**Міністерство освіти і науки України**
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету )
Кафедра кібербезпеки
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Квантові та постквантові засоби та методи захисту інформації

Виконав: студент VI курсу, групи СБм-61
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

(підпис) Пащак С.А (прізвище та ініціали)

Керівник (підпис) Кульчицький Т.Р. (прізвище та ініціали)

Нормоконтроль (підпис) Лечаченко Т.А. (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В. (прізвище та ініціали)

Рецензент (підпис) Осухівська Г.М. (прізвище та ініціали)

Тернопіль - 2023

Міністерство освіти і науки України
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«____» _____ 2023 р.

# ЗАВДАННЯ
## НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Пащаку Станіславу Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Квантові та постквантові засоби та методи захисту інформації

Керівник роботи Кульчицький Т.Р. Ph.D в галузі правва
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*16* » *листопада* 2023 року *№ № 4/7-1061*

2. Термін подання студентом завершеної роботи 16.12.2033

3. Вихідні дані до роботи Технічна документація, наукова література, інтернет-джерела

4. Зміст роботи (перелік питань, які потрібно розробити): Вступ. Розділ 1. Аналіз загроз криптографічних систем та їх ефективність. 1.1 Аналіз математичної частини загрози. 1.2 Оцінка обчислювальних потужностей. 1.3 Прогнозування загроз пов'язаних із захистом інформації. 1.4 Висновок до першого розділу. Розділ 2. Дослідження перспективних методів постквантовго і квантого захисту інформації 2.1 SPHINCS+. 2.2 Falcon 2.3 Kyber 2.4 Квантова теорія, квантові схеми. 2.5 Незалежний від вимірювального пристрою квантовий розподіл ключів(MDI-QKD) 2.6 Спутниковий QKD(SatQKD) 2.7 Висновок до другого розділу Роздл 3. Розробка PQC openssl build'a та його практичне застосування 3.1 Створення версії openssl, яка містить PQC алгоритми 3.2 Аналіз швидкодії постквантових алгоритмів. 3.3 Висновок до розділу. 4 Охорона праці та безпека в надзвичайних ситуаціях. 4.1 Охорона праці, Додатки 4.2 Безпека в надзвичайних ситуаціях, Висновки, Перелік використаних джерел.

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

1. Титульна сторінка 2. Тема. Мета.Об'єкт. Предмет дослідження . 3.Завдання дослідження. 4. Відношення між класами P та NP.5.ДМТ та НДМТ. 6. Кількості кубітів у КК. 7. Стійкість RSA-2048. 8. Графік енергоспоживання ОС. 9.Алгоритм SPHINCS+ 256 SHA256. 10.Схема роботи WOTS+ та FORS. 11.Бенчмарк SPHINCS+ Haraka. 12.Ftree. 13.FastFurie. 14.Рек.пар.алгоритму FALCON. 15.Складність атаки звичайного комп'ютера і квантовго. 16.KeyGen. 17.Enc. 18.Dec. 19.CCAKEM.KeyGen. 20.CCAKEM.Enc. 21.CCAKEM.Dec. 22Прод. Kyber та Kyber-90s. 23.Стійкість Kyber. 24. Функції f1, f2, f3 і f4. 25.Схема HSHT для X. 26. Схема H та $\oplus$. 27.Схема $\oplus$. 28.Відображення $x \wedge y = a \oplus b$. 29.Залежність $\theta$ від x,y. 30.Схема оптимальної стратегії. 31. Наглядний результат функції, UxU. 32.Схематичне зображення DI-QK. 33.Схема MDI-QKD. 34.Імовірнісний відгук BSA. 35.Принцип SatQKD. 36.Графік ефективності SayQKD. 37.apt update. 38.git clone. 39.Build. 40.oqs-provider. 41.curl. 42.Перевірка oqs-provider.

43.FALCON-512 та Kyber768. 44.SHAKE – 128 та Kyber768. 45.Продуктивність Falcon 512. 46.Продуктивність SPHINCS+-128. 47. Продуктивність RSA512. 48.Продуктивність Kyber512. 49.Продуктивність RSA2048. 50.Висновок. 51.Апробація кваліфікаційної роботи.

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| Охорона праці | Осухівська Г.М., к.т.н., доцент | | |
| Безпека в надзвичайних ситуаціях | Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва | | |

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | | *Виконано* |
| 2. | Підбір наукових джерел про постквантову та квантову | | |
| | криптографію | | *Виконано* |
| 3. | Переклад та опрацювання наукових джерел про | | |
| | квантову криптографію | | *Виконано* |
| 4. | Виконання дослідження щодо аналіз квантових | | |
| | та постквантових систем захисту інформації | | *Виконано* |
| 5. | Оформлення розділу «Аналіз загроз криптографічних | | |
| | систем та їх ефективність» | | *Виконано* |
| 6. | Оформлення розділу «Дослідження перспективних | | |
| | методів постквантовго і квантового захисту | | |
| | інформації» | | *Виконано* |
| 7. | Оформлення розділу «Розробка PQC openssl build’a та | | |
| | його практичне застосування» | | *Виконано* |
| 8. | Виконання завдання до підрозділу «Охорона праці» | | *Виконано* |
| 9. | Виконання завдання до підрозділу «Безпека в надзви- | | |
| | чай них ситуаціях» | | *Виконано* |
| 10. | Оформлення кваліфікаційної роботи | | *Виконано* |
| 11. | Нормоконтроль | | *Виконано* |
| 12. | Перевірка на плагіат | | *Виконано* |
| 13. | Попередній захист кваліфікаційної роботи | | *Виконано* |
| 14. | Захист кваліфікаційної роботи | | |

Студент _____ Пащак С.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Кульчицький Т.Р.
(підпис) (прізвище та ініціали)

# АНОТАЦІЯ

Квантові і постквантові методи та засоби захисту інформації // Qualification work of the educational level “Master” Degree Thesis // Пащак Станіслав Анатолійович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2023 // С. 109, рис. – 28 , табл. – 18 , додат. 3– , бібліогр. – 22.

Ключові слова: КРИПТОГРАФІЯ, КВАНТОВИЙ КОМП’ЮТЕР, АЛГОРИТМ, КУБІТ, СТІЙКІСТЬ, ЙМОВІРНІСТЬ, ПІДПИС, ШИФРУВАННЯ.

Ця кваліфікаційна робота є результатом дослідження проблем традиційної криптографії та провідних систем безпечної передачі інформації на яких покладено вирішення даних проблем.

У першому розділі описується загроза для всіх криптографічних систем та порівнюються обчислювальні можливості традиційного так квантового комп’ютерів, їхня роль у захисті інформації.

У другому розділі розглядаються найбільш перспективні алгоритми постквантовго шифрування та квантові системи здатні підтримувати безпечний обмін інформацією

Третій розділ – практична частина, у якому описаний процес інтеграції постквантових криптографічних алгоритмів у інструмент шифрування повідомлень. Перевірка ефективності інтегрованих алгоритмів.

# ANNOTATION

Quantum and post-quantum methods and tools of information security // Qualification work of the educational level "Master" Degree Thesis // Pashchak Stanislav Anatoliiovych // Ternopil National Technical University named after Ivan Pului, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group SBm-61 // Ternopil, 2023 // P. 109, fig. - 28, tables - 18, appendix - 3, bibliography - 22.

Keywords: CRYPTOGRAPHY, QUANTUM COMPUTER, ALGORITHM, QUBIT, RESISTANCE, PROBABILITY, SIGNATURE, ENCRYPTION.

This qualification work is the result of a study of the problems of traditional cryptography and the leading systems of secure information communication that are responsible for solving these problems.

The first chapter describes the threat to all cryptographic systems and compares the computing performance of traditional and quantum computers and their role in information security.

The second section discusses the most promising post-quantum encryption algorithms and quantum systems capable of supporting secure information exchange.

The third section is the practical part, which describes the process of integrating post-quantum cryptographic algorithms into a message encryption tool. Testing the performance of the integrated algorithms.

ЗМІСТ

# ВСТУП

**Актуальність теми.** Безсумнівно, XXI століття – початок інформаційної ери. Більшість людей за всю історію людської цивілізації навіть собі уявити не могли, що в їхніх нащадків буде можливість дістати із кишені, так званий, смартфон та за лічені секунди дізнатись останні події які відбуваються у світі. Велика кількість людей які виросли в цифру епоху, можливо, навіть не розуміють яке було життя без глобальної мережі. Обмін невеликою кількістю інформації став практично миттєвим, навіть саме поняття «невелика кількість інформації» змінилось. Кожен хто має доступ до глобальної мережі має можливість обмінюватись інформації із іншою людиною, у вигляді повідомлення та навіть відео зв'язку у реальному часі. Можливість швидкого обмінну інформації дає змогу ділитись досвідом різних людей, що в свою чергу пришвидшує досягнення в різних сферах людської діяльності, як-от науки, економіки, мистецтва та літератури. Все це можливо завдяки досягненням науки, математики та цифрових технологій. Но окрім того, що інформації є багато, вона повинна бути доцільна, має зберігати цілісність, автентичність і у деяких випадках конфіденційність. Використання дезінформації тільки іноді у рідкісних випадках буває доцільним, тому були розроблені методи і підходи для збереження цих якостей інформації. Один із таких способів, криптографія – наука про математичні способи та методи обробки інформації які надають їй автентичність, конфіденційність та цілісність.

Криптографією займаються вже дуже давно, найперші згадування про це датуються 3-4 тисячоліттям до н.е.. В давнину використання криптографії були набагато більш примітивним та використовувались в особливо рідкісних випадках для збереження секретної інформації. В сьогоденні криптографія є невід'ємною частиною обміну інформації, практично вся інформація або шифрується, або підписується. Важко собі уявити, що би сталось якби математичні принципи, на яких базується криптографія, втратили би свою ліквідність. Для прикладу: якби був знайдений алгоритм швидкого(за

поліноміальний час) дискретного логарифмування, то можна було би розшифрувати більшість повідомлень між користувачами або публікувати інформації від чужого іменні. На перший погляд в цьому немає нічого страшного але все набагато гірше ніж здається, навіть коли математичні методи на яких стоїть криптографія лишаються ефективними, зловмисниками не одноразово вдавалось викрасти секретну інформацію, що в свою чергу призводило до дуже плачевних наслідків.

Якщо взяти нинішню ситуацію в світі то наприклад, існує низка тоталітарних країн, які скоюють замах на анонімність, конфіденційність та секретну інформацію в глобальній мережі. Більшість користувачів здатні безпечно висловлювати свою думку, але в таких країнах як Китай ця безпека досягається за рахунок анонімності, но високоймовірно як такі країни дізнались способи швидкої факторизації та дискретного логарифмування то стало би питання про глобальну цифрову безпеку, або навіть фізичну безпеку. Найгірше те, що ми зараз не можемо дізнатись чи існує спосіб швидкого розв'язування цих математичних задач, проте вже відомо як мінімум про один такий алгоритм з використанням квантового комп'ютера. Тому ще відносно недавно з'явилися нові розділи криптографії, як-от постквантова та квантова криптографія. Ці криптографічні системи мають бути стійкі як мінімум до швидких алгоритмів дискретного логарифмування та факторизації, щоб при появі такого алгоритму власнику не стала доступна практично вся інформації із мережі інтернет.

«Хто володіє інформацією, той володіє світом»(Натан Маєр Ротшильд).

Отже, **метою** даної роботи є дослідження способів, методів та технологій для захисту інформації під загрозою існування алгоритмів швидкого розв'язування задач, нерозв'язаність за поліноміальній час яких є гарантом стійкості більшості сучасності криптографічних систем. А саме дослідження квантових та постквантових криптографічних систем.

Для досягнення цієї **мети** потрібно виконати низку наступних задач:

- Розглянути загрози сучасній криптографії, проаналізувати можливості сучасних обчислювальних пристроїв в тому числі квантових комп'ютерів.
- Вибірково дослідити постквантові та квантові системи захисту інформації.

- Провести аналіз розглянутих систем, порівняти їх.
- Оцінити дані розглянутих систем із точки зору перспективи та ресурсозатратності.

**Об’єкт дослідження** – криптологія.

**Предмет дослідження** – постквантові та квантові системи захисту інформації.

**Наукова новизна одержаних результатів** кваліфікаційної роботи полягає у тому, що розглянуті методи захисту інформації є новітніми, мало поширеними та малодослідженими . Квантово криптографічні системи працюють на основі квантової механіки, більшість знань із якої є теоретичними. Та постквантові системи, новий погляд на старі алгоритми дозволив створити методи шифрування, безпечніші та ефективніші ніж поширенні в сьогоденні.

**Практичне значення одержаних результатів.** В першу чергу переконання людей в потребі удосконалення та дослідження методів захисту інформації, як за допомогою криптології так і квантової механіки та фізики. Також отриманні результати вносять ясність в роботу сертифікованих постквантовх шифрів та квантових систем захисту інформації. Це все в свою підвищує безпеку обміну інформації та людський добробут.

**Апробація результатів кваліфікаційної роботи**. Окремі результати проведених досліджень доповідались на XI науково-технічна конференція «Інформаційні моделі, системи та технології», Тернопіль, ТНТУ 13-14 грудня 2023 р.

**Публікації.** За темою роботи з викладенням її основних результатів опубліковано 1-а наукова праця, що являє собою тези в збірнику матеріалів науково-практичних конференцій (див. Додаток А).

# 1 АНАЛІЗ ЗАГРОЗ КРИПТОГРАФІЧНИХ СИСТЕМ ТА ЇХ ЕФЕКТИВНІСТЬ

## 1.1 Аналіз математичної частини загрози

Світ не стоїть на місці, виникають нові проблеми і відповідно знаходяться вирішення таких проблем. У випадку криптографії, математичний апарат, в свій час, дозволив сотворити ефективні засоби захисту інформації але не зміг дозволити розробку ефективних засобів атаки. Проте вже зараз стало питання чи криптографічні системи асиметричного шифрування, початок яким був даний ще в 80-х роках, дійсно математично стійкі до атак чи в один момент нас може очікувати колапс всієї глобальної мережі.

Проблема навкруги якої обертається тема дипломної – проблема рівності класів P і NP(P versus NP problem), одна із найважливіших проблем тисячоліття, яка ставить під питання чи можливо вирішувати задачі класу NP за поліноміальний час або швидше. Якщо взяти до уваги що клас P – клас задача, які можна вирішити за поліноміальний час тоді проблему можна сформулювати коротко «Чи P = NP ?».

