щоб досягти будь-якої суттєвої безпеки в контексті атак бічними каналами.

NIST оголосив, що процес стандартизації PQC продовжується четвертим раундом, при цьому наступні KEM все ще знаходяться на розгляді: BIKE, Classic McEliece, HQC і SIKE. Однак на розгляді не залишилося жодного кандидата на цифровий підпис. Таким чином, NIST опублікував заклик до додаткових пропозицій щодо цифрового підпису, які слід розглянути в процесі стандартизації PQC. Прийом документів завершився 1 червня 2023 р.

17 липня 2023 р. NIST оголосив про додаткових кандидатів на цифровий підпис для процесу стандартизації PQC [1].

NIST насамперед зацікавлений у додаткових схемах підписів загального призначення, які не базуються на структурованих решітках. Для певних застосувань, таких як прозорість сертифікатів, NIST також може бути зацікавлений у схемах підписів, які мають короткі підписи та швидку перевірку. NIST відкритий для отримання додаткових матеріалів на основі структурованих решіток, але має намір урізноманітнити стандарти постквантових підписів. Таким чином, будь-яка пропозиція підпису на основі структурованої решітки повинна буде значно перевершувати CRYSTALS-Dilithium [2] і FALCON [3] у відповідних додатках і/або забезпечувати значні додаткові властивості безпеки, які будуть розглянуті для стандартизації.

Метою статті є аналіз, оцінка та порівняння алгоритмів ЕП, в основі яких лежить криптографія на решітках, додаткового конкурсу NIST США. Зокрема, розглянуто алгоритми EagleSign [4], дві версії алгоритму EHTv3 та EHTv4 [5], HAETAE [6], HAWK [7], HuFu [8] та Raccoon [9].

1. Попередні визначення та критерії порівняння

На даний момент NIST обрав 40 алгоритмів-кандидатів ЕП [1]. Серед них є (табл. 1): 6 алгоритмів ЕП на основі кодів, один алгоритм ЕП на основі ізогеній, 7 алгоритмів ЕП, в основі яких лежать операції на решітках, 7 кандидатів на роль алгоритму ЕП на основі методу MPC-in-the-Head та 10 алгоритмів, в основі яких лежать багатоваріативні перетворення, на основі симетричних криптоперетворень було обрано 4 схеми ЕП, та ще 5 кандидатів, що базуються на інших видах криптографічних перетворень.

Таблиця 1

Кандидати на роль криптоперетворення типу ЕП

На основі кодів	На основі ізогеній	На решітках	На основі MPC-in-the-Head	На основі багатоваріативних криптоперетворень	На основі симетричних криптоперетворень	Інше
CROSS	SQIsign	EagleSign	Biscuit	3WISE	AIMer	ALTEQ
Enhanced pqsigRM		EHTv3 та EHTv4	MIRA	DME-Sign	Ascon-Sign	eMLE-Sig 2.0
FuLeeca		HAETAE	MiRitH	HPPC	FAEST	KAZ-SIGN
LESS		HAWK	MQOM	MAYO	SPHINCS-alpha	Preon
MEDS		HuFu	PERK	PROV		Xifrat1-Sign.I
Wave		Raccoon	RYDE	QR-UOV		
		SQUIRRELS	SDitH	SNOVA		
				TUOV		
				UOV		
				VOX		

Фіналістами конкурсу PQC NIST обрав три електронні підписи, серед яких два підписи на решітках. А саме:

- Falcon [3] – це стандарт цифрових підписів PQC (Post Quantum Cryptography), затверджений NIST. Він походить від NTRU і є методами на основі решітки для квантового надійного цифрового підпису. Falcon базується на методі Гентрі, Пейкерта та Вайкунтанатана для створення схем підпису на основі решітки, з використанням швидкої вибірки Фур'є, що дозволяє дуже швидко реалізувати, тисячі підписів за секунду на звичайному комп'ютері; а верифікація відбувається в п'ять-десять разів швидше.

Вибираємо три параметри: N, p і q. Щоб обчислити пару ключів, ми вибираємо два поліноми: f і g. Після цього можемо обчислити $F = f_q = f^{-1}\left(\mathrm{mod}\, q\right)$, де f і f_q – це особисті ключі. Відкритий ключ визначається як $h = p \cdot f_q \cdot f\left(\mathrm{mod}\, q\right)$.

Алгоритм використовує справжню вибірку Гауса, що гарантує незначний витік інформації про секретний ключ до практично нескінченної кількості підписів (більше 2^{64}). Завдяки використанню решіток NTRU підписи значно коротші, ніж у будь-якій схемі підпису на основі решітки з тими самими гарантіями безпеки, тоді як відкриті ключі приблизно однакового розміру. Операції мають вартість $O\left(n \log n\right)$ для ступеня n, що дозволяє використовувати дуже довгострокові параметри безпеки за помірних витрат. Falcon сумісний із невеликими вбудованими пристроями з обмеженим обсягом пам'яті.

