

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)

Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему Дослідження алгоритмів цифрового підпису різного призначення
(назва теми)

Виконав(ла): студент(ка) ІІ курсу,
групи БКЗ-813м

Спеціальності 125 Кібербезпека та захист інформації

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Безпека інформаційних і комунікаційних систем

АБГАРЯН Ю.С.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЗІНА Г.Л.

(ПРИЗВИЩЕ та ініціали)

Рецензент МОРОЗ Г.В.

(ПРИЗВИЩЕ та ініціали)

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій

Кафедра інформаційної безпеки та наноелектроніки

Ступінь вищої освіти магістр

Спеціальність 125 Кібербезпека та захист інформації

(код і найменування)

Освітня програма (спеціалізація) Безпека інформаційних і комунікаційних систем

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ІБтаН

Андрій КОРОТУН

«___» _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

АБГАРЯНА Юри Серьожайовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження алгоритмів цифрового підпису різного
призначення

Research into digital signature algorithms for various purposes

керівник проєкту (роботи) к.ф.-м.н., доцент КОЗІНА Галина Леонідівна,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «10» грудня 2024 року №522

2. Строк подання студентом проєкту (роботи) 10.12.2024

3. Вихідні дані до проєкту (роботи) алгоритми постквантової криптографії

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Огляд алгоритмів цифрових підписів; алгоритми цифрового підпису в
постквантової криптографії

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень,
кількість слайдів, плакатів)

Презентація доповіді (в MS PowerPoint), 17 слайдів.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1 – 2	КОЗИНА Г.Л., доцент кафедри ІБтаН	02.09.2024	05.12.2024
Нормоконтроль	КОРОЛЬКОВ Р. Ю., доцент кафедри ІБтаН		09.12.2024

7. Дата видачі завдання «02» вересня 2024року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1.	Збір та аналіз інформації про алгоритми цифрових підписів	02.09.24 – 16.09.24	виконано
2.	Систематизація літературних даних	17.09.24 – 23.09.24	виконано
3.	Складання і затвердження наукового завдання	24.09.24 – 29.09.24	виконано
4.	Формування та уточнення наукового завдання	30.09.24 – 05.10.24	виконано
5.	Аналіз продуктивності алгоритмів постквантової криптографії у механізмі цифрового підпису	06.10.24 – 31.10.24	виконано
6.	Оформлення графічної частини	01.11.24 – 19.11.24	виконано
7.	Оформлення ПЗ	20.11.24 – 30.11.24	виконано

Студент(ка)

Юра АБГАРЯН

(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Галина КОЗИНА

(підпис) (Ім'я ПРИЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до магістерської роботи: 50 с., 1 табл., 15 рис., 29 джерел.

ВІДКРИТИЙ КЛЮЧ, ПОСТКВАНТОВА КРИПТОГРАФІЯ, СЕКРЕТНИЙ КЛЮЧ, ХЕШ-ФУНКЦІЯ, ЦИФРОВИЙ ПІДПИС

Мета роботи: дослідження продуктивності генерації пари ключів для цифрового підпису за допомогою алгоритмів пост квантової криптографії.

Об'єкт та предмет дослідження: об'єктом дослідження є алгоритми цифрового підпису; предметом дослідження є продуктивність постквантових алгоритмів у механізмі цифрового підпису.

Методи дослідження: описово-аналітичний.

Результати: результатом дослідження є визначення продуктивності постквантових алгоритмів при генерації пари ключів для цифрового підпису.

Рекомендації щодо впровадження: робота носить прикладний характер, а її результати можуть знайти застосування для вибору найкращих алгоритмів для цифрових підписів.

Практична цінність: визначено середній час підписання файлів різного розміру за допомогою різних алгоритмів.

ABSTRACT

Explanatory note to the master's thesis: 50 p., 1 table, 15 figure, 29 sources.

**PUBLIC KEY, POST-QUANTUM CRYPTOGRAPHY, SECRET KEY,
HASH FUNCTION, DIGITAL SIGNATURE**

The purpose of the work: to study the performance of generating a key pair for a digital signature using post-quantum cryptography algorithms.

Object and subject of research: the object of the study is digital signature algorithms; the subject of the study is the performance of post-quantum algorithms in the digital signature mechanism.

Research methods: descriptive-analytical.

Results: the result of the study is to determine the performance of post-quantum algorithms when generating a key pair for a digital signature.

Recommendations for implementation: the work is of an applied nature, and its results can be used to select the best algorithms for digital signatures.

Practical value: the average time for signing files of different sizes using different algorithms has been determined.

ЗМІСТ

	С.
Вступ.	7
1 Огляд алгоритмів цифрових підписів.	9
1.1 Визначення поняття цифрового підпису.	9
1.2 Властивості цифрового підпису.	12
1.3 Інфраструктура відкритих ключів.	14
1.4 Електронний цифровий підпис: криптографія з відкритим ключем .	17
1.5 Технології цифрового підпису на практиці.	19
2 Алгоритми цифрового підпису в постквантовій криптографії.	24
2.1 Цифровий підпис і сучасна криптографія.	24
2.2 Алгоритми постквантової криптографії.	26
2.3 Реалізація алглритмів постквантової криптографії.	29
2.4 Аналіз ефективності алгоритмів.	33
2.5 Результати вимірювань.	35
Висновки.	45
Перелік джерел посилення.	47

ВСТУП

Завдяки використанню Інтернету в поєднанні з можливостями комп'ютерних технологій можна зробити комп'ютери невеликими за розміром, але вони можуть обробляти дані швидше і більше. Швидкий розвиток технологій сьогодні може спростити різні роботи, однією з яких є використання цифрових підписів.

Отже виникла можливість підписати багато файлів або документи за допомогою підключених електронних пристроїв, таких як смартфони або комп'ютери, які підключені до мережі Інтернет. Такі прориви, безумовно, дуже корисні в таких роботах, які вимагають великої кількості підписів документів або листування. Ця умова стала можливою під час пандемії Covid-19 завдяки виконанню віддаленої роботи.

Компанії роблять багато міжнародних транзакцій та вимагають підписання файлу, тому знаходження працівників у місці зберігання файлів більше не є обов'язковим.

Різноманітні банківські ризики при використанні Інтернету справді потребують передбачення та альтернативних рішень для подолання ризику невизначеності що виникає. Існування електронного підпису (е-підпису) є однією з відповідей на ці ризики. Електронний підпис є одним із засобів аутентифікації, технології, яка вважається достатньо безпечною для заміни звичайного підпису.

В даний час цифровий підпис є частиною повідомлення, яке підтверджує правильне джерело та вказує, що повідомлення не змінилося в процесі підписання документа. Дані зашифровані унікальним чином і ідентифікують відправника, забезпечуючи належну цілісність документа, оскільки технічно лише одержувачі з правильним програмним забезпеченням

можуть читати код підпису. Таким чином, одержувач документа с цифровим підписом може гарантувати, що документ не змінився на шляху до нього.

В зв'язку з вищенаведеним, проблема надання можливостей цифровим підписам протистояти новим загрозам, зокрема пов'язаним із потенціалом квантових комп'ютерів, є актуальною.

1 ОГЛЯД АЛГОРИТМІВ ЦИФРОВИХ ПІДПИСІВ

1.1 Визначення поняття цифрового підпису

Термін цифровий підпис охоплює безліч типів «підписів». Електронні підписи – це просто електронне підтвердження особи. Це визначення завідомо достатньо широке, щоб охоплювати всі види електронної ідентифікації, від біометричних підписів, таких як сканування райдужної оболонки ока та відбитків пальців, до небіометричних підписів, таких як цифрові підписи.

Електронні підписи часто поділяють на високо захищені та незахищені. Цифровий підпис має виконувати єдині основні функції, які ми очікуємо від документів, підписаних власноручними підписами, а саме цілісність, незаперечність, автентифікацію та конфіденційність. Незаперечність є важливою для транзакцій електронної комерції, особливо якщо контракти укладаються в електронному вигляді. Процес цифрового підпису починається з отримання математичного коду (званого хеш-кодом) чека. Цей хеш-код може бути унікальним цифровим відбитком чека. Якщо хоч трохи змінити чек, хеш-код різко зміниться. Наступним етапом створення цифрового підпису є підпис хеш-коду разом із вашим закритим ключем. Цей підписаний хеш-код потім додається до чека.

Як діє такий підпис? Одержувач вашого чека може перевірити надісланий вами хеш-код, використовуючи ваш відкритий ключ [1,2]. У той самий час з отриманого чека створюється абсолютно новий хеш-код, який порівнюється з першим підписаним хеш-кодом. Якщо хеш-коди схожі, то одержувач переконався, що чек не було змінено. Одержувач також знає, що лише ви надішлете чек, оскільки лише ви маєте закритий ключ, який підписав перший хеш-код.

Оскільки цифровий підпис має запропонувати автентифікацію джерела даних (і невідомність), зрозуміло, що сам цифровий підпис має бути частиною інформації, яка залежить від повідомлення, і не може бути повністю окремим

ідентифікатором. Його можна надіслати як окрему частину інформації в повідомленні, але його обчислення має включати повідомлення. Секретний параметр знає лише підписувач.

Внаслідок того, що цифровий підпис має забезпечувати неспростовність, його обчислення має включати секретний параметр, відомий лише підписувачу. Єдиним можливим винятком із цього правила є те, що протилежна особа повністю довіряє всім сторонам, залученим до підписання та перевірки цифрових підписів.

Цифрові підписи дозволяють людям підписувати цифрові документи, надаючи властивості власноручного підпису. Вони повинні заповнити п'ять переконливих атрибутів рукописних підписів, перерахованих у [3]. Власноручні підписи є автентичними, непробачними, не підлягають повторному використанню, незмінними та не можуть бути скасовані у випадку власноручних підписів. І підпис, і, отже, документ є фізичними речами, що ускладнює підписувачу заявити, що підпис не є його власним. Щоб створити безпечну схему електронного підпису, ці атрибути повинні бути задоволені.

Технології електронного підпису містять PIN-коди, ідентифікатори користувачів і паролі, цифрові підписи, оцифровані підписи, а також апаратні та біометричні маркери. Тому необхідно розрізняти електронні та цифрові підписи. Цифрові підписи — це підмножина технологій електронного підпису, які використовують ключі та криптографічні алгоритми для підписання документів.

Рисунок 1.1 ілюструє використання PKI для генерації цифрових підписів, далі наведено приклад сценарію цифрового підпису. Боб (відправник) хоче надіслати Алісі (одержувач) текстове повідомлення з цифровим підписом. Спочатку Боб читає текстове повідомлення, яке потрібно підписати, і перетворює хешоване повідомлення за допомогою функції дайджесту повідомлення (наприклад, MD5, SHA1 тощо). Функція дайджесту

повідомлення може бути функцією, яка генерує 162-бітовий хеш початкового повідомлення; цей хеш не може мати звичку повторно генерувати перше повідомлення. Тому хешоване повідомлення безпечне та унікальне. Коли Боб отримує хешоване повідомлення, він використовує алгоритм цифрового підпису загального відкритого ключа та свій закритий ключ, щоб підписати хеш, щоб отримати цифровий підпис для конкретного документа.