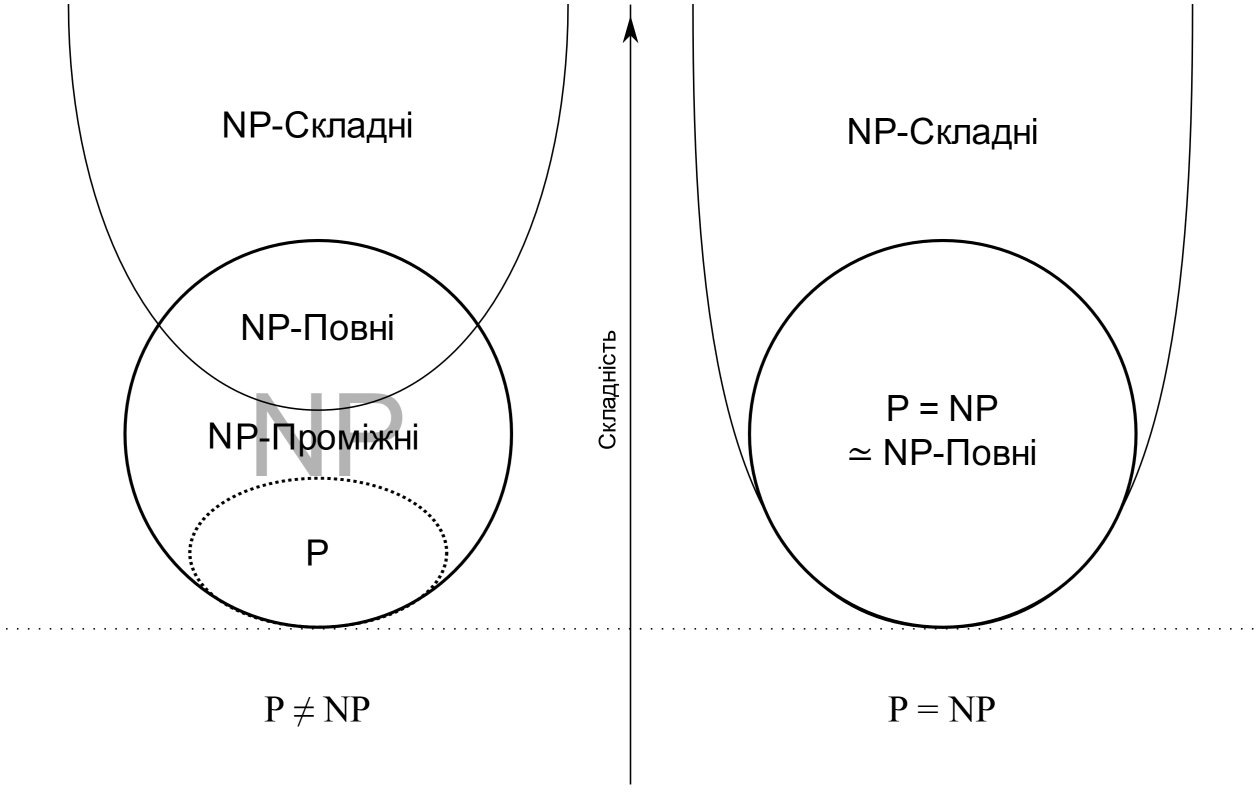


Рисунок 1.1 – Відношення між класами P та NP

*P клас.* P(polynomial-time) клас– клас задач виконання яких можливий за поліноміальний час, тобто за час $\leq O(n^k)$. Однієї із властивостей алгоритмів за поліноміальний час є композиційна закритість. Тобто деякі задачі або знаходження функцій можуть бути розбитті на під задачі або складові функції які також розв'язуються не більше як за поліноміальний час. Це теоретично дає можливість виконання будь якого такого алгоритму навіть на самому слабкому детермінованому комп'ютері за час небільший поліноміального.

В деяких випадках існує підтвердження існування швидкого алгоритму для певної задачі, іншими словами доказ того що ця задача знаходиться в класі P, але сам алгоритм ще не був знайдений. Одним із прикладів є теорема Робертсона-Сеймура про скінченний список заборонених мінорів, ця теорема також стверджує що існує алгоритм складності $O(n^3)$ для перевірки мінора. Таке неконструктивне доведення підтверджує існування алгоритму але сам алгоритм невідомий.

Із прикладів складних алгоритмів класу P: множення матриці(найбільш сучасні оптимізований $O(n^{2.3728596})$, тест простоти AKS(складність $O(\log^6 n)$), алгоритм Дейкстри або найкортошого шляху(складність $O(V^2)$, де V - вершини графу)

*NP клас.* NP(*nondeterministic polynomial-time*) клас – клас задач які мають дві основні властивості що роблять їх належними цьому класу:

- Алгоритм швидкого розв'язку доступний тільки для недетермінованого комп'ютера(машини Тюрінга), окрім задач P класу які входять у NP;
- Алгоритм швидкої перевірки доступний на звичному детермінованому комп'ютері.

Проте, якщо P = NP то всі властивості класу NP дублюються із класом P, точніше ці класи ідентичні.

*Недетермінована машина Тюрінга* – теоретичний пристрій здатний виконувати не детерміновані алгоритми. Якщо звичайна машина Тюрінга змінює один стан на інший послідовно, в не детермінованій машині Тюрінга правила допускають більше як одну дію в конкретній ситуації. Детермінована машина

Тюрінга для знаходження відповіді для NP задачі повинна проходити всі можливі ситуації та їх обраховувати, недетермінована машина Тюрінга в свою чергу здатна паралельно обраховувати багато задач паралельно використовуючи ресурси близькі як для обрахування однієї задачі. Іншими словами ДМТ це рух однією дорогою, повернення до перехрестя і потім рух іншою, а НДМТ це рух всіма дорогами одночасно  із такою самою швидкістю як у випадку ДМТ. Симуляція НДМТ є задачею експоненціальної складності $O(x^n)$.

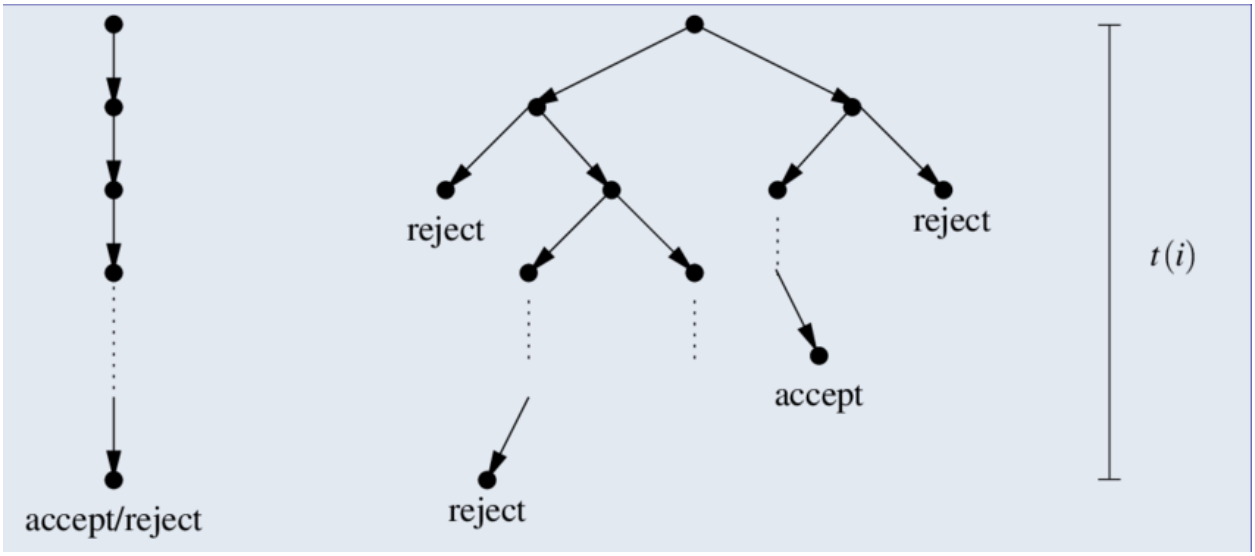


Рисунок 1.2 – ДМТ та НДМТ

Приклади NP задач: P задачі, задача комівояжера, перевірка розв’язаності логічного виразу, пакування рюкзаку.

*NP-Повні.* NP – повні(*nondeterministic polynomial-time complete*) це найскладніші задачі які входять до класу NP. Зараз відомо що тільки НДМТ здатна швидко обраховувати задачі цього класу. Було доведено що любу задачу NP класу можна швидко звести у NP-Повну задачу, також задачі цього класу швидко зводяться одна в іншу. Відповідно якщо знайти алгоритм швидкого розв’язку або довести його існування для звичайного детермінованому комп’ютері для любої із задач NP-Повного класу це підтвердить рівність P=NP.

Приклади NP-Повних задач: здійсненність булевих функцій(SAT), задача листоноші, задача модуляроності мереж.

*NP-Проміжні*. Задачі класу NP які не змогли віднести до класів P та NP-Повні. Існування цього класу можливе при умові P≠NP, відповідно доведення

того що даний клас порожній за теоремою Ладнера доводить що P=NP. Всі задачі які прийнято рахувати NP-Проміжними ймовірно знаходяться в цьому класі, можливо в подальшому будуть віднесені до класу P або NP-Повні.

Приклад NP-Проміжних задач: задача тривіальності вузлів, задача визначення ізоморфізму двох графів, задача лінійної подільності.

*NP-Складні*. Клас задачі для яких невідомі алгоритми розв'язку ні детерміновані, ні недетерміновані, ні ймовірнісні, окрім NP-Повних задач які входять у клас NP-Складних. Алан Тюрінг ще в 1936 році довів що не існує алгебраїчного розв'язку проблеми зупинки. Проблема зупинки є однією із доведено нерозв'язних задач, тому якщо задача зводиться до проблеми зупинки, ця задача входить у NP-Складний клас.

Приклад NP-Складних задач: NP-Повні завдання, десята проблема Гільберта(задача цілочислових розв'язків Діофантових рівнянь), проблема зупинки(halting problem)

Також окремо варто виділити класи BPP та BQP які безпосередньо зачіпають тему роботи. $P \subseteq BPP \subseteq BQP$

BPP(*bounded-error, probabilistic, polynomial-time*) клас - задачі які вирішуються за поліноміальний час ймовірнісними алгоритмами, помилка при використанні цих алгоритмів повинні бути не більші 1/3, тобто точністю неменше 2/3. На практиці такі алгоритми є дуже ефективні в вирішенні деяких складних завдань.

BQP(*bounded-error quantum polynomial-time*) клас – задачі які швидко вирішуються ймовірнісним методами на квантовому комп'ютері із точністю не менше 2/3.

## 1.2 Оцінка обчислювальних потужностей.

Винайдення квантового комп'ютера та поява алгоритму Шора дали поштовх для розробки постквантові криптографії. В сьогоденні не відомо підтвердженого ймовірнісного квантового алгоритму знаходження дискретного логарифму, проте відомо алгоритм факторизації цілих чисел за час $O(\log^3 n)$, n –

число яке потрібно розкласти на множники. Для зрівняня одні з найшвидших алгоритмів множення мають складність O(n log(n) log(log(n)) ), де n – кількість бітів найбільшого із множників. В свій час Петер Шор висунув припущення що підхід застосований в його алгоритмі може бути використаний для знаходження дискретного логарифму за поліноміальний час. Все частіше і частіше з'являються публікації робіт математиків які намагаються довести існування алгоритмів швидкого обчислення дискретного логарифму та дискретного логарифму на еліптичних кривих.

Одним із відкритих питань є питання рівності потужностей обчислення НДМТ та Квантового комп'ютера,  це ставить під питання безпеку сучасних криптографічних систем. Якщо виявиться, що P = NP або квантовий комп'ютер є альтернативою НДМТ тоді навіть методи і способи які використовуються в пост квантовій криптографії не будуть мати ніякого значення. В такому випадку доцільним буде використання тільки квантової криптографії. Однак велика кількість авторитетних математиків схиляється до домку що все-таки P≠NP, а квантовий комп'ютер набагато слабший за НДМТ. Проте ні перше ні друге твердження не було доведене, тому ризики залишаються і насправді ризик дуже великий.

Рекомендований модуль для RSA складає 4096 біта, відповідно розмір простого числа буде розміром не більше 2048 біт, якщо грубо порівняти то складність факторизації буде приблизно у $5 \times 10^5$-раза складніше. Щоб квантовий комп'ютер зміг швидко факторизувати 4096-бітне число йому потрібно приблизно у два рази більше кубітів. Проте із року в рік квантові комп'ютери збільшують кількість кубітів. Осінню 2022 року найпотужніший комп'ютер IBM Osprey використовував 433 кубіти, в цьому, 2023 році також осінню з'явився комп'ютер який містить більше ніж у два рази кубітів, 1180 кубітів, Atom Computing's.

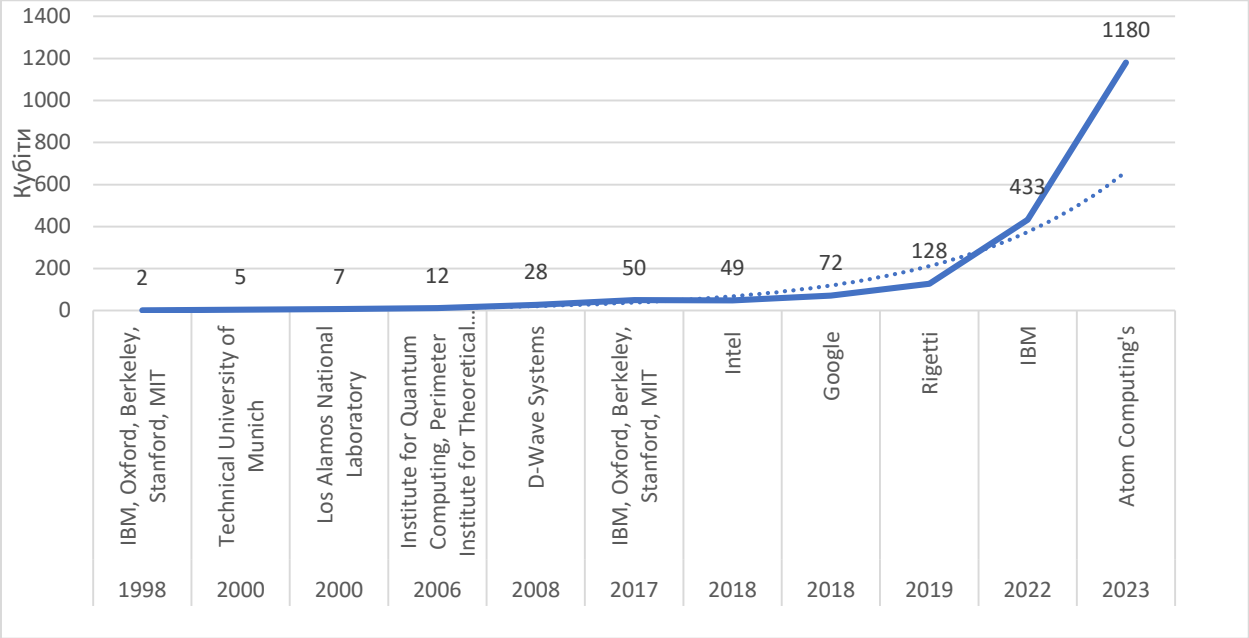


Рисунок 1.3 – Кількості кубітів у КК

Як видно із графіку починаючи із 2019 року почався стрімкий ріст кількості кубітів які використовуються в квантових комп'ютерах. З іншої сторони із року в рік змінюється оцінка кількості ресурсів потрібних для злому ключа, та змінюється вона на користь КК. Як от, у 2009 році за оцінкою Ван Метра та його колег, КК із $6.5 \times 10^9$ фізичних кубітів зможе зламати ключ RSA-2048 за 410 днів, вже у 2012 році Фаулер виснув припущення що із $10^9$ кубітами ключ може бути зламаний за 1.1 дня, а у 2021 році оцінили, що $2 \times 10^7$ кубітів будуть в змозі зламати ключ, всього лиш, за 0.31 дня. Як бачимо із оцінка ефективності КК все вища і вища, при тому, що кількість кубітів потрібних для злому падає але швидкість злому все зростає.