- Dilithium [2]. На зараз CRYSTALS (Cryptographic Suite for Algebraic Lattices) підтримує два квантово надійні механізми: Kyber для механізму інкапсуляції ключів (KEM) і обміну ключами; і Dilithium для алгоритму цифрового підпису. CRYSTALS Dilithium використовує схеми Фіата–Шаміра на основі решітки та створює один з найменших підписів серед усіх

постквантових методів із відносно малими розмірами відкритого та закритого ключів. Три основні реалізації параметрів: Dilithium 2, Dilithium 3 і Dilithium 5. Загалом, Dilithium 3 еквівалентний 128-бітному підпису і, можливо, є відправною точкою для реалізації.

2. Алгоритм ЕП EagleSign

Як було сказано вище, NIST стандартизував два методи Dilithium і Falcon для цифрового підпису постквантової криптографії. Dilithium – це підпис на основі MLWE (Module-Learning With Errors), тоді як Falcon використовує підпис на основі NTRU. Тепер є нові методи решітки, які розвиваються як частина раунду додаткового підпису. Одним із них є EagleSign, визначений в [4], який використовує варіацію методу підпису Ель-Гамаля, без переривання, але використовує структуровані решітки.

Більшість відомих методів зламу RLWE та NTRU не можна тривіально узагальнити до відкритого ключа схеми EagleSign. Автори [4] стверджують, що використання разом MNTRU та MLWE в одному відкритому ключі дозволить зробити більш складними алгебраїчні та геометричні властивості основної решітки, і, таким чином, трохи віддалитися від сильних структурованих решіток. У підписі властивість нульового знання гарантує, що процес підписання не розкриває жодної інформації про секретний ключ, пов'язаний із відкритим ключем, який використовується в процесі верифікації.

Підпис є безпечним у ROM [4]. Безпека в ROM випливає із загальної структури, використовуючи лему про розгалуження. EagleSign забезпечує більшу гнучкість для легкого оновлення рівня безпеки в майбутньому

Слід зазначити, що EagleSign швидший і більш простий, ніж Falcon і Dilithium, запропоновані NIST для стандартизації. Для рекомендованих параметрів розміри EagleSign менші, ніж розміри Dilithium, але розміри EaglaSign подібні до розмірів Dilithium для рівнів 2 і 5. Наразі для рівня безпеки 1 алгоритм ще не є реалізованим, як зазначають автори в [4], саме тому в розд. 9 в табл. 5 нижче швидкодія алгоритму наведена лише для 3 та 5-го рівнів безпеки.

Авторами підпису EagleSign [4] доведено безпеку у моделі випадкового оракула, проте не в моделі квантового випадкового оракула. Безпека в моделі випадкового оракула випливає із загальної структури підпису, використовуючи лему про розгалуження. EagleSign забезпечує більшу гнучкість для легкого оновлення рівня безпеки в майбутньому.

3. Алгоритми ЕП EHTv3 та EHTv4

EHT визначає метод цифрових підписів на основі постквантової криптографії. Перша версія алгоритму електронного підпису EHTv1, представлена в [11]. Друга версія EHTv2 з'явилася в матеріалах NISK 2022 [12]. Актуальна версія EHTv3 в основному відрізняється вибором матриці C. Крім того, в [10] представлено EHTv4, який дуже схожий на EHTv3, але використовує арифметику в кільці скінченної групи Z_q замість Z_q. Це забезпечує в цілому більш ефективний алгоритм для порівнянних рівнів безпеки за рахунок більшого розміру підпису. Та якщо порівнювати обидві версії алгоритму EHTv3 та EHTv4 [5] з криптосистемою Dilithium, то в вони мають набагато менші підписи в порівнянні з Dilithium.

Схеми [5] прозорі та прості для розуміння та реалізації. Відкритий ключ генерується за допомогою фактично однієї інверсії матриці та двох множень матриця-матриця. Підпис генерується за допомогою по суті трьох множень матриці-вектору. Верифікація виконується одним множенням матриці-вектору. Оскільки множення матриці легко розпаралелити, реалізація розпаралелювана.

Особистий ключ може бути згенерований із 48-байтового початкового числа, щоб задовольнити рівень безпеки NIST, можна навіть взяти коротше початкове число. У EHTv3 з міркувань ефективності також потрібно зберігати характеристичний поліном (не обов'язково секретний) підматриці C_1 з C. У EHTv4 замість цього можна зберегти інверсію кількох кільцевих елементів групи.