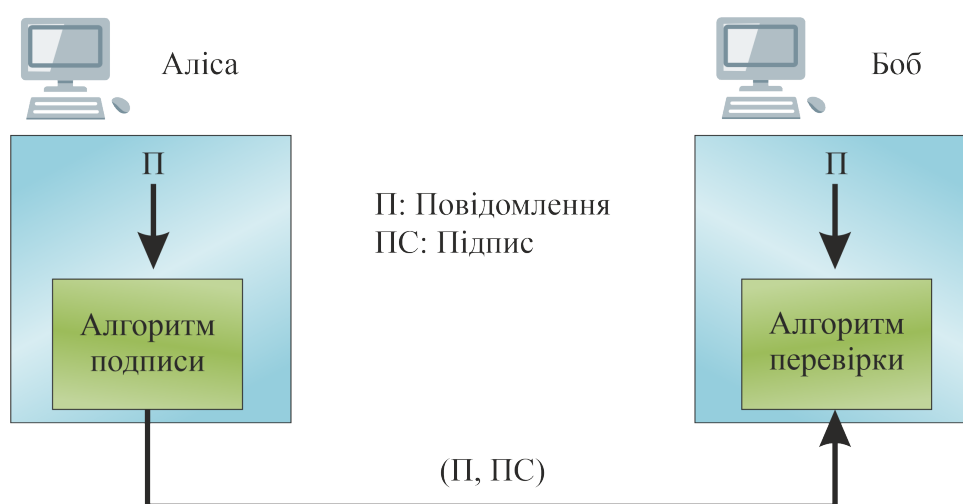


Рисунок 1.1 – Процес цифрового підпису

Коли Аліса отримає цифровий підпис і, отже, відповідне текстове повідомлення, їй доведеться обчислити два окремих значення. Спочатку хешоване повідомлення отриманого тексту обчислюється за ідентичним алгоритмом хешування. Тоді, коли буде отримане хеш-значення, вона використовуватиме алгоритм дешифрування з відкритим ключем і цифровим підписом Боба, щоб отримати підписаний хеш. Якщо Аліса зможе розшифрувати цифровий підпис, це говорить про те, що приватний ключ Боба був використаний для шифрування хешованого повідомлення. Остаточним кроком для Аліси є свірка хешу, який вона обчислила, з хешем, який вона отримала в процесі дешифрування. Якщо ці два хешованих повідомлення збігаються, це означає, що вона отримала перше повідомлення, яке підписав

Боб (таким чином зберігаючи цілісність повідомлення). Створення та розповсюдження ключів є найбільшими проблемами при розгортанні PKI. Відповідь полягає в тому, щоб використовувати довірений центр сертифікації (ЦС), який називається Certification Authority (CA) у PKI. ЦС може бути довіреною організацією, яка приймає заявки на сертифікати від організацій, автентифікує програми, видає сертифікати користувачам і пристроям у PKI, а також підтримує та надає інформацію про статус сертифікатів. Якщо СА керує великою, географічно розсіяною популяцією, він використовує місцеві органи реєстрації (LRA), які забезпечують прямі фізичні контакти з суб'єктами [4].

1.2 Властивості цифрового підпису

Щоб бути дійсними, цифрові підписи вимагають таких властивостей [5]:

Автентичність: звуковий підпис передбачає, що підписувач навмисно підписав пов'язане повідомлення.

Непідробність: лише підписувач може надати дійсний підпис для пов'язаного повідомлення.

Неможливість повторного використання: підпис документа не можна використовувати на іншому документі.

Невідмова: підписувач не може заперечити, що підписав документ, який містить дійсний підпис.

Цілісність: треба переконатися, що вміст не було змінено.

Таким чином, цифровий підпис відтворює бажані характеристики власноручного підпису та пропонує ще більш надійні типи автентифікації, якщо під час обробки секретної інформації дотримуються правильних процедур. Наприклад, власноручні підписи можуть бути перевірені лише

експертами, які практикують те, що можна назвати лише неточною наукою. Навпаки, математична процедура для перевірки цифрових підписів перевіряється будь-якою кількістю незалежних агентів, і між цими агентами немає місця для розбіжностей. Цифри або правильні, або ні.

Зараз існує як мінімум два поширених методи генерації цифрових підписів. Один з них може бути запатентованою технологією RSA Data Security Inc. і названа цифровим підписом RSA на честь її винахідників Ріввеста, Шаміра та Адлемана. Ця технологія розгортається в дуже значній кількості додатків, і її ліцензували майже всі великі телекомунікаційні та комп'ютерні компанії в США, включаючи AT&T, Apple, IBM, Microsoft, Novell і Sun Microsystems. єдиним серйозним конкурентом підписів RSA є алгоритм цифрового підпису (DSA), який був запропонований NIST як федеральний стандарт інформаційної науки під назвою Стандарт цифрового підпису (DSS).

Підписи RSA та DSA використовують подібні типи обчислень, і вони можуть виконуватися апаратними та/або програмними рішеннями. Вони покладаються на поняття «центру сертифікації ключів» (CA), який відповідає за видачу та/або сертифікацію ключів. Перша роль центру сертифікації ключів полягає в тому, щоб переконатися, що відкритий ключ користувача точний. ЦС не обов'язково бути онлайн, щоб відповісти на запитання щодо законності ключів. Натомість, коли користувачеві видається (або вибирається та реєструється) його пара відкритих/приватних ключів, центр сертифікації просто видає цифровий підпис цих ключів, щоб засвідчити, що вони повинні бути визнані такими, що були видані фактичному користувачеві. Ці облікові дані, як правило, мають термін дії та повинні передавати іншу інформацію, наприклад, завдання, на яке користувач має сертифікат. Коли цифровий підпис створюється користувачем, облікові дані для ключів, які звикли створювати цифровий підпис, також можуть бути включені до цифрового підпису (але це не обов'язково). Користувач може отримати кілька облікових даних для свого

відкритого ключа, і користувачеві не потрібно отримувати кілька пар відкритих/приватних ключів для кількох програм. Зокрема, не повинно бути необхідності, щоб пара відкритий/приватний ключ користувача не могла використовуватися для підписання документів про стан здоров'я, його електронної пошти або транзакцій його платіжної картки. Для користувача, який отримує доступ до системи, що належить даній лікарні, він отримає облікові дані для свого відкритого ключа від самої лікарні. Щоб отримати доступ до системи в ролі медсестри, вони захочуть надати облікові дані для ідентичних ключів, які були видані агентством з акредитації для медсестер [7-9].

Окрім базової сертифікації ключів за допомогою цифрових підписів за власним відкритим ключем центру сертифікації, центр сертифікації ключів може робити:

- Побудову та видачу персональних та відкритих ключів для користувачів.
- Повторну сертифікація ключів, термін дії сертифікатів яких закінчився.
- Видачу списків відкликаних ключів для зламаних ключів.

Немає причин володіти секретними ключами користувача, які зберігаються на машині центру сертифікації, і є вагомими причини для того, щоб вони там не зберігалися.

1.3 Інфраструктура відкритих ключів

Інфраструктура відкритих ключів (PKI) підтримує розповсюдження та ідентифікацію відкритих ключів шифрування, дозволяючи користувачам і

комп'ютерам як безпечно обмінюватися даними через мережі, так само як і через Інтернет, і перевіряти особу альтернативної сторони [3].

Без РКІ конфіденційну інформацію все ще можна зашифрувати (забезпечуючи конфіденційність) і обмінюватися, але не було б гарантії ідентифікації (автентифікації) альтернативної сторони. Будь-який тип конфіденційних даних, якими обмінюються в Інтернеті, залежить від безпеки РКІ.

Типовий РКІ складається з апаратного забезпечення, програмного забезпечення, політик і стандартів для керування створенням, адмініструванням, розповсюдженням і відкликанням ключів і цифрових сертифікатів. Цифрові сертифікати знаходяться всередині РКІ, оскільки вони підтверджують особу суб'єкта сертифіката та прив'язують цю особу до загального відкритого ключа, що міститься в сертифікаті.

Типовий РКІ включає такі ключові елементи:

- Довірена сторона, яка називається центром сертифікації (СА), діє через корінь довіри та надає послуги, які підтверджують автентичність людей, комп'ютерів та інших об'єктів.
- Реєстраційний орган, який часто називають підпорядкованим ЦС, сертифікований кореневим ЦС для видачі сертифікатів для певного використання, дозволеного основою.
- База даних сертифікатів, яка зберігає запити на сертифікати, видає та скасовує сертифікати.
- Сховище сертифікатів, яке знаходиться на сусідньому комп'ютері як область для зберігання виданих сертифікатів і особистих ключів.

ЦС видає цифрові сертифікати юридичним і фізичним особам після перевірки їхньої особи. Він підписує ці сертифікати за допомогою свого закритого ключа; його відкриті ключі, доступні для будь-якої або всіх зацікавлених сторін під час самопідписання сертифіката ЦС. Центри

сертифікації використовують цей довірений кореневий сертифікат для створення «ланцюжка довіри» — багато корневих сертифікатів вбудовано у веб-браузери, щоб вони мали вбудовану довіру цих ЦС. Веб-сервери, клієнти електронної пошти, смартфони та багато іншого апаратного та програмного забезпечення також підтримують PKI та містять надійні кореневі сертифікати від ключових ЦС.

Включаючи відкритий ключ суб'єкта чи фізичної особи, цифрові сертифікати включають інформацію про алгоритм, за допомогою якого створено підпис, ідентифіковану особу чи організацію, цифровий підпис ЦС, який перевірів дані суб'єкта та видав сертифікат, мету остаточного шифрування відкритого ключа. , підпис і підписання сертифіката, подібно до діапазону дат, протягом якого сертифікат вважатиметься дійсним. Орган сертифікації має бути загальновідомим і надійним і повинен мати чітко визначений процес ідентифікації перед видачею сертифіката. СА надає онлайн-доступ до всіх виданих сертифікатів і надає онлайн-доступ до списку відкликаних сертифікатів. ЦС відображає в Інтернеті ліцензію, видану Контролером, і відображає в Інтернеті схвалену заяву про практику сертифікації (CPS). Він має відповідати закону/правилам/положенням і інструкціям про IT

Приватний ключ СА або ССА вимагає найвищого рівня безпеки. Тут для зберігання закритого ключа використовується апаратний модуль безпеки (HSM). Для підпису потрібна більше ніж одна особа. HSM розміщується в камері зберігання з відеоспостереженням 24x7 [10].

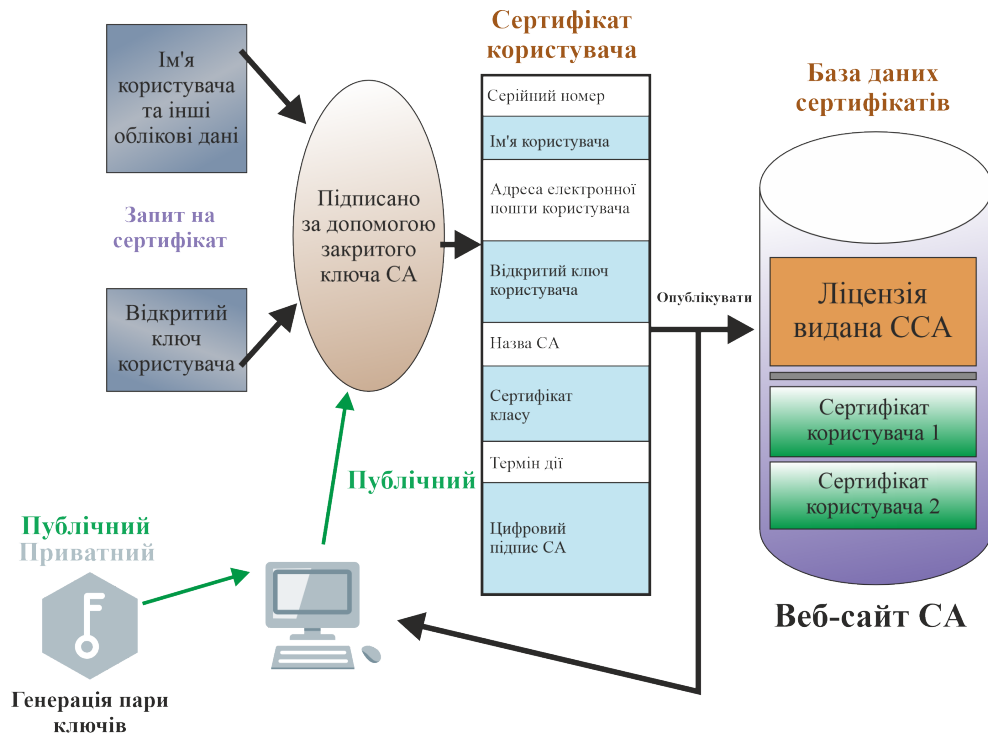


Рисунок 1.2 – Орган сертифікації

1.4 Електронний цифровий підпис: криптографія з відкритим ключем

Як і всі інші методи цифрового підпису також мають свої переваги та недоліки.