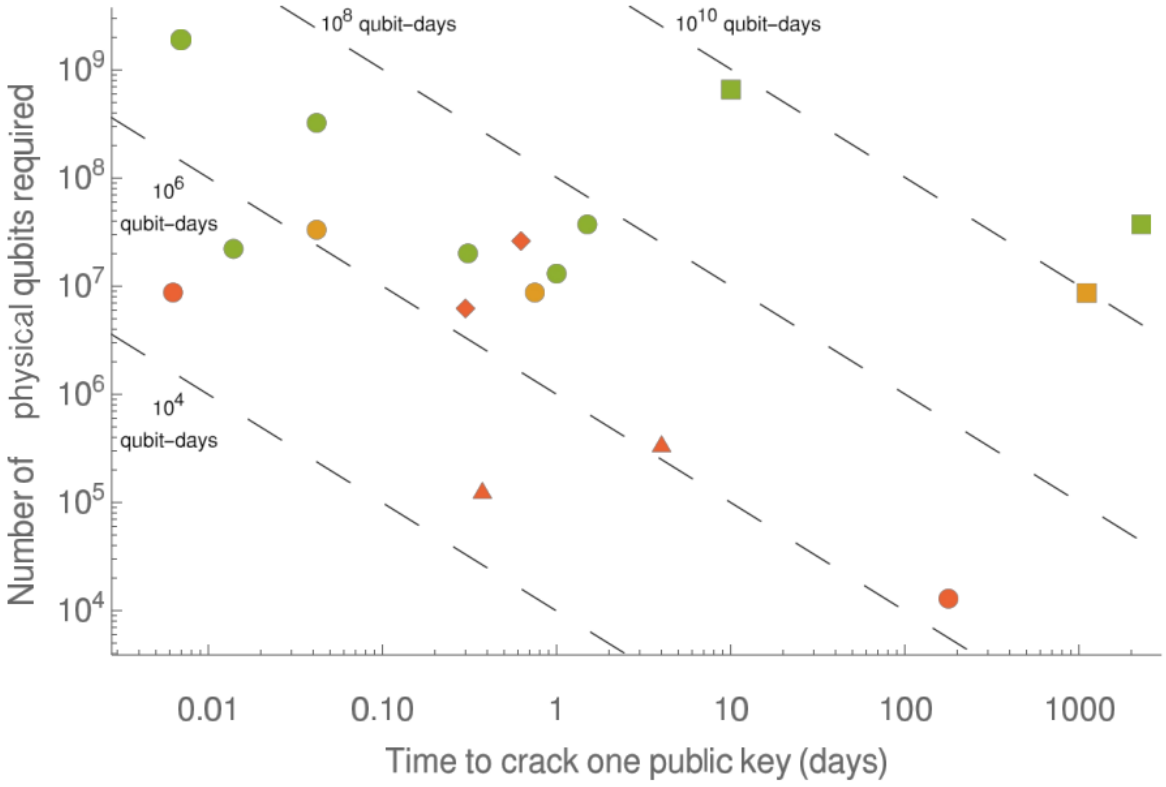


Рисунок 1.4 – Стійкість RSA-2048

Кольори маркерів позначають три сценарії розвитку технологій КК: амбітні сценарії (зелений), більш амбітні сценарії (жовтий) або найамбітніші сценарії (червоний). Фігури ринків позначають архітектури кубітів: надпровідний-трансмон (кола), захоплений іон(квадрати), топологічна (ромби) або котяча(трикутники).

Закон Мура стверджує що кількість транзисторів збільшується у двічі кожних 2 роки. Можна зробити висновок що достатньо збільшувати розмір ключа шифрування, щоб зробить шифри стійкими проти квантових комп’ютерів, проте кількість транзисторів на одинцю об’єму не може зростати нескінченно. Також є проблема із енергоспоживаннями, хоча ефективність транзисторів і економність сильно зростає це не покриває зростаючі потреби в енергоспоживання обчислювальних систем.

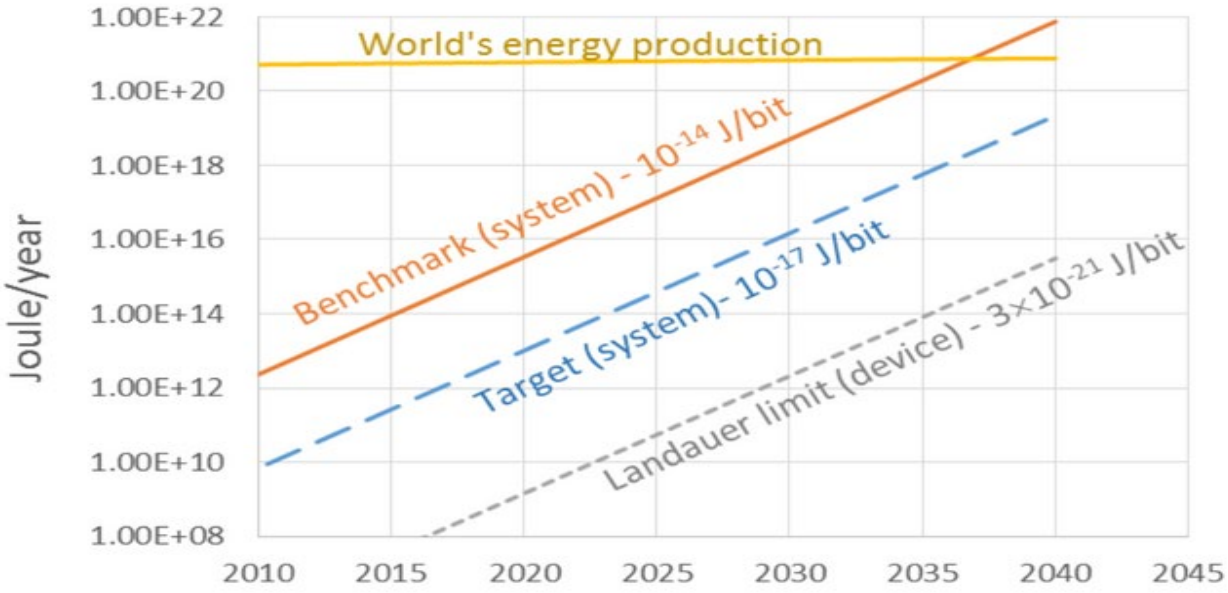


Рисунок 1.5 – Графік енергоспоживання ОС

На графіку видно, що такий розвиток обчислювальних систем в 2038 році, без підвищення генерації енергії та оптимізації, будуть поглинали всю енергію яку виробляє людство, а збільшення ключів для шифрування тільки буде збільшувало енергоспоживання. Але обчислювальні системи, які забезпечують працездатність глобальної мережі, не є першими у пріоритеті надання енергії. Іншими словами у майбутньому кількість обчислень може бути обмежена нестачею енергії для роботи цих систем.

## 1.3 Прогнозування загроз пов’язаних із захистом інформації

Всі вище вказані вразливості та можливі загрози, можливо, і не потребувало би негайної уваги, проте людство зараз знаходиться на межі так званої техногенної синуглярності, різкого стрибку людства у розвитку науки та технологій. Якби не це то можна було би залишити цю проблему без такої кількості уваги, все це знаючи що такі проблеми як рівність класів P і NP і інші проблеми тисячоліття потребують відносно велику кількість часу для вирішення. Причинна техногенної синуглярності тільки одна, розвиток штучного інтелекту. Є дуже багато прикладів тому що ШІ може бути розумніший і ефективніший у вирішенні деяких задач ніж людина. Багато фахівців із ШІ та науковців

наголошують на тому що іменно сам ШІ і є найбільшою небезпекою, проте небезпеку несуть і знання які можна получити від ШІ. Математичний ШІ вже здатний вирішувати  деякі головоломки та дав відповідь на питання, на які людство довго не могло відповісти. На певні задачі важко відповісти по причинні антропоморфного фактору, тобто більшість людей схильні мислити однаково, тому навіть на перший погляд прості задачі залишаються нерозв'язаними. ШІ такої вади не має або може її легко перебороти, у зв'язку з можливістю в реальному часі перелаштовувати свої принципи обчислення інформації. Еволюція ШІ йде в реальному часі, тому швидше за все ШІ зможе покращувати сам себе і це буде відбувалось дуже швидко. Відповідно, кількість нової і не зрозумілої інформації буде дуже велика, це несе велику загрозу так як людство інтелектуально не готові використовувати можливості надані ШІ-ом[1]. Можливо ШІ доведе, що P≠NP але в той самий час придумає першу НДМТ, можливо ШІ придумає більш сучасні криптографічні методи та в свою чергу і сам знайде спосіб їх обійти. Тому потрібно підстрахуватись принаймні від загроз які вже видно на горизонті, інакше це приведе до колапсу інформаційно-комунікаційної системи або не приведе до такого швидкого колапсу.

Існування вище вказаних загроз несе небезпеку людському добробуту, так як порушує функціонал низки галузей: ведення комерційної діяльності, державні справи, військові дії та особисті справи залежать від існування загальноприйнятих засобів автентифікації особи, повноважень, власності, ліцензії, підпису, нотаріального засвідчення, дати дії, отримання тощо. У минулому це майже повністю залежало від фізичних документів і протоколів створення цих документів, тобто від їхньої автентифікації. Суспільство розвивається і приймає складні набори юридичних і криміналістичних процедур, що майже повністю залежать від речових доказів, притаманних самим документам, для вирішення спорів щодо автентичності. В інформаційну епоху, однак, володіння, контроль, передача або доступ до реальних активів часто ґрунтується на електронній інформації, а ліцензія на використання, модифікацію або поширення цінної інформації визначається аналогічним чином. Таким чином, важливо, щоб внутрішні докази були присутні в самій інформації - оскільки це

єдине, відомий допустимий варіант. Сучасна криптологія, таким чином, повинна забезпечувати всі функції, які зараз виконують документи - публічні та приватні. Насправді, вона часто повинна робити більше. Коли хтось надсилає документ поштою з проханням надати квитанцію про доставку, квитанція лише доводить, що конверт було доставлено; вона нічого не говорить про вміст. Однак цифрові сертифікати походження та цифрові квитанції нерозривно пов'язані з кожним електронним документом. Багато інших функцій, таких як підписи, також набагато складніші в цифровому середовищі. Тому загроза із боку КК є загрозою для багатьох аспектів суспільного життя які опираються на методи криптології.

Пост квантова криптографія є відповіддю на загрозу яку несе квантовий комп'ютер. Практичне використання КК зараз є сильно обмеженим, проте все може змінитись під впливом ШІ та стрімкого росту наукових досягнень. Було придумано велику кількість алгоритмів ПКШ проте кожен із цих алгоритмів має недоліки які не дають змоги на практиці використовувати їх так легко як стандартні алгоритми асинхронного шифрування. Однієї із найбільших вад є то що при умові не поганої стійкості до атаки із використанням КК(квантового комп'ютера) ці алгоритми є не стійкі для звичних атак та потребують великого розміру ключа. Поки, що NIST(The National Institute of Standards and Technology) стандартизував 4 алгоритми PQC. За 3 раунди 2016-2019, 2019-2020 та 2020-2022 було відібрано для стандартизації тільки один алгоритм поширення ключа та 3 алгоритми підпису[2]:

Алгоритми цифрового підпису: CRYSTALS-Dilithium , FALCON (на основі ґраток); SPHINCS+(на основі хешу).

Алгоритми KEM(інкапсуляції ключа): CRYSTALS-Kyber(на основі ґраток).

На 4 раунд, який почнеться вже відібрано декілька кандидатів: McEliece, HQC, SIKE, BIKE та можливо їх появиться більше.

## 1.4 Висновок до першого розділу

В першому розділі описано загрози сучасним криптографічним системам, та їхня ефективність, також розглянуті питання які на пряму не зв'язані із темою

роботи але в недалекому майбутньому можуть нести небезпеку для систем захисту інформації.

Досліджено теоретичну частину проблеми «P=NP?», порівняно ефективність квантових та звичайних комп’ютерів, використання квантового комп’ютера для атак на поширенні алгоритми шифрування.

Незалежно чи з точки зору математики, чи з технологій, все-одно виникає загроза сучасним криптографічним системам. Якщо криптографія розглядає загрозу існування квантових комп’ютерів, які здатні вирішувати задачі далеко за межами P класу, як загрозу, то з точки зору математики сама загроза є стан недоведеності твердження P≠NP. Також на фоні всього  всіх вище вказаних проблем, є одна яка пов’язана  із стрімким розвитком технологій та ШІ.

# 2 ДОСЛІДЖЕННЯ ПЕРСПЕКТИВНИХ МЕТОДІВ ПОСТКВАНТОВГО І КВАНТОГО ЗАХИСТУ ІНФОРМАЦІЇ

У даному розділі будуть розглянуті провідні постквантові алгоритми шифрування та квантові системи які забезпечують безпечний обмін інформацією.

Із постквантових алгоритмів варта відзначити 4 алгоритми які пройшли сертифікацію NIST, та є готовими для глобального використання: SPHINCS+, FALCON, CRYSTALS-Kyber та CRYSTALS-Dilithium. Перші 3 алгоритми будуть вивчені та проаналізовані в у цьому розділі.

Є дві теоретичні квантові системи, які є найбільш реалізаційна перспективними : MDI-QKD та супутниковий QKD(SatQKD). Ці системи будуть проаналізовані із точки зору квантової механіки та квантових схем.

## 2.1 SPHINCS+

SPHICNS+ - це відносно новий алгоритм підпису(DSA), за стандартом EUF-CMA, який був розроблений Шнайдером, Бернштейном, Ланге, Хопвудом, Нідерхагеном, Хюльсінгом, Папакрістодулу, Швабе та Вілкоксом-О'Хірном як схема підпису на основі хеш-функції і була першою схемою підпису, яка запропонувала параметри, стійкі до квантового криптоаналізу[3]. SPHINCS використовує багато компонентів із схеми підпису Меркеля(XMSS), але працює з більшими ключами і підписами для усунення недоліків XMSS. Стандартизований NIST’ом у 2022 році.

Основна ідея полягає в тому, щоб автентифікувати величезну кількість пар ключів з одноразовим підписом (FTS) пар ключів за допомогою так званого гіпердерева. Схеми FTS - це схеми підпису, які дозволяють парі ключів створювати невелику кількість підписів, наприклад, порядку десяти для наших наборів параметрів. Для кожного нового повідомлення вибирається (псевдовипадкова) пара ключів FTS для підписання повідомлення. Підпис складається з підпису FTS та автентифікаційної інформації для цієї пари ключів FTS. Автентифікаційна інформація є приблизно гіпердеревовидним підписом,

тобто підписом з використанням сертифікаційного дерева підписів дерева Меркла.

Більш конкретно, гіпердерево - це дерево багаторазових підписів на основі хешу (MTS). Ці багаторазові підписи дозволяють парі ключів підписувати фіксовану кількість повідомлень N - для SPHINCS+ N є степенем 2, наприклад $2^8 = 256$. Самі пари ключів MTS організовані у вигляді N-арного дерево з d рівнями. На верхньому рівні d - 1 знаходиться одна пара ключів MTS, яка використовується для підпису відкритих ключів N пар ключів MTS, які утворюють рівень d - 2. Кожна з цих N пар ключів MTS використовується для підписання інших N відкритих ключів MTS, що утворюють рівень d - 3. І так далі до Nd-1 пар ключів на нижньому рівні, які використовуються для підписання N відкритих ключів FTS, кожна з яких призводить до загальна кількість Nd автентифікованих пар ключів FTS. Автентифікаційна інформація для пари ключів FTS складається з d підписів MTS, які будують шлях від пари ключів FTS до вершини MTS-дерева.