Обидві схеми дозволяють гнучко обирати параметри для підвищення рівня безпеки, якщо це необхідно.

EHTv3 може добре працювати на 8-розрядних платформах, оскільки його арифметика є модулем відносно невеликого додатного цілого числа . Оскільки секретний ключ EHTv3 може бути створений із вихідного коду, здається, що сучасні смарт-карти мають обчислювальні ресурси для реалізації алгоритму генерації підпису EHTv3. Подібне стосується підписів EHTv4. Цей напрямок потребує подальшого вивчення.

4. Алгоритм ЕП HAETAE

HAETAE [6] – це метод постквантової криптографії (PQC) на основі решітки, який базується на методах криптоалгоритму Dilithium, безпека якої базується на складності модульних версій задач LWE та SIS. Таким чином, він використовує підхід «Фіат–Шамір з переривання-ми» [14, 15], що ґрунтується на вибірці відхилення: вибірка відхилення використовується для перетворення пробного підпису, вибірка якого залежить від конфіденційної інформації, у підпис, вибірку якого можна відкрито моделювати. Підпис HAETAE частково схожий на Crystals-Dilithium [2], але відрізняється двома аспектами: бімодальним розподілом для вибірки відхилення (подібно до схеми підпису BLISS [13]) і вибіркою з рівномірного розподілу гіперкулі та відхилення до нього.

Таким чином розміри підписів на 30 – 40 % менші, ніж у Dilithium за порівнянних рівнів безпеки, а ключі перевірки на 20 – 25 % менші. Також неодмінною перевагою є доволі легкі реалізація та впровадження схеми, оскільки весь процес підписання можна реалізувати за допомогою арифметики з фіксованою комою, і значна частина підписання, яка не залежить від повідомлень, може виконуватися «офлайн» для рандомізованої версії схеми.

На відміну від таких алгоритмів як Falcon та Mitaka, що покладаються на цілочисельну вибірку Гауса із довільними центрами, HAETAE має набагато простіший процес генерації ключів, оскільки покладається лише на (з нульовим центром) неперервну вибірку Гауса, що використовується для рівномірної вибірки в гіперкулях. Виклики до нього також можна масово розпаралелювати. Ця відмінність робить HAETAE можливим мати алгоритм підпису з фіксованою комою та легші маскування.

Але необхідно зауважити, що хоча HAETAE простіший з точки зору впровадження, його ключ перевірки та розмір підпису більші, ніж у Falcon, що видно з табл. 2 – 4 відповідно. Та у порівняння з Dilithium алгоритм HAETAE є повільнішим, оскільки алгоритм генерації ключа перезапускається, якщо секретний ключ не задовольняє умову відхилення ключа.

5. Алгоритм ЕП HAWK

HAWK [7] – це метод підпису на основі решітки, який створює підписи за допомогою проблеми ізоморфізму решітки (LIP). Він є швидшим, ніж Dilithium для підписання та верифікації. Він також займає мало пам’яті та підтримується різними апаратними засобами. Наразі немає функції маскування, яка може бути підозрілою для аналізу криптоаналітичних атак бічними каналами. Є деякі занепокоєння щодо доказів безпеки, пов’язаних з HAWK. HAWK схожий на метод FALCON, але використовує іншу складну задачу. Це зводить проблему решітки до задачі найкоротшого вектору, де підпис визначає здатність розв’язувати загальну проблему найближчого вектору. Загалом, однак, він використовує більшу частину коду FALCON для генерації ключів для вирішення рівняння NTRU.

Для генерації ключів у HAWK потрібні вибірки з центрованого біноміального розподілу, який легко отримати з джерела однорідних бітів. Підписування вимагає дискретної гаусової вибірки з фіксованою шириною з двох сумісних класів цілочисельної решітки, що легко досягається за допомогою двох фіксованих попередньо обчислених таблиць достатньої точності.

Схема ЕП HAWK є SUF-CMA безпечною у моделі випадкового оракула за умови складності проблеми omSVP (варіація проблеми пошуку найменшого вектора). Задача відновлення секретного ключа безпосередньо з відкритого ключа є прикладом проблеми ізоморфізму

модульних решіток (smLIP). HAWK має невелике використання пам'яті. Наприклад, HAWK-512 вимагає не більше 14 кілобайтів оперативної пам'яті для будь-якого алгоритму, включаючи більш швидкі варіанти підпису та верифікації. Якщо ключі можуть бути згенеровані ззовні та складно закодовані на пристрої, тоді HAWK-512 і HAWK-1024 можуть підписувати та перевіряти, використовуючи лише 6 КБ і 11 КБ оперативної пам'яті відповідно. Це робить HAWK гарним кандидатом для багатьох вбудованих платформ на основі ядер ARM Cortex-M0(+): продукти в цьому діапазоні включають, наприклад, серію LPC800 від NXP, STM32F0 від ST або XMC1000 від Infineon (16 КіБ SRAM). Окрім цього, будь-яка функція в HAWK, яка залежить від секретних даних, має час роботи, незалежний від цих даних. HAWK добре підходить для різного обладнання, оскільки він не залежить від арифметики з плаваючою комою. Обчислення з плаваючою комою (подвійна точність) не потрібні. Це дозволяє запускати HAWK на багатьох (обмежених) пристроях, не обладнаних таким FPU.

Ефективного маскування для HAWK (поки що) немає. Незважаючи на простоту вимоги лише двох фіксованих дискретних розподілів Гауса для вибірки під час підписання, це відкрита дослідницька проблема для створення ефективного методу маскування на основі таблиці, який використовується. Позитивним моментом є те, що крім цього компонента відомо, як ефективно маскувати решту конструкції HAWK.

6. Алгоритм ЕП HuFu

Існує два основних підходи до підписів на основі решітки: Fiat-Shamir і Hash-and-sign. Загалом, Dilithium використовує підхід Fiat-Shamir, тоді як Falcon використовує підхід Hash-and-sign, використовуючи структуру на решітках GPV. Новим запропонованим стандартом є HuFu.

HuFu [8] – це схема електронного підпису, безпека якої базується на складності стандартних найгірших проблем на загальних решітках. Крім того, що HuFu не використовує структурованих решіток, він має досить інакший дизайн порівняно з Crystals-Dilithium [2] і Falcon [38]. На високому рівні HuFu – це схема підпису типу «геш-і-підпис», запропонована Гентрі, Пейкертом і Вайкунтатаном [18]. Його екземпляр створюється на складних випадкових решітках відповідно до конструкції входу гаджета [19] і використовуючи техніку компактного гаджета [20] для досягнення загальної хорошої продуктивності. У двох словах складові HuFu можна описати так:

HuFu = фреймворк GPV + складні випадкові решітки + компактний гаджет.

Як і у випадку з Falcon, він використовує метод геш-підпису з GPV. Falcon використовує решітку NTRU, тоді як HuFu використовує гаджетний підхід для представлення решітки. За допомогою гаджетного підходу вхід створюється за допомогою лінійного відношення між загальнодоступною решіткою та решіткою гаджета (яка не є повною основою решітки). На жаль, гаджетний підхід призводить до значно більших відкритих і секретних ключів, але може бути основою інших криптографічних примітивів (наприклад, для шифрування на основі ідентифікації та сукупних підписів).

Безпеки алгоритму ЕП HuFu базується на проблемах SIS та LWE, які є принаймні такими ж складними, як стандартні найгірші задачі решітки на загальних решітках. Такі консервативні припущення безпеки уникають ризику алгебраїчних атак на проблеми ідеальної решітки та атаки на підрешітку проти NTRU.

HuFu створено в рамках моделі Micciancio-Peikert [19]. Як результат, HuFu має онлайн/офлайн структуру, а його онлайн-операції прості, швидкі та повністю над цілими числами. Ця функція буде дуже корисною для певних випадків використання. Крім того, структура гаджета забезпечує потужну універсальність, що веде до широкого спектру вдосконалених криптосистем, таких як шифрування на основі атрибутів, підписи груп, сліпі підписи, тощо. Завдяки цьому HuFu легше адаптувати для надання розширеної функціональності.

7. Алгоритм ЕП Raccoon

Raccoon [9] – це схема постквантового підпису на основі решітки, яка використовує метод Fiat Shamir без переривань (на відміну від методу Dilithium, який виконує перетворення Фіат–Шаміра із перериваннями). Цей метод дозволяє підтримувати розподілені порогові підписи [21], а також забезпечує покращену підтримку атак на бічних каналах. Raccoon був розроблений PQShield і був представлений на конкурс NIST PQC для отримання додаткових підписів.

Оскільки структура Raccoon дуже схожа на Dilithium, можна використовувати дуже схожі стратегії реалізації та оптимізації. Особливо, коли використовується 32-розрядна арифметична інструкція «CRT», код NTT для Raccoon по суті еквівалентний коду Dilithium як на мікроконтролерах, так і на SIMD високого класу.

Основний принцип дизайну Raccoon – піддатливість до маскування. По суті, Raccoon можна замаскувати в порядку $d-1$ за допомогою $O(d\log d)$. Це дозволяє маскувати Raccoon на високих рівнях з невеликим впливом на ефективність.

При високих порядках маскування споживання пам'яті стає новим вузьким місцем ефективності через необхідність зберігати поліноми, замасковані у високих порядках. Автори [9] вирішують це за допомогою методів, які дозволяють значно зменшити вартість пам'яті маскованих значень.

Raccoon покладається на (варіанти) припущень на решітках, які добре зрозумілі. А саме Module-LWE та Module-SIS, подібно до (вибраного) основного стандарту Dilithium. Зауважте, що для евклідової норми алгоритм покладається на Module-SIS, на відміну від трохи менш звичайної норми нескінченності, яка використовується в Dilithium.

Ще однією з переваг є проста і портативна реалізація., що є двома основними ідеями дизайну Raccoon. Наприклад, розподіли помилок базуються на рівномірних розподілах по $\{0,\ldots,2^u-1\}$; це робить впровадження простим на широкому спектрі платформ. Подібним чином 49-бітний модуль можна розділити на два 24-бітний і 25-бітний модулі; це полегшує впровадження на 32-розрядних архітектурах.

На відміну від багатьох інших схем, Raccoon не потребує замаскованих реалізацій симетричних криптографічних компонентів, таких як SHA-3/SHAKE. Кількість окремих гаджетів маскування є відносно невеликою, що призводить до простішого та легшого для перевірки мікропрограмного та апаратного забезпечення.

Окрім масштабованості безпеки та теоретичної обґрунтованості, важливою перевагою маскувальних контрзаходів є те, що вони менш залежать від фізичних деталей реалізації порівняно з методами логічного рівня. Таким чином, реалізації – певною мірою – портативні.

Однак алгоритм ЕП Raccoon має більші розміри, ніж Falcon і Dilithium. Завдяки видаленню вибірки відхилень, розмір підпису Raccoon значно більший, ніж Dilithium, незважаючи на те, що він має подібну структуру та базується на подібних припущеннях. Розміри ключів перевірки подібні до розмірів Dilithium. Тобто, якщо порівнювати підпис з Dilithium, то хоча відкритий (перевіряючий) ключ має подібний розмір, особистий (підписуючий) ключ і розмір підпису приблизно в п'ять разів більші у Raccoon.

Це збільшення розміру пов'язане з тим, що розміри підписів Raccoon масштабуються логарифмічно залежно від кількості запитів. На даний момент набори параметрів і відповідні перевірки безпеки для NIST рівнів I, III і V охоплюють максимальну кількість запитів Q_s, що дорівнює 2^{53}, 2^{51} і 2^{55} відповідно.

Ще одним недоліком схеми ЕП Raccoon є те, що вона немає стійкості до атак помилками. Незважаючи на те, що дизайн Raccoon робить його більш стійким до атак із сторонніх каналів, атаки помилками також є серйозною загрозою в реальному конкурентному середовищі.

8. Порівняння алгоритмів ЕП на решітках

У цьому розділі у табл. 2, 3 та 4 наведено результати порівняння розмірів підписів, відкритих та особистих ключів відповідно, вже стандартизованих NIST алгоритмів ЕП на решітках Crystals Dilithium та Falcon з розглянутими вище алгоритмами в додаткового конкурсу NIST. А також обчислення швидкодії розглянутих у попередніх розділах алгоритмів ЕП у табл. 5. Порівняння було проведено для 1, 3 та 5-го рівнів безпеки відповідно. Усі значення в табл. 2 – 4 наведено в байтах, а швидкодія в табл. 5 у циклах процесору.

Таблиця 2

Порівняння розмірів підписів алгоритмів ЕП
для 1, 3 та 5 рівнів безпеки

Схема підпису ЕП	Розмір підпису (у байтах)		
	Рівень безпеки 1	Рівень безпеки 3	Рівень безпеки 5
Crystals Dilithium	2420	3293	4595
Falcon	690	–	1330
EagleSign	2144	2336	3488
EHTv3	169	255	344
EHTv4	369	–	875
HAETAE	1463	2337	2908
HAWK	555	–	1221
HuFu (байт)	2495	3580	4560
Raccoon	11524	14544	20330

Таблиця 3

Порівняння розмірів відкритих ключів алгоритмів ЕП
для 1, 3 та 5 рівнів безпеки

Схема підпису ЕП	Розмір відкритого ключа (у байтах)		
	Рівень безпеки 1	Рівень безпеки 3	Рівень безпеки 5
Crystals Dilithium	1312	1952	2592
Falcon	897	–	1793
EagleSign	1824	1824	3616
EHTv3	83,490	191,574	348,975
EHTv4	1107	–	2623
HAETAE	992	1472	2080
HAWK	1024	–	2440
HuFu (байти)	1083424	2228256	3657888
Raccoon (байтів)	2256	3160	4064

Таблиця 4

Порівняння розмірів секретних ключів алгоритмів ЕП
для 1, 3 та 5 рівнів безпеки

Схема підпису ЕП	Розмір секретного ключа (у байтах)		
	Рівень безпеки 1	Рівень безпеки 3	Рівень безпеки 5
Crystals Dilithium	2528	4000	4864
Falcon	1281	–	2305
EagleSign	573	573	1600
EHTv3	368	532	701
EHTv4	419	–	925
HAETAE	1408	2112	2752
HAWK	184	–	360
HuFu (байти)	11417440	23172960	37418720
Raccoon (байти)	14800	18840	26016

Порівняння швидкодії підписів. Усі запуски алгоритмів та середні оцінки часу було здійснено та розраховано на комп’ютері з 64-розрядною операційною системою Windows 10

на процесорі Intel(R) Core(TM) i7-10510U CPU @ на 2.30 GHz. Дані є усередними над близько 50 запусками кожного алгоритму.

Таблиця 5

Порівняння швидкодії генерації ключів, підпису
та верифікації алгоритмів ЕП для 1, 3 та 5 рівнів безпеки

Схема підпису ЕП	Рівень безпеки	Швидкодія (у циклах)		
		Генерація ключів	Підписання	Верифікація
Crystals Dilithium	1	300,751	1,081,174	327,362
	3	544,232	1,712,783	522,267
	5	819,475	2,383,399	871,609
Falcon	3	19,872,000	396,678	82,339
	5	63,135,000	961,208	205,128
EagleSign	3	1,020,723	1,283,454	955,956
	5	3,443,617	2,358,603	1,602,340
EHTv3	1	465,600,000	181,920,000	1,968,000
	3	1,432,800,000	494,400,000	4,272,000
	5	3,672,000,000	732,000,000	7,584,000
EHTv4	1	29,040,000	21,600,000	9,240,000
	5	276,000,000	142,320,000	62,880,000
HAETAE	1	3,823,188	20,578,698	1,527,304
	3	12,164,364	32,672,248	2,456,500
	5	22,121,374	58,188,178	3,686,662
HAWK	1	8,430,000	85,400	181,000
	5	43,700,000	148,000	303,000
HuFu	1	1,193,896,000	7,322,000	1,804,000
	3	8,916,915,000	18,413,000	6,105,000
	5	9,727,510,000	31,896,000	10,424,000
Raccoon	1	2,256,000	4,817,000	1,757,000
	3	3,252,000	6,860,000	2,764,000
	5	5,199,000	10,062,000	4,554,000

Як видно з таблиць, алгоритми EagleSign та HAWK є доволі швидкими в порівнянні з іншими кандидатами, а HAWK також виграє поміж інших кандидатів для підписання та верифікації. Він також займає мало пам'яті та підтримується різними апаратними засобами. Алгоритм EHTv3 має великі розміри відкритих ключів для всіх рівнів безпеки, але EHTv4 виправляє цей недолік і має розміри ключів та підписів, навіть, менші ніж у Falcon і Dilithium, але в той же час втрачають у швидкодії. Також ми бачимо, що HuFu має найбільші розміри ключів, до чого, на жаль, призводить гаджетний підхід. Але даний алгоритм може бути гарною основою інших криптографічних примітивів (наприклад, для шифрування на основі ідентифікації та сукупних підписів), що потребує подальшого дослідження.

Висновки

1. Розглянуто та проведено порівняння алгоритмів ЕП, в основі яких лежить криптографія на решітках, додаткового конкурсу NIST США [1]. Зокрема, в роботі розглянуто алгоритми EagleSign [4], дві версії алгоритму EHTv3 та EHTv4 [5], HAETAE [6], HAWK [7], HuFu [8] та Raccoon [9]. І в результаті було зроблено наступні висновки.

2. EagleSign простіший і швидший (у деяких випадках), ніж Falcon і Dilithium. Розміри подібні до розмірів Dilithium, але для рекомендованих параметрів розміри EagleSign менші, ніж у Dilithium. Але він має ті самі обмеження, що й будь-який електронний підпис на основі решіток, щодо довгострокової безпеки.

3. Безперечною перевагою EHTv3 і EHTv4 є короткі підписи, створені схемами. Відкритий ключ EHTv4 всього в кілька разів більший за сам підпис. Це явна перевага EHTv4. Але у той же час відкритий ключ EHTv3 досить великий, і це є помітним обмеженням у порівнянні з EHTv4 та деякими іншими схемами підпису на основі решітки.

4. Схема ЕП HAETAE відрізняється від Dilithium двома основними аспектами: використовується бімодальний розподіл для вибірки з відхиленням, як у схемі підпису BLISS, замість «унімодального» розподілу, такого як Dilithium, обираються та відхилюються рівномірні розподіли гіперкуль замість дискретних рівномірних розподілів гіперкуба. Розміри підписів HAETAE на 30 – 40 % менші, ніж у Dilithium за порівнянних рівнів безпеки, а ключі перевірки на 20 – 25 % менші. З точки зору реалізації, незважаючи на те, що обґрунтування її конструкції відрізняється від обґрунтування конструкції Dilithium, схема залишається зручною для впровадження.

5. Схема ЕП HAWK є SUF-CMA безпечною у моделі випадкового оракула за умови складності проблеми omSVP (варіація проблеми пошуку найменшого вектора). Задача відновлення секретного ключа безпосередньо з відкритого ключа є прикладом проблеми ізоморфізму модульних решіток (smLIP). HAWK має невелике використання пам'яті і добре підходить для різного обладнання, оскільки він не залежить від арифметики з плаваючою комою. Обчислення з плаваючою комою (подвійна точність) не потрібні. Це дозволяє запускати HAWK на багатьох (обмежених) пристроях, не обладнаних таким FPU.

6. Алгоритм HuFu має короткі підписи та швидку реалізація: характерний розмір HuFu відповідає розміру Crystals-Dilithium, тоді як HuFu не базується на структурованих решітках. Підписання та верифікація HuFu є ефективними. HuFu також добре розпаралелюється, що дає певний простір для оптимізації. Крім того, його онлайн/офлайн-структура дозволяє ще більше скоротити онлайн-час виконання та обчислювальний ресурс. Але цей алгоритм також має і ряд недоліків. Як видно з табл. 3, 4 HuFu має великі відкриті ключі: розмір відкритого ключа HuFu становить від 1 до 3,5 мегабайт для трьох різних рівнів безпеки, тому HuFu може бути не дуже бажаним для багатьох програм. Тим не менш, HuFu цілком придатний для випадків використання, коли ключі не передаються часто. У той час як онлайн фаза в процедурі підписання реалізована повністю над цілими числами, автономна фаза все ще часто використовує арифметику з плаваючою комою. Це може бути обмеженням, коли режим онлайн/офлайн вимкнено, особливо для реалізацій на пристроях обмежень. Однак це можна вирішити за допомогою техніки інтегрального розкладання за Грамом.

7. Схема підпису Raccoon заснована на перетворенні Фіата–Шаміра з підпису на основі решітки Шнорра з тонким аналізом, щоб запобігти використанню методу відхилення вибірки та вибірки Гауса, дозволяючи використовувати методи маскування для секретних ключів під час алгоритму підпису, як контрзахід проти атак сторонніми каналами. Крім того, безпека Raccoon ґрунтується на стандартних припущеннях щодо модульних решіток, які використовуються в (вибраному) стандартизованому Dilithium. Ці припущення добре вивчені протягом десяти років і показали свою надійність, пропонуючи справедливий розмір параметрів. Одним з основних недоліків схеми ЕП Raccoon є те, що вона немає стійкості до атак помилками. Незважаючи на те, що дизайн Raccoon робить його більш стійким до атак із сторонніх каналів, атаки помилками також є серйозною загрозою в реальному середовищі.

8. Підсумовуючи порівняльний аналіз в розділі 6 можна сказати, що алгоритми EagleSign та HAWK є доволі швидкими в порівнянні з іншими кандидатами, а HAWK також виграє поміж інших кандидатів для підписання та верифікації. Він також займає мало пам'яті та підтримується різними апаратними засобами, що робить його доволі гарним кандидатом. Алгоритм EHTv4 має розміри ключів та підписів, навіть, менші ніж у Falcon і Dilithium, але в той же час втрачають у швидкодії. Також слід зазначити, що HuFu має найбільші розміри ключів, до чого, на жаль, призводить гаджетний підхід. Але даний алгоритм може бути гарною основою інших криптографічних примітивів (наприклад, для шифрування на основі ідентифікації та сукупних підписів), що потребує подальшого дослідження.

9. Алгоритм ЕП Raccoon має більші розміри, ніж Falcon і Dilithium. Завдяки видаленню вибірки відхилень, розмір підпису Raccoon значно більший, ніж Dilithium, незважаючи на те, що він має подібну структуру та базується на подібних припущеннях. Розміри ключів перевірки подібні до розмірів Dilithium. Тобто, якщо порівнювати підпис з Dilithium, то хоча відкритий ключ має подібний розмір, особистий ключ і розмір підпису у Raccoon приблизно в п'ять разів більші. Це збільшення розміру пов'язане з тим, що розміри підписів Raccoon масштабуються логарифмічно залежно від кількості запитів.

Список літератури:

1. NIST standardization process “Post-Quantum Cryptography: Digital Signature Schemes”. Access mode: https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
2. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehlé. “CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme”. 2022.
3. Thomas Prest; Pierre-Alain Fouque; Jeffrey Hoffstein; Paul Kirchner. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. Specification v1.2 – 01/10/2020.
4. Hounkpevi A.C., Djimnaibeye S., Seck M. EagleSign: A new post-quantum ElGamal-like signature over lattices. Submission to the NIST's post-quantum cryptography standardization process. (2023).
5. Semaev I., Feussner M. Digital Signature Algorithms EHTV3 and EHTV4 submission to NIST PQC. Submission to the NIST’s post-quantum cryptography standardization process. 2023
6. Cheon J. H., Choe H., Devevey J., Güneysu T., Hong D., Krausz M., Yi M. Haetae: Shorter lattice-based fiat-shamir signatures // Cryptology ePrint Archive. 2023.
7. Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, Wessel van Woerden. HAWK. version 1.0 (June 1, 2023). [Electronic resource]. Access mode: https://hawk-sign.info.
8. Yang Yu, Huiwen Jia, Leibo Li, Delong Ran, Zhiyuan Qiu, Shiduo Zhang, Xiuhan Lin, and Xiaoyun Wang. HuFu: Hash-and-Sign Signatures From Powerful Gadgets. Algorithm Specifications and Supporting Documentation. 2023.
9. Rafael del Pino, Shuichi Katsumata, Thomas Prest, Mélissa Rossi. Raccoon: A Masking-Friendly Signature Proven in the Probing Model. CRYPTO, 2024.
10. A. Becker, N. Gama, A. Joux, Solving shortest and closest vector problems: The decomposition approach, IACR Cryptology ePrint Archive, 2013/685.
11. I. Semaev. New Digital Signature Algorithm EHT, Cryptology ePrint Atchive, 2022/339.
12. I. Semaev. New Digital Signature Algorithm EHTv2, NISK 2022, 28.11-1.12.2022, Kristiansand, Norway.
13. Leo Ducas, Alain Durmus, Tancrede Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians // Ran Canetti and Juan A. Garay, editors, Advances in Cryptology – CRYPTO, pages 40–56. Springer, 2013.
14. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures // Mitsuru Matsui, editor, Advances in Cryptology – ASIACRYPT, pages 598–616. Springer, 2009.
15. Vadim Lyubashevsky. Lattice signatures without trapdoors // David Pointcheval and Thomas Johansson, editors, Advances in Cryptology – EUROCRYPT, pages 738–755. Springer, 2012.
16. Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Juliane Kramer, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qTESLA // Cryptology ePrint Archive, Number 2019/085, 2019. [Electronic resource]. – Access mode: https://eprint.iacr.org/2019/085.
17. Melissa Azouaoui, Olivier Bronchain, Gaetan Cassiers, Clement Hoffmann, Yulia Kuzovkova, Joost Renes, Markus Schonauer, Tobias Schneider, Francois-Xavier Standaert, and Christine van Vredendaal. Leveling Dilithium against leakage: Revisited sensitivity analysis and improved implementations // Cryptology ePrint Archive, Report 2022/1406, 2022. [Electronic resource]. – Access mode: https://eprint.iacr.org/2022/1406.
18. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions // Richard E. Ladner and Cynthia Dwork, editors, 40th ACM STOC, pages 197–206. ACM Press, May 2008.
19. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller // David Pointcheval and Thomas Johansson, editors, EUROCRYPT 2012, volume 7237 of LNCS, pages 700–718. Springer, Heidelberg, April 2012.
20. Yang Yu, Huiwen Jia, and Xiaoyun Wang. Compact lattice gadget and its applications to hash-and-sign signatures // CRYPTO 2023, page (to appear), 2023.
21. Del Pino R., Katsumata S., Maller M., Mouhartem F., Prest T., & Saarinen M. J. (2024, May). Threshold raccoon: Practical threshold signatures from standard lattice assumptions // Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 219–248). Cham: Springer Nature Switzerland.

Надійшла до редколегії 27.05.2024

Відомості про авторів:

Горбенко Юрій Іванович – канд. техн. наук, АТ «Інститут Інформаційних Технологій», перший заступник, головного конструктора, Україна; e-mail: gorbenkou@iit.kharkov.ua, ORCID: https://orcid.org/0000-0003-0073-9107

Острянська Єлизавета Вадимівна – Харківський національний університет імені В.Н. Каразіна, молодший науковий працівник, АТ «Інститут Інформаційних Технологій», аналітик з систем захисту інформації; Україна; e-mail: antelizza@gmail.com