Нижче наведено переваги криптографії з симетричним ключем

Швидкість: використовуючи DS business, не потрібно чекати, поки паперові документи будуть надіслані будь-якими поштовими службами. Контракти пишуться, завершуються та підписуються всіма зацікавленими сторонами за короткий проміжок часу.

Витрати: передача по мережі дешевша за поштові послуги. А якщо це робиться за допомогою цифрового підпису, то набагато дешевше, ніж інші.

Безпека: використання цифрових підписів і електронних документів зменшує ризики декодування, читання, видалення або зміни документів під час передачі.

Невідмовність: передача електронного документа в цифровому вигляді ідентифікує вас як підписанта, і це не може бути пізніше спростовано.

Запобігання самозванцям: жодна інша особа не може закріпити ваш цифровий підпис або надіслати електронний документ, неправильно оскаржуючи його підписання вами.

Мітка часу: за допомогою мітки часу ваших цифрових підписів ви отримаєте правильний час підписання документів.

Автентичність: і паперовий штамп, і цифровий штамп мають однакове значення автентичності.

Нижче наведено недоліки цифрового підпису:

Термін дії: цифрові підписи також схожі на інші електронні носії, і ми всі знаємо, що кожен з них має обмежений час. Таким чином, це показує, що DS також надходить із закінченням терміну дії.

Сертифікати: для ефективного використання цифрового підпису як відправник, так і отримувач повинні придбати авторизовані сертифікати.

Програмне забезпечення: і відправнику, і одержувачу також необхідно придбати авторизоване програмне забезпечення, щоб зробити передачу плавнішою та легшою.

Закон: у деяких штатах і країнах заповіді щодо проблем, створених комп'ютером і технологіями, є слабкими або взагалі відсутні. Обмін у таких юрисдикціях стає дуже ризикованим для тих, хто використовує електронні документи з цифровим підписом.

Сумісність: під час використання цифрового підпису на різних платформах також виникає багато проблем із сумісністю.

Процес генерації та процес перевірки цифрового підпису вимагає значної кількості часу. Отже, для регулярного обміну повідомленнями швидкість зв'язку зменшиться.

Якщо користувач змінює свій приватний ключ після кожної фіксованої перерви періоду, тоді записи про всі ці зміни повинні бути зарезервовані. Якщо виникає суперечка щодо раніше надісланого повідомлення, то потрібно звернутися до старої пари ключів. Таким чином, завантаження всіх попередніх ключів є додатковою витратою.

1.5 Технології цифрового підпису на практиці

Технології цифрового підпису використовуються з тією ж метою, що й традиційні підписи [11]. Як правило, алгоритми цифрового підпису складаються з трьох різних етапів: алгоритм генерації ключів, алгоритм підпису та алгоритм перевірки. Алгоритм генерації ключа використовується для вибору та виведення закритого ключа та відкритого ключа, який відповідає закритому ключу. Закритий ключ вибирається випадковим чином із групи ймовірних закритих ключів. Ключ — це параметр у криптографії, який використовується для визначення функціонального результату криптографічного алгоритму, або, простіше кажучи, зашифрованого тексту. Алгоритм підпису використовує повідомлення та закритий ключ для створення цифрового підпису. Нарешті, алгоритм перевірки підпису, який виконується на стороні одержувача, використовується для перевірки автентичності документа на основі повідомлення, відкритого ключа та підпису. Одержувач отримує повідомлення разом із підписом і відкритим ключем, який доступний для

перевірки автентичності повідомлення. Це проводиться шляхом порівняння отриманого підпису з розрахованим підписом.

Цей метод, заснований на асиметричній криптографії, де користувачі мають пару ключів, відкритий і закритий, також називається методом відкритого ключа [12]. У цьому методі кожен користувач має два унікальних асоційованих криптографічних ключа. Відкритий ключ доступний у загальнодоступному сховищі, тоді як закритий ключ зберігається у користувача ключів. Метод відкритого ключа базується на інфраструктурі відкритих ключів, яка створена для видачі, підтримки та відкликання сертифікатів відкритих ключів.

У випадку цифрових підписів закритий ключ використовується для обчислення цифрового підпису, який можна перевірити за допомогою відповідного відкритого ключа. Закритий ключ також можна використовувати для відкриття повідомлень, зашифрованих відкритим ключем, і навпаки. Відкритий ключ у випадку цифрових підписів використовується для перевірки цифрових підписів, підписаних за допомогою закритого ключа.

Відкритий ключ — це криптографічний ключ, який використовується з асиметричним алгоритмом і пов'язаний з окремим закритим ключем. Закритий ключ, також званий секретним ключем, використовується в методі відкритого ключа для створення цифрових підписів, які можна перевірити за допомогою відповідного відкритого ключа. Їх також можна використовувати для відкриття повідомлень, зашифрованих за допомогою відкритого ключа.

Цифрові підписи з використанням методу відкритого ключа пропонують механізм для перевірки автентифікації джерела, цілісності даних і невідмовності підписувача, якщо вони належним чином реалізовані. Невідмовність означає, що цілісність і автентичність можна підтвердити за допомогою сертифіката, відповідного ключу.

Ключовою особливістю послуг цифрового підпису є автентифікація. Автентифікація як електронний процес, який дозволяє електронну ідентифікацію фізичної чи юридичної особи або підтвердження походження та цілісності даних в електронній формі. У регламенті eIDAS є три рівні цифрових підписів, які визначаються методами їх автентифікації: простий, розширений і кваліфікований. На практиці простий електронний підпис може являти собою прапорець із позначкою «Я приймаю» на веб-сайті [13]. Другий рівень цифрових підписів є «розширеним» і визначається в регламенті eIDAS як електронний підпис, який відповідає таким вимогам: (1) «він однозначно пов'язаний з підписантом», (2) «він здатний ідентифікувати підписанта», (3) «він створений за допомогою засобів, які підписант може зберігати під своїм одноосібним контролем» і що (4) «він пов'язаний з даними, до яких він відноситься, таким чином, що будь-яка наступна зміна даних можна виявити». На практиці це означає, що підписант має бути ідентифікований за допомогою ідентифікаційної картки, банківського ідентифікатора або мобільного ідентифікатора. Найвищий рівень кваліфікованого цифрового підпису наразі можливий лише за допомогою організаційних карток, наданих Digital and Population Data Services Agency (DVV), і недоступний для комерційного використання. Тому, коли це стосується цифрових підписів із сильною автентифікацією, воно стосується цифрових підписів, які задовольняють критеріям розширеного електронного підпису, визначеним положенням eIDAS. Термін «строга автентифікація» означає автентифікацію, яка використовує служби надійної ідентифікації.

Надійні методи ідентифікації доступні споживачам через банки та постачальників телекомунікацій. Такими засобами є або мобільні сертифікати (фінською мовою: mobiilivarmenne), надані постачальниками телекомунікацій, або банківські ідентифікатори, надані банками. Окремо від них кошти, надані

центром реєстрації населення , який надає довідки громадянина та довідки організації.

У службі мобільного сертифіката (також називається Mobile ID) сертифікат особи додається до SIM-карти, яка використовується в мобільному пристрої. Послуга мобільних сертифікатів базується на сертифікатах X.509 і методі відкритих ключів, де приватні ключі, пов'язані з сертифікатами, знаходяться на SIM-картці, захищеній PIN-кодом. Мобільний сертифікат використовує унікальний електронний ідентифікаційний номер. Під час подання заявки на мобільний сертифікат первинна ідентифікація заявника проводиться або за допомогою ідентифікаційного номера банку, або особисто за допомогою паспорта чи посвідчення особи.

Ідентифікатор банку як засіб електронної ідентифікації базується на банківських облікових даних, які надаються особі банками. Початкова ідентифікація для банківських облікових даних проводиться за допомогою дійсного документа, що посвідчує особу, виданого поліцією, паспорта або документа, що посвідчує особу, який можна використовувати як проїзний документ. Якщо заявник не має прийнятного документа, поліція також може подбати про процес ідентифікації.

Всі постачальники послуг надійної ідентифікації надають свою ідентифікацію як послугу низці постачальників послуг у різних галузях, яким потрібно ідентифікувати своїх клієнтів

Загалом процес підпису в службах цифрового підпису, складається приблизно з наступних кроків: сторона, яка збирає підписи, надсилає підписанту запрошення підписати документ. Загальними засобами для цього запрошення є електронна пошта та SMS. Далі підписант відкриває запрошення, яке отримав. Щоб відкрити документ, користувач часто повинен надати певну форму облікових даних, наприклад пароль, наданий відправником запрошення.

Одержавши доступ до документа, особа, яка підписується, може його переглянути.

Щоб перейти до підписання документа, підписант використовує метод ідентифікації, який залежить від використовуваної послуги та методів, які дозволив відправник. Це може бути як мінімум комбінація електронної пошти та пароля, і в цьому випадку рівень надійності схеми ідентифікації низький, але зазвичай служби прагнуть використовувати метод, який забезпечує значний рівень гарантії, наприклад мобільний сертифікат або банківський ідентифікатор. Сильна автентифікація є цифровим еквівалентом показу ідентифікаційної картки. Саму дію підписання можна завершити, натиснувши кнопку або намалювавши підпис комп'ютерною мишкою. Після того, як усі підписані сторони підписали документ, він доступний для архівування. Залежно від використовуваної послуги, документ можна заархівувати в службі та/або завантажити та зберегти на власному пристрої. Послуги архівування є однією з відмінних рис цих послуг, оскільки деякі служби пропонують довгострокове архівування, а інші не надають архівування взагалі. Сервіси зазвичай зберігають ідентифікаційні дані, такі як ір -адреси та позначки часу, які потім можна використовувати для перевірки справжності підпису.

Популярність цифрового підпису стрімко зросла через пандемію COVID-19, яка змусила обмежити контакти між людьми. За даними Visma Solutions, кількість підписів, зроблених за допомогою їхнього сервісу Visma Sign, подвоїлася лише за десять тижнів після карантину, спричиненого пандемією.

Кількість досліджень щодо прийняття користувачами цифрових підписів досить обмежена. В роботі [14] використовували інтегровану рамкову модель прийняття технологій (ТАМ) і технологічно -організаційне середовище, щоб визначити сім рушійних факторів використання цифрового підпису: захист безпеки, внутрішні потреби, навчання та освіта, урядова політика, підтримка постачальників, передбачувана простота використання та передбачувана корисність.

2 АЛГОРИТМИ ЦИФРОВОГО ПІДПISУ В ПОСТКВАНТОВІЙ КРИПТОГРАФІЇ

2.1 Цифровий підпис і сучасна криптографія

Криптографія стала невід'ємною частиною сучасної комунікації. Вона істотно еволюціонувала протягом століть, що призвело до розвитку двох основних категорій: симетричної та асиметричної криптографії. Симетрична криптографія використовує тільки один ключ, за допомогою якого виконується як шифрування, так і дешифрування. Це простий і надзвичайно ефективний метод, але він не позбавлений недоліків. Серед них найбільшим є спосіб обміну або встановлення ключа між сторонами. На щастя, це завдання можна довірити й іншим методам. Наприклад, для передачі ключа можна використовувати асиметричну криптографію або кнопку розподілу квантового ключа [15]. За умови конфіденційності та цілісності ключів симетрична криптографія може запропонувати розумну швидкість операцій і безпеку.