Підпис MTS - це просто класичний підпис дерева Меркла у випадку SPHINCS+. Він складається з одноразового підпису (OTS) на даному повідомленні плюс шлях аутентифікації в бінарному хеш-дереві, що авентифікує N пар ключів OTS однієї пари ключів MTS. Відкритий ключ SPHINCS+ по суті є відкритим ключем MTS верхнього рівня, який є лише кореневим вузлом його двійкового хеш-дерева, а отже, єдиним хеш-значенням. Однак, фактичні SPHINCS+ додатково містять публічне початкове значення тієї ж довжини, що і кореневий вузл. Це пов'язано із деякими технічними причинами. Секретний ключ SPHINCS+ - це лише одне секретне початкове значення. Таким чином, всі OTS, OTS-секретні ключі і 5 секретних ключів FTS генеруються псевдовипадковим чином. OTS-секретні ключі та FTS разом повністю визначають всю віртуальну структуру пари ключів SPHINCS+. Знову ж таки, фактичні секретні ключі SPHINCS+ містять додаткове секретне значення того ж розміру, що і секретне насіння(SR.seed),а також і копію відкритого ключа. Додаткове значення використовується для PRF-ключа, який використовується в псевдовипадковому хешуванні.

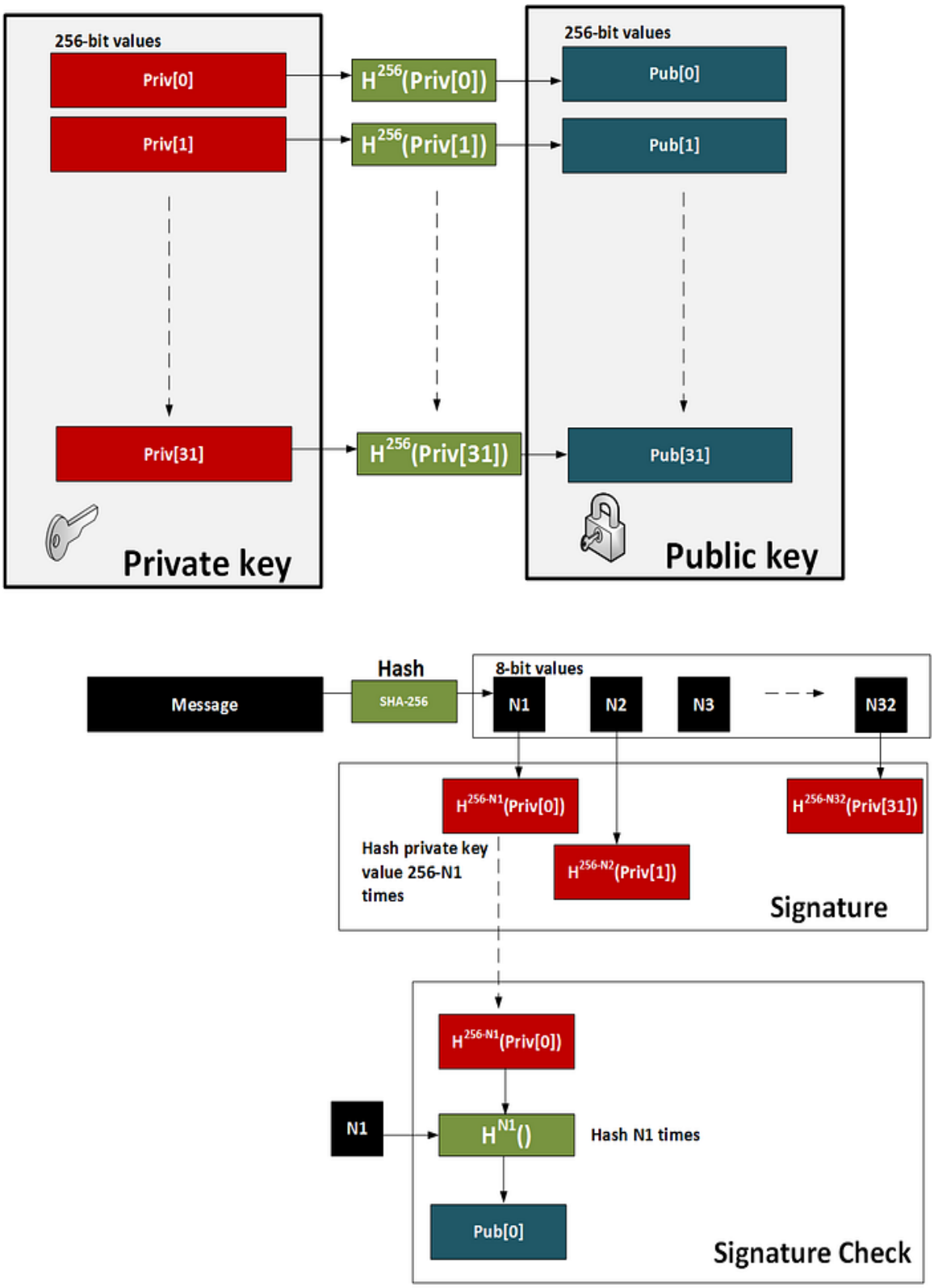


Рисунок 2.1.1 – Алгоритм SPHINCS+ 256 SHA256

Основні параметри криптосистеми: *n* – основний секретний параметри(байт), є вихідною довжиною всіх криптографічних функцій, тому визначає, якого рівня безпеки досягає набір параметрів;

- $h$ – висота гіпердерева, визначає кількість екземплярів FORS ;
- $w$ – параметр Вінтренітза, визначає кількість і довжину хеш-ланцюжків для кожного екземпляру WOTS+;
- $d$ – кількість шарів гіпердерева XMSS, є параметром продуктивності, не впливає на безпеку;
- $k$ – кількість дерев в FORS;
- $t$ – кількість рівні FORS-дерев, $t = 2^a$ , $a > 0$, $a \subset Z$;
- m – довжина повідомлення;
- len – n-байтовий стрічковий елемент;
- $m = \lfloor (k \log t + 7)/8 \rfloor + \lfloor (h - h/d + 7)/8 + \lfloor (h/d + 7)/8 \rfloor$;
- len = len1 + len2, $len1 = \left\lceil \frac{8n}{\log(w)} \right\rceil$, $len2 = \left\lfloor \frac{\log(len1(w-1))}{\log(w)} \right\rfloor + 1$;

Дальше будуть розглянуті основі модулі підпису SPHINCS+: WOTS+, FORS та Hypertree.

*WOTS+*. Базовою схемою є WOTS+, одноразовий підпис Вінтерніца - розширення одноразового підпису Лампорта, що використовується для підписання групи байтів фіксованої довжини за допомогою рекурсивної ланцюгової функції, яка викликає змінну хеш-функцію F, і контрольної суми з метою зробити підпис неможливим для підробки[4].

WOTS+- це схема одноразового підпису: приватний ключ повинен бути використовуватися для підписання лише одного повідомлення. При повторному використанні для підписання декількох повідомлень, безпека швидко погіршується.

WOTS+ має два параметри n і w. n - це параметр безпеки; це довжина повідомлення, а також довжина елемента приватного ключа, елемента відкритого ключа і елемента підпису в бітах. w - це параметр Вінтерніца; він може бути використаний для досягнення оптимізації між часом підписання і розміром підпису: більше значення означає менший, повільніший підпис. w зазвичай обмежується значеннями 4, 16 або 256.. len, представляє кількість n-бітових значень у нестисненому WOTS+ закритому ключі, відкритому ключі та підписі.

У контексті SPHINCS+ закритий ключ WOTS+ отримується з секретного насіння SK.seed, яке є частиною закритого ключа SPHINCS+, та адреси пари ключів WOTS+ у гіпердереві, використовуючи PRF. Відповідний відкритий ключ отримується шляхом ітеративного застосування F або w повторень до кожного з n-бітових значень у закритому ключі, ефективно створюючи len хеш-ланцюги. Тут F параметризується адресою пари ключів WOTS+, а також висотою виклику F і його конкретного ланцюжка, на додаток до seed PK.seed, який є частиною відкритого ключа SPHINCS+. Ми не використовуємо так звані ℓ-дерева для стиснення відкритого ключа WOTS+. Замість цього, відкритий ключ стискається до n-бітового значення. за допомогою одного виклику $Th_{len}$-хеш-функції з можливістю налаштування. Ми використовуємо "відкритий ключ WOTS+" для позначення стисненого відкритого ключа.

Для підпису та перевірки вхідне повідомлення m інтерпретується як len1 цілих чисел $m_i$ , від 0 до w - 1. Ми обчислюємо контрольну суму $C = \sum_{i=1}^{len_1}(w - 1 - m_i)$ над цими значеннями, представлену у вигляді рядка з len2 значень з основою w C = (C1, ... ,Clen2). Ця контрольна сума необхідна для того, щоб запобігти підробці повідомлення: збільшення хоча б одного mi призводить до зменшення хоча б одного $C_i$ і навпаки. Використовуючи ці len цілих чисел як довжину ланцюга, до елементів закритого ключа застосовується ланцюгова функція F. Це призводить до len n-бітових значень, які складають підпис. Потім верифікатор може перерахувати контрольну суму, отримати довжини ланцюжків і застосувати F для завершення кожного ланцюжка до його повної довжини. Це призводить до отримання голів ланцюжка, які хешуються за допомогою $Th_{len}$ для обчислення n-бітового відкритого ключа

*Гіпердерево(Hypertree).* Одне дерево. Щоб мати можливість підписати $2^h$ повідомлень, підписувач отримує $2^{h'}$ відкритих ключів WOTS+. Ми використовуємо ці ключі як вершини листя. Для побудови бінарного дерева ми багаторазово застосовуємо H до пар вузлів, параметри яких містять унікальну адресу цього застосування H, а також публічне насіння PK.seed. Ми вважаємо, що таке дерево має висоту $h'$ , що відповідає кількості застосувань H для переходу від листя до кореня. Корінь цього дерева є тим, що зараз буде коротко

слугувати публічним ключем схеми одного дерева. Один з вузлів листя WOTS+ використовується для створення підпису на n-бітному повідомленні. Простої публікації підпису WOTS+ недостатньо, оскільки для перевірки також потрібна інформація про решту дерева. Для цього підписувач включає так званий "шлях автентифікації". Верифікатор спочатку отримує відкритий ключ WOTS+ з підпису, а потім використовує вузли, включені в шляху автентифікації, щоб відновити кореневий вузол.

Для того, щоб зробити малоймовірним, що випадковий вибір листового вузла неодноразово призводить до того, що той самий листовий вузол буде обрано SPHINCS-дерево має бути значно більшим за розміром. Замість того, щоб збільшувати h (і нести непереборні витрати на обчислення $2^{h'}$ WOTS+ відкритих ключів на кожну операцію підпису), ми створюємо гіпердерево. Ця конструкція слугує деревом сертифікації. Листяні вузли WOTS+ дерев на нижньому рівні використовуються для підписання повідомлень (або, в нашому випадку, відкритих ключів FTS), тоді як листові вузли дерев на всіх інших рівнях використовуються для підписання кореневих вузлів дерев, розташованих нижче. Підписи WOTS+ та шляхи автентифікації від листка внизу гіпердерева до кореня самого верхнього дерева утворюють шлях автентифікації. Важливо, що всі вузли листя всіх проміжних дерев мають детерміновано згенеровані відкриті ключі WOTS+, які не залежать від жодного з дерев, розташованих нижче. Це означає, що повне гіпердерево є суто віртуальним: його ніколи не потрібно обчислювати повністю. Під час генерації ключа для отримання відкритого ключа обчислюється лише найверхнє піддерево. Ми визначаємо загальну висоту дерева h і кількість проміжних шарів d, заднім числом встановлюючи $h' = h/d$.

*FORS.* В якості схеми з кількома сигнатурами у SPHINCS+ ми визначаємо FORS: Ліс випадкових підмножин (Forest of Random Subsets), що є покращенням HORST із SPHINCS. FORS визначається в межах цілих чисел k і $t = 2^a$, і може бути використана для підписання рядків з k-а бітів.

Закритий ключ FORS складається з kt випадкових n-бітних значень, згрупованих у k наборів по t значень у кожному. У контексті SPHINCS+ ці значення детерміновано отримуються з SK.seed за допомогою PRF та адреси

ключа у гіпердереві. Для побудови відкритого ключа FORS ми спочатку будуємо k двійкових хеш-дерев над наборами елементів закритого ключа. Кожне з t значень використовується як листковий вузол, в результаті чого утворюється k дерев висотою a. Ми використовуємо H, що адресується за розташуванням пари ключів FORS у гіпердереві та унікальною позицією виклику хеш-функції в дереві FORS. Як і у WOTS+, ми стискаємо кореневі вузли за допомогою виклику $Th_k$. Отримане n-бітне значення є відкритим ключем FORS.

За заданим повідомленням з ka бітів ми виділяємо k рядків з a бітів. Кожен з цих рядків бітів інтерпретується як індекс однолистої вершини у кожному з k дерев FORS. Підпис складається з цих вузлів і відповідних шляхів автентифікації. Верифікатор реконструює кожен з кореневих вузлів, використовуючи шляхи автентифікації, і використовує $Th_k$ для реконструкції відкритого ключа. В рамках SPHINCS+ підпис FORS ніколи не перевіряється в явному вигляді. Замість цього, отриманий відкритий ключ використовується як повідомлення, яке неявно перевіряється разом з підписом WOTS+.

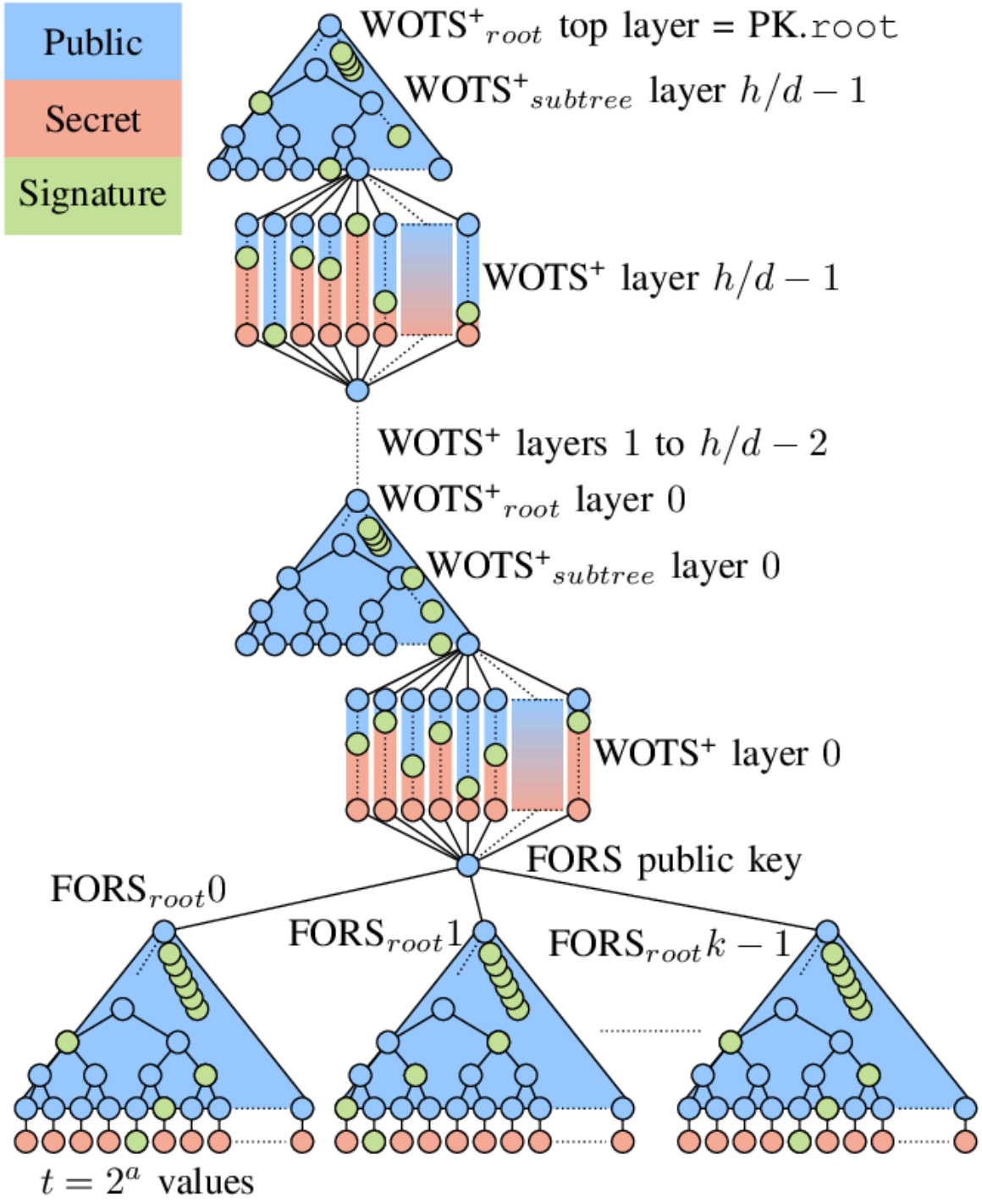


Рисунок 2.1.2 – Схема роботи WOTS+ та FORS

*SPHINCS+*.Майже всі елементи, які складають пару ключів SPHINCS+, були неявно розписані вище. Відкритий ключ складається з двох n-бітових значень: кореневого вузла верхнього дерева гіпердерева та випадкового відкритого ключа PK.seed. Крім того, приватний ключ складається ще з двох n-бітних випадкових значень: SK.seed, для генерації секретних ключів WOTS+ і FORS, та SK.prf, що використовується нижче для випадкового дайджесту повідомлень[5].

Загальний підпис складається з підпису FORS на дайджесті повідомлення, підпису WOTS+ на відповідному відкритому ключі FORS і серії шляхів автентифікації та підписів WOTS+ для автентифікації цього відкритого ключа WOTS+. Щоб перевірити цей ланцюжок шляхів і підписів, верифікатор ітеративно реконструює відкриті ключі і кореневі вузли, поки не буде досягнутий кореневий вузол на вершині гіпердерева SPHINCS+.

Спочатку ми псевдовипадковим чином генеруємо R на основі повідомлення та SK.prf. R можна зробити недетермінованим, додавши хаотичність використавши випадкове число як аргумент для генерації(Rand) R. Це може протидіяти атакам побічних каналів. Встановлення цього значення у повністю нульовий рядок (або використання значення з низькою ентропією) не впливає негативно на псевдовипадковість R. Формально, ми говоримо, що R = PRF(SK.prf, Rand, M). R є частиною підпису. Використовуючи R, ми отримуємо індекс вузла листа, який буде використано, а також дайджест повідомлення у вигляді (MD || idx) = $H_{msg}$(R, PK.seed, PK.root, M). На відміну від SPHINCS, цей метод вибору індексу є публічно верифікованим, що не дозволяє зловмиснику вільно вибирати випадковий індекс і комбінувати його з повідомленням на свій розсуд. Дуже важливо, що це протидіє багатоцільовим атакам на схему підпису декількох разів. Оскільки індекс тепер може бути обчислений верифікатором верифікатором, він більше не включається в підпис.

Для хешування WOTS+ та FORS можна використовувати довільну хеш функцію. Розмір хешу фнукції має бути рівним параметру n. Наприклад, можна вибрати вже нові сертифіковані NIST, як-от, SHA-3, BLAKE2[5]. Рекомендується використовував швидкі хеш функції, швидкість обчислення хеш функції експоненціально впливає на швидкість SPHINCS+ також для досягнення повної стійкості до квантового криптоаналізу бажано використовувати пост квантові хеш функції, як приклад, Haraka[5].

Таблиця 2.1.1 – Бенчмарк SPHINCS+ Haraka

| | key generation | signing | verification |
|---|---|---|---|
| SPHINCS $^+$-Haraka-128s-simple | 30075604 | 240763926 | 308774 |
| SPHINCS $^+$-Haraka-128s-robust | 37113806 | 304905780 | 432066 |
| SPHINCS $^+$-Haraka-128f-simple | 482332 | 12196792 | 799808 |
| SPHINCS $^+$-Haraka-128f-robust | 587548 | 15176760 | 1072774 |
| SPHINCS $^+$-Haraka-192s-simple | 46369950 | 481682614 | 480264 |
| SPHINCS $^+$-Haraka-192s-robust | 63387838 | 718896354 | 759952 |
| SPHINCS $^+$-Haraka-192f-simple | 732770 | 21433286 | 1205698 |
| SPHINCS $^+$-Haraka-192f-robust | 998446 | 30866288 | 1799300 |
| SPHINCS $^+$-Haraka-256s-simple | 28822310 | 451164660 | 696980 |
| SPHINCS $^+$-Haraka-256s-robust | 40954800 | 677039436 | 1046096 |
| SPHINCS $^+$-Haraka-256f-simple | 1809078 | 41973226 | 1252598 |
| SPHINCS $^+$-Haraka-256f-robust | 2599368 | 61706762 | 1854540 |

Як видно із Таблиці 2.1.3 SPHINCS+ є повільним алгоритмом підписання, також недоліком є розмір підпису.

Із плюсів можна відмітити:

- Малий розмір відкритого так закритого ключа.
- Стійкість до State-of-the-art attacks та квантового алгоритму Ґровера.
- SPHINCS+ спирається на давно відомі математчині принципи які вже були перевірені часом, тому відомий вектор атак які можуть здійснюватись.
- Швидкість сильно залежить від хеш функції, тим самим підбір швидкої хеш функції сильно пришвидшує SPHINCS+.
- Теоретично XMSS може бути повністю замінена на SPHINCS+.
- Стійкість до колізій.
- Багаторазове використання вже сформованих блоків шифру.

Аналіз стійкості алгоритму. Distinct-Function Multi-Target Second-Preimage(DFMTSP) resistance - це властивість хеш-функції, яка означає, що обчислювально неможливо знайти будь-який другий відмінний вхід, який має такий самий вихід, як і даний вхід, для будь-якої з функцій, які використовуються в хеш-функції. Ця властивість є важливою для безпеки деяких криптографічних схем, таких як схеми цифрового підпису.

Для оцінки складності типових атак на властивості хеш-функцій, хеш-функції зазвичай моделюються як (сімейство) випадкових функцій. Для

випадкових функцій немає різниці між стійкістю до окремих функцій та стійкістю до багатофункціональних функцій. Кожен ключ просто вибирає нову функцію пробігу, незалежно від того, чи є ключ випадковим, чи ні. Ймовірність успіху будь-якого класичного $q_{hash}$-запиту проти DFMTSP до другого перетворення випадкової функції з діапазону $\{0, 1\}^{8n}$ дорівнює $\frac{q_{\text{hash}}+1}{2^{8n}}$. Для квантових комп'ютерів з $q_{hash}$-запитами ймовірність успіху дорівнює $\Theta\left(\frac{(q_{\text{hash}}+1)^2}{2^{8n}}\right)$. Такою самою буде і складність атаки на псевдовипадковий генератор.

## 2.2 Falcon

Falcon – алгоритм підпису(DSA), за стандартом EUF-CMA був розроблений шляхом об'єднання декількох робіт, включаючи схему підпису на основі решітки NTRUSign (криптосистеми GGH, де першими запропонували схему підпису на основі решітки), яка мала недолік у своїй процедурі підпису. У 2008 році був запропонували метод, який виправляє цей недолік і забезпечує загальну основу для побудови безпечних схем підпису на основі хешу і решітки. Пізніше був об'єднаний фреймворк GPV з NTRU для створення достовірно безпечного підпису на основ решіток, потім був запропонований новий алгоритм для вирішення проблеми повільної роботи алгоритму. У 2022р. стандартизований NIST'ом.

Стійкість алгортму Falcon спирається на алгебраїчні - решітки NTRU. Це дозволяє звести ключі до поліномів степеня n ($n = 2^k$). Обчислення виконуються за модулем нормованого полінома $\phi$ степеня n, $\phi = x^n + 1$. В алгоритмі поліноми розглядаються як вектори та матриці. Для малого простого числа $q \in N$, у FALCON $q = 12289$, нехай Zq - кільце коефіцієнтів Z/qZ. $\sigma$ – стандартне відхилення, $\sigma \approx 1.55\sqrt{q} \ll q$, знаходимо $\lfloor \beta^2 \rfloor$, $\beta = 1.1 \cdot \sigma\sqrt{2n}$ NTRU-ґратки будуються за допомогою поліномів f, g, F, G $\in Z[x]/(\phi)$ та рівняння NTRU $fG - gF = q(mod\phi)$, якщо $f^{-1}$ обернене до $f$ існує то відкритий ключ обраховується за

формулою $h = gf^{-1} mod(\phi, q)$. Також щоб можна було знай G, F. Має бути задовільна така умова[6]:

$$\gamma = max\left\{\| (g, -f) \|, \left\|\left(\frac{qf^{\star}}{ff^{\star}+gg^{\star}}, \frac{qg^{\star}}{ff^{\star}+gg^{\star}}\right)\right\|\right\} \leq 1.17\sqrt{q} \qquad (2.2.1)$$

Потім розв'язується рівняння NTRU для знаходження відповідних F і G. Щоб знайти F та G проводимо такі дії:

Якщо n = 1:

- Розвязуємо $uf - vg = \gcd(f, g)$ за допомогою розширеного алгоритму Евкліда, знаходимо u, v. Якщо gcd(f,g) ≠ 1, тоді генеруємо f, g заново.
- (F, G) ← (vq, uq)
- Якщо n ≠ 1.
- $N_{L/K}(f) = f(x) \cdot f(-x)$, шукаєм $\begin{matrix} f' \leftarrow N(f) \\ g' \leftarrow N(g) \end{matrix}$.

  Якщо, n≠1 тоді n = n/2вертаємось до 1 умови, та шукаєм (F', G') в іншому випадку рекурсивно повертаємось до 1 пункту, повторюємо поки n≠1.Коли знайдем (F', G'), переходимо до наступного кроку.
- $\begin{matrix} F \leftarrow F'(x^2)g(-x) \\ G \leftarrow G'(x^2)f(-x) \end{matrix}$.
- $k \leftarrow \left\lfloor \frac{Ff^{\star}+Gg^{\star}}{ff^{\star}+gg^{\star}} \right\rceil$, $\begin{matrix} F \leftarrow F - kf \\ G \leftarrow G - kg \end{matrix}$ виконуємо це поки k ≠ 0. Коли k = 1, виконуємо ще раз і получаємо (F,G).

Згенеровані поліноми потім зберігаються у так званому FALCON дереві , для якого потрібно обчислити G = LDL* матриці G = BB*, $\mathbf{B} = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$

- Для того щоб знайти дерево потрібно скористатись так званим LDL-розкладом: $G = LDL*$, L – нижня одинична трикутна матриця, D – діагональна матриця. $D_j = G_{jj} - \sum_{k=1}^{j-1} L_{jk}^2 D_k$ , $L_{ij} = \frac{1}{D_j}\left(G_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_k\right)$, якщо j < i.
- Знаходимо $G = LDL*$.

- Ми зберігаємо L у T.value, яке є значенням кореня з T. Оскільки L має вигляд $L = \begin{bmatrix} 1 & 0 \\ L_{10} & 1 \end{bmatrix}$, для оптимізації зберігаємо тільки $L_{10}$.
- Потім ми розбиваємо кожен діагональний елемент D на матрицю менших елементів. Точніше, для діагонального елемента $d \in Q[x]/(x^n + 1)$ ми розглядаємо відповідний ендоморфізм $\psi_d : z \in Q[x]/(x^n + 1) \rightarrow dz$ і записуємо його матрицю перетворення над меншим кільцем $Q[x]/(x^{n/2} + 1)$.
- $$\begin{bmatrix} d_0 & d_1 \\ xd_1 & d_0 \end{bmatrix} \left( = \begin{bmatrix} d_0 & d_1 \\ d_1^\star & d_0 \end{bmatrix} \right) \quad (2.2.1)$$
- В алгоритмі генерації дерева FALCON вхідна матриця завжди розміру 2×2.

Таблиця 2.2.1 − Алгоритм Ftree

| | **Algorithm FTree** $_{\mathbf{n}}(\mathbf{G})$ |
|---|---|
| **1:** | $(\mathbf{L}, \mathbf{D}) \leftarrow \mathbf{LDL}^{\star}(\mathbf{G}), \mathbf{L} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \boldsymbol{L_{10}} & \mathbf{1} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \boldsymbol{D_{00}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{D_{11}} \end{bmatrix}$ |
| 2: | T.value $\leftarrow \boldsymbol{L_{10}}$ |
| 3: | if $(\boldsymbol{n = 2})$ then |
| 4: | T.leftchild $\leftarrow \boldsymbol{D_{00}}$ |
| 5: | T.rightchild $\leftarrow \boldsymbol{D_{11}}$ |
| 6: | return $\boldsymbol{T}$ |
| 7: | else |
| 8: | $\boldsymbol{d_{00}, d_{01}} \leftarrow \mathbf{split}\boldsymbol{FF}(\boldsymbol{D_{00}})$ |
| 9: | $\boldsymbol{d_{10}, d_{11}} \leftarrow \mathbf{splitFF}(\boldsymbol{D_{11}})$ |
| 10: | $\mathbf{G_0} \leftarrow \begin{bmatrix} \boldsymbol{d_{00}} & \boldsymbol{d_{01}} \\ \boldsymbol{d_{01}^{\star}} & \boldsymbol{d_{00}} \end{bmatrix}, \mathbf{G_1} \leftarrow \begin{bmatrix} \boldsymbol{d_{10}} & \boldsymbol{d_{11}} \\ \boldsymbol{d_{11}^{\star}} & \boldsymbol{d_{10}} \end{bmatrix}$ |
| 11: | T.leftchild $\leftarrow \boldsymbol{FTree}(\mathbf{G_0})$ |
| 12: | T.rightchild $\leftarrow \boldsymbol{Ftree}(\mathbf{G_1})$ |
| 13: | return $\boldsymbol{T}$ |

Принцип роботи алгоритму генерації підпису простий: спочатку обчислюється хеш-значення c з повідомлення m та a, а потім, використовуючи знання секретного ключа f, g, F, G, обчислюються два коротких значення $s_1$, $s_2$, $s_1 + s_2h = c \pmod q$.

Не вірний, спосіб знайти такі короткі значення ($s_1$, $s_2$) - обчислити $t \leftarrow (c, 0)\times B^{-1}$, округлити його до вектора $z = \lfloor t \rceil$ і вивести $(s_1, s_2) \leftarrow (t-z)B$; він задовольняє усім вимогам, щоб бути легітимним підписом, але цей метод є небезпечним і призводить до витоку закритого ключа.

Правильний, спосіб згенерувати ($s_1$, $s_2$) без витоку закритого ключа полягає у використанні генератора з пасткою. У FALCON ми використовуємо генератор проб з пасткою, який називають швидким перетворенням Фур'є. Обчислення дерева T за допомогою вище вказаного алгоритму FTree під час генерації ключової пари є кроком ініціалізації цієї пастки.

Серцем алгоритму є швидке перетворення Фур'є для FALCON, яке застосовує рандомізоване округлення до коефіцієнтів t. Але робить це в адаптивний спосіб, використовуючи інформацію, що зберігається в дереві FALCON.

Таблиця 2.2.2 − Алгоритм FastFurie

| | **Algorithm FastFurie** $_n(\mathbf{t}, \mathbf{T})$ |
|---|---|
| 1: | if $\boldsymbol{n = 1}$ then |
| 2: | $\boldsymbol{\sigma'} \leftarrow$ T.value |
| 3: | $\boldsymbol{z_0} \leftarrow \mathbf{SampZ}(\boldsymbol{t_0}, \boldsymbol{\sigma'})$ |
| 4: | $\boldsymbol{z_1} \leftarrow \mathbf{SampZ}(\boldsymbol{t_1}, \boldsymbol{\sigma'})$ |
| 5: | return $\mathbf{z} = (\boldsymbol{z_0}, \boldsymbol{z_1})$ |
| 6: | $(\boldsymbol{\ell}, \mathbf{T_0}, \mathbf{T_1}) \leftarrow$ ( T.value, T.leftchild, T.rightchild ) |
| 7: | $\mathbf{t_1} \leftarrow \boldsymbol{split}\mathbf{FF}(\boldsymbol{z_1})$ |
| 8: | $\boldsymbol{z_1} \leftarrow \mathbf{FastFurie_{n/2}}(\boldsymbol{t_1}, \boldsymbol{T_1})$ |
| 9: | $\boldsymbol{z_1} \leftarrow \mathbf{mergeFF}(\mathbf{z_1})$ |
| 10: | $\boldsymbol{t'_0} \leftarrow \boldsymbol{t_0} + (\boldsymbol{t_1} - \boldsymbol{z_1}) \odot \boldsymbol{\ell}$ |
| 11: | $\mathbf{t_0} \leftarrow \mathbf{split}\boldsymbol{FF}(\boldsymbol{t'_0})$ |
| 12: | $\mathbf{z_0} \leftarrow$ FastFurie $_{n/2}(\mathbf{t_0}, \mathbf{T_0})$ |
| 13: | $\boldsymbol{z_0} \leftarrow \mathbf{merge}\boldsymbol{FF}(\mathbf{z_0})$ |
| 14: | return $\mathbf{z} = (\boldsymbol{z_0}, \boldsymbol{z_1})$ |

SampZ$(t_0, \sigma')$ - получає значення з рухомою комою μ, σ′ ∈ R такі, що σ ′ ∈ [$\sigma_{min}$, $\sigma_{max}$] та дає на виході ціле число z ∈ Z, вибране з розподілу, дуже близького до $D_{Z,\mu,\sigma.}$

merge$FF(\mathbf{z}_0)$ - еквівалентна mergeF$F(f_0, f_1) = f_0(x^2) + xf_1(x^2)$.

split$FF(t'_0)$ - протилежна mergeFF тобото split$FF(f) = (f_0, f_1)$, обчислюється за формулами $f_0 = \sum_{0 \le i < n/2} a_{2i} x^i$ та $f_1 = \sum_{0 \le i < n/2} a_{2i+1} x^i$.

Формально підписання можна розбити на 4 етапи:

- Випадкове число r генерується рівномірно на проміжку $0…2^{320}$. Потім конкатенований рядок (r || m) хешується до точки *c* використовується для інтерпретації фрагмента b бітів, витягнутого з SHAKE-256, в ціле число діапазону від 0 до $2^b$ - 1 (перший з b бітів має число вагою в $2^b$ - 1 , останній має вагу 1)).
- Обчислюється, попередньо, t із c, яке потім подається на вхід алгоритму швидкого перетворення Фур'є(FastFurie $_n(t, T)$ ), який виводить два коротких поліноми $s_1$, $s_2$, такі, що $s_1 + s_2h = c \pmod q$.
- Кодується $s_2$ (стискається) до бітового рядка s.
- Получаємо пару (r,s).
- Етапи розшифрування:
- Повідомлення m конкатенуються у рядок (r ‖ m), який хешується до полінома c.
- s декодується (розпаковується) до полінома $s_2$.
- Обчислюється значення $s_1 = c - s_2 \times h \pmod q$.
- Якщо $\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$, то підпис вважається дійсним.

Таблиця 2.2.3 – Рекомендовані параметри FALCON

| | FALCON-512 | FALCON-1024 |
|---|---|---|
| Рівень захисту по шкалі NIST | I | V |
| $n$ | 512 | 1024 |
| $q$ | 12289 | |
| $\sigma_{\min}$ | 165.736617183 | 168.388571447 |
| $\sigma_{\max}$ | 1.277833697 | 1.298280334 |
| $\lfloor \beta^2 \rfloor_{\max}$ | 34034726 | 70265242 |
| Розмір відкритого ключа в байтах | 897 | 1793 |
| Довжина підпису в байтах | 666 | 1280 |

Аналіз відомих теоретично можливих атак:

Атака на знаходження ключа. Розглянемо ґратку $\begin{bmatrix} q & h \\ 0 & 1 \end{bmatrix}$, знайшовши короткі базиси на цій основі, ми перерахуємо

всі точки ґратки у кулі радіусом $\sqrt{2n} \cdot \sigma_{\{f,g\}}$ з центром на початку координат. Зі значною ймовірністю ми можемо знайти $[g \mid f]$. Якщо $\lambda$ = (2n - B) норма Грама-Шмідта, яка приблизно дорівнює нормі найкоротшого вектора решітки, утвореного останніми B векторами, спроектованими ортогонально на перші (2n - B – 1) вектори. Алгоритм ґратчастого просіювання(lattice sieving), що виконується на цій спроектованій ґратці, можеш обчислити всі вектор з нормою $<\sqrt{4/3}\lambda$. Припустимо, що серед низ є проекція ключа тоді $\sqrt{B}\sigma_{\{f,g\}} \leq \sqrt{4/3}\lambda$, та ми можемо знайти секретний ключ із його проекції використовуючи алгоритм наближеного розв’язку задачі знаходження найближчого вектора(CVP), як-от,

алгоритм Бабая(BNP) на всіх просіяних векторах із високою ймовірністю знайти секретний ключ[7].

Найкращий відомий алгоритм пошуку найкоротший базисів, DBKZ(Self-Dual BKZ) дає, що $\lambda = \left(\frac{B}{2\pi e}\right)^{1-n/B} \sqrt{q}$ , відповідно $(B/2\pi e)^{1-n/B}\sqrt{q} = \sqrt{3/4B}\sigma_{\{f,g\}}$ . В цьому випадку ми прирівняли швидкість ґратчастого просіювання до швидкості алгоритму DBKZ, що в свою чергу є далеким в дійсності. Знайти B і вирахувати складність атаки дальше дуже просто, але враховуючи всі нюанси, ця атака на ключ не є перспективною.

Атака підробки підпису. Підробка підпису може бути виконана шляхом знаходження точки решітки на відстані, обмеженій β від випадкової точки, у тій самій решітці, що розглядалась вище. Цю задачу також можна розв'язати алгоритмом знаходження найкоротших базисів. Однією з можливостей є використання вкладання Каннана, тобто додавання (H(r||m), 0, K) до базису решітки, розширеного рядком нулів, що дає наступну матрицю: $\begin{bmatrix} q & h & H(r \parallel m) \\ 0 & 1 & 0 \\ 0 & 0 & K \end{bmatrix}$ . Також можна скористатися особливістю алгоритму ґратчастого просіювання(lattice sieving) та згенерувати багато коротких векторів і , якщо серед них є вектори типу (c, * ,K) то можна дізнатись і точки ґратки H(r||m) – c.

Якщо припустити, що K ≈$\sqrt{q}$, для DBKZ алгоритму то, $\left(\frac{B}{2\pi e}\right)^{n/B} \sqrt{q} \leq \beta$. При обчислені $\beta$ використовується q, тому параметр q, практично, не впливає на стійкість до атаки підробки підпису. За допомогою методики New Hope можна обчислити bit-security(кількість бітів закритого ключа має бути відомих для атакуючого щоб повністю зламати алгоритм).

Таблиця 2.2.4 – Складність атаки звичайного комп’ютера і квантовго[7]

| | Знаходження ключа | | | | Підробка підпису | | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | $B$ | $B'$ | Звичайний | Квантовий | $B$ | $B'$ | Звичайний | Квантовий |
| 512 | 458 | 418 | 133 | 121 | 411 | 374 | 120 | 108 |
| 1024 | 936 | 869 | 273 | 248 | 952 | 884 | 277 | 252 |

Можна замітити, що ефективність квантового комп’ютера теоретично в 1.11 раз більша, але на практиці, це занадто мала різниця щоб використовувати квантовий комп’ютер в таких цілях. Атака із показником B=411 в наведеному прикладі є самою ефективною.

Ресурси потрібні для атаки на FALCON-512, B=411.Кількість операцій на просторі n алгоритму ґратчастого просіювання, для знаходження найкоротшого вектора рівна $\frac{n^3}{4B^2}$. Якщо взяти перший асимптотичний член ми получає $\frac{n^3}{4B^2} \cdot (\sqrt{1.5})^{B'} \approx 2^{120}$ операцій. Ця оцінка є дуже поверхневою, тому, що ми нехтуємо деякими членами нижчих порядків.

Із плюсів варта відмітити малий розмір як відкритого так і закритого ключа, та порівняно із попереднім алгоритмом не великий розмір підпису та вища швидкість роботи.

## 2.3 Kyber

Kyber. CRYSTAL-Kyber - це захищений за стандартом IND-CCA2 механізм інкапсуляції ключів (KEM)[8]. Безпека Kyber базується на складності розв'язання проблеми навчання з помилками в модульних решітках (проблема MLWE). Побудова Kyber відбувається у два етапи: спочатку ми вводимо INDCPA-безпечну схему шифрування з відкритим ключем, що шифрує повідомлення фіксованої довжини 32 байти, яку ми називаємо Kyber.CPAPKE. Потім використовується модифіковане перетворення Фудзісакі-Окамото (FO) для побудови IND-CCA2-безпечного KEM. Сама процедура складається із 2 під

етапів: Kyber.CPAPKE та Kyber.CCAKEM. Kyber – перший PQC алгоритм сертифікований NIST.

Модулі Kyber.CPAPKE та Kyber.CCAKEM окремо мають по 3 функції: генерацію ключів, шифрування і розшифрування( подібний до SPHINCS+). Для того, щоб розглядати ці функції варта розглянути допоміжні функції.

Рівно ймовірна вибірка(Parse): Kyber використовує детермінований підхід для вибірки елементів в Rq, які статистично близькі до рівномірно випадкового розподілу. Для такої вибірки використовується функція Parse, яка отримує на вхід потік байт B = $b_0$, $b_1$, $b_2$, ... і обчислює NTT-представлення $\hat{a} = \hat{a}_0 + \hat{a}_1 X + \cdots + \hat{a}_{n-1} X^{n-1} \in R_q$ для a ∈ $R_q$. Суть Parse полягає у тому, що якщо вхідний байтовий масив статистично близький до рівномірно випадкового байтового масиву, то вихідний поліном статистично близький до рівномірно випадкового елемента $R_q$. Він представляє рівномірно випадковий поліном у $R_q$, оскільки NTT є бієктивним і, таким чином, відображає поліноми з рівномірно випадковими коефіцієнтами у поліноми з знову ж таки рівномірно випадковими коефіцієнтами.

Вибірка з біноміального розподілу(CBD): Шум у Kyber береться із центрованого біноміального розподілу $B_\eta$ для η = 2 або η = 3. $(a_1, \dots, a_\eta, b_1, \dots, b_\eta) \rightarrow \sum_{i=1}^{\eta} (a_i - b_i)$

Для специфікації Kyber нам потрібно визначити, як поліном f ∈ $R_q$ вибирається відповідно до $B_\eta$ детерміновано з 64η байт вихідних даних псевдовипадкової функції. Для цього баритовий рядок перетворюємо в бітовий та почергово робимо вибірку у $(a_1, \dots, a_\eta, b_1, \dots, b_\eta)$, віднімаєм відповідні елементи і тепер вони переходять в $f_0 + f_1 X + f_2 X^2 + \cdots + f_{255} X^{255}$.[9]

Кодування та декодування(Decode): Існує два типи даних, які Kyber має перетворити у байт-масиви: інші байт-масиви та (вектори) поліноми. Масиви байт легко перетворюються за допомогою індексної відповідності, потрібно визначити, як це буде відбувлось. $\mathcal{B}^{32\ell} \rightarrow (\beta_0, \dots, \beta_{256\ell-1})$, після чого рахуємо $f_i := \sum_{j=0}^{\ell-1} \beta_{i\ell+j} 2^j$. Після того додаєм і позлучається поліном, $f_0 + f_1 X + f_2 X^2 + \cdots + f_{255} X^{255}$.

Для оптимізації обчислень поліномів, використовується NTT – дискретна форма перетворення Фур'є .

$\mathrm{NTT}(f) = \hat{f} = \left(\hat{f}_0 + \hat{f}_1 X, \hat{f}_2 + \hat{f}_3 X, \ldots, \hat{f}_{254} + \hat{f}_{255} X\right)$ , де $\hat{f}_{2i} = \sum_{j=0}^{127} f_{2j}\zeta^{(2\mathrm{br}_7(i)+1)j}$ , $\hat{f}_{2i+1} = \sum_{j=0}^{127} f_{2j+1}\zeta^{(2\mathrm{br}_7(i)+1)j}$ , в якому, br7(i) для i = (0, . . . ,127) - обернення бітів 7-бітового цілого числа без знаку ***i***. Особливе значення для нас має $\mathrm{NTT}^{-1}$ , обчислюється як $NTT^{-1}(f) = \frac{1}{n} \cdot NTT(f \cdot g)$, де n - кількість елементів у послідовності, а g - обернений елемент до f в полі цілих чисел. Обернений елемент можна знайти за допомогою розширеного алгоритму Евкліда.

Compress, Decompress. $\mathrm{Compress}_q(x, d) = \lceil (2^d/q) \cdot x \rfloor mod^+ 2^d$, $\mathrm{Decompress}_q(x, d) = \lceil (q/2^d) \cdot x \rfloor$.

Протилежна функція(Encode). Дістає із $f_i \rightarrow \beta_0, \ldots, \beta_{256\ell-1} \rightarrow \mathcal{B}^{32\ell}$.

Основні параметри: цілі числа n, k, q, $\eta_1$, $\eta_2$, $d_u$ та $d_v$., n = 259, q = 3329. η1, η2, $d_u$ та $d_v$ обираються для досягнення балансу між безпекою, розміром шифрованого тексту та ймовірністю збою. Всі три набори параметрів забезпечують ймовірність збою < $2^{-128}$. Параметр q = 3329, був підібраний для простоти множення у NTT. Параметр $\boldsymbol{\eta_1}$ визначає шум **s**, **e** та **r**. Параметр $\eta_2$ визначає шум $\mathbf{e_1}$ та $\mathbf{e_2}$.

Функції та їх реалізація:

- G – SHA3-512
- XOF – Shake-128
- PRF – Shake-256(s||b)
- H – SHA3-256
- KDF – Shake-256

Алгоритм KeyGen():

- Генерація 32-байтового масиву.
- Получений масив проходить хеш-функції G(**d**) (SHA3-512), та обчислюється середнє відхилення $\boldsymbol{\sigma}$ та $\boldsymbol{p}$.
- Після чого за допомогою XOF(Shake-128) рівноймовірної вибірки формується матриця **A**.

- Генеруємо поліноми s, e. Для кожного елементу $s_i$ та $e_i$ получаєм за допомогою біноміального розподілу $\eta_1$-випробувань PRF(σ, N) (Shake-256(σ ||N)), де N – індекс відповідного йому елемента $(s_0, \dots, s_{k-1}, e_k, \dots, e_{2k-2})$.
- Обчислюємо $\hat{t} \coloneqq \hat{A} \circ \hat{s} + \hat{e}$
- Обчислюємо *pk*(відкритий ключ) та *sk*(закритий ключ), $pk = (\hat{t} \bmod^+ q)$, $sk = (\hat{s} \bmod^+ q)$
- Перетворюємо в байтовий масив обидва ключі, до відкритого ключа конкатенуємо *p*, $pk = pk \,||p$

Таблиця 2.3.1 − Алгоритм KeyGen

| **Algorithm KeyGen():** | |
|---|---|
| 1: $\boldsymbol{d} \leftarrow \mathcal{B}^{32}$ | |
| 2: $(\boldsymbol{\rho}, \boldsymbol{\sigma}) \coloneqq \mathbf{G}(\boldsymbol{d})$ | G(d) = SHA3-512(d) |
| 3: $\boldsymbol{N} \coloneqq \mathbf{0}$ | |
| 4: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k}-\mathbf{1}$ do | Створення матриці $\hat{\mathbf{A}}$ в |
| 5: for $\boldsymbol{j}$ from 0 to $\boldsymbol{k}-\mathbf{1}$ do | просторі NTT. |
| 6: $\hat{\mathbf{A}}[\boldsymbol{i}][\boldsymbol{j}] \coloneqq \mathbf{Parse}(\mathbf{XOF}(\boldsymbol{\rho}, \boldsymbol{j}, \boldsymbol{i}))$ | Створення вибірки s із $B_{\eta 1}$ |
| 7: end for | |
| 8: end for | |
| 9: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k}-\mathbf{1}$ do | |
| 10: $\mathbf{s}[\boldsymbol{i}] \coloneqq \mathbf{CBD}_{\boldsymbol{\eta_1}}(\mathbf{PRF}(\boldsymbol{\sigma}, \boldsymbol{N}))$ | |
| 11: $\boldsymbol{N} \coloneqq \boldsymbol{N}+\mathbf{1}$ | |
| 12: end for | |
| 13: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k}-\mathbf{1}$ do | |
| 14: $\mathbf{e}[\boldsymbol{i}] \coloneqq \mathbf{CBD}_{\boldsymbol{\eta_1}}(\mathbf{PRF}(\boldsymbol{\sigma}, \boldsymbol{N}))$ | |
| 15: $\boldsymbol{N} \coloneqq \boldsymbol{N}+\mathbf{1}$ | |
| 16: end for | |
| 17: $\hat{\mathbf{s}} \coloneqq \mathbf{NTT}(\mathbf{s})$ | |
| 18: $\hat{\mathbf{e}} \coloneqq \mathbf{NTT}(\mathbf{e})$ | |
| 19: $\hat{\mathbf{t}} \coloneqq \hat{\mathbf{A}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$ | |
| 20: $\boldsymbol{pk} \coloneqq (\mathbf{Encode}_{\mathbf{12}}(\hat{\mathbf{t}} \bmod^{+} \boldsymbol{q}) \parallel \boldsymbol{\rho})$ | |
| 21: $\boldsymbol{sk} \coloneqq \mathbf{Encode}_{\mathbf{12}}(\hat{\mathbf{s}} \bmod^{+} \boldsymbol{q})$ | |
| 22: return $(\boldsymbol{pk}, \boldsymbol{sk})$ | |

Алгоритм Enc():

- Перетворюємо масив бітів відкритого ключа у поліном $\hat{\mathrm{t}}$.
- Шукаєм $\rho := pk + 12 \cdot k \cdot n/8$ .
- Знаходимо $\hat{\mathrm{A}}^T$ як у випадку KeyGen() але обертаємо матрицю навколо діагоналі.
- Обчислюємо $r$, $e_1$, $e_2$ так само як і $s$ та $e$ у KeyGen(), але для $e_1$, $e_2$ , проводимо $\eta_2$-випробувань для біноміального розподілу, замість $\sigma$ використовуємо параметр r. Знаходимо $\hat{\mathrm{r}} := \mathrm{NTT}(\mathrm{r})$.
- Обчислюємо $u$ та $v$, $\mathrm{u} := \mathrm{NTT}^{-1}\left(\hat{\mathrm{A}}^T \circ \hat{\mathrm{r}}\right) + \mathrm{e}_1$ , $v := \mathrm{NTT}^{-1}(\hat{\mathrm{t}}^T \circ \hat{\mathrm{r}}) + e_2 + \mathrm{Decompress}_q(\mathrm{Decode}_1(m),1))$.
- Виконуємо для $\boldsymbol{u}$ та $\boldsymbol{v}$, функцію Compress, перетворюємо в множину байтів та конкатенуємо разом.

Таблиця 2.3.2 − Алгоритм Enc

| **Algorithm Enc (pk,m,r)** | |
|---|---|
| **1:** $\boldsymbol{N} := \mathbf{0}$ | |
| 2: $\hat{\mathbf{t}} := \mathbf{Decode_{12}}(\boldsymbol{pk})$ | |
| **3:** $\boldsymbol{\rho} := \boldsymbol{pk} + \mathbf{12} \cdot \boldsymbol{k} \cdot \boldsymbol{n}/\mathbf{8}$ | |
| 4: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k} - \mathbf{1}$ do | XOF = SHAKE-128 |
| 5: for $\boldsymbol{j}$ from 0 to $\boldsymbol{k} - \mathbf{1}$ do | |
| 6: $\hat{\mathbf{A}}^{\boldsymbol{T}}[\boldsymbol{i}][\boldsymbol{j}] := \mathbf{Parse}(\mathbf{XOF}(\boldsymbol{\rho}, \boldsymbol{i}, \boldsymbol{j}))$ | |
| 7: end for | |
| 8: end for | |
| 9: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k} - \mathbf{1}$ do | |
| 10: $\mathbf{r}[\boldsymbol{i}] := \mathbf{CBD}_{\boldsymbol{\eta_1}}(\mathbf{PRF}(\boldsymbol{r}, \boldsymbol{N}))$ | |
| 11: $\boldsymbol{N} := \boldsymbol{N} + \mathbf{1}$ | PRF(s,b) =SHAKE256(s\|\|b) |
| 12: end for | |
| 13: for $\boldsymbol{i}$ from 0 to $\boldsymbol{k} - \mathbf{1}$ do | |
| 14: $\mathbf{e_1}[\boldsymbol{i}] := \mathbf{CBD}_{\boldsymbol{\eta_2}}(\mathbf{PRF}(\boldsymbol{r}, \boldsymbol{N}))$ | $\mathrm{Decompress}_q(x,d) = \lceil (q/2^d) \cdot x \rfloor$ |
| 15: $\boldsymbol{N} := \boldsymbol{N} + \mathbf{1}$ | $\mathrm{Compress}_q(x,d) = \lceil (2^d/q) \cdot x \rfloor mod^+ 2^d$ |
| 16: end for | |
| 17: $\boldsymbol{e_2} := \mathbf{CBD}_{\boldsymbol{\eta_2}}(\mathbf{PRF}(\boldsymbol{r}, \boldsymbol{N}))$ | |
| 18: $\hat{\mathbf{r}} := \mathbf{NTT}(\mathbf{r})$ | |
| **19:** $\mathbf{u} := \mathbf{NTT^{-1}}(\hat{\mathbf{A}}^{\boldsymbol{T}} \circ \hat{\mathbf{r}}) + \mathbf{e_1}$ | |
| **20:** $\boldsymbol{v}$: $= \mathbf{NTT^{-1}}(\hat{\mathbf{t}}^{\boldsymbol{T}} \circ \hat{\mathbf{r}}) + \boldsymbol{e_2} + \mathbf{Decompress}_{\boldsymbol{q}}(\mathbf{Decode_1}(\boldsymbol{m}), \mathbf{1}))$ | |
| 21: $\boldsymbol{c_1} := \text{Encode}_{\boldsymbol{d_u}}(\mathbf{Compress}_{\boldsymbol{q}}(\mathbf{u}, \boldsymbol{d_u}))$ | |

Продовження таблиці 2.3.3

| **Algorithm Enc (pk,m,r)** | |
|---|---|
| 22: $\boldsymbol{c_2} := \mathbf{Encode}_{d_v}\left(\mathbf{Compress}_q(\boldsymbol{v}, \boldsymbol{d_v})\right)$<br>23: return $\boldsymbol{c} = (\boldsymbol{c_1} \parallel \boldsymbol{c_2})$ | |

Алгоритм Dec():

- Відновлюємо *u* та *v*, $\mathrm{u} := \mathrm{Decompress}_q\left(\mathrm{Decode}_{d_u}(c), d_u\right), \quad v :=$ $\mathrm{Decompress}_q\left(\mathrm{Decode}_{d_v}(c + d_u \cdot k \cdot n/8), d_v\right)$.
- Перетворюємо закритий ключ із байт масиву в поліном.
- Розшифровуємо повідомлення,$m := \mathrm{Encode}_1\left(\mathrm{Compress}_q(v - \mathrm{NTT}^{-1}(\hat{\mathrm{s}}^T \circ \mathrm{NTT}(\mathrm{u})), 1)\right)$.

Таблиця 2.3.3 − Алгоритм Dec

| **Algorithm Dec** $(\boldsymbol{sk}, \boldsymbol{c})$ **:** |
|---|
| 1: $\mathbf{u} := \mathbf{Decompress}_q\left(\mathbf{Decode}_{d_u}(\boldsymbol{c}), \boldsymbol{d_u}\right)$ |
| 2: $\boldsymbol{v} := \mathbf{Decompress}_q\left(\mathbf{Decode}_{d_v}(\boldsymbol{c} + \boldsymbol{d_u} \cdot \boldsymbol{k} \cdot \boldsymbol{n}/\mathbf{8}), \boldsymbol{d_v}\right)$ |
| 3: $\hat{\mathbf{s}} := \mathbf{Decode}_{12}(\boldsymbol{sk})$ |
| 4: $\boldsymbol{m} := \mathbf{Encode}_1\left(\mathbf{Compress}_q(\boldsymbol{v} - \mathbf{NTT}^{-1}(\hat{\mathbf{s}}^{\boldsymbol{T}} \circ \mathbf{NTT}(\mathbf{u})), \mathbf{1})\right)$ |
| 5: return $\boldsymbol{m}$ |

Дальше буде описана робота основного модуля Kyber.CCAKEM. Цей модуль фактично використовує тільки функції модуля Kyber.CPAPKE та допоміжні функції тому для нього не потрібно особливого пояснення.

Як видно із алгоритма CCAKEM.KeyGen() відкритий ключ із модуля Kyber.CPAPKE так і залишається загальним відкритим ключом змінюється тільки закритий ключ.

Таблиця 2.3.4 − Алгоритм CCAKEM.KeyGen

| | **Algorithm CCAKEM.KeyGen()** |
|---|---|
| 1: | $\mathbf{z} \leftarrow \boldsymbol{\mathcal{B}}^{\mathbf{32}}$ |
| 2: | $(\mathbf{pk}, \mathbf{sk}') := \text{KeyGen}()$ |
| 3: | $\mathbf{sk} := (\mathbf{sk}' \parallel \mathbf{pk} \parallel \mathbf{H}(\mathbf{pk}) \parallel \mathbf{z})$ |
| 4: | return $(\mathbf{pk}, \mathbf{sk})$ |

Параметр ***r***, який був нам потрібен для біноміального розподілу в Dec(), задається ззовні функції. Для безпеки системи, потрібно слідкувати за можливістю витоку H(m).

Таблиця 2.3.5 Алгоритм CCAKEM.Enc

| | **Algorithm CCAKEM. Enc(pk)** |
|---|---|
| **1:** | $\boldsymbol{m} \leftarrow \boldsymbol{\mathcal{B}}^{\mathbf{32}}$ |
| **2:** | $\boldsymbol{m} \leftarrow \mathbf{H}(\boldsymbol{m})$ |
| **3:** | $(\bar{\boldsymbol{K}}, \boldsymbol{r}) := \mathbf{G}(\boldsymbol{m} \parallel \mathbf{H}(\boldsymbol{pk}))$ |
| **4:** | $\boldsymbol{c} := \mathbf{Enc}\ (\boldsymbol{pk}, \boldsymbol{m}, \boldsymbol{r})$ |
| **5:** | $\boldsymbol{K} := \mathbf{KDF}(\bar{\boldsymbol{K}} \parallel \mathbf{H}(\boldsymbol{c}))$ |
| 6: | return $(\boldsymbol{c}, \boldsymbol{K})$ |

Таблиця 2.3.6 − Алгоритм CCAKEM.Dec

| **Algorithm CCAKEM.Dec** $(\boldsymbol{c}, \boldsymbol{sk})$ |
|---|
| 1: $\boldsymbol{pk} := \boldsymbol{sk} + \mathbf{12} \cdot \boldsymbol{k} \cdot \boldsymbol{n}/\mathbf{8}$ |
| 2: $\boldsymbol{h} := \boldsymbol{sk} + \mathbf{24} \cdot \boldsymbol{k} \cdot \boldsymbol{n}/\mathbf{8} + \mathbf{32} \in \mathcal{B}^{\mathbf{32}}$ |
| 3: $\boldsymbol{z} := \boldsymbol{sk} + \mathbf{24} \cdot \boldsymbol{k} \cdot \boldsymbol{n}/\mathbf{8} + \mathbf{64}$ |
| 4: $\mathbf{Dec}(\mathbf{s}, (\mathbf{u}, \boldsymbol{v}))$ |
| 5: $(\bar{\boldsymbol{K}}', \boldsymbol{r}') := \mathbf{G}(\boldsymbol{m}' \parallel \boldsymbol{h})$ |
| 6: $\boldsymbol{c}' :=$Enc $(\boldsymbol{pk}, \boldsymbol{m}', \boldsymbol{r}')$ |
| 7: if $\boldsymbol{c} = \boldsymbol{c}'$ then |
| 8: return $\boldsymbol{K} := \mathbf{KDF}(\bar{\boldsymbol{K}}' \parallel \mathbf{H}(\boldsymbol{c}))$ |
| 9: else |
| 10: return $\boldsymbol{K} := \mathbf{KDF}(\boldsymbol{z} \parallel \mathbf{H}(\boldsymbol{c}))$ |
| 11: end if |
| 12: return $\boldsymbol{K}$ |

Таблиця 2.3.7 – Продуктивність Kyber та Kyber-90s

| " KYBER512 | | | | | |
|---|---|---|---|---|---|
| Sizes (in Bytes) | | Haswell Cycles (ref) | | Haswell Cycles (AVX2) | |
| sk: | 1632 (or 32 ) | gen: | 122684 | gen: | 33856 |
| pk: | 800 | enc: | 154524 | enc: | 45200 |
| ct: | 768 | dec: | 187960 (or ≈ **288912**) | dec: | 34572 (or ≈ **59088**) |
| KYBER512-90s | | | | | |
| Sizes(inBytes)HaswellCycles(ref)HaswellCycles(AVX2) | | | | | |
| sk: | 1632 (or 32 ) | gen: | 213156 | gen: | 21880 |
| pk: | 800 | enc: | 249084 | enc: | 28592 |
| ct: | 768 | dec: | 277612 (or ≈ **405268**) | dec: | 20980 (or ≈ **38752** ) |
| KYBER768 | | | | | |
| Sizes(inBytes)HaswellCycles(ref)HaswellCycles(AVX2) | | | | | |
| sk: | 2400 (or 32 ) | gen: | 199408 | gen: | 52732 |
| pk: | 1184 | enc: | 235260 | enc: | 67624 |
| ct: | 1088 | dec: | 274900 (or ≈ **425492**) | dec: | **53156**( or ≈ **82220**) |
| KYBER768-90s | | | | | |
| Sizes(inBytes)HaswellCycles(ref)HaswellCycles(AVX2) | | | | | |
| sk: | 2400 (or 32 ) | gen: | 389760 | gen: | 30460 |
| pk: | 1184 | enc: | 432764 | enc: | 40140 |
| ct: | 1088 | dec: | **473984**( or ≈ **671864**) | dec: | **30108**( or ≈ **51512**) |
| " KYBER1024 | | | | | |
| Sizes(inBytes)HaswellCycles(ref)HaswellCycles(AVX2) | | | | | |
| sk: | 3168 (or 32 ) | gen: | 307148 | gen: | 73544 |
| pk: | 1568 | enc: | 346648 | enc: | 97324 |
| ct: | 1568 | dec: | 396584 (or ≈ **617848**) | dec: | 79128 (or ≈ **115332**) |
| KYBER1024-90s | | | | | |
| Sizes(inBytes)HaswellCycles(ref)HaswellCycles(AVX2) | | | | | |
| sk: | 3168 (or 32 ) | gen: | 636380 | gen: | 43212 |
| pk: | 1568 | enc: | 672644 | enc: | 56556 |
| ct: | 1568 | dec: | 724144 (or ≈ **1009448** ) | dec: | **44328**( or ≈ **71180**) |

Рівень захищеності алгоритму базується на оцінках ресурсозатратності атак на основну проблему - модульного навчання з помилками (MLWE). Найвідоміші атаки на основну проблему MLWE в Kyber не використовують модель ґратки. Тому ми проаналізуємо складність проблеми MLWE як проблеми LWE.

Існує два алгоритмічні підходи для оракула SVP в BKZ: алгоритм перебору та алгоритми ґратчастого просіювання. Ці два класи алгоритмів мають дуже різні

характеристики продуктивності, і, зокрема, для просіювання, важко передбачити, як продуктивність масштабується від розмірів решітки, з якими успішно справлялися, до більших розмірів, які є актуальними в атаках на криптосистеми на кшталт Kyber. Відправною точкою такого аналізу є той факт, що алгоритми перебору мають супер-експоненціальний час роботи, в той час як алгоритми просіювання мають лише експоненціальний час роботи. Експериментальні дані з типових реалізацій BKZ показують, що алгоритми перебору більш ефективні на "малих" розмірностях, тому питання полягає в тому, на якій розмірності просіювання стає більш ефективним. Найвідоміші методи просіювання були повільнішими на практиці для доступних розмірностей до $b \approx 130$[10].

Більшість алгоритмів ґратчасого просіювання виграють від використання алгоритму квантового пошуку Гровера, знижуючи складність до $2^{0.265b+o(b)}$ . Однак, навіть якщо використовувати оптимізацію для квантового комп'ютера, це все одно дає дуже мізерний результат. Складність для звичайного комп'ютера, $2^{0.292b+o(b)}$.

Формально, стійкість алгоритмів FALCON та Kyber, ґрунтується на похожих принципах, відповідно складність злому майже еквівалентна.

Таблиця 2.3.8 – Стійкість Kyber

| | KYBER512 | KYBER768 | KYBER1024 |
|---|---|---|---|
| Рівень безпеки поNIST | 1 | 3 | 5 |
| Розмірнсть атаки на ґратки . $d$ | 1003 | 1424 | 1885 |
| Розмір блокуBKZ $\beta$ | 403 | 625 | 877 |
| core-SVP класична складність | 118 | 182 | 256 |
| core-SVP квантова складність | 107 | 165 | 232 |
| Розмірнсть атаки на ґратки $d$ | 1025 | 1467 | 1918 |
| BKZ-blocksize $\beta$ | 413 | 637 | 894 |
| Розмір просіювання $\beta' = \beta - d_{4\mathrm{f}}$ | 375 | 586 | 829 |
| $\log_2$ (вентилі) | 151.5 | 215.1 | 287.3 |
| $\log_2$ (пам'ять в бітах) | 93.8 | 138.5 | 189.7 |

## 2.4 Квантова теорія, квантові схеми

Квантова криптографія - це галузь науки, яка використовує закони квантової механіки для забезпечення безпечного обміну інформацією. Однією з основних особливостей квантової криптографії є використання квантових станів, таких як фотони, для передачі інформації. Квантові стани мають властивість суперпозиції, що означає, що вони можуть існувати в комбінації двох або більше можливих значень, таких як горизонтальна або вертикальна поляризація. Коли квантовий стан вимірюється, він розпадається до одного з можливих значень, і результат є випадковим і непередбачуваним. Це робить квантові стани ідеальними для генерації випадкових ключів і виявлення спроб підслуховування.

Ще однією особливістю квантової криптографії є використання квантової заплутаності - явища, коли два або більше квантових станів пов'язані таким чином, що їхні властивості корелюють, навіть коли вони розділені великими відстанями. Коли вимірюються два переплетені квантові стани, їхні результати завжди пов'язані, незалежно від відстані або проміжного середовища. Це робить квантову заплутаність корисною для створення захищених каналів зв'язку та для здійснення квантової телепортації, тобто перенесення квантових станів з одного місця в інше.

Одним з найбільш широко використовуваних протоколів у квантовій криптографії є квантовий розподіл ключів (QKD) - метод генерації та розподілу секретних ключів з використанням квантових станів. QKD складається з двох фаз: квантової та класичної. У квантовій фазі відправник (Аліса) і одержувач (Боб) обмінюються квантовими станами, такими як поляризовані фотони, через квантовий канал, наприклад, оптичне волокно. У класичній фазі Аліса і Боб спілкуються через загальнодоступний канал, наприклад, інтернет, щоб виконати корекцію помилок і посилення конфіденційності, які є методами усунення помилок і зменшення інформації, доступної для підслуховування (Єва). В кінці протоколу QKD Аліса і Боб обмінюються секретним ключем, який можна використовувати для шифрування і дешифрування повідомлень.

Однією з головних проблем квантової криптографії є забезпечення безпеки квантового каналу та квантових пристроїв. Квантовий канал може бути схильний до шуму, втрат і перешкод, які можуть вплинути на якість і цілісність квантових станів. Квантові пристрої, такі як джерела фотонів, детектори та модулятори, можуть бути схильні до недосконалостей, дефектів та атак, які можуть поставити під загрозу функціональність та безпеку протоколу QKD. Тому квантова криптографія вимагає ретельного теоретичного аналізу та експериментальної перевірки, щоб гарантувати її безпеку і надійність. Найбільш популярні види QKD:

- QKD одиночних фотонів. Цей вид QKD використовує джерела, які випромінюють один фотон за раз, який кодує інформацію за допомогою поляризації, фази або частоти. Приймач вимірює фотон за допомогою поляризаційних фільтрів, фазових модуляторів або інтерферометрів. Цей вид QKD включає такі протоколи, як BB84, B92, SARG04, DPS-QKD та інші.
- QKD на основі заплутаних фотонів. Це спосіб QKD, який використовує джерела, що випускають пари фотонів, які перебувають у заплутаному стані, тобто мають кореляцію незалежно від відстані між ними. Приймач визначає фотони за допомогою поляризаційних фільтрів, фазових

модуляторів або інтерферометрів. До цього способу QKD належать такі протоколи, як E91, BBM92, COW-QKD та інші.

- QKD квантових неперервних змінних. Цей тип QKD використовує джерело, яке випромінює слабкий лазерний імпульс, який кодує інформацію за допомогою амплітуди або фази. Приймач вимірює пульсацію за допомогою тактового або гетеродинного детектора. Він включає такі протоколи, як QKD, CV-QKD, CV-MDI-QKD та CV-QKD.
- QKD неполяризованого світла. Цей метод не вимагає використання поляризаційних фільтрів або модуляторів, що спрощує і знижує вартість квантових пристроїв. Натомість він використовує неполяризовану властивість, що дає змогу розкласти його на два квадратичні компоненти(0,1).
- QKD цифрової обробки сигналів. Даний метод не вимагає використання фотонів для передачі сигналів, натомість він використовує спеціальні математичні функції для кодування інформації, яка може передаватися через оптичне волокно. Цей метод також дозволяє розробляти недорогі та компактні квантові пристрої.
- QKD квантових точок. Квантові точки - це наночастинки, які мають властивість випромінювати світло певної довжини хвилі при збудженні. Вони можуть бути використані для створення квантових ключів, які можуть бути вбудовані в різні носії, такі як папір, пластик або текстиль. Це збільшує гнучкість та безпеку QKD.

Для того, щоб розглянути новітні системи QKD потрібно познайомитись із деякою теорією. Моделюю теорії квантового обчислення є квантова схема - математичний апарат який дозволяє моделювати квантові стани, квантові вентилі та їх перетворення.

Розглянемо, що таке стан одного кубіта. Назвемо систему, що розглядається, X, і будемо використовувати символ Σ для позначення множини класичних станів X. Крім припущення, що Σ скінченна, припускаємо, що Σ не є порожньою. Хоча є сенс розглядати фізичні системи з нескінченною кількістю

класичних станів, поки що ігноруватимемо цю можливість. Наприклад: X – біт, $\Sigma = \{0,1\}$.

Однак часто при обробці інформації наші знання про X є невизначеними. Ми представляємо наші знання про класичні стани X шляхом присвоєння ймовірностей кожному класичному стану, в результаті чого отримуємо ймовірнісний стан. Наприклад, припустимо, що X - це біт. Виходячи з того, що ми знаємо або очікуємо про те, що відбувалося з X в минулому, можливо, ми вважаємо, що X перебуває в класичному стані 0 з ймовірністю 3/4 і в стані 1 з ймовірністю 1/4, запишемо як $\Pr(X = 0) = \frac{3}{4}$ та $\Pr(X = 1) = \frac{1}{4}$. Практичніше записувати у вигляді вектор-стовпця, $\begin{pmatrix} \frac{3}{4} \\ \frac{1}{4} \end{pmatrix}$ де верхнє значення ймовірність 0, а нижнє 1 у $\{0,1\}$.

Позначимо щойно описаний вектор, тобто вектор, який має 1 у позиції, що відповідає a, і 0 у всіх інших позиціях, через $|a\rangle$. Цей вектор читається як «кет a». Вектори такого типу також називають векторами зі стандартним базисом. Наприклад стандартний вектор задано як: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ та $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, відповідно $\begin{pmatrix} \frac{3}{4} \\ \frac{1}{4} \end{pmatrix} = \frac{3}{4}|0\rangle + \frac{1}{4}|1\rangle$.

Якщо $\Sigma=\{0,1\}$, то існує чотири функції такого вигляду: $f_1$, $f_2$, $f_3$ і $f_4$, які можна подати у вигляді таблиць значень наступним чином:

Таблиця 2.4.1 – Функції $f_1$, $f_2$, $f_3$ і $f_4$

| *a* | $f_1(a)$ | *a* | $f_2(a)$ | *a* | $f_3(a)$ | *a* | $f_4(a)$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

Перша та остання з цих функцій є константними: $f_1(a) = 0$, $f_4(a) = 1$. Функція $f_2(a)$ є функцією тотожності. А $f_3$ - це функція $f_3(a) = 1$ і $f_3(a) = 0$, яка більш відома як функція NOT. Дії детермінованих операцій над імовірнісними станами можна представити матрично-векторним множенням. Зокрема, матриця M, яка представляє задану функцію $f : \Sigma \to \Sigma$, є такою, що задовольняє наступним умовам: $M\ |a\rangle = |f(a)\rangle$. Матриці $M_1$...$M_4$, що відповідають функціям $f_1$...f4, наведеним вище, мають наступний вигляд:

$$M_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},\ M_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ M_3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ M_4 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}. \qquad (2.4.1)$$

Зручним способом представлення матриць цих та інших форм є використання позначень для векторів-рядків, аналогічних до позначень для векторів-стовпців, розглянутих раніше. Позначимо через ⟨a| вектор-рядок, який має 1 у стовпчику, що відповідає a, і нуль у всіх інших стовпчиках, для кожного $a \in \Sigma$. Цей вектор читається як «бра a». Наприклад: ⟨0|=(1 0) та ⟨1|=(0 1).

Для довільного вибору класичної множини станів Σ, розглядаючи вектори рядків та вектори стовпців як матриці та виконуючи матричне множення |b⟩⟨a|, отримуємо квадратну матрицю, яка має 1 в елементі, що відповідає парі (b,a), що означає, що рядок елемента відповідає b, а стовпець - a, і 0 для всіх інших елементів. Приклад: $|0\rangle\langle 1| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ . Тепер відобразити M як, $M = \sum_{a\in\Sigma} |f(a)\rangle\langle a|$, що показує зв'язок між простою функції і квантовою функцією.

Тепер, якщо ми знову подумаємо про вектори як про матриці, але цього разу розглянемо множення ⟨a||b⟩, то отримаємо матрицю 1×1, яку ми можемо розглядати як скаляр (тобто число). Запишемо цей добуток як ⟨a|b⟩, а не ⟨a||b⟩. Це задовольняє наступну формулу: $M|b\rangle = (\sum_{a\in\Sigma} |f(a)\rangle\langle a|)|b\rangle = \sum_{a\in\Sigma} |f(a)\rangle\langle a \mid b\rangle = |f(b)\rangle$.

Тепер перейдем до ймовірнісних операцій. Розглянемо операцію над бітом , якщо класичний стан біта дорівнює 0, то його залишають у спокої, а якщо класичний стан біта дорівнює 1, то його перевертають у 0 з ймовірністю 1/2. Ця операція описується матрицею: $\begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}$.

Імовірнісні операції на інтуїтивному рівні - це операції, в яких випадковість може бути якимось чином використана або введена під час операції, як у наведеному вище прикладі. Кожен стовпчик можна розглядати як векторне представлення ймовірнісного стану, який генерується при будь-якому класичному вхідному стані, що відповідає цьому стовпі. Інший спосіб думати про імовірнісні операції полягає в тому, що вони є випадковим вибором детермінованих операцій. Наприклад, ми можемо думати про операцію у наведеному вище прикладі як про застосування або функції тотожності, або функції постійного 0, кожна з яких має ймовірність 1/2. Як у цьому випадку: $\begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$.

Що стосується комбінації ймовірнісних операцій то просто відповідні матриці множаться, відповідні такі комбінації є асоціативними через асоціативність множення матриць: $M_2M_1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$, $M_1M_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$.

Розглянемо основні позначення зв’язаних квантових станів та, що вони означають. X та Y - біти, так що їх відповідні класичні множини станів {0,1} відповідно. Ось імовірнісний стан пари (X,Y):

$$\Pr((\mathrm{X},\mathrm{Y}) = (0,0)) = 1/2$$
$$\Pr((\mathrm{X},\mathrm{Y}) = (0,1)) = 0$$
$$\Pr((\mathrm{X},\mathrm{Y}) = (1,0)) = 0$$
$$\Pr((\mathrm{X},\mathrm{Y}) = (1,1)) = 1/2 \qquad (2.4.2)$$

Це імовірнісний стан, в якому і X, і Y є випадковими бітами - кожен з них дорівнює 0 з ймовірністю 1/2 і 1 з ймовірністю 1/2 - але класичні стани цих двох бітів завжди збігаються. Це приклад кореляції між цими системами. Вимірювання змінює наші знання про систему, а отже, змінює ймовірнісний стан, який ми пов'язуємо з цією системою: якщо ми визнаємо, що X перебуває в класичному стані a ∈ Σ, то новий вектор ймовірності, який представляє наші знання про X, стає вектором з 1 у елементі, що відповідає a, і 0 у всіх інших елементах. Цей вектор вказує на те, що X перебуває в класичному стані a з вірогідністю, яку ми знаємо, щойно розