МЕТОДИ ПЕРСПЕКТИВНИХ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ
МЕТОДЫ ПЕРСПЕКТИВНЫХ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ
METHODS OF PROMISING CRYPTOGRAPHIC TRANSFORMATIONS

УДК 004.056.55 **DOI:10.30837/rt.2021.4.207.01**

І.Д. ГОРБЕНКО, д-р техн. наук, О.Г. КАЧКО, канд. техн. наук, О.В. ПОТІЙ, д-р техн. наук,
Ю.І. ГОРБЕНКО, канд. техн. наук, В.А. ПОНОМАР, канд. техн. наук,
М.В. ЄСІНА, канд. техн. наук, І.В. СТЕЛЬНИК, С.О. КАНДІЙ, К.О.КУЗНЕЦОВА

ОБҐРУНТУВАННЯ ТА ПРОПОЗИЦІЇ ЩОДО ВИБОРУ, УДОСКОНАЛЕННЯ ТА СТАНДАРТИЗАЦІЇ МЕХАНІЗМУ ПОСТКВАНТОВОГО ЕЛЕКТРОННОГО ПІДПИСУ НА НАЦІОНАЛЬНОМУ ТА МІЖНАРОДНОМУ РІВНЯХ

Вступ

Наразі, та очевидно в перспективі, для криптографічного захисту інформації (КЗІ) застосовуються та будуть застосовуватись математичні методи, механізми та алгоритми стандартизованих асиметричних криптоперетворень типу електронний підпис (ЕП). Він є основною та суттєвою складовою забезпечення кібербезпеки у сенсі якісного надання таких послуг з безпеки інформації як цілісність, неспростовність та автентичність інформації та даних, що обробляються [1 – 4]. Але є реально обґрунтовані підозри, що у постквантовий період існуючі стандарти ЕП будуть зламуватись та компрометуватись з використанням класичних та квантових криптоаналітичних систем з відповідним математичним, програмним та апаратно-програмним забезпеченням [5 – 13].

Аналіз підтвердив, що уже практично розроблені, виготовлені та застосовуються квантові комп'ютери. Стан створення та можливості застосування квантових комп'ютерів для вирішення задач криптоаналізу можна оцінити наступним чином [1 – 13].

Згідно з [9 – 11] IBM розробила та представила квантовий 127-кубітний процесор Eagle. Він прийшов на зміну 65-кубітному квантовому процесору Hummingbird, що відповідає дорожній карті квантових технологій від IBM [10].

Є відомості [10, 12] про наміри IBM представити 433-кубітний процесор Osprey в 2022 р., а 1121-кубітний процесор Condor – в 2023 р. Вказане відповідає дорожній карті, наведеній в [10]. В [10] також зазначено, що відмінність Eagle від попередніх процесорів полягає в тому, що він потребує, завдяки застосуванню мультиплексування зчитування, на кубіт регістрі значно меншої кількості електроніки для контролю та зчитування. Також IBM повідомляє про наміри щодо побудови на основі покращених чіпів нової інтегрованої квантової обчислювальної системи IBM Quantum System Two замість вже існуючої системи IBM Quantum System One.

Компанія D-Wave [13], що відома своїми розробками в сфері побудови псевдоквантових (гібридних) комп'ютерів з великою загальною кількістю кубітів (понад 2000 кубітів на початку та понад 5000 кубітів сьогодні), повідомила про наміри представити машину із загальною кількістю кубітів понад 7000 приблизно в 2023 – 2024 рр. та про наміри щодо розробки власних комп'ютерів для провідникових квантових машин гейтового типу (розробкою яких наразі займаються IBM, Google та інші).

На наш погляд, цим даним можна довіряти, але зрозуміло, що фактичний стан розроблення та застосування потужних квантових комп'ютерів та їх математичного і програмного забезпечень є, очевидно, строго конфіденційним та надійно захищається, а розголошуються тільки явно відомі дані про квантові комп'ютери та їх можливості застосування в криптології. Дані, що наведені вище, певною мірою нами перевірені практично [8].

Вирішення вказаної проблеми кібербезпеки та безпеки інформації у цілому в перехідний та постквантовий періоди може бути здійснено на основі розроблення, прийняття та застосування як на міжнародному, так і національному рівнях, в тому числі стандартизованих постквантових ЕП. Вказане може досягатись суттєвим обґрунтуванням, розробкою пропозицій щодо застосування нових математичних методів та механізмів криптоперетворень типу ЕП [1 – 4, 23 – 28] на основі існуючих альтернатив та відповідних моделей безпеки [14 – 19].

Таким чином, зважаючи на можливості та перспективи зламу існуючих асиметричних криптосистем типу ЕП, Національний інститут стандартів і технологій (NIST) США закінчив та прийняв рішення у вигляді проєкту стандарту NIST 8309 щодо 2-го раунду конкурсу на перспективні стандарти асиметричних криптоперетворень [4, 24 – 28]. Визначені фіналісти типу ЕП 2-го етапу конкурсу ґрунтуються на методах Crystals-Dilithium [24, 25, 30 – 33], Falcon [26] та Rainbow [28]. Визначено також три альтернативні кандидати на міжнародний постквантовий стандарт ЕП – GeMSS, Picnic та SPHINCS+ [4], які потребують більш детальних досліджень, скоріше всього на 4-му етапі конкурсу.

Попередній аналіз показав, що в Україні є розуміння існування загроз кібербезпеці та безпеці інформації у випадку застосування у перехідний та постквантовий періоди існуючих стандартизованих ЕП. Розроблено та прийнято національний стандарт «Алгоритми асиметричного шифрування та інкапсуляції ключів» (ДСТУ 8961-2019), що побудований на основі застосування криптоперетворень на алгебраїчних решітках. Особливістю цього стандарту є суттєве підвищення криптографічної стійкості асиметричного шифрування та інкапсуляції ключів у перехідний та постквантовий період проти усіх відомих класичних, квантових, спеціальних атак та атак на основі помилок [17 – 19]. Тому, по суті, наразі одним із основних проблемних питань щодо забезпечення необхідних рівнів безпеки в перехідний та постквантовий періоди є також розробка та прийняття постквантових стандартів ЕП.

Метою цієї статті є обґрунтування, порівняння альтернатив та розробка пропозицій щодо вибору та стандартизації постквантових стандартів ЕП на міжнародному та національному рівнях з урахування результатів 2-го та 3-го раундів конкурсу NIST США [4] та національних досліджень [6 – 8, 24, 27].

1. Аналіз альтернатив та вибір методу (схеми) розробки та прийняття постквантового стандарту ЕП на національному та міжнародному рівнях

Аналіз показав [4, 20 – 28], що серед постквантових механізмів (схем) ЕП суттєві переваги надані проєктам ЕП Crystals-Dilithium [23, 30, 31], Falcon [26] та Rainbow [28]. Вони рекомендовані для подальшого дослідження та стандартизації в процесі 3-го етапу конкурсу та, по суті, є фіналістами конкурсу NIST США. Причому кращими визначено як математичні основи, так і алгоритми ЕП Crystals-Dilithium [6, 11 – 13]. ЕП Falcon досліджувався на 2-му раунді як один із трьох проєктів стандартів ЕП. Основним та домінуючим підходом до проєктування механізму ЕП Falcon є використання парадигми «геш і підпис» [26]. Його перевагою є доказова стійкість в межах моделі квантового випадкового оракула. Але його аналізу присвячено значно менше робіт, ніж, наприклад, щодо Crystals-Dilithium та Rainbow.

У [26, 27] представлено удосконалений варіант схеми ЕП Falcon. Вказана удосконалена реалізація варіанту ЕП Falcon забезпечує постійний час виконання операцій перетворень. Вона може реалізуватись без використання апаратного забезпечення, хоча при його використанні може досягатись вища продуктивність. В новій реалізації необхідний менший обсяг оперативної пам’яті, вона більш ефективна як у великих системах (x86 з ядрами Skylake, POWER8 тощо), так і на малих мікроконтролерах (ARM Cortex M4) [26].

Основними властивостями ЕП Falcon є наступні [26, 27].

1. В ЕП Falcon для криптоперетворень використовується вибірка Гауса, розроблена Престом, Рікоссетом та Россі. Така вибірка використовує вибірку з відхиленнями з ретельно налаштованими параметрами таким чином, що спостереження за швидкістю відхилення не дає ніякої статистично корисної інформації щодо особистих ключів при усіх параметрах безпеки.

2. Проблемним було використання плаваючої точки, що приводило до появи неконстантних операцій, наприклад множення поліномів. Щоб уникнути будь-яких неконстантних операцій під час ЕП, їх було оптимізовано. Зокрема, ділення та квадратні корені використовуються лише при генерації ключів і не можуть викрити навіть незначної інформації про особистий ключ.

3. Весь доступ до пам'яті засобів реалізації здійснюється за несекретними адресами. Причому шаблон доступу до пам'яті не залежить від будь-якої секретної інформації, як для генерування ключів, так і для вироблення ЕП.

4. В реалізації ЕП Falcon не потрібна підтримка операцій з плаваючою точкою. Хоча така реалізація може використовувати апаратне забезпечення з плаваючою точкою, коли вона доступна. Вона включає код емуляції з плаваючою точкою, який забезпечує використання лише операції з цілими числами. Такий код емуляції характеризується повністю постійним часом виконання перетворень для всіх значень, які можуть з'являтися, і переноситься на всі платформи, що мають, наприклад, компілятор C99 і звичайні типи цілочисельних значень фіксованої ширини (uint32_t, uint64_t).

5. Нова реалізація Falcon є більш швидкою та ефективна при роботі з RAM. Так, коли підтримуються операційні коди AVX2, то така реалізація може їх використовувати в Falcon-512 на Intel Core i7-6567U на частоті 3,3 ГГц [26]. При цьому, якщо використовується одне ядро, то за секунду можна генерувати приблизно 7700 ЕП. Також можна реалізувати варіант зі зменшеним обсягом RAM. За цієї умови ЕП може бути обчислений для Falcon-512 у межах менше, ніж 40 кБ оперативної пам’яті. Це досягається для тимчасових значень та незначного використання простору стека, при цьому досягається приблизно 3800 підписів в секунду.

6. З використанням процесорів ARM Cortex M4 для підвищення продуктивності були додані спеціальні вбудовані процедури збірки; варіант із зменшеним використанням оперативної пам’яті. При його застосуванні можна виконати ЕП «Сокіл» та Falcon-512 приблизно за 21,2 мільйона тактів. Але це досягається при застосуванні «розширеного особистого ключа». Безпосередньо розширений ключ обчислюється зі звичайного особистого ключа приблизно за 16,2 млн. тактів і використовує 57,3 кБ оперативної пам’яті.

7. Удосконалена реалізація ЕП «Сокіл» та ЕП Falcon має відкритий код (ліцензія MIT щодо Falcon) та доступна на веб-сайті: https://FALCON-sign.info.

Разом з тим, в удосконаленій версії враховано основні відмінності версії 2.0 Falcon специфікації від версії 1.0 специфікації? Вони полягають у наступному [26].

1) Видалено набір параметрів II – III рівня, що спричинило видалення n=768 та $\varphi=x^n-x^{n/2}+1$. Детально версію 1.1 специфікації, а також еталонну реалізацію наведено в [27].

2) Наведено опис режиму відновлення ключів, який робить Falcon ще більш конкурентоспроможним з точки зору компактності.

Зроблено кілька інших доповнень, які по суті складаються з уточнення та деталізації кількох моментів.

Крім того, під час розроблення обох версій ЕП Falcon, були прийняті обмеження щодо рівнів безпеки, максимально 256 біт проти класичного та 128 біт – проти квантового криптоаналізу [4, 26, 27]. Такі обмеження залишились для проєкту ЕП Falcon також на 3-му етапі міжнародного конкурсу NIST США. Ці обмеження, на наш погляд, пов’язані зі складністю обчислення загальносистемних параметрів, а також із суттєвим впливом їх збільшення параметрів на швидкодію ЕП. Тобто, для безпечного використання ЕП Falcon повинні бути знайдені набори загальносистемних параметрів, за яких забезпечується стійкість до всіх відомих та потенційних атак, тобто класичних, квантових, на основі помилок та спеціальних атак.

В процесі формування вимог до ЕП NIST у рамках конкурсу був зацікавлений тільки в наборах загальносистемних параметрів до 256 біт класичної безпеки включно. Проте, на нашу думку, на перспективу доцільним є забезпечення не менше 384 і 512 біт безпеки проти класичного криптоаналізу та не менше 192 та 256 біт безпеки проти квантового криптоаналізу [29]. Але, як показали дослідження, як з точки зору теорії, так і практики, генерація загальносистемних параметрів для використання 384 і 512 біт безпеки проти класичного криптоаналізу та 192 і 256 біт безпеки проти квантового криптоаналізу є важливою задачею.

Пропозиції щодо розв’язку такої проблеми у вигляді проєкту Національного стандарту у наступному [27, 29].

Розроблення проєкту стандарту має за мету визначення криптографічних алгоритмів ЕП на алгебраїчних решітках для забезпечення послуг цілісності, справжності, доступності та неспростовності інформації та ресурсів, які служать основою надання електронних довірчих послуг під час взаємодії двох або більше суб'єктів, які довіряють надавачу електронних довірчих послуг. Стандарт призначено для використання під час розробки систем, комплексів та засобів криптографічного захисту інформації для надання користувачам послуг цілісності, справжності та неспростовності підписувача в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах, в тому числі для захисту від спеціальних атак, а також у постквантовий період.

Основною перевагою проєкту стандарту є доказова стійкість в межах моделі квантового випадкового оракула. Іншими перевагами ЕП «Сокіл » є те, що він забезпечує, у порівнянні з усіма іншими алгоритмами ЕП, мінімальну суму розміру відкритого ключа та розміру підпису, а також ефективні алгоритми підписання та перевірки ЕП, хоча генерація ключів в них відбувається повільніше. Також ЕП «Сокіл» без проблем може вводиться в існуючі протоколи та додатки та забезпечує прийнятну загальну продуктивність. Під час застосування ЕП «Сокіл» є можливість реалізувати його за технологією обробки з відновленням повідомлень.

В алгоритмі асиметричного ЕП «Сокіл» використовують асиметричну пару ключів – особистий (секретний) ключ та відкритий ключ. Для ЕП інформації (даних) використовують особистий (секретний) ключ підписувача, а для перевірки ЕП використовують відкритий ключ перевірки ЕП. На основі перевірки ЕП інформації (даних) отримувачем (перевірником) встановлюється та приймається рішення щодо її цілісності, справжності та неспростовності підписувача тощо.

Прирозробленні стандарту мають бути враховані вимоги щодо забезпечення криптографічної стійкості проти спеціальних атак на основі витоку по технічних каналах, а також потенційних класичних та квантових атак, в тому числі у перехідний та постквантовий періоди. Стандарт розроблено з урахуванням досвіду створення та застосування національних стандартів ДСТУ 7564:2014, ДСТУ 7624:2014, ДСТУ 8845:2019 та ДСТУ 8961:2019.

Стандарт у залежності від рівнів криптографічної стійкості проти класичних та квантових атак, атак сторонніми каналами та на основі помилок, які необхідно забезпечити, можна застосовувати в трьох режимах роботи:

– 128 біт захисту від класичних атак та 64 біт захисту від квантових атак, захист від спеціальних атак сторонніми каналами та на основі помилок (запас стійкості відповідає ДСТУ 7624:2014 (128), AES (128)) (1 режим);

– 256 біт захисту від класичних атак та 128 біт захисту від квантових атак, захист від спеціальних атак сторонніми каналами та на основі помилок (запас стійкості відповідає ДСТУ 7624:2014 (256), AES (256)) (2 режим);

– 512 біт захисту від класичних атак та 256 біт захисту від квантових атак, захист від спеціальних атак сторонніми каналами та на основі помилок (запас стійкості відповідає ДСТУ 7624:2014 (512)) (4 режим).

В кожному із режимів роботи необхідно використовувати криптографічні перетворення та відповідні функції гешування з необхідними розмірами параметрів та ключів.

Важливим є те, що в кожному з вказаних режимів роботи, за рахунок застосування ЕП, для надання послуг цілісності, справжності, доступності та неспростовності використовують алгоритми криптографічних перетворень з відкритими ключами на основі алгебраїчних NTRU решітках з заданою вибіркою, що дозволяє у залежності від вимог, отримати різні рівні безпеки та техніко-економічні і техніко-експлуатаційні характеристики (показники). В цій статті наводяться відповідні результати досліджень з використанням комплексної методики [14 – 16]. Їх аналіз показує, що розроблення та впровадження проєкту національного стандарту «Інформаційні технології. Криптографічний захист інформації. Алгоритми електронного підпису на алгебраїчних NTRU решітках з заданою вибіркою» є актуальною та надзвичайно важливою науково-технічною задачею, вирішення якої спрямоване на забезпечення належного виконання вимог Положення про порядок здійснення криптографічного захисту інформації в Україні, затвердженого Указом Президента України від 22.05.98 № 505, а також тісно пов'язане із виконанням завдань та основних положень Концепції національної безпеки

України, законів України «Про основні засади забезпечення кібербезпеки України» [32], «Про електронні довірчі послуги», «Про захист інформації в інформаційно-телекомунікаційних системах» та інших нормативно-правових актів із захисту національного інформаційного простору України.

Перевагами проєкту стандарту «Сокіл» є доказова стійкість, так як він в межах моделі квантового випадкового оракула. При його використанні будуть забезпечуватись, у порівнянні з усіма іншими алгоритмами ЕП постквантового періоду, мінімальна сума розміру відкритого ключа та розміру підпису, а також ефективні алгоритми підписання та перевірки ЕП, хоча генерація ключів в них відбувається повільніше. Проєкт стандарту ЕП «Сокіл» без проблем може застосовуватись в існуючих протоколах та додатках, забезпечуючи при цьому прийнятну загальну продуктивність. Також ЕП «Сокіл» може ефективно застосовуватись за технологією обробки інформації з відновленням повідомлень.

Наведені результати досліджень дозволили прийняти рішення про перспективність та можливість застосування математичного методу (схеми) Falcon [26] для створення проєкту постквантового національного стандарту ЕП «Сокіл» [27].

2. Сутність математичних методів (схем) (механізмів) ЕП Falcon та «Сокіл»

Проєкти ЕП Falcon [26] та «Сокіл»[27] – це засновані на алгебраїчній решітці механізми ЕП, що використовують парадигму «геш і підпис». Їх стійкість ґрунтується на складності проблеми SIS (коротке ціле рішення) над решітками NTRU, а докази безпеки є як у класичній випадковій моделі оракула (ROM), так і у квантовій моделі оракула QROM. Інтегрально стосовно схеми необхідно відмітити наступне [26, 27]. Необхідно також погодитись, що однією з основних переваг схеми Falcon є те, що вона забезпечує, у порівнянні з усіма іншими схемами ЕП, що були представлені до 2-го раунду конкурсу NIST США та досліджуються на 3-му етапі, мінімальну суму розміру відкритого ключа та розміру ЕП. Також, схема ЕП Falcon забезпечує постійний час виконання операцій перетворень, що дозволяє забезпечити захист і від спеціальних атак. ЕП Falcon та «Сокіл» ефективні при підписанні та перевірці ЕП, хоча генерація ключів в них виконується, у порівнянні, наприклад, з Crystals-Dilithium, повільніше [23, 30, 31]. Також ЕП Falcon [4, 26] та «Сокіл» [27] можуть без особливих проблем вводитися в існуючі протоколи та додатки та забезпечувати загальну продуктивність, що вимагається. Але, ЕП Falcon, при його застосуванні для реалізації ЕП «Сокіл», є складнішим для впровадження, ніж Crystals-Dilithium. Так, він вимагає використання деревоподібних структур даних, операцій з плаваючою комою та випадкової вибірки, по суті, для викривлення коефіцієнтів поліномів, з використанням дискретних гаусових розподілів. Крім того, щодо схеми ЕП на першому етапі її реалізації у детермінованій процедурі ЕП [26] спостерігався потенційний витік (підозри) інформації про особистий ключ.

Генеалогія механізму ЕП Falcon може бути подана у вигляді генеалогічного дерева, що наведене на рис. 1. Безпосередньо схема (метод) та його математична основа є результатом багаторічної праці криптологів та математиків [26]. Значне число досліджень та розробок поступово привели до нинішньої технології ЕП Falcon.

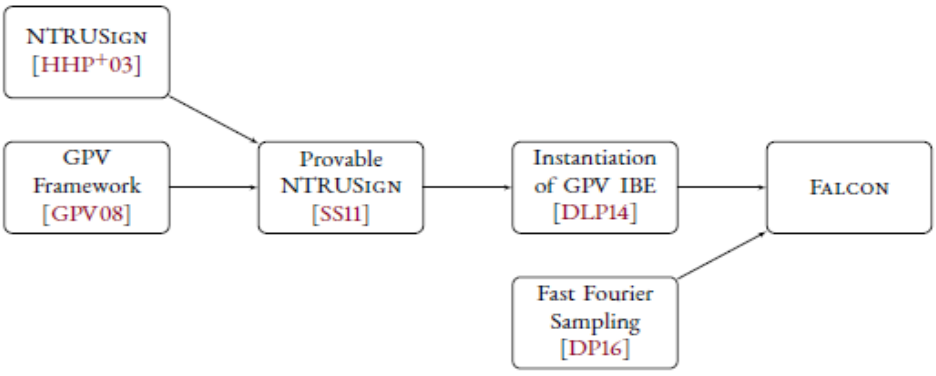

Рис. 1. Генеалогічне дерево Falcon

У 2008 р. Gentry, Peikert і Vaikuntanathan [34, 35] запропонували метод, який дозволив суттєво зменшив вразливість процедури ЕП NTRU. Більше того, зробив це доказово стійким способом. Результатом застосування вказаного способу стала основна платформа (GPV-платформа) побудови захищених ЕП на основі концепції «геш та підпис». Ця платформа будується на основі використання двох специфічних інгредієнтів:

1) Класу криптографічних решіток у вигляді NTRU решіток.
2) Нової методики підвищення безпеки, яка була названа «швидкою вибіркою Фур'є».

Подальший розвиток схеми наведено в [26], де було запропоновано поєднати платформу GPV з решітками NTRU. Результат такого поєднання привів до доказово стійкого NTRUSign. У цілому, коротко, схему підпису на основі схеми Falcon можна подати у такому вигляді:

Falcon = платформа GPV + решітки NTRU + швидка вибірка Фур'є.

Таким чином, схема Falcon ґрунтується на решітках, що уже були випробувані часом в NTRU асиметричному криптоперетворенні. Усі операції в такій конкретній NTRU решітці виконуються над поліномами $\mathbb{Z}[X]$ по модулю $\phi = X^n + 1$, де n – степінь двійки ($n = 512$ для Falcon-512, 1024 – для Falcon-1024 та додатково 2048 – для «Сокіл»). В схемах Falcon та «Сокіл» відкриті ключі, особисті ключі та підписи подаються також у вигляді поліномів, коефіцієнти яких цілі числа, тобто в вигляді $\mathbb{Z}[X]$. Але все ж таки певні проміжні значення перетворень є поліномами, коефіцієнти яких не є цілими числами.

Особистий ключ складається з чотирьох поліномів f, g, F і G, які задовольняють (вирішують) рівняння NTRU виду [26, 27]:

$$fG - gF = q \bmod \phi, \tag{1}$$

де $q = 12289 = 3 \times 2^{12} + 1$ – просте ціле число, що вибрано у такому вигляді для забезпечення використання при множенні поліномів, швидкого NTT множення. Причому коефіцієнти елементів особистого ключа є обмеженими по значенням (малі) цілі числа, їх величину рекомендується змінювати в межах від -127 до +127 [25, 26].

Відкритий ключ подається у вигляді поліному $h = \mathbb{Z}[X]$, коефіцієнти якого можуть змінюватись в інтервалі від 0 до q-1, тобто приймають значення, у порівнянні з особистим ключем та поліномом, що є ЕП, у найбільшому інтервалі – від 0 до 12289, причому

$$fh = g \bmod \phi \bmod q \tag{2}$$

або

$$h = (g / f) \bmod \phi \bmod q. \tag{3}$$

В схемі Falcon значення ЕП також подається у вигляді поліному, але коефіцієнти поліному ЕП рекомендується змінювати в інтервалі від -1080 до +1080 [26, 25].

Таким чином, коефіцієнти полінома особистого ключа рекомендується змінювати в інтервалі від -127 до +127, безпосередньо значення ЕП в інтервалі -1080 до +1080, а значення відкритого ключа у найбільшому інтервалі – від 0 до 1228, а також в менших інтервалах зі вказаними обмеженнями. По суті, застосування у поліномах названих сутностей трьох модулів дозволяє у цілому збільшити ентропію криптографічного перетворення схеми Falcon.

Необхідно також відмітити, що особистий ключ – поліном $f \in \mathbb{Z}[X] \bmod(\phi)$ можна подати [26] у вигляді матриці розміром $n \times n$, кожен *i*-й рядок якої складається з коефіцієнтів

$$f_i = x^i \bmod(\phi(x)). \tag{4}$$

Додавання та множення таких матриць є однозначним додаванню та множенню відповідних поліномів, взятих по модулю $\phi(x)$.

Аналіз показав, що особистий ключ насправді є коротким базисом для решітки $2n \times 2n$ та може бути поданим у матричному вигляді [26]:

$$B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}. \quad (5)$$

Також відкритий ключ ж може бути поданий у вигляді основи для тієї ж самої алгебраїчної решітки, але зі збільшеними векторами:

$$p = \begin{bmatrix} -h & I_n \\ qI_n & O_n \end{bmatrix}, \quad (6)$$

де I_n і O_n – одинична і нульова матриці з $n \times n$ розміром відповідно.

Безпосередньо ЕП повідомлення M виконується у такій послідовності:

1) Підписувач, що має доступ до особистого ключа, під час ЕП кожен раз генерує нове випадкове значення (сіль) *r* відповідної довжини.

2) Здійснюється конкатенація та наступне гешування при ЕП кожного нового значення r та повідомлення *M*, що підписується, яке перетворюється в поліном $c \in \left[X\right]/(\phi(x))$ з коефіцієнтами в діапазоні від 0 до $q-1$.

3) Підписувач, використовуючи свій особистий ключ чи знання про свій особистий ключ, обчислює два поліноми s_1 та s_2, але такі, щоби вони задовольняли рівнянню:

$$s_1 + s_2 h = c (\mathrm{mod}\, \phi \,\mathrm{mod}\, q). \quad (7)$$

Безпосередньо ЕП повідомлення *M* є поліномом s_2 та випадкове чи псевдовипадкове *r* значення, що визначає ентропію перетворення типу ЕП.

При перевірці ЕП повідомлення *M*, спочатку обчислюється геш-значення конкатенації значення r та повідомлення *M*, які отримані перевірником, потім воно перетворюється в поліном *c* з коефіцієнтами в діапазоні від 0 до $q-1$. Далі, використовуючи обчислене значення *c*, *в*ідкритий ключ *h* та значення ЕП s_2, обчислюємо

$$s_1 = c - s_2 h \,\mathrm{mod}\, \phi \,\mathrm{mod}\, q. \quad (8)$$

3) Наостанок перевіряється, що складений вектор поліномів (s_1, s_2) розмірності $2n$ дійсно має достатньо низьку (необхідну) норму.

Перевірку ЕП можна виконати повністю за допомогою обчислень над цілими числами по модулю q, а оскільки $2n$ ділить значення $q-1=12288$, то для прискорення множення при перевірці можна використати швидке NTT множення (перетворення). За рахунок цього підвищується швидкодія перевірки ЕП та більш ефективно може використовуватись оперативна пам'ять засобу перевірки.

3. Алгоритми обчислення ключів, основні положення вироблення та перевірки ЕП

3.1. Алгоритм обчислення ключів

В цьому параграфі наводиться системно зібраний алгоритм із доступних джерел та подається з єдиних системних позицій алгоритм генерування ключів для проєктів стандартів ЕП Falcon та «Сокіл» [26, 27].

Етапи обчислення асиметричної пари ключа – особистої та відкритої частин.

Особистий ключ складається з поліномів *f, g, F, G*.

Відкритий ключ складається з поліному *h*.

Далі наведено етапи виконання алгоритму обчислення ключів.

3.1.1. Обчислення компонентів особистого ключа (поліноми *f, g*)

1) Обчислення відкритого ключа (полінома $h=g/f(\mathrm{mod}\ q, \mathrm{mod}\ x^n+1)$);

2) Обчислення компонентів особистого ключа (поліноми *F*, *G*, що повинні бути розв'язком *NTRU* рівняння

$$f*G - g*F = q\ (mod\ x^n+1) \quad (9)$$

Алгоритм обчислення поліномів *f, g*.

Коефіцієнти поліномів *f, g* повинні бути випадковими, знаходитись в інтервалі [-($2^{SK\_0\_BITS}$-1, ($2^{SK\_0\_BITS}$-1], задовольняти розподілу Гауса для $\sigma = 1.17\sqrt{\frac{q}{2n}}$, μ =0, що забезпечує властивості для полінома *h*, як для випадкового полінома [26, 25].

Сума коефіцієнтів для поліному (*f*) та поліному (*g*) повинна бути непарною.

Квадратична норма вектору [*f, g*] не повинна перевищувати *max_norma* = 1.17^2q.

Квадратична норма вектору [*f*', *g*'], ортогонального до вектору [*f, g*], не повинна перевищувати *max_norma* = 1.17^2q.

3.1.2. Алгоритм обчислення відкритого ключа (поліном *h*)

Формула для обчислення:

$$h=g/f \ (mod\ q, mod\ x^n+1). \qquad (10)$$

Поліном *f* повинен мати інверсію, що забезпечує можливість обчислення поліному *h*.

3.1.3. Алгоритм обчислення особистого ключа – поліномів *F, G*

Поліноми *F, G* повинні задовольняти NTRU рівнянню (9)

Загальний алгоритм обчислення поліномів *F, G*:

1. Для поліномів *f, g* виконати покроковий перехід від поля x^n+1 до полів $x^{n/2}+1$, $x^{n/4}+1$, … x^1+1. В результаті отримаємо поліноми:

$f^{(n/2)}$ – поліном *f* для поля $x^{n/2}+1$, який має *n*/2 коефіцієнтів;

$g^{(n/2)}$ – поліном *g* для поля $x^{n/2}+1$, який має *n*/2 коефіцієнтів;

$f^{(n/4)}$ – поліном *f* для поля $x^{n/4}+1$, який має *n*/4 коефіцієнтів;

$g^{(n/4)}$ – поліном *g* для поля $x^{n/4}+1$, який має *n*/4 коефіцієнтів;

…

$f^{(1)}$ – поліном *f* для поля *x*+1, який має один коефіцієнт;

$g^{(1)}$ – поліном *g* для поля *x*+1, який має один коефіцієнт;

Виконується $log_2 n$ кроків, всі кроки виконуються ідентично.

2. Вирішити діафантове рівняння $f^{(1)}*G'^{(1)} - g^{(1)}*F'^{(1)} = 1$ відносно невідомих $F'^{(1)}$, $G'^{(1)}$; (умова існування рішення в цілих числах – $GCD\ (f^{(1)}, g^{(1)}) = 1$).

3. Обчислити $F^{(1)} = q * F'^{(1)}$; $G^{(1)} = q * G'^{(1)}$; $F^{(1)}$, $G^{(1)}$ – рішення рівняння $f^{(1)}*G^{(1)} - g^{(1)}*F^{(1)} = q$, поліноми $F^{(1)}$, $G^{(1)}$ мають по одному коефіцієнту.

4. Для пари поліномів $F^{(1)}$ $G^{(1)}$, виконати покроковий перехід від поля x^1+1 до полів x^2+1, x^4+1, …, x^n+1. В результаті отримаємо поліноми:

$F^{(2)}$ – поліном, який має 2 коефіцієнта;

$G^{(2)}$ – поліном, який має 2 коефіцієнта;

$F^{(4)}$ – поліном, який має 4 коефіцієнта;

$G^{(4)}$ – поліном, який має 4 коефіцієнта;

…

$F^{(n)}$ – поліном, який має *n* коефіцієнтів;

$G^{(n)}$ – поліном, який має *n* коефіцієнтів;

Якщо усі коефіцієнти $F^{(n)}$, $G^{(n)}$ задовольняють вимогам щодо допустимих розмірів, то приймається рішення $F^{(n)}=F$, $G^{(n)}=G$. Якщо ні, то процес починається з обчислення складових особистого ключа *f, g* повторно.

Виконується $log_2 n$ кроків, всі кроки виконуються ідентично.

3.1.4. Перетворення переходу від поля x^t+1 до поля $x^{t/2}+1$

Вхід: поліном *a* з коефіцієнтами $a_0, a_1, \ldots, a_{t-1}$;

Вихід: поліном a з коефіцієнтами $b_0, b_1, \ldots, b_{t/2-1}$;

Для фіксованого *t* необхідно виконати кроки 1–3.

1. Формування двох поліномів *e*, *o*, перший з коефіцієнтами з парними номерами, другий з непарними вхідного полінома *a*:

$e_j = a_{2j}$ (j =0, 1, $t/2 - 1$);

$o_j = a_{2j+1}$ (j =0, 1, $t/2 - 1$);

2. Обчислення поліномів *e*2, *o*2:

$e2 = e^2 \ mod \ x^{t/2}+1$;

$o2 = o^2 \ mod \ x^{t/2}+1$;

3. Обчислення коефіцієнту поліному b_0:

$b_0 = e2_0 + o2_{t/2-1}$

4 Обчислення решти коефіцієнтів з індексами 1, 2, $t/2–1$

$b_j = e2_j – o2_{j-1}$

При виконанні кроків 2–4 застосовують довгі числа.

Перехід від поля x^t+1 до поля $x^{2t}+1$

Включає:

- обчислення поліномів F для поля $x^{2t}+1$;
- обчислення поліномів G для поля $x^{2t}+1$.

Для обчислення поліному $F'^{(2t)}$ для поля $x^{2t}+1$ застосовують поліном $F^{(t)}$ для поля x^t+1 та поліном $g^{(2t)}$ для поля $x^{2t}+1$, який отримано на етапі переходу від поля $2t$ до t.

Для формування поліному $G'^{(2t)}$ для поля $x^{2t}+1$ застосовують поліном $G^{(t)}$ для поля x^t+1 та поліном $f^{(2t)}$ для поля $x^{2t}+1$, який отримано на етапі переходу від поля $2t$ до t

Для перетворення застосовують однакові операції:

Вхід:

- поліном $A^{(t)}$ з коефіцієнтами $A^{(t)}_0, A^{(t)}_1, \ldots, A^{(t)}_{t-1}$;
- поліном $b^{(2t)}$ з коефіцієнтами $b^{(2t)}_0, b^{(2t)}_1, \ldots, b^{(2t)}_{2t-1}$.

Вихід:

- поліном $C^{(2t)}$ з коефіцієнтами $C^{(t)}_0, C^{(t)}_1, \ldots, C^{(t)}_{2t-1}$.

3.1.5. Прикінцеве отримання складових ключа – поліномів *F, G*

Для отримання коефіцієнтів поліному C необхідно виконати кроки 1–3.

1. Формування поліному u для поля $x^{2t}+1$: $u_{2j}=A^{(t)}_j$; $u_{2j+1}=0$, (j = 0, 1, …t-1);
2. Формування поліному v для поля $x^{2t}+1$: $v_{2j}= b^{(2t)}_{2j}$; $v_{2j+1}= -b^{(2t)}_{2j+1}$, ($j$ = 0, 1, …t-1);
3. Обчислення поліному C для поля $x^{2t}+1$: $C = u * v \ mod \ x^{2t}+1$

Пункти 1 – 3 необхідно виконати для обчислення поліномів $F'^{(2t)}$, $G'^{(2t)}$.

4. Редукція поліномів $F'^{(2t)}$, $G'^{(2t)}$.

Обчислюють значення цілого коефіцієнту k:

$$k = \left\lfloor \left(\mathrm{F}'^{(2t)} f^{*(2t)} + \mathrm{G}'^{(2t)} g^{*(2t)}\right) / \left(f^{(2t)} f^{*(2t)} + g^{(2t)} g^{*(2t)}\right) \right\rfloor.$$

Обчислюють нові значення $F'^{(2t)}$, $G'^{(2t)}$

$$(F'^{(2t)}, G'^{(2t)}) = (F'^{(2t)} - k * f^{(2t)}, G'^{(2t)} - k * g^{(2t)}).$$

Редукція продовжується до тих пір, поки $k \neq 0$.

5. В якості поточних значень $f^{(2t)}$, $g^{(2t)}$ приймаються обчислені значення $F'^{(2t)}$, $G'^{(2t)}$

$$f^{(2t)}=F'^{(2t)}; \ g^{(2t)}=G'^{(2t)};$$

Після обчислення поліномів для поля x^2+1, x^4+1, … x^n+1 отримані значення – поліноми F, G, що разом із особистими (секретними) складовими f та g і є особистим (секретним) ключем. Відмітимо також, що наведений вище метод розв'язку порівняння (1) може бути оптимізований по критерію складності і це потребує окремого розгляду.

3.2. Алгоритм вироблення ЕП

Вироблення ЕП виконується в два етапи: розгортання особистого ключа (однократно) та вироблення електронного підпису на сеансі зв'язку.

Алгоритм вироблення ЕП здійснюється виконанням таких складових етапу:

- розгортання особистого ключа, в процесі якого на його основі формується базис решітки у *FFT* форматі та обчислюється дерево для базису решітки;

- алгоритм обчислення компонентів ЕП, в процесі виконання якого виконується генерація псевдовипадкового поліному r по модулю q (сеансового ключа для ЕП), обчислюються безпосередньо компоненти електронного підпису s_1 та s_2 поліноми та відповідно розподілу Гауса застосовується функція вибірки та відбракування (sampler).

3.2.1. Алгоритм розгортання особистого ключа

Параметри:

n – визначає степінь полінома:

$\sigma 2$ – визначає дисперсію для розподілу Гауса.

Вхід:

- складові особистого ключа – поліноми f, g, F, G.

Вихід:

- базис решітки у вигляді матриці B, побудований згідно з особистим ключем;
- дерево для решітки, яке застосовується при обчисленні компонентів ЕП.

Формування на основі особистого ключа базису решітки.

Базис решітки у вигляді матриці B:

$$B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}.$$

Визначник цієї матриці дорівнює $f*G - g*F = q$, де f, g, F, G – є відповідні поліноми.

Для забезпечення швидких обчислень базис може задаватися в *FFT* форматі, тобто у вигляді

$$\hat{B} = \begin{bmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & B_{1,1} \end{bmatrix} = \begin{bmatrix} FFT(g) & -FFT(f) \\ FFT(G) & -FFT(F) \end{bmatrix}.$$

Формування матриці Грама.

Матриця Грама $\hat{G} = \hat{B} * B^*$, де B^* матриця є результатом транспонування матриці $\hat{B}$, та заміни комплексних даних на комплексно зв'язані дані:

$$\hat{G} = \begin{bmatrix} G_{0,0} & G_{0,1} \\ G_{1,0} & G_{1,1} \end{bmatrix} = \hat{B} * B^* = \begin{bmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & B_{1,1} \end{bmatrix} * \begin{bmatrix} B_{0,0}{}^* & B_{1,0}{}^* \\ B_{0,1}{}^* & B_{1,1}{}^* \end{bmatrix}.$$

Так як $G_{0,1} = G_{1,0}$, тому в подальшому будуть застосовуватись $G_{0,0}, G_{0,1}, G_{1,1}$

3.2.2. Формування дерева швидких обчислень для базису решітки

Компоненти ЕП залежать від особистого ключа та повідомлення, яке підписується. Усі обчислення, які необхідні для особистого ключа, можна виконати заздалегідь та оформити у вигляді двійкового дерева, яке має $\log_2 n$ рівнів.

Для побудови кожного наступного рівня $logn = log_2 n, log_2 n\text{-}1, \ldots, 2$ виконуються операції:

Матриця G для рівня $logn$ подається у вигляді добутку:

$$G=L*D*L^*,$$

де компоненти матриці G – поліноми порядку 2^{logn}; L – нижня трикутна матриця з одиничними діагональними елементами, при цьому $L_{0,0} = L_{1,1} = 1$; $L_{0,1} = 0$; $L_{1,0}$ належить обчисленню; D – діагональна матриця, тобто $D_{0,1} = D_{1,0} = 0$; елементи діагоналі $D_{0,0}, D_{1,0}$ належать обчисленню.

Показано, що таке подання існує та воно єдине.

Значення $L_{1,0}$ розглядається як вузол дерева наступного рівня.

Значення $D_{0,0}$ та $D_{1,1}$ застосовуються для побудови вузлів дерева кожного наступного рівня ($D_{0,0}$, $D_{1,1}$ – поліноми для поля x^t+1).

$D_{0,0}$ перетворюється в 2 поліноми порядку 2^{logn-1} (позначимо їх *left*1, *right*1) з урахуванням *FFT* представлення кожного поліному.

$D_{1,1}$ перетворюється в 2 поліноми порядку 2^{logn-1}(позначимо їх *left2*, *right*2) з урахуванням *FFT* представлення кожного поліному.

Значення *left*1, *right*1, *left*1 застосовуються в якості елементів матриці G для побудови лівого піддерева, тобто $G_{0,0} = left1, G_{0,1} = right1, G_{1,1} = left1$.

Значення *left*2, *right*2, *left*2 застосовуються в якості елементів матриці G для побудови правого піддерева, тобто $G_{0,0} = left2, G_{0,1} = right2, G_{1,1} = left2$.

Для першого рівня *logn*=1 значення елементу обчислюється за формулою: $\frac{\sqrt{G_{0,0}[0]}}{\sigma 2}$ ($G_{0,0}[0]$ – значення нульового (єдиного) коефіцієнту поліному $G_{0,0}$, σ2 – параметр).

Далі для обчислення ЕП застосовується особистий ключ key з полями *key*.*B*00, *key*.*B*01, *key*.*B*10, *key*.*B*11, *key*.*tree*.

3.2.3. Алгоритм обчислення компонентів електронного підпису

Спочатку обчислюється геш для випадкового (псевдовипадкового) рядка октетів *nonce* та повідомлення *m* для підпису, отриманий рядок перетворюється в поліном *r* з коефіцієнтами по модулю *q*. Далі обчислюються поліноми s_1 та s_2, такі що

$$s_1+s_2*h=r\ (mod\ q\ mod\ \varphi), \tag{11}$$

де *h* – відкритий ключ.

Для обчислення s_1 та s_2 застосовують розгорнутий особистий ключ та поліном *c*. При обчисленні додається випадковий шум, для створення якого застосовують генератор псевдовипадкових даних. Для ініціалізації цього генератору застосовують випадкову послідовність октетів, в подальшому позначену як *seed*.

Значення s_1 може бути обчислено з формули (11) при відомих s_2, *h*, *c*, тому в якості ЕП застосовують значення *nonce* та s_2.

Параметри та вхідні дані.

Вхідними параметрами являються довжини випадкових компонентів та гешу*:*

SEED_OCTETS – довжина *seed*;

NONCE_OCTETS – довжина *nonce*;

HASH_OCTETS – довжина hash.

Вхідними даними для обчислення компонентів ЕП є:

- базис на основі особистого ключа у вигляді матриці *B*;
- дерево *tree* для базису на основі особистого ключа;
- вхідне повідомлення *m* завдовжки *mlen*.

Вихід:

- компоненти підпису (*nonce*, s_2).

В алгоритмі вироблення компонентів ЕП застосовується функція *gen_r* для обчислення псевдовипадкового поліному *r* з коефіцієнтами за модулем *q*, та функція *sign_tree* для безпосереднього обчислення s_1, s_2 (визначені нижче).

Обчислення компонентів електронного підпису.

1. Генерація *seed* завдовжки *SEED_OCTETS* (випадкові або псевдовипадкові).

2. Генерація *nonce* завдовжки *NONCE_OCTETS* (випадкові або псевдовипадкові).

3. Ініціалізація генератору псевдовипадкових даних (*ctx*1) на основі геш-значення *H=Hash*(*nonce*||*m*). Конкретна реалізація залежить від генератору псевдовипадкових послідовностей, що застосовується.

4. Обчислення поліному *r* з застосуванням функції *r* = *gen_r* (*ctx*1).

5. Ініціалізація генератору псевдовипадкових даних (*ctx*2) для створення ЕП з застосуванням *seed*. Конкретна реалізація залежить від генератору псевдовипадкових послідовностей, що застосовується.

6. Обчислення s_1, s_2 (функція *sign_tree*)

s_1, s_2 = sign_tree (*ctx*2, *r*, *key*)

Генерація псевдовипадкового поліному *r* по модулю *q*. Функція *gen_r*.

Параметри:

n – визначає степінь полінома;

q – просте число.

Передобчислена константа:

$c = \lfloor 65536 \,/\, q \rfloor * q$.

Вхід:

ctx – контекст для генератору псевдовипадкових чисел.

Вихід:

r – псевдовипадковий поліном.

Для усіх коефіцієнтів поліному:

1. Генерація псевдовипадкових байтів (2 байта): *low*, *high*.
2. Обчислення цілого числа *temp* = *high* * 256 + *low*.
3. *if temp* < *c then*

поточний коефіцієнт полінома = *temp mod q*

else

перехід на крок 1

end if

Обчислення s_1, s_2. Функція *sign_tree*.

Параметри:

n – визначає степінь поліному;

β – граничне значення квадратичної норми для ЕП.

В функції *sign_tree* для вибірки значень застосовується функція *sampling_fft,* яка визначена нижче.

Вхід:

ctx – контекст генератору псевдовипадкових чисел;

r – псевдовипадковий поліном;

key – особистий ключ після завантаження, який містить матрицю $\hat{B}$ та дерево.

Для прискорення операцій над поліномами застосовують *FFT* формат.

Вихід:

s_1, s_2 – компоненти ЕП.

Алгоритм.

1.1 Формування псевдовипадкових даних завдовжки 64 байта за допомогою генератору з заданий контекстом (*ctx*).

1.2 Ініціалізація генератору сформованими даними (ctx2).

1.3 Обчислення вектору *t* з компонентами *t*0, *t*1, такого, що $t * \hat{B} = r$, тобто

$$\hat{B}^{-1}=\begin{bmatrix} FFT(g) & -FFT(f) \\ FFT(G) & -FFT(F) \end{bmatrix}^{-1} = \frac{1}{q}\begin{bmatrix} -FFT(F) & FFT(f) \\ -FFT(G) & FFT(g) \end{bmatrix};$$

$$\begin{bmatrix} FFT(t0) \\ FFT(t1) \end{bmatrix} = [FFT(r) \quad 0] * \frac{1}{q}\begin{bmatrix} -FFT(F) & FFT(f) \\ -FFT(G) & FFT(g) \end{bmatrix} == \frac{1}{q}\begin{bmatrix} FFT(r) * FFT(-F) \\ FFT(r) * FFT(f) \end{bmatrix};$$

$$FFT(t0) = \frac{1}{q}\, FFT(r) * FFT(-F);\; FFT(t1) = \frac{1}{q}\, FFT(r) * FFT(f).$$

4. Обчислення вектору $\begin{bmatrix} FFT(tx) \\ FFT(ty) \end{bmatrix}$ за допомогою функції *sampling_fft* (додавання випадкового шуму)

$$\begin{bmatrix} FFT(tx) \\ FFT(ty) \end{bmatrix} = sampling\_fft\ (ctx, tree, FFT(\text{t0}), FFT\ (\text{t1})).$$

5. Обчислення вектору $[FFT(t0') \quad FFT(t1')]$ з урахування шума, який внесла функція *sampling_fft*

$$[FFT(t0') \quad FFT(t1')] = \begin{bmatrix} FFT(tx) \\ FFT(ty) \end{bmatrix} * \hat{B}.$$

6. Відновлення вектору [*t*0', *t*1'] (зворотне перетворення з *FFT* формату)

$$t0' = FFT^{-1}(FFT(t0')); \qquad t1' = FFT^{-1}(FFT(t1')).$$

7. Обчислення вектору *s* (s_1, s_2): $s_1 = r - t0'$; $s_2 = 0 - t1'$.

8. Якщо квадратична норма (s_1, s_2) не перевищує β^2 (малі поліноми), закінчити, інакше перейти на крок 4.

Функція вибірки (*sampling_fft*).

Параметри:

n – степінь поліному.

Функція рекурсивна, виконується для поліномів степені *n*, *n*/2, *n*/4, …, 1, тобто загальна кількість кроків $log_2 n+1$.

Спочатку визначаються дії для усіх кроків крім останнього, а потім для останнього кроку.

Вхід:

logn – номер рівня, приймає значення $log_2 n$, $log_2 n - 1, \ldots, 0$.

ctx – контекст генератору псевдовипадкових чисел;

*t*0, *t*1 – поліноми в *FFT* форматі;

tree – гілка дерева, яка визначена поточним рівнем.

Вихід:

*t*x, *t*y – поліноми.

Для усіх рівнів крім останнього виконуються кроки 1–7.

1. Визначається степінь полінома для поточного рівня $t=2^{logn}$.

2. Поліном *t*1 степені *t* перетворюється в два поліноми *t*10 та *t*11 степені *t*/2 з урахуванням *FFT* формату. Значення *t*10, *t*11 застосовуються в якості вхідних даних для наступного рівня функції *sampling_fft*, тобто для поліномів степені *t*/2. Позначимо результат роботи функції *sampling_fft* для *t*10, *t*11 як *ty*10 та *ty*11.

3. Поліноми *ty*10 та *ty*11 порядку *t*/2 об'єднуються в один поліном порядку *t* з урахуванням *FFT* формату. Позначимо отриманий результат *ty*.

4. Коригування *t*0 з урахуванням зміни *t*1 та відповідного листа дерева для приватного ключа: *t*0' = *t*0 + (*t*1 – *t*y) * *tree*.

5. Поліном *t*0' степені *t* перетворюється в два поліноми *t*00 та *t*01 з урахуванням *FFT* формату. Значення *t*00, *t*01 – застосовуються в якості вхідних даних для наступного рівня функції *sampling_fft*, тобто для поліномів степені *t*/2. Позначимо результат роботи функції *sampling_fft* для *t*00, *t*01 як *tx*00 та *tx*01.

6. Поліноми *tx*00 та *tx*01 порядку *t*/2 об'єднуються в один поліном порядку *t* з урахуванням *FFT* формату. Позначимо отриманий результат *tx*.

7. Значення *tx*, *ty* – результат роботи функції для рівня *logn*.

Для останнього кроку (*logn* = 0, *t* = 1, поліноми *t*0, *t*1 та відповідний лист дерева містять по одному елементу) застосовується функція *sampler* (вибірка та відбракування).

1. σ := *tree* Відповідний лист дерева.

2. *tx* := *sampler*(*ctx*, *t*0, σ); Вибірка наступного елементу

3. *ty* := *sampler*(*ctx*, *t*1, σ); Вибірка наступного елементу

Значення *tx*, *ty* – результат роботи функції для рівня logn

Функція вибірки та відбраківки (sampler)

Дозволяє виконати вибірку згідно розподілу Гауса $D_{\mu,\sigma}$ для заданих μ (μ = *t*0 або μ = *t*1) та σ, яке обчислене при формуванні дерева для секретного базису.

Параметри:

σ_{min} – мінімальне значення для відхилення;

σ_{max} – максимальне значення для відхилення.

Вхід:

ctx – контекст генератору псевдовипадкових чисел;

μ – математичне очікування;

σ – середнє квадратичне відхилення.

Вихід:

z – значення, яке задовольняє розподілу Гауса.

1. Для значення μ виділяється ціла частина (*i*μ), яка не перевищує задане значення (функція floor) та дробова *d*μ (μ = *i*μ + *d*μ, *d*μ≥ 0).

2. Для *d*μ генерується випадкове *z*0, яке задовольняє розподілу Гауса для значення σ_{max}.

3. Перевіряється, можливість застосування *z*0 з урахуванням поточного значення σ. Якщо значення *z*0 треба відхилити, алгоритм повертається на крок 2.

4. Повертається значення z = *i*μ + *z*0.

3.3. Алгоритм перевірки електронного підпису

Параметри:

n – визначає степінь полінома;

q – просте число;

NONCE_OCTETS – розмір *nonce* (байтів);

β^2 – константа, яка обмежує значення норми для підпису.

Вхід:

повідомлення *m* завдовжки *mlen*;

nonce завдовжки NONCE_OCTETS;

s_2 – компонент електронного підпису;

h – відкритий ключ.

Вихід:

OK – електронний підпис правильний;

ERROR – електронний підпис не правильний.

Алгоритм перевірки ЕП зводиться до ініціалізації на основі геш-значення *H* генератору псевдовипадкових даних, обчислення поліному *r*, обчислення s_1 та перевірка, що квадратична норма s_1, s_2 не перевищує β^2.

1. Ініціалізація генератору псевдовипадкових даних (*ctx*1) на основі геш значення *H*=*Hash*(*nonce*||*m*). Конкретна реалізація залежить від генератору псевдовипадкових послідовностей, що застосовується.

2. Обчислення поліному *r* з застосуванням функції *r* = *gen_r* (*ctx*1).

3. Обчислення $s_1=(r-s_2*h) \bmod q \bmod \varphi$.

4. Перевірка, що квадратична норма s_1, s_2 не перевищує β^2 $(\sum_{i=0}^{n-1}(s1_i{}^2 + s2_i{}^2) < \beta^2)$. Якщо умова виконується, то повертає *OK*, інакше *ERROR*.

4. Аналіз потенційних атак на ЕП Falcon та «Сокіл»

На основі обґрунтованих та вибраних моделей порушника, загроз та моделі безпеки [14 – 19], основними атаками, стосовно яких повинно бути здійснено захист, є такі:

- атаки на ЕП Falcon та «Сокіл» на основі функції гешування H;
- атаки на відновлення особистого ключа з відкритого ключа ЕП Falcon та «Сокіл»;
- атаки на основі підробки ЕП Falcon та «Сокіл».

Нижче розглядаються та аналізуються вказані атаки на предмет їх складності.

4.1. Атаки на ЕП Falcon та «Сокіл» на основі функції гешування H

Перетворення GPV [34], яке застосовується в схемах ЕП Falcon та «Сокіл», вимагає щоб функція гешування H була захищена від колізій. Це означає, що розмір початкової ентропії (солі) в бітах повинен бути не меншим за 2λ, де λ – рівень безпеки, що вимагається. Але, враховуючи те, що згідно з вимогами NIST [2, 4] кількість запитів на вироблення ЕП (signature

queries) повинно бути не більшим за $q_s = 2^{64}$, а ми приймаємо також таке обмеження, тому реальне значення солі повинне бути не менше ніж

$$\lambda + \log_2(q_s). \tag{12}$$

Для 5 – 7 рівнів безпеки це дає значення, що наведені в табл. 1 [29].

Таблиця 1

Розмір (ентропія) початкового значення *r* невизначеності (солі) в бітах

Безпека	Розмір *r*	Розмір *r* з врахуванням вимог NIST
256	512	320
384	768	448
512	1024	576

Аналіз показав, що основними атаками щодо ЕП Falcon та ЕП «Сокіл» є атаки на відновлення особистого (закритого) ключа з відкритого ключа та атаки на підробку ЕП. Розглянемо ці атаки детальніше.

4.2. Атаки на відновлення особистого ключа з відкритого ключа щодо ЕП «Сокіл»

Атаки на відновлення особистого (секретного) ключа з відкритого ключа можуть зводиться до вирішення проблеми NTRU [34, 35]. У ряді схем, стійкість яких ґрунтуються на проблемі NTRU, поліноми *f, g* мають коефіцієнти з множини значень {0, 1, -1}. Це робить можливим реалізувати різні комбінаторні атаки. Наприклад, для ДСТУ 8961:2019 «Скеля» найефективнішою атакою є гібридна атака, яка знаходить частину вектора комбінаторними шляхами. Для схем ЕП Falcon та «Сокіл» такі атаки неможливі, оскільки поліноми *f, g* змінюються (точніше, семплуються) згідно з нормальним розподілом з заданими параметрами. За даного перетворення простір можливих значень поліномів збільшується настільки, що застосування комбінаторних методів стає неефективним. Залишається прямий шлях відновлення особистого ключа з відкритого засобом редукції базису решітки. При цьому, чим менші значення має норма найменшого вектора (*f, g*), тим більша криптостійкість системи. В криптосистемі Falcon поліноми генеруються над полем

$$\square_q [X] / (\phi(x)), \deg(\phi) = n \tag{13}$$

з математичним очікуванням рівним 0. Перетворення спираються на результати роботи [26], у якій детально досліджувалися можливості застосування алгоритмів семплування нормально розподілених величин. В [26] було показано, що алгоритм семплування Клейна може давати вектори розміру $\approx \sqrt{\frac{qe}{2}}$, що є дуже близьким до теоретичного мінімуму $\sqrt{q}$. Відповідно, щоб отримати такий розмір, кожний коефіцієнт отримується з розподілу з середньоквадратичним відхиленням [29]

$$\sigma' = \sqrt{\frac{qe}{2}} * \frac{1}{\sqrt{2n}}. \tag{14}$$

Найменший вектор може бути знайдений, якщо його проєкція на простір, що натягнутий на перші B векторів $b_1^*, b_2^*, ..., b_B^*$ буде менша за b_{2n-B}^*. Згідно з [26, 29] ця проєкція може бути оцінена як

$$\sqrt{\gamma_B} * \det(\Lambda_{[b_1^*,...,b_B^*]}) \approx \sqrt{\frac{3}{4}} * \sigma' \sqrt{B} = \sqrt{\frac{3}{4}} * \sqrt{\frac{qe}{2}} * \frac{1}{\sqrt{2n}} * \sqrt{B}. \tag{15}$$

Водночас, b_{2n-B}^* може бути оцінений як [29]

$$\left\|b_{2n-B}^{*}\right\| \approx GH(B)^{\frac{2n+1-2(2n-B)}{2(B-1)}} \det(\Lambda)^{\frac{1}{2n}} = GH(B)^{\frac{2B-2n+1}{2B-2}} \sqrt{q} = \left(\frac{B}{2\pi e}\right)^{\frac{2B-2n+1}{2B-2}} \sqrt{q}\,. \qquad (16)$$

Таким чином, маємо умову щоб

$$\left(\frac{B}{2\pi e}\right)^{\frac{2B-2n+1}{2B-2}} \sqrt{q} < \sqrt{\frac{3eB}{8n}} * \sqrt{q}\,. \qquad (17)$$

Далі, завдяки вибору $\sigma' = \sqrt{\frac{qe}{2}} * \frac{1}{\sqrt{2n}}$ з обох сторін рівняння маємо множник $\sqrt{q}$, який можна скоротити. Тому умова захисту від атак на відновлення особистого ключа з відкритого шляхом редукції виглядає наступним чином:

$$\left(\frac{B}{2\pi e}\right)^{\frac{2B-2n+1}{2B-2}} < \sqrt{\frac{3eB}{8n}}\,. \qquad (18)$$

4.3. Атаки на підробку ЕП «Сокіл»

Атаки на безпосередньо підробку ЕП можуть бути найбільш загрозливими. Тому іншим вектором атаки є атака підробки ЕП. При реалізації такої атаки потрібно знайти достатньо короткий вектор *s*. Відповідно, це можливо зробити, редукувавши базис так, щоб виконувалася умова [25, 26]

$$\left\|b_{1}^{*}\right\| < \beta\,. \qquad (19)$$

Причому оцінити $\left\|b_{1}^{*}\right\|$ можливо таким же чином, що і у попередньому випадку, тобто у такій послідовності:

$$\left\|b_{1}^{*}\right\| \approx GH(B)^{\frac{2n+1-2}{2(B-1)}} \det(\Lambda)^{\frac{1}{2n}} = GH(B)^{\frac{2n-1}{2B-2}} \sqrt{q} = \left(\frac{B}{2\pi e}\right)^{\frac{2n-1}{2B-2}} \sqrt{q}\,. \qquad (20)$$

Отримуємо, що умовою захисту від атак на підробку підпису є

$$\left(\frac{B}{2\pi e}\right)^{\frac{2n-1}{2B-2}} \sqrt{q} < \beta\,. \qquad (21)$$

Для практичного прийняття рішення необхідно визначитись щодо того, як обирати параметр β. Розробники ЕП Falcon пропонують використовувати значення $\sigma = 1.55\sqrt{q}$ для полінома $\phi = x^n + 1$ і $\sigma = 1.32 * 2^{1/4} * \sqrt{q}$ для полінома $\phi = x^n - x^{n/2} - 1$. Параметр β для полінома $\phi = x^n + 1$ обчислюється як

$$\beta = 1.2 * \sigma\sqrt{2nq}\,. \qquad (22)$$

Для полінома $x^n - x^{n/2} - 1$ β обчислюється таким чином:

$$\beta^2 = \frac{(1.2 * \sigma * 2n\sqrt{q})^2}{n}\,. \qquad (23)$$

Формули відрізняються тому, що для полінома $x^n - x^{n/2} - 1$ замість L2 норми обчислення виконуються за допомогою embedding norm під час генерації ключів та підпису.

Якщо підставити значення β у рівняння для оцінки захищеності від атак на підробку підпису, то також обидві сторони будуть пропорційні значенню $\sqrt{q}$. Таким чином, середньоквадратичні відхилення при семплуванні поліномів з нормального розподілу підібрані таким чином, щоб від q складність атаки не залежала. Проте на параметр q існує безліч інших обмежень, які впливають на його вибір.

Параметр q обирається згідно наступних міркувань [6, 8]:

- для захисту від алгебраїчних атак q має бути простим числом;
- якщо q буде занадто малим (порядку $q \approx n$), то будуть можливі BKW атаки;
- якщо q буде занадто великим ($q \approx n^{2.83}$), то будуть можливі атаки на підполе;
- якщо використовується поле $x^n + 1$, то для реалізації ефективного множення повинне виконуватися рівняння $q \equiv 1 \bmod 2n$;
- якщо використовується поле $x^n - x^{n/2} - 1$, то для реалізації ефективного множення повинне виконуватися рівняння $q \equiv 1 \bmod 3n$.

П р и м і т к а : Стійкість найкращого алгоритму пошуку найменшого вектору оцінюється як 2^0.292B, де B – розмір блоку при редукції. Якщо при криптоаналізі застосовувати алгоритм Гровера, то нижня оцінка класичної стійкості в 256 біт складає 2^0.265B квантової стійкості (при класичній стійкості в 256 біт). Тому, для ЕП на решітках квантова стійкість при класичній стійкості 256 біт набагато більше, ніж 128 біт.

5. Результати порівняння перспективних механізмів ЕП, що засновані на перетвореннях на алгебраїчних решітках

Оцінка та порівняння міжнародних та національних проєктів постквантових стандартів ЕП.

В табл. 2 наведено характеристики обраних для порівняння алгоритмів (значення швидкості криптоперетворень та генерації ключів наведено в тактах). В порівнянні брали участь проєкти стандартів «Вершина» та «Сокіл», а також алгоритм Dilithium, який за попередніми дослідженнями мав кращі результати серед постквантових алгоритмів підпису, що засновані на перетвореннях на алгебраїчних решітках. Стійкість алгоритмів «Вершини» 128 біт відповідає 3-му рівню стійкості NIST, 256 – 5-му, тому пропорційно для виконання порівняння згідно шкали оцінок попарного порівняння параметрам 384 був наданий 7-й рівень, а 512 – 9-й.

Таблиця 2

Характеристики алгоритмів ЕП, що засновані на перетвореннях на алгебраїчних решітках

Алгоритми	$I_{ст.}$	$l_{в.к.}$	$l_{о.к.}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$	$T_{гк.}$
Dilithium_round3_sec2	2	1 312	3 504	2 420	259 172	118 412	124 031
Dilithium_round3_sec3	3	1 952	3 856	3 293	428 587	179 424	256 403
Dilithium_round3_sec5	5	2 592	5 792	4595	538 986	279 936	298 050
Вершина_1_128 («Вершина»)	3	1 472	3 488	2 693	133 340	109 818	90 328
Вершина_1_256 («Вершина»)	5	2 624	5 792	5 345	259 103	233 712	229 669
Вершина_1_384 («Вершина»)	7	4 528	9 088	6762	411 040	398 029	317 324
Вершина_1_512 («Вершина»)	9	5 824	11 008	10708	643 744	620 989	485 471
Вершина_2_128 («Сокіл»)	3	897	4097	666	655 672	139 620	33 696 000
Вершина_2_256 («Сокіл»)	5	1 793	8193	1 280	1 338 825	285 714	107 055 000
Вершина_2_512 («Сокіл»)	9	3 585	5121	2 515	2 600 053	265 416	28 493 603 229

В табл. 3 наведено результати досліджень – відносна перевага алгоритмів ЕП, що отримана методом попарних порівнянь за кожною з характеристик.

На рис. 2 відображено гістограму загальної відносної переваги алгоритмів ЕП з урахуванням вагових коефіцієнтів характеристик.

Як видно, найбільшу перевагу має алгоритм «Вершина» з параметрами стійкості 128 біт, для більш стійких параметрів перевага вже у алгоритм «Сокіл».

В подальшому порівнювалися алгоритми, що показали кращі результати в попередньому етапі – SPHINCS+_s, «Вершина» та «Сокіл» (через те, що для різних рівнів стійкості перевага у різних алгоритмів), а також Rainbow (оптимізована реалізація зі стандартними параметрами).

Таблиця 3

Відносна перевага алгоритмів ЕП, отримана методом попарних порівнянь, за кожною з характеристик

Алгоритми	$I_{ст.}$	$l_{в.к.}$	$l_{о.к.}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$	$T_{гк.}$
Dilithium_round3_sec2	0,0198	0,1770	0,1816	0,1131	0,1583	0,1857	0,2090
Dilithium_round3_sec3	0,0299	0,0965	0,1475	0,0606	0,0849	0,0984	0,1082
Dilithium_round3_sec5	0,0697	0,0655	0,0768	0,0507	0,0666	0,0506	0,0915
Вершина_1_128 («Вершина»)	0,0299	0,1395	0,1816	0,0800	0,3006	0,2195	0,2696
Вершина_1_256 («Вершина»)	0,0697	0,0655	0,0768	0,0339	0,1583	0,0716	0,1388
Вершина_1_384 («Вершина»)	0,1453	0,0327	0,0407	0,0261	0,0975	0,0348	0,0797
Вершина_1_512 («Вершина»)	0,2681	0,0233	0,0296	0,0173	0,0479	0,0218	0,0608
Вершина_2_128 («Сокіл»)	0,0299	0,2487	0,1212	0,3211	0,0479	0,1466	0,0192
Вершина_2_256 («Сокіл»)	0,0697	0,1108	0,0467	0,1989	0,0238	0,1130	0,0146
Вершина_2_512 («Сокіл»)	0,2681	0,0406	0,0973	0,0984	0,0143	0,0581	0,0086

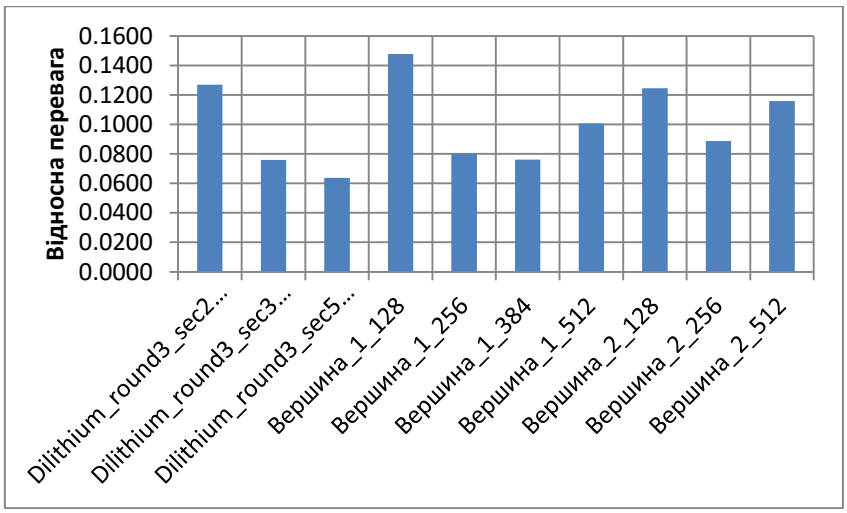

Рис. 2. Переваги алгоритмів ЕП

В табл. 4 наведено результати досліджень – відносна перевага алгоритмів ЕП, що отримана методом попарних порівнянь за кожною з характеристик.

Таблиця 4

Відносна перевага алгоритмів ЕП, отримана методом попарних порівнянь, за кожною з характеристик

Алгоритми	$I_{ст.}$	$l_{в.к.}$	$l_{о.к.}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$	$T_{гк.}$
SPHINCS+_128s	0,0155	0,2658	0,2675	0,0189	0,0090	0,0117	0,0164
SPHINCS+_192s	0,0331	0,2289	0,2304	0,0107	0,0090	0,0102	0,0139
SPHINCS+_256s	0,0794	0,1953	0,1984	0,0082	0,0090	0,0079	0,0101
Вершина_1_128 («Вершина»)	0,0331	0,0599	0,0664	0,0385	0,2036	0,1525	0,2908
Вершина_1_256 («Вершина»)	0,0794	0,0416	0,0433	0,0240	0,1340	0,0713	0,2232
Вершина_1_512 («Вершина»)	0,2596	0,0279	0,0274	0,0142	0,0619	0,0305	0,1712
RAINBOW_I_round3_avx	0,0155	0,0117	0,0120	0,2924	0,2864	0,2913	0,0960
RAINBOW_III_round3_avx	0,0331	0,0082	0,0082	0,2043	0,1148	0,1229	0,0500
RAINBOW_VI_round3_avx	0,0794	0,0064	0,0064	0,1746	0,0494	0,0438	0,0263
Вершина_2_128 («Сокіл»)	0,0331	0,0770	0,0578	0,1007	0,0619	0,1095	0,0622
Вершина_2_256 («Сокіл»)	0,0794	0,0495	0,0337	0,0683	0,0363	0,0898	0,0341
Вершина_2_512 («Сокіл»)	0,2596	0,0279	0,0486	0,0451	0,0247	0,0586	0,0057

На рис. 3 відображено гістограму загальної відносної переваги алгоритмів ЕП з урахуванням вагових коефіцієнтів характеристик.

Серед алгоритмів усіх алгоритмів кращий результат у RAINBOW_I_round3_avx (за рахунок малої довжини підпису та великої швидкодії). Але при використанні параметрів, що гарантують більшу стійкість, цей алгоритм вже на останньому місці. Якщо ж брати всі можливі параметри алгоритмів, то на першому місці «Вершина» (який в порівнянні з алгори-

тмами, що засновані на інших математичних апаратах по сукупності оцінок обійшов «Сокіл».

Тобто, якщо необхідний мінімально задовільний рівень захисту, то кращі результати у Rainbow, а в якості універсального алгоритму краще «Вершина». До того ж для «Вершини» не були представлені параметри для рівнів 1-2 NIST. Тобто, якщо б були представлені параметри для даних рівнів, то можливо такі параметри обійшли б і Rainbow.

Кращий результат у алгоритму «Вершина», друге місце у алгоритму Dilithium, а третє у «Сокіл». Але серед параметрів максимальної стійкості 512 біт вже перевага у «Сокіл».

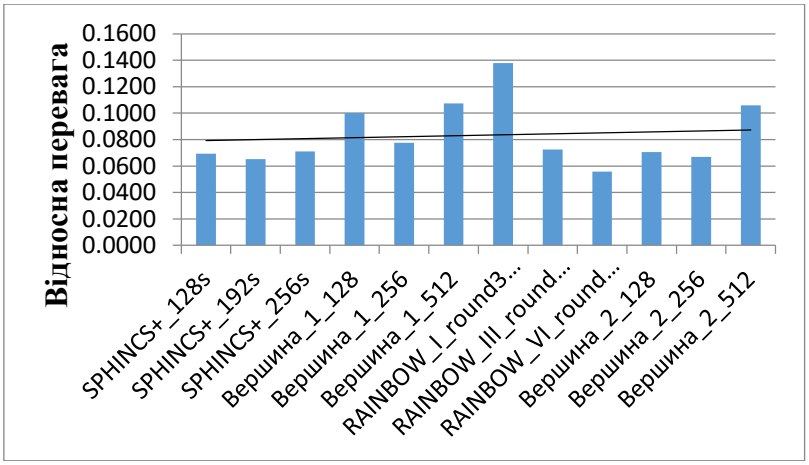

Рис. 3. Переваги алгоритмів ЕП

При порівнянні методом ранжування з'ясувалося, що кращий результат мають алгоритми, які побудовані на основі перетворень в решеті числового поля (рис. 4).

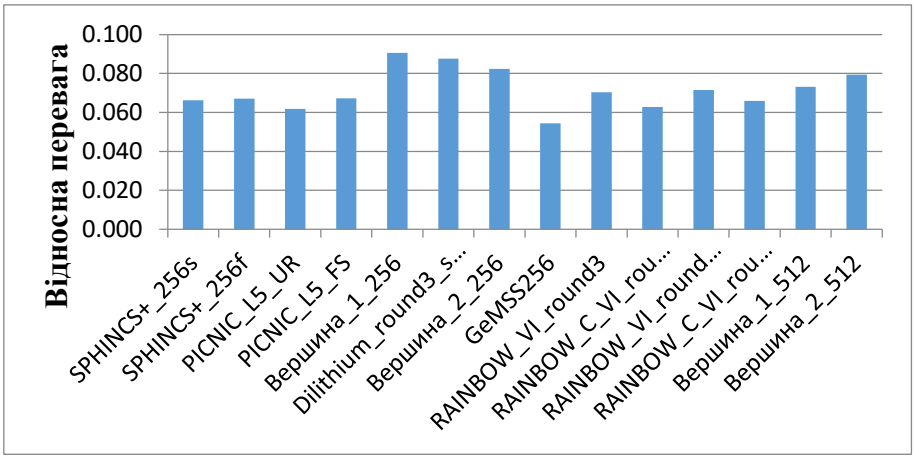

Рис. 4. Переваги алгоритмів ЕП, високого рівня захисту

Висновки

1. Механізми ЕП на решітках є основними претендентами на перемогу в конкурсі NIST PQC. Тому, їх подальший детальний аналіз та порівняння щодо основних характеристик стійкості є першочерговою задачею. Схема Falcon, як фіналіст 2-го етапу, потребує особливої уваги, оскільки має нетипову конструкцію, що використовує арифметику з плаваючою крапкою.

2. На практиці застосовано методику порівняння перспективних постквантових криптоалгоритмів ЕП. Порівнювалися алгоритми, що пройшли до 3-го етапу NIST, а також алгоритми проєктів стандарту «Вершина» та «Сокіл». При порівнянні використовувалися два методи порівняння для отримання більш точної оцінки в залежності від вимог до алгоритмів ЕП.

3. Кращі показники у алгоритмів «Вершина» та «Сокіл» («Сокіл» на даному етапі програє «Вершині» через свою низьку швидкодію, якщо його реалізація буде більш оптимізова-

на, то вже «Сокіл» буде на першому місці), в якості альтернативи можна розглядати алгоритм Rainbow. Крім того «compressed» варіант параметрів Rainbow дозволяє розглядати даний алгоритм в якості повноцінної альтернативи, навіть для систем з обмеженням до розмірів ключів. Але найменші розміри ключів у алгоритму SPHINCS+_s, який досить сильно програє по швидкодії.

4. Криптостійкість ЕП «Сокіл» як ЕП Falcon залежить від двох добре вивчених – NTRU- та SIS-проблем. Завдяки використанню семплування особистих ключів, застосуванням нормального розподілу, гібридні атаки для зламу недоцільні, а NTRU-атаки можуть зводиться до прямої атаки редукцією решітки. Також SIS-проблема, в свою чергу, може вирішуватись за допомогою редукції решітки.

5. Завдяки використанню циклотомічних поліномів розробники проєкту Falcon досягли гарної швидкодії вироблення та перевірки ЕП з використанням бінарних та тернарних дерев. Проте недоліком такого підходу є недостатня гнучкість при генерації загальносистемних параметрів. Криптостійкість здебільшого залежить від параметра n, який має або бути ступенем двійки, або невеликим кратним до ступеня двійки. Можливі значення параметра лежать у невеликій множині, що сильно обмежує вибір параметрів.

6. Криптостійкість ЕП «Сокіл» як і ЕП Falcon сильно залежить від того, наскільки малі вектори можливо семплувати з нормального розподілу. Розробники Falcon детально вивчили відомі алгоритми та обрали алгоритм Клейна, оскільки він у порівнянні з іншими алгоритмами дає найменші вектори. У подальшому дослідження та розробка нових алгоритмів семплування можуть дати можливість зменшити розміри ЕП та підвищити швидкодію. Для підвищення стійкості в ній при криптоперетвореннях використовується вибірка Гауса, що розроблена Престом, Рікоссетом та Россі. Причому, вибірка з відхиленням ретельно налаштована з параметрами таким чином, що спостереження за швидкістю відхилення не дає при усіх параметрах безпеки жодної статистично корисної інформації щодо особистих ключів.

7. Оцінки криптостійкості отримані при зведенні відповідних решіток для розміру блоку B, як 0,265B та 0,292B. Такий підхід вважається класичним і використовувався як авторами ЕП Dilithium, так і авторами ЕП Falcon.

8. Для отримання оцінок щодо складності задач криптоаналізу SIS та NTRU було використано зведення проблеми до редукції решіток. Причому через недостатню гнучкість механізму ЕП «Сокіл», як і для ЕП Falcon, при параметрах 384 біт класичної та 192 біт квантової стійкості. В подальшому такий режим був знятий, і як наслідок поліном $x^n–x^{n/2}–1$ в реалізації ЕП «Сокіл» попередньо не використовується. Але необхідно провести аналіз можливостей реалізації і цього варіанту побудови ЕП.

9. До основного проблемного питання щодо недоліку ЕП Falcon необхідно віднести використання арифметики з плаваючою крапкою. Разом з використанням деревоподібних структур це ускладнює аналіз схеми до атак сторонніми каналами. Іншою проблемою є складність реалізації на малоресурсних пристроях.

10. Обмеження щодо ЕП Falcon на 3-му етапі міжнародного конкурсу NIST США, на наш погляд, пов’язані зі складністю обчислення загальносистемних параметрів, а також із суттєвим впливом їх збільшення на швидкодію ЕП. Тобто, для безпечного використання ЕП Falcon повинні бути знайдені набори загальносистемних параметрів, за яких забезпечується стійкість до всіх відомих та потенційних атак, тобто класичних, квантових, на основі помилок та спеціальних атак.

11. На перспективу доцільним є забезпечення не менше 384 і 512 біт безпеки проти класичного криптоаналізу та не менше 192 і 256 біт безпеки проти квантового криптоаналізу. Але, як показали дослідження, як з точки зору теорії, так і практики, генерація загальносистемних параметрів для використання 384 і 512 біт безпеки проти класичного криптоаналізу та 192 і 256 біт безпеки проти квантового криптоаналізу, є важливою задачею.

Список літератури:

1. Neal Koblitz, Alfred J. Menezes A Riddle Wrapped in an Enigma. [Електронний ресурс]. Режим доступу: https://eprint.iacr.org/2015/1018.pdf.

2. Lily Chen Report on Post-Quatum Cryptography. NISTIR 8105 (DRAFT) / Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone. Режим доступу: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.

3. NIST IR 8240 Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. Режим доступу: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

4. NIST IR 8309 Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. Режим доступу: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

5. Квантовые компьютеры. [Електронний ресурс]. Режим доступу: http://www.nkj.ru/archive/articles/5309/.

6. 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.

7. Горбенко І. Д. Постквантова криптографія та механізми її реалізації / І. Д. Горбенко, О. О. Кузнецов, О. В. Потій, Ю. І. Горбенко, Р. С., Ганзя, В. А. Пономар // Радіотехніка. 2017. Вып. 186. С. 32–52.

8. Каптьол Є. Ю. Аналіз можливостей та особливості програмування задач криптології на квантовому компютері / Є. Ю. Каптьол, І.Д. Горбенко // Радіотехніка. 2020. Вип. 202. С. 37-48.

9. IBM Quantum breaks the 100-qubit processor barrier. IBM Research Blog. [Електронний ресурс]. Режим доступу: https://research.ibm.com/blog/127-qubit-quantum-processor-eagle.

10. IBM's roadmap for scaling quantum technology. IBM Research Blog. [Електронний ресурс]. Режим доступу: https://research.ibm.com/blog/ibm-quantum-roadmap.

11. First quantum computer to pack 100 qubits enters crowded race. [Електронний ресурс]. Режим доступу: https://www.nature.com/articles/d41586-021-03476-5.

12. IBM claims advance in quantum computing. BBC News. Paul Rincon. [Електронний ресурс]. Режим доступу: https://www.bbc.com/news/science-environment-59320073.

13. D-Wave plans to build a gate-model quantum computer. TechCrunch. Frederic Lardinois. [Електронний ресурс]. Режим доступу: https://techcrunch.com/2021/10/05/d-wave-plans-to-build-a-gate-model-quantum-computer/

14. Горбенко Ю. І. Модель порушника систем електронних цифрових підписів в умовах квантового криптоаналізу / Ю. І. Горбенко, О. В. Шевцов, Т. Ю. Кузнецова // Радіотехніка. 2016. Вип. 186. С. 53-69.

15. Горбенко Ю. І. Моделі загроз щодо асиметричних криптоперетворень перспективного електронного підпису / Ю. І. Горбенко, М. В. Єсіна, В. В. Онопрієнко, Г. А. Малєєва // Радіотехніка. 2020. Вип. 202. С. 72-78.

16. 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.

17. EUF-CMA and SUF-CMA. [Електронний ресурс]. Режим доступу: https://blog.cryptographyengineering.com/euf-cma-and-suf-cma.

18. Yesina M. Comparative Analysis of Key Encapsulation Mechanisms / Maryna Yesina, Mikolaj Karpinski, Volodymyr Ponomar, Yurij Gorbenko, Tomasz Gancarczyk, Uliana Iatsykovska // Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). September 18-21, 2019, Metz, France. Vol. P. 7-12.

19. Єсіна М. В. Моделі безпеки постквантових криптографічних примітивів // Міжнар. наук. симпозіум "Питання оптимізації обчислень (ПОО-XLVI)", 2019 р. Математичне на комп'ютерне моделювання. Сер.: Технічні науки. Вип. 19. С. 49-55.

20. Наказ «Про встановлення вимог з безпеки та захисту інформації до кваліфікованих надавачів електронних довірчих послуг та їхніх відокремлених пунктів реєстрації», №269 від 14.05.2020 р. Режим доступу: https://zakon.rada.gov.ua/laws/show/z0668-20#Text.

21. ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння». Режим доступу: https://dbn.co.ua/load/normativy/dstu/4145/5-1-0-1798.

22. ДСТУ ISO/IEC 14888-3:2019 «Інформаційні технології. Методи захисту. Цифрові підписи з доповненням. Ч. 3. Механізми, що ґрунтуються на дискретному логарифмі». Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=83556.

23. Dilithium 3. ЕП Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehlé Crystals-Dilithium: Algorithm Specifications and Supporting Documentation. Режим доступу: https://pq-crystals.org/dilithium/data/dilithium-specification.pdf.

24. Олексійчук А. М. Обґрунтування перспективного постквантового національного стандарту електронного підпису на основі решіток / А. М. Олексійчук, В. А. Кулібаба, М. В. Єсіна, С. О. Кандій, Є. В. Острянська, І. Д. Горбенко // Радіотехніка. 2020. Вип. 200 С. 5–14.

25. Ducas L. et al. Crystals-Dilithium: digital signatures from module lattices. Режим доступу: https://cryptojedi.org/papers/dilithium-20170617.pdf.

26. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU Specification v1.2 – 01/10/2020. Pierre-Alain Fouque Jeffrey Hoffstein Paul Kirchner. [Електронний ресурс]. Режим доступу: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

27. Горбенко І. Д. Основні положення та результати порівняння властивостей електронних підписів постквантового періоду на алгебраїчних решітках / І. Д. Горбенко, О. Г. Качко, О. В. Потій, А. М. Олексійчук, Ю. І. Горбенко, М. В. Єсіна, І. В. Стельник, В. А. Пономар // Радіотехніка. 2021. Вип. 205 С. 5-21.

28. Rainbow Signature. Режим доступу: https://www.pqcrainbow.org/.

29. Горбенко І. Д. Генерація загальносистемних параметрів для криптосистеми Falcon для 256, 384, 512 біт безпеки / І. Д. Горбенко, С. О. Кандій, М. В. Єсіна, Є. В. Острянська // Радіотехніка. 2020. Вип. 202. С. 57-63.

30. Горбенко І. Д. Методи обчислення системних параметрів для електронного підпису «Crystals-Dilithium» 128, 256, 384 та 512 біт рівнів безпеки / І. Д. Горбенко, А. М. Олексійчук, О. Г. Качко, Ю. І. Горбенко, М. В. Єсіна, С. О. Кандій // Радіотехніка. 2020. Вип. 202 С. 5-27.

31. Горбенко Ю. І. Аналіз стійкості постквантового електронного підпису Dilithium до атак на помилки / Ю. І. Горбенко, О. С. Дроздова // Радіотехніка. 2020. Вип. 202 С. 49-56.

32. Закон України «Про основні засади забезпечення кібербезпеки України» (Відомості Верховної Ради (ВВР), 2017, № 45, ст. 403). Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#Text.

33. Горбенко І. Д. Прикладна криптологія : монографія ; вид. 2-ге / І. Д. Горбенко, Ю. І. Горбенко // Харків : Форт, 2012. 868 с.

34. Craig Gentry Trapdoors for hard lattices and new cryptographic constructions / Craig Gentry, Chris Peikert, Vinod Vaikuntanathan // Richard E. Ladner and Cynthia Dwork, editors, 40th ACM STOC, p. 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

35. Léo Ducas Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures / Léo Ducas, Phong Q. Nguyen // Wang and Sako, p. 433–450.

Надійшла до редколегії 11.10.2021

Відомості про авторів:

Горбенко Іван Дмитрович – д-р техн. наук, професор, Харківський національний університет імені В.Н. Каразіна, професор кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, головний конструктор АТ «Інститут інформаційних технологій»; Україна; e-mail: GorbenkoI@iit.kharkov.ua; ORCID: https://orcid.org/0000-0003-4616-3449

Качко Олена Григорівна – канд. техн. наук, Харківський національний університет радіоелектроніки, професор кафедри програмної інженерії, факультет комп'ютерних наук, начальник відділу програмування АТ «Інститут інформаційних технологій», Україна, e-mail: iit@iit.kharkov.ua, ORCID: https://orcid.org/0000-0001-9249-0497

Потій Олександр Володимирович – д-р техн. наук., професор, заступник Голови Державної служби спеціального зв'язку та захисту інформації; Україна; e-mail: potav@ua.fm; ORCID: https://orcid.org/0000-0002-2366-0541

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

Пономар Володимир Андрійович – канд. техн. наук, Харківський національний університет імені В.Н. Каразіна, науковий співробітник кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: Laedaa@gmail.com; ORCID: https://orcid.org/0000-0001-5271-2251

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

Стельник Ігор Валерійович – Адміністрація Державної служби спеціального зв'язку та захисту інформації України; директор Департаменту.

Кандій Сергій Олегович – АТ «Інститут інформаційних технологій», технік-конструктор; Україна; e-mail: sergeykandy@gmail.com

Кузнецова Катерина Олександрівна – Харківський національний університет імені В.Н. Каразіна; студентка; Україна; e-mail: kate7smith12@gmail.com