Асиметрична криптографія вирішує найбільшу проблему симетричної криптографії, а саме безпечний обмін ключами або встановлення ключів. Для шифрування та дешифрування використовується пара ключів, відкритий і закритий. Приватний ключ обов'язково залишається конфіденційним, оскільки лише за допомогою цього ключа власник може розшифрувати повідомлення, адресовані йому. Однак відкритий ключ є спільним, щоб надати відправникам можливість шифрувати повідомлення. Безпека асиметричної криптографії фундаментально базується на складності математичних задач. Серед них можна виділити факторизацію великих простих чисел, яка є основою RSA, і системи на основі еліптичних кривих, наприклад ECC. Асиметрична криптографія також лежить в основі цифрових підписів.

Цифрові підписи є одним із основних застосувань асиметричної криптографії. Цифрові підписи забезпечують неспростовність даних і їх цілісність. Для того, щоб створити підпис, необхідні кілька елементів: закритий ключ, алгоритм підпису і, як правило, хеш-функція. Схема цифрового підпису представлена на рис. 2.1. По-перше, обчислюється хеш (1) для підпису файлу – цей процес не є обов’язковим, але підтримує ефективність процесу підпису. Згодом використовується алгоритм підпису. Його вхідними даними є хеш файлу, який потрібно підписати, та закритий ключ підписувача (2). Результатом цієї дії є підпис, який підтверджує незаперечне авторство файлу. Ця колекція — вихідні дані та підпис — надсилається одержувачу (3).



Рисунок 2.1 – Схема цифрового підпису

Особа, яка перевіряє підпис, окрім файлу та його підпису, також повинна мати інформацію про те, яку функцію хешування було використано та який алгоритм підпису застосовано. Процес перевірки починається із застосування алгоритму підпису разом із відкритим ключем підписувача до отриманого

підпису (4). Результатом цієї операції є хеш вихідних даних. Наступним кроком є окреме обчислення хешу підписаного файлу за допомогою відомої хеш-функції (5). Потім ці два хеші необхідно порівняти (6). Якщо вони рівні, це підтверджує неспростовність і цілісність підписаних даних. Процес перевірки також представлений на рис. 2.1.

2.2 Алгоритми постквантової криптографії

Квантові комп'ютери являють собою прорив у сфері обчислень. На відміну від класичних пристроїв, вони використовують принципи квантової механіки для обробки інформації. Найменша одиниця даних у звичайних комп'ютерах називається бітом; воно може приймати значення 0 або 1. Однак у квантових комп'ютерах це замінено кубітами, які можуть перебувати в суперпозиції станів 0 і 1. Це дозволяє їм виконувати багато операцій одночасно, що призводить до збільшення швидкості. Проблеми, з якими стикаються поточні прототипи квантових комп'ютерів, полягають у їх стабільності та масштабованості. Щоб підтримувати квантовий стан, необхідно підтримувати надзвичайно низькі температури, якомога ближче до абсолютного нуля. Наступними факторами, які негативно впливають на кубіти, є всілякі вібрації або магнітні поля [16]. Незважаючи на те, що квантові комп'ютери все ще потребують багато досліджень, вони становлять потенційну загрозу для сучасної криптографії. Отже, є підстави для становлення та розвитку постквантової криптографії.

Однією з найвідоміших концепцій у галузі квантових технологій є алгоритм Шора. Це квантовий алгоритм, представлений Пітером Шором у 1994 році [17]. Його було розроблено з метою розкладання на множники цілих чисел

. В даний час ця проблема є надто складною, щоб класичні комп'ютери могли виконувати обчислення за вимірний час. Обчислювальна складність класичних алгоритмів експоненціальна, тоді як алгоритм Шора пропонує поліноміальну складність, що призводить до значно швидших обчислень. Це рішення поєднує звичайні математичні підходи та принципи квантової механіки [18]. Нині широко використовуваний алгоритм RSA базується саме на складності проблеми факторизації великих чисел.

Іншим прикладом є алгоритм Гровера. Він був розроблений Ловом Гровером у 1996 році, і його метою є пошук у неупорядкованих базах даних [19]. Тоді як це завдання з класичним алгоритмом має обчислювальну складність $O(N)$, алгоритм Гровера досягає складності $O(\sqrt{N})$. Причиною цього є використання квантової суперпозиції, що дає можливість перевіряти декілька потенційних рішень одночасно [20]. Його практичне застосування може загрожувати алгоритмам криптографії, але стрибок у швидкості не такий значний. Вважається, що подовження ключів є достатнім, щоб створити стійкість до атак з його участю. Незважаючи на це, навіть якщо алгоритм Гровера не становить такої серйозної загрози, як алгоритм Шора, він також відіграє помітну роль щодо ризику для поточної кібербезпеки.

Постквантова криптографія є проактивною відповіддю на загрози, спричинені появою алгоритмів Шора та Гровера. Через потенційну здатність зламати широко використовувані алгоритми, такі як RSA або ECC, повинні бути рішення, які підтримують асиметричне шифрування даних, але стійкі до атак квантових комп'ютерів. Саме це і є метою розробки постквантових алгоритмів. Вони поділяються на чотири основні категорії.

- Граткова криптографія є одним із провідних кандидатів постквантової криптографії. Вони базуються на складності задач, таких як проблема найкоротшого вектора (SVP), а також навчання з помилками (LWE) [21]. Безпека цих рішень походить від труднощів розв'язування деяких решіток у

багатовимірних просторах. Проблема, наприклад, полягає в знаходженні констант, на які слід помножити базисний вектор, що є непростим завданням для квантових комп'ютерів. Деякі з відомих алгоритмів, що належать до цієї групи, це NTRU, CRYSTALS- Dilithium і Falcon.

- Криптографія на основі хешування — це група, яка базує свою безпеку на відповідних хеш-функціях і схемах, таких як схема підпису Меркла [21]. Наразі невідомо жодного ефективного квантового алгоритму, здатного зламати криптографічну хеш-функцію; однак недоліком такого рішення може бути обмеження кількості підписів, згенерованих за допомогою конкретної пари ключів. Прикладом алгоритму, який належить до цієї групи, є SPHINCS+.

- Криптографія на основі коду — це група, яка базує свою безпеку на кодах виправлення [22]. Прикладами алгоритмів з такими властивостями є McEliece і його варіант Niederre – iter . Складність цих рішень полягає в декодуванні загального лінійного коду.

- Криптографія на основі ізогенії є останньою з чотирьох основних груп постквантових алгоритмів. Розв'язки в ньому базуються на властивостях еліптичних кривих і задачі пошуку кігтя. Це передбачає роботу з еліптичними кривими, які мають властивість ізогенії [15].

Коротко обговоримо алгоритми CRYSTALS- Dilithium , Falcon і SPHINCS+, кожен з яких представляє різний підхід до постквантової криптографії.

- CRYSTALS- Dilithium — це алгоритм на основі ґратки. Він був розроблений для цифрових підписів і став головним фіналістом конкурсу NIST саме в цій категорії, що також є його головною перевагою. Безпека цього алгоритму полягає в рівні складності ґраткових задач над модульними ґратками [23]. Як алгоритм, заснований на ґратці, він виграє від складності задач найкоротшого вектора та проблем навчання з помилками, які також вважаються незламними за допомогою квантових комп'ютерів.

– Falcon — це аббревіатура від «Fast-Fourier Lattice Compact Signatures over NTRU». Це також був ще один кандидат у конкурсі NIST. Він використовує структуру решітки NTRU та швидке перетворення Фур'є (FFT) для покращення продуктивності. Його основною перевагою є малий розмір створюваних підписів, що особливо важливо в контексті передачі даних великого обсягу [24].

– SPHINCS+ представляє інший підхід до попередніх прикладів, оскільки це алгоритм, заснований на хеш-функціях. Його перевагою є здатність базуватися на різних хеш-функціях, наприклад, SHA2 або SHAKE, що дозволяє адаптувати його відповідно до конкретних вимог безпеки. З іншого боку, його недоліком є відносно великий розмір генерованих підписів [25].

Кожен із зазначених алгоритмів являє собою достатній рівень безпеки. Ці три алгоритми були обрані NIST під час процесу стандартизації алгоритмів, призначених для цифрових підписів. Враховуючи доступність усіх трьох представлених алгоритмів, вони були реалізовані для використання з цифровим підписом і протестовані.

2.3 Реалізація алглритмів постквантової криптографії

Аналіз продуктивності алгоритмів постквантової криптографії базується на реалізованому додатку CLI та бібліотеці liboqs [26]. Liboqs надає елементи, пов'язані з криптографією, а Click відповідає за створення інтерфейсу командного рядка. Усі елементи поєднуються, утворюючи середовище розробки. Ці рішення не тільки забезпечують функціональний інтерфейс, але й полегшують ефективне програмування.

Бібліотека `liboqs` — це набір постквантових алгоритмів, підготовлених у рамках проекту Open Quantum Safe [27]. Бібліотека була розроблена, щоб функціонувати як комплексний набір інструментів для тестування та оцінки для дослідників і розробників, надаючи їм можливість експериментувати з низкою криптографічних схем, які вважаються стійкими до потенційних атак з квантових комп'ютерів.

Бібліотека написана на C, забезпечуючи оптимізовану продуктивність основу для криптографічних операцій. Щоб забезпечити ширшу доступність та інтеграцію в різні стеки програмного забезпечення, `liboqs` надає мовні оболонки, включаючи оболонку Python. Ця оболонка дозволяє використовувати постквантові алгоритми в програмах Python і була включена в реалізацію. Щоб використовувати `liboqs -python`, спочатку необхідно створити та встановити бібліотеку `liboqs`, а потім належним чином налаштувати віртуальне середовище Python.

Реалізована програма може підписувати вибраний файл і перевіряти цифрові підписи за допомогою постквантових алгоритмів. Його основна функція базується на команді `rq -sign`, яку можна використовувати в терміналі. Приклад знака та процесу перевірки представлено на рис.2.2. Важливою функцією є `generate-keypair`, яка генерує закритий ключ, який використовується для підпису документів, і відкритий ключ, який можна використовувати для перевірки підписів. Команда підписати є центральною функцією програми — вона дозволяє користувачам цифрово підписувати документи, забезпечуючи їх автентичність і цілісність. Перевірка цифрового підпису здійснюється командою `rq -sign verify`. Крім того, команда `rq -sign ls` розроблена для надання вичерпного списку всіх постквантових криптографічних алгоритмів, які зараз підтримуються програмою.

```

filip@filip-post-quantum:~$ pq-sign sign example.txt Dilithium2_private_key.key
Signing example.txt file with Dilithium2...
Signed
filip@filip-post-quantum:~$ pq-sign verify example.txt signature_example_Dilithium2.sig Dilithium2_public_key.pub
Is signature valid: True

```

Рисунок 2.2 – Підпис і правильна перевірка

Таблиця 2.1 містить специфікації доступних алгоритмів [26]. Він докладно описує назви алгоритмів разом із їхніми відповідними рівнями безпеки, представленими рівнями безпеки NIST [28]. Рівні безпеки криптографічних алгоритмів або систем визначаються на основі числового значення, пов'язаного з обсягом роботи або операцій, необхідних для їх компрометації. Ці рівні варіюються від рівня 1, який пропонує найнижчий рівень безпеки, до рівня 5, який забезпечує найвищий рівень безпеки [29]. Цифри в кінці назви алгоритму Dilithium відповідають заявленому рівню безпеки NIST. У таблиці також наведено розміри відкритих ключів, секретних ключів і підписів, усі виражені в байтах. І саме розмір згенерованого підпису є вирішальним для розрізнення алгоритмів SPHINCS+. Важливо відзначити, що розуміння компромісів між рівнем безпеки та вимогами до ресурсів є обов'язковим для користувачів. Це розуміння може допомогти у виборі відповідного алгоритму для конкретного контексту безпеки.

Таблиця 2.1 – Специфікації доступних алгоритмів [19]

Алгоритм	Заявлено Рівень NIST	Розмір відкритого ключа (байти)	Розмір секретного ключа (байти)	Розмір підпису (байти)
Dilithium 2	2	1312	2528	2420
Dilithium 3	3	1952 рік	4000	3293
Dilithium 5	5	2592	4864	4595
Falcon -512	1	897	1281	666
Falcon -1024	5	1793 рік	2305	1280
SPHINCS+-SHA2-128f-простий	1	32	64	17 088
SPHINCS+-SHA2-128s-простий	1	32	64	7856
SPHINCS+-SHA2-192f-простий	3	48	96	35,664
SPHINCS+-SHA2-192s-простий	3	48	96	16,224
SPHINCS+-SHA2-256f-простий	5	64	128	49,856
SPHINCS+-SHA2-256s-простий	5	64	128	29,792
SPHINCS+-SHAKE-128f-простий	1	32	64	17 088
SPHINCS+-SHAKE-128s-простий	1	32	64	7856
SPHINCS+-SHAKE-192f-простий	3	48	96	35,664
SPHINCS+-SHAKE-192s-простий	3	48	96	16,224
SPHINCS+-SHAKE-256f-простий	5	64	128	49,856
SPHINCS+-SHAKE-256s-простий	5	64	128	29,792

2.4 Аналіз ефективності алгоритмів

Проведемо аналіз ефективності, особливо зосереджуючись на часових метриках. Ідея тестування продуктивності полягає у всебічному дослідженні постквантових алгоритмів у запропонованій реалізації. Існує кілька причин, чому час є ключовим фактором щодо криптографічних алгоритмів. Найважливіші з них перераховані нижче.

- Масштабованість — алгоритми зі швидшим часом обробки краще підходять для керування великими наборами даних або сценаріями з високим трафіком. Якщо алгоритм ефективний за часом, вся система зможе обробляти більше операцій.

- Взаємодія з користувачем — криптографічні алгоритми мають бути достатньо швидкими, щоб забезпечити плавне використання програм. Затримка суттєво впливає на досвід використання конкретної служби. Крім того, деякі програми потребують роботи в режимі реального часу, що не підтримується неефективними алгоритмами криптографії.

- Безпека — якщо час виконання певного алгоритму є надмірно довгим, це може створити можливість для зломисника зібрати інформацію про час, потрібний цій службі. Вимірюючи час у різних прикладах вхідних даних, можна отримати таку інформацію, як довжина використаних ключів. Щоб запобігти цій уразливості, необхідно переконатися, що використовувані рішення оптимізовані за часом і розроблені таким чином, щоб час виконання був постійним, що означає, що вони не залежать від вхідних даних.

- Продуктивність системи — криптографічні операції майже завжди є частиною більших систем, і тому їх ефективність у часі перетворюється на сукупну продуктивність складних служб.

Більше того, алгоритми криптографії, що використовуються на практиці, вимагають балансування надійності безпеки та ефективності часу. Правильний компроміс є вирішальним, беручи до уваги зручність використання таких рішень.

Щоб перевірити продуктивність постквантових алгоритмів у механізмі цифрового підпису, необхідно було мати файли, які були б підписані. Передбачалося, що цілі файли зашифровані закритими ключами кожного перевіреного алгоритму. Щоб отримати репрезентативний спектр результатів, було створено три файли розміром 10 МБ, 100 МБ і 1 ГБ. Це було досягнуто за допомогою рідної команди Linux `dd`, яка використовується для копіювання та перетворення даних. За допомогою цієї команди було скопійовано псевдовипадкові дані з файлу `/dev/urandom`.

Кожен функціонал був протестований для кожного алгоритму. Кожне вимірювання повторювалося 101 раз. Час першого вимірювання для заданого алгоритму був помітно довшим і не був включений у графіки та подальший аналіз. Це пов'язано з тим, що дані зчитуються з диска та завантажуються в кеш-пам'ять. Пропустивши перше вимірювання в кожній серії, можна було виключити коефіцієнт швидкості читання диска з компонента вимірювання та отримати відповідний час, необхідний для виконання перевіреного алгоритму. Отже, представлені графіки показують середній час виконання алгоритму, розрахований за 100 ітераціями заданої операції. Смуги похибок, що показують стандартну помилку середнього (SEM), також включені для кожного алгоритму. Перед початком вимірювань необхідно було підготувати різні компоненти. Для процесів підписання та перевірки підпису було підготовлено три файли розміром 10 МБ, 100 МБ та 1 ГБ. Для кожного алгоритму також була створена відповідна пара ключів. Під час оцінки генерації пари ключів файли, що містять пари ключів, видалялися після кожної ітерації. Що стосується тестів підписання файлів, кожен ітерацію запускали для одного файлу, і, як і у

випадку з генерацією пари ключів, отриманий файл видалявся після кожної ітерації. Дотримуючись тієї самої процедури, тести перевірки підпису були виконані на одному файлі для кожного алгоритму, щоб порівняти результати вимірювань.

2.5 Результати вимірювань

Спочатку була перевірена продуктивність генерації пари ключів для різних постквантових криптографічних алгоритмів. На рис. 2.3 представлені результати цього сценарію. Можна помітити, що алгоритми Dilithium мали найкращу часову продуктивність. Алгоритми з групи SPHINCS+ із суфіксом «f» також досягли приємного результату, вирізнившись із другої частини цієї групи. Однак слід пам'ятати, що вони оптимізовані за швидкістю, а не за розміром. Тому вони генерують довший підпис, ніж ті, що мають суфікс 's', як показано в табл. 2.1. Обидві ці групи отримали рази на порядок більші, ніж алгоритми Falcon і SPHINCS+ із суфіксом 's'. Проте спонукає до роздумів те, що алгоритми SPHINCS+SHA-192s-simple і SPHINCS+SHAKE-192s-simple продемонстрували довший час виконання порівняно з аналогами з довгими ключами, а саме SPHINCS+SHA-256s-simple і SPHINCS+SHAKE- 256s-прості відповідно. Ця тенденція продуктивності суперечила поведінці інших алгоритмів. Наприклад, алгоритми SPHINCS+ із суфіксом «f» і алгоритми Falcon продемонстрували залежність: чим більший розмір ключа, тим довший час виконання. Така несподівана поведінка може бути прямим наслідком того, як певні алгоритми реалізовано в бібліотеці.

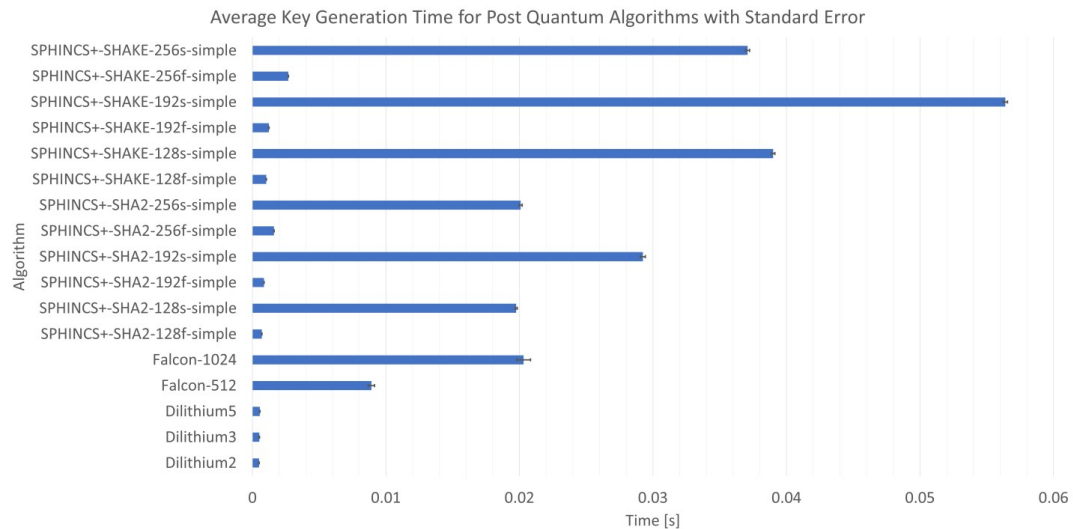


Рисунок 2.3 – Середній час генерації ключа

Рис. 2.4-2.6 стосуються функції підпису файлу. Вони показують середній час виконання, отриманий кожним алгоритмом під час підписання файлів розміром 10 МБ, 100 МБ і 1 ГБ відповідно. Розумна тенденція полягає в тому, що чим більший розмір файлу, який потрібно підписати, тим довше триває ця операція. У всіх трьох випадках групи алгоритмів Dilithium і Falcon досягли найкращих часових показників. Подібно до рис. 2.3, де показано середній час генерації пари ключів, розбіжність між алгоритмами SPHINCS+ із суфіксами «s» і «f» була помітною в перевазі оптимізованих за швидкістю. Цей результат відповідає очікуванням. Однак різниця зменшилася зі збільшенням розміру підписаного файлу. Як на рис. 2.3 також були присутні результати, які суперечать інтуїції, а саме в алгоритмах групи SPHINCS+ було видно приклади кожного розміру файлу, коли алгоритм, що працює з коротшим ключем, досяг гіршого часу, ніж його аналог, що працює з довшим ключем. Оскільки продуктивність алгоритмів у двох групах, Dilithium і Falcon, була подібною, було проведено додатковий аналіз. Мета цього додаткового дослідження полягала в тому, щоб отримати детальне уявлення про їх продуктивність. Результати представлені на рис. 2.7. На основі аналізу результатів можна

зробити висновок, що алгоритми Dilithium досягають кращої продуктивності навіть для відносно невеликих розмірів даних зі знаком. Згідно з інтуїцією та вимірюваннями на більших файлах, алгоритм Falcon-1024 працював майже вдвічі довше, ніж його еквівалент Falcon-512 через вдвічі довший ключ.

На рис. 2.8-2.10 показано середній час перевірки для підписаних файлів розміром 10 МБ, 100 МБ і 1 ГБ відповідно. Рис. 2.8 посилаючись на випадок із найменшим розміром файлу, показує, що алгоритми Dilithium показали найгірші результати, що повністю відрізняється від попередніх результатів. У цьому випадку найкращого результату досягли алгоритми SPHINCS+SHA2-192 і SPHINCS+SHA2-256. Алгоритм SPHINCS+ на основі хеш-функції SHAKE досяг порівнянних результатів незалежно від розміру ключа. В операції з файлом розміром 100 МБ, як показано на рис. 2.9 простий алгоритм SPHINCS+SHAKE-256s зафіксував найдовшу тривалість. Трійка алгоритмів SPHINCS+SHA2 також показала невеликі варіації. Однак, окрім цього, результати були відносно порівнянними. Результати, показані на рис. 2.10 (файл 1 ГБ) були найбільш послідовними в усіх оцінених сценаріях. Викидом був набір із чотирьох алгоритмів із групи SPHINCS+, зокрема по два варіанти SPHINCS+SHA2-192 і SPHINCS+SHA2-256, які зафіксували значно кращі результати порівняно з іншими протестованими прикладами. Тим не менш, дисперсія не перевищувала 30% порівняно з іншими алгоритмами.

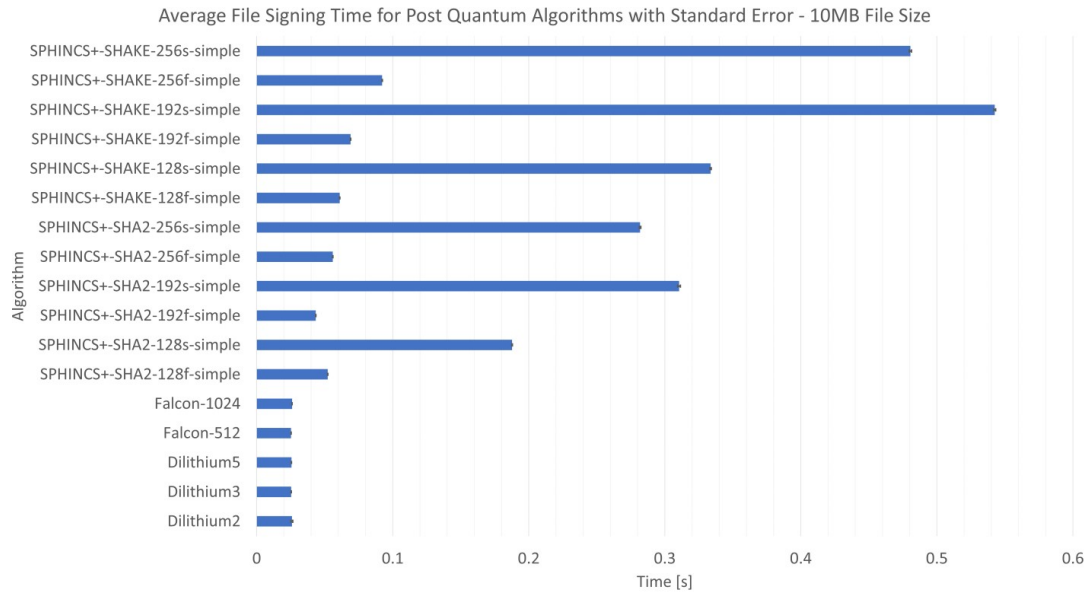


Рисунок 2.4 – Середній час підписання файлу — розмір файлу 10 МБ

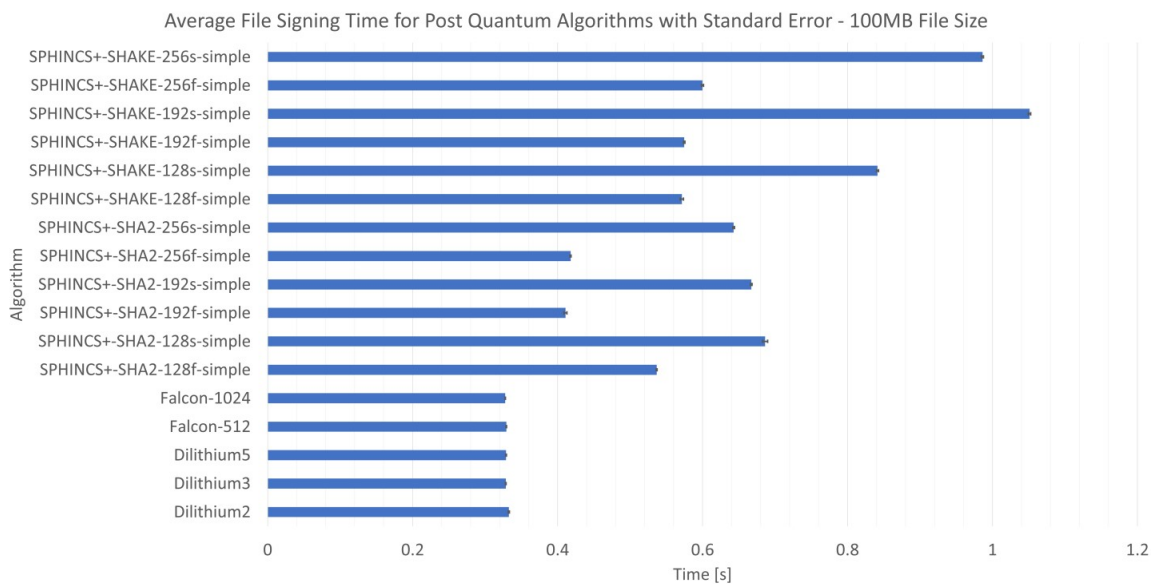


Рисунок 2.5 – Середній час підписання файлу — розмір файлу 100 МБ

Висновок, який можна зробити щодо групових алгоритмів SPHINCS+, полягає в тому, що для підписання файлів великого розміру кращим вибором будуть алгоритми з суфіксом 's'. Причина цього полягає в тому, що під час обробки великих файлів розбіжність між алгоритмами, оптимізованими для швидкості, і алгоритмами, оптимізованими для розміру, зменшується. Однак алгоритми з суфіксом 's' зберігають перевагу значно меншого розміру

згенерованого підпису. Незважаючи на те, що алгоритми з суфіксом «f» значно скорочують час генерації пари ключів, у звичайному випадку для підписання кількох файлів використовується одноразово згенерована пара ключів. Крім того, варто згадати ще одну відмітну особливість групи алгоритмів SPHINCS+: незалежно від розміру файлу, розбіжність у часі перевірки між варіантами, оптимізованими за швидкістю та оптимізованими за розміром, незначна.

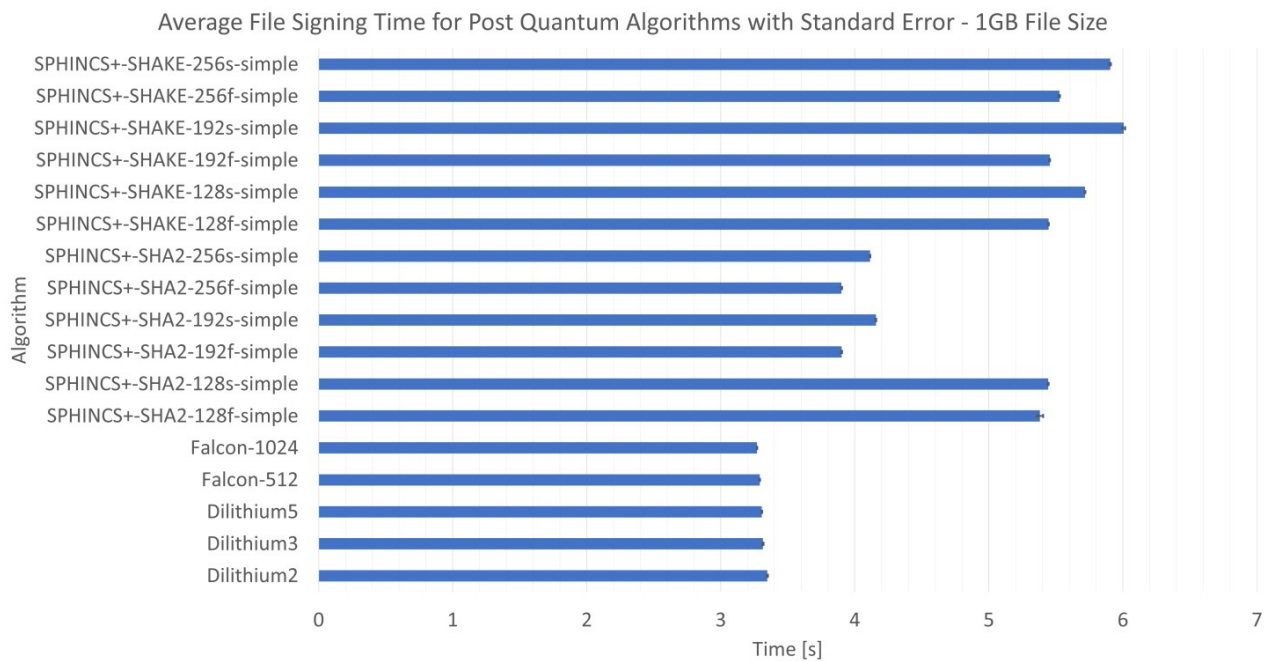


Рисунок 2.6 – Середній час підписання файлу — розмір файлу 1 ГБ

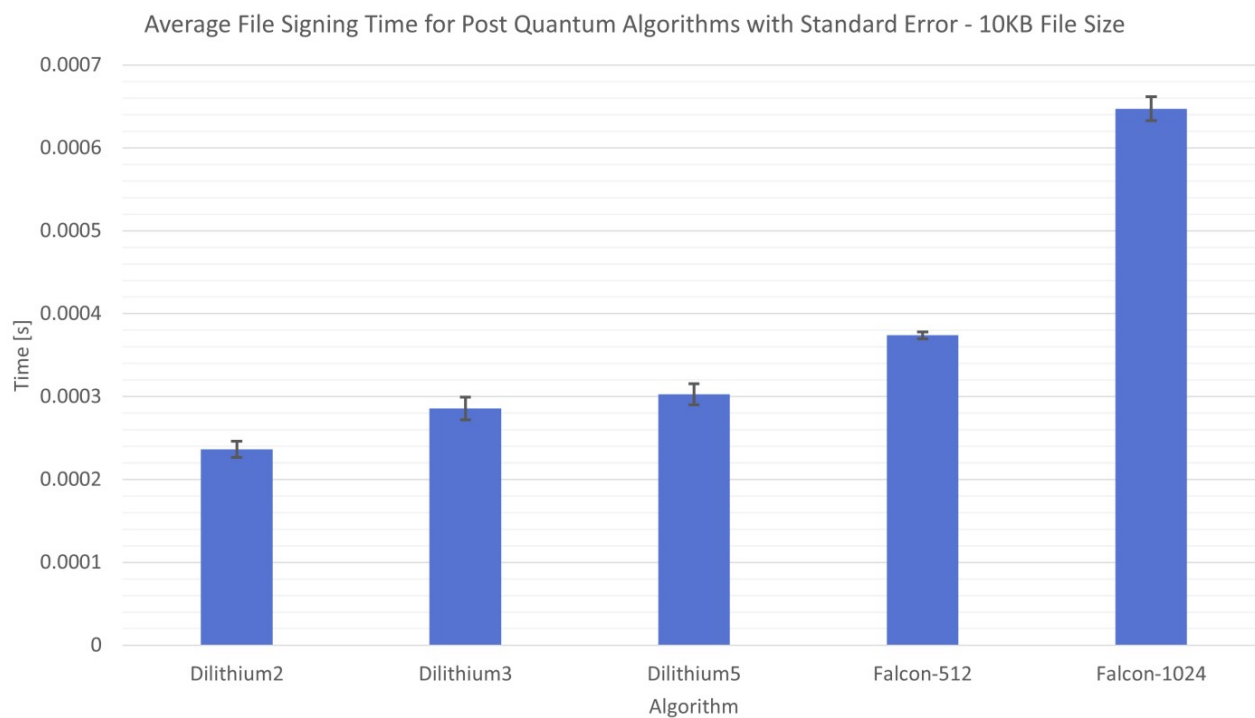


Рисунок 2.7 – Середній час підписання файлу — розмір файлу 10 КБ

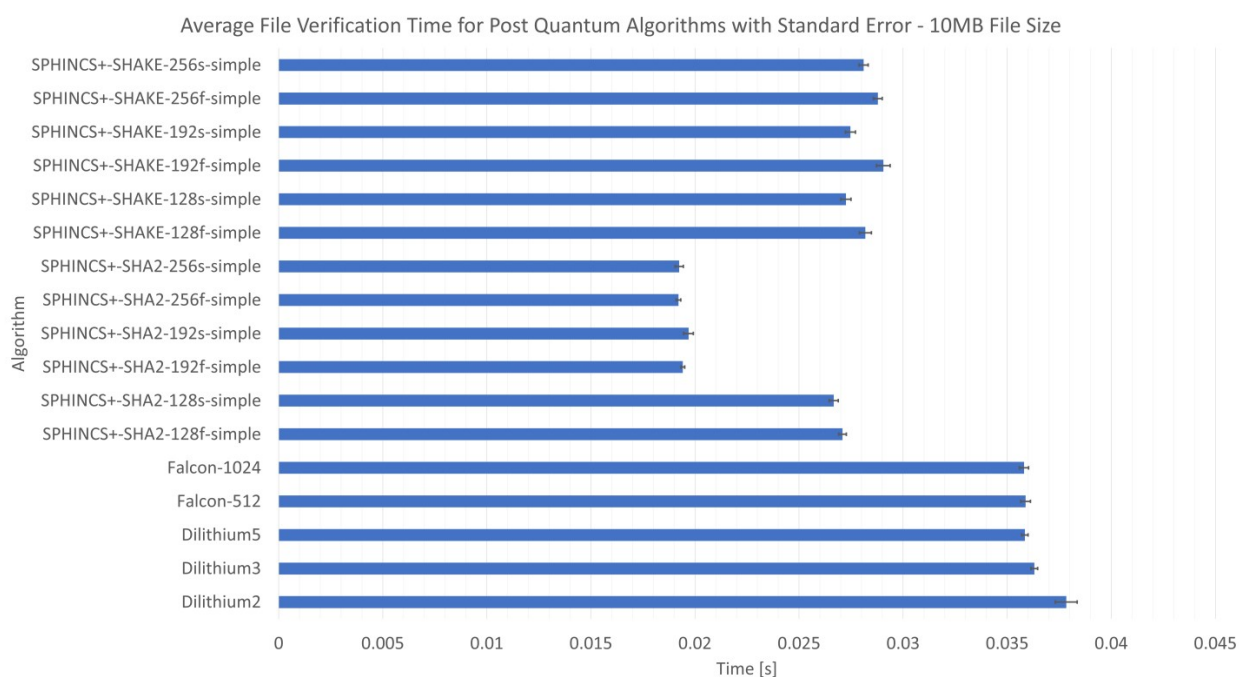


Рисунок 2.8 – Середній час перевірки файлу — розмір файлу 10 МБ

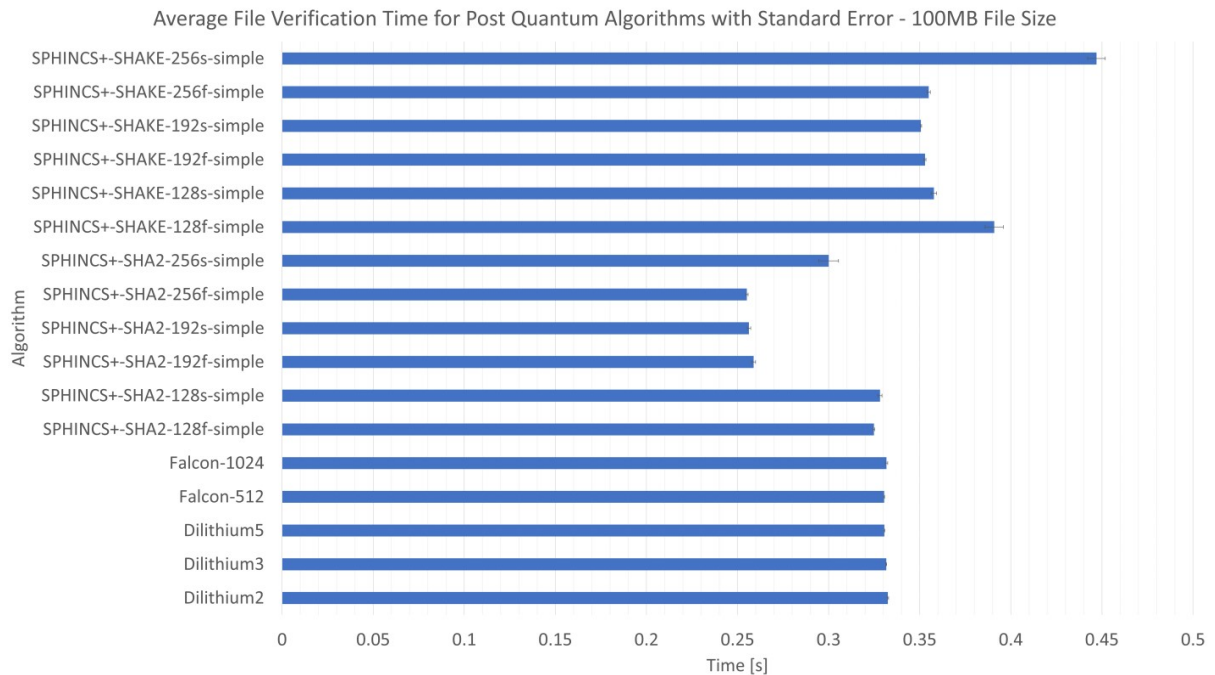


Рисунок 2.9 – Середній час перевірки файлу — розмір файлу 100 МБ

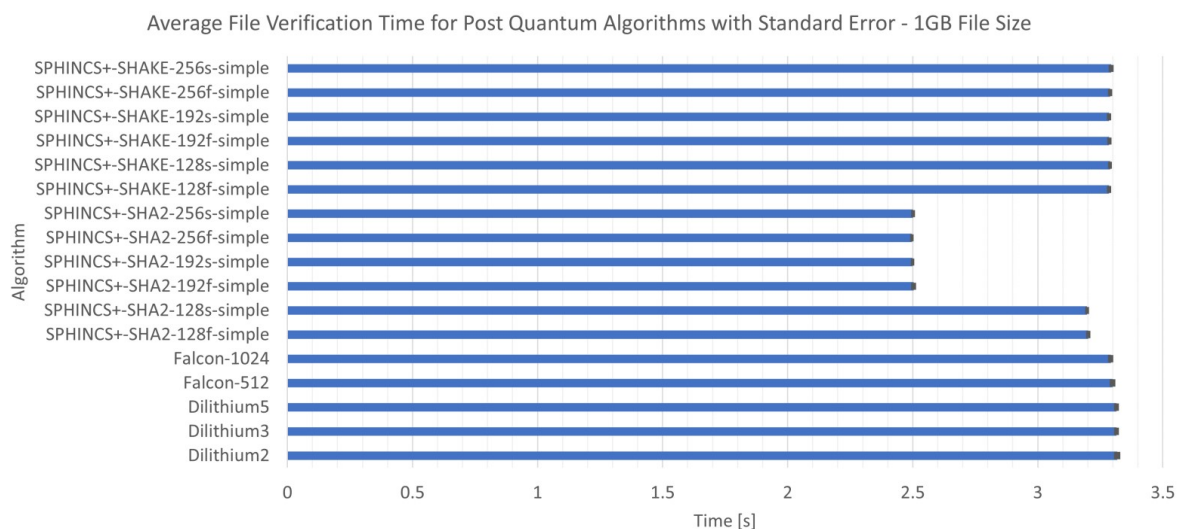


Рисунок 2.10 – Середній час перевірки файлу — розмір файлу 1 ГБ

Представимо порівняння вимірюваних постквантових алгоритмів із добре відомим алгоритмом RSA, наріжним каменем сучасних криптографічних систем, але яким загрожує бачення квантових комп'ютерів. Це порівняння спрямоване на оцінку продуктивності та зручності використання постквантових криптографічних рішень у контексті реальних додатків. Він

досліджує порівняння постквантових алгоритмів із популярним асиметричним шифром у сучасних програмах. Одним із головних моментів цього аналізу буде спроба відповісти на ключове питання: чи виправдані витрати часу на постквантові алгоритми для досягнення квантової стійкості? Це питання має першочергове значення, оскільки перехід до постквантової криптографії може стати необхідним. Щоб отримати результати, відповідні для порівняння, була використана реалізація RSA, що походить від «криптографії» бібліотеки Python. Розмір використовуваного ключа був 2048 байт, а експонента була встановлена на 65 537. Вимірювання проводилися в тому ж середовищі, що й вимірювання постквантового алгоритму.

На рис. 2.11-2.13 представлено порівняння середнього часу підписання файлів постквантових алгоритмів і алгоритмів RSA. Через складність постквантових алгоритмів очікуються витрати часу порівняно з RSA. Приклади з розміром файлу 100 МБ і 1 ГБ відповідають цій гіпотезі, тоді як приклад із розміром файлу 10 МБ показує, що постквантовий алгоритм може бути навіть трохи швидшим, ніж RSA. Дивлячись на повний контекст цих трьох сценаріїв, можна помітити, що відмінності не є приголомшливими. Тоді залишається питання, чи варте значне покращення безпеки витрати часу? Алгоритми Dilithium , включно з найкращим — позначеним як Dilithium5, здаються придатними кандидатами для заміни RSA. Він має рівень безпеки NIST 5, заявлений OpenQuantumSafe , тоді як тестована версія RSA оцінюється як рівень 1. У проведених вимірюваннях для найбільшого файлу (1 ГБ) Dilithium5 досяг середнього результату 3,31 с, тоді як RSA отримав 2,56. с. Це означає накладні витрати приблизно на 27,7% в обмін на стійкість до квантового криптоаналізу.

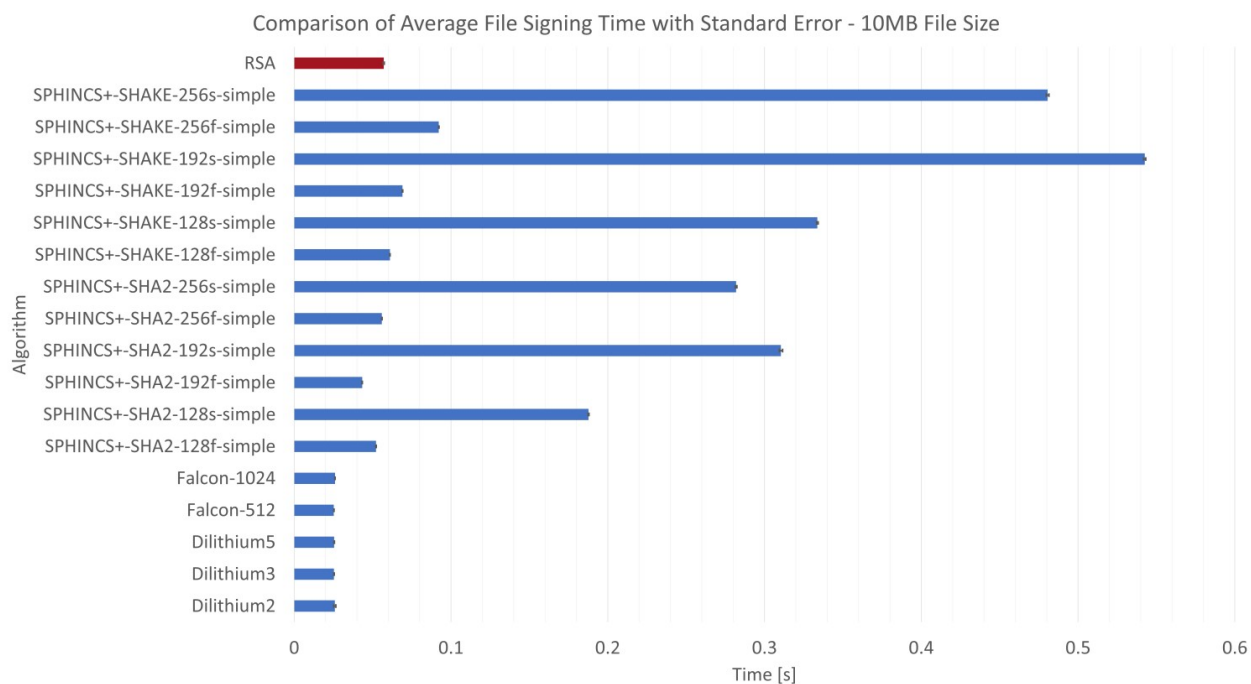


Рисунок 2.11 – Порівняння середнього часу підписання файлу — розмір файлу 10 МБ

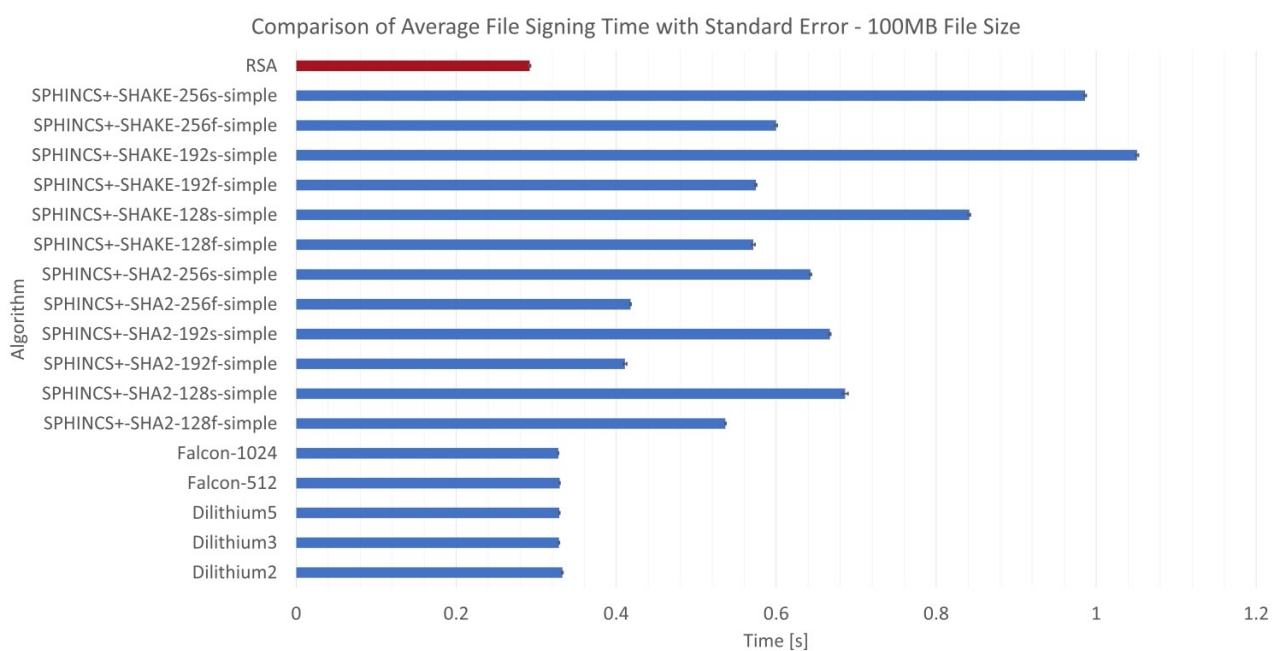


Рисунок 2.12 – Порівняння середнього часу підписання файлу — розмір файлу 100 МБ



Рисунок 2.13 – Порівняння середнього часу підписання файлу — розмір файлу
1 ГБ

ВИСНОВКИ

Продemonстровано фундаментальну роль постквантової криптографії в покращенні цифрових підписів проти нових загроз, пов'язаних із потенціалом квантових комп'ютерів. Завдяки впровадженню та тестуванню вибраних алгоритмів було продемонстровано здатність постквантових алгоритмів у застосуванні до цифрових підписів. Вони, безсумнівно, є альтернативою криптографічним рішенням, які зараз використовуються, і перспективними кандидатами на їх заміну.

Реалізації постквантових алгоритмів для цифрових підписів були позитивно перевірені в кількох сценаріях використання.

Результати узгоджуються з порівняльним аналізом бібліотеки `liboqs`. При аналізі часових показників алгоритм Dilithium (особливо Dilithium5) виявився найкращим кандидатом для цифрових підписів. Хоча він трохи відставав від деяких конкурентів під час перевірки підпису, він явно домінував під час створення пари ключів і процесів підписання. Додатковим фактом на його користь є розмір генерованих підписів. Усі три доступні варіанти алгоритму Dilithium генерують підпис у кілька разів менший, ніж групові алгоритми SPHINCS+. Але якщо розмір згенерованого підпису є основним критерієм, алгоритм Falcon, є фаворитом, досягаючи найменшого часу підписів.

Представляючи підхід хеш-функції, показано, що алгоритми в групі SPHINCS+ демонструють велику різноманітність завдяки своїй здатності співпрацювати з різними хеш-функціями. При виборі між реалізаціями з цієї конкретної групи кращим вибором будуть алгоритми, оптимізовані для розміру підпису. Це має місце тому, що зі збільшенням розміру файлу розбіжність між двома варіантами алгоритму SPHINCS+ зменшується.

Встановлено, що існують алгоритми, які можуть замінити RSA з граничними накладними витратами в часі, діаметрально підвищуючи безпеку, тоді як вибір конкретного алгоритму може визначатися потребами конкретного випадку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Contributors to Wikimedia projects. Key (cryptography) - Wikipedia. *Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Key_\(cryptography\)](https://en.wikipedia.org/wiki/Key_(cryptography)).
2. Contributors to Wikimedia projects. Key generation - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Key_generation.
3. Design and implementation of a digital signature solution for a healthcare enterprise / Bengisu Tulu та ін. *10th americas conference on information systems* : Conference, м. New York, 6 серп. 2004 р. URL: https://www.researchgate.net/publication/220891165_Design_and_Implementation_of_a_Digital_Signature_Solution_for_a_Healthcare_Enterprise.
4. Fruhlinger J. What is PKI? And how it secures just about everything online. *CSO Online*. URL: <https://www.csoonline.com/article/567339/what-is-pki-and-how-it-secures-just-about-everything-online.html>.
5. Lake J. What are digital signatures and how do they work? | comparitech. *Comparitech*. URL: <https://www.comparitech.com/blog/information-security/digital-signatures/>.
6. Contributors to Wikimedia projects. Digital signature - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Digital_signature.
7. Bardon A. Public key certificates: importance in cybersecurity | LU. *Lindenwood University Online*. URL: <https://online.lindenwood.edu/blog/public-key-certificates-what-it-is-the-importance-in-cybersecurity/>.
8. Goldwasser S., Micali S., Rivest R. L. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM journal on computing*. 1988. Т. 17, № 2. С. 281–308. URL: <https://doi.org/10.1137/0217017>.

9. Vaudenay S. Digital signature schemes with domain parameters. *Information security and privacy*. Berlin, Heidelberg, 2004. C. 188–199. URL: https://doi.org/10.1007/978-3-540-27800-9_17.
10. Public key certificate in subject area: computer science. *ScienceDirect*. URL: <https://www.sciencedirect.com/topics/computer-science/public-key-certificate>.
11. On the availability and effectiveness of open source software for digital signing of PDF documents / J. Gamalielsson та ін. *Open source systems: adoption and impact*. Cham, 2015. C. 71–80. URL: https://doi.org/10.1007/978-3-319-17837-0_7.
12. *Security Customer Support Site*. URL: https://support.trust.telia.com/repository/Telia_Mobiiliasiointivarmenne_CPS_v2.2.pdf.
13. Aleksi VÃ¤hima. NÃ¤in toimii sÃ¤hkÃ¶linen allekirjoitus â testissÃ¤ 6 ratkaisua. *TÃ¤rkeimmÃ¤t talousuutiset | Kauppalehti*. URL: <https://www.kauppalehti.fi/uutiset/nain-toimii-sahkoinen-allekirjoitus-testissa-6-ratkaisua/5572c4d8-cacb-495a-8da7-bf2de6fa1294>.
14. Haryanto B., Gandhi A., Giri Sucahyo Y. The determinant factors in utilizing electronic signature using the TAM and TOE framework. *2020 fifth international conference on informatics and computing (ICIC)*, м. Gorontalo, Indonesia, 3–4 листоп. 2020 р. 2020. URL: <https://doi.org/10.1109/icic50835.2020.9288623>.
15. Quantum key distribution / M. Mehic та ін. *ACM computing surveys*. 2020. Т. 53, № 5. С. 1–41. URL: <https://doi.org/10.1145/3402192>.
16. Quantum computer architecture toward full-stack quantum accelerators / K. Bertels та ін. *IEEE transactions on quantum engineering*. 2020. Т. 1. С. 1–17. URL: <https://doi.org/10.1109/tqe.2020.2981074>.
17. Shor P. W. Algorithms for quantum computation: discrete logarithms and factoring. *35th annual symposium on foundations of computer science*, м. Santa Fe, NM, USA. URL: <https://doi.org/10.1109/sfcs.1994.365700>.

18. Hlukhov V. Quantum-Inspired computing: shor's algorithm and euler's totient function. *2023 IEEE 12th international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, м. Dortmund, Germany, 7–9 вец. 2023 p. 2023. URL: <https://doi.org/10.1109/idaacs58523.2023.10348718>.
19. Grover L. K. A fast quantum mechanical algorithm for database search. *The twenty-eighth annual ACM symposium*, м. Philadelphia, Pennsylvania, United States, 22–24 тpав. 1996 p. New York, New York, USA, 1996. URL: <https://doi.org/10.1145/237814.237866>.
20. Shrivastava P., Soni K. K., Rasool A. Evolution of quantum computing based on grover's search algorithm. *2019 10th international conference on computing, communication and networking technologies (ICCCNT)*, м. Kanpur, India, 6–8 лип. 2019 p. 2019. URL: <https://doi.org/10.1109/icccnt45670.2019.8944676>.
21. Wong D. Real-World cryptography. Manning Publications Co. LLC, 2021.
22. Code-Based schemes for post-quantum digital signatures / A. Kuznetsov та iн. *2019 10th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, м. Metz, France, 18–21 вец. 2019 p. 2019. URL: <https://doi.org/10.1109/idaacs.2019.8924271>.
23. Cryptographic suite for algebraic lattices. *CRYSTALS*. URL: <https://pq-crystals.org/>.
24. Fast-Fourier lattice-based compact signatures over NTRU. *Falcon*. URL: <https://falcon-sign.info/>.
25. Ananda Mohan P. V. Hash-based Digital Signatures- A tutorial review. *2023 IEEE international conference on public key infrastructure and its applications (PKIA)*, м. Bangalore, India, 8–9 вец. 2023 p. 2023. URL: <https://doi.org/10.1109/pkia58446.2023.10262777>.

26. Open source liboqs library. *Open Quantum Safe*. URL: <https://openquantumsafe.org/liboqs/>.
27. Stebila D., Mosca M. Post-quantum key exchange for the internet and the open quantum safe project. *Lecture notes in computer science*. Cham, 2017. C. 14–37. URL: https://doi.org/10.1007/978-3-319-69453-5_2.
28. Post-Quantum Cryptography | CSRC. *NIST Computer Security Resource Center | CSRC*. URL: <https://csrc.nist.gov/projects/post-quantum-cryptography>.
29. Security requirements for cryptographic modules. Gaithersburg, MD : National Institute of Standards and Technology, 2001. URL: <https://doi.org/10.6028/nist.fips.140-2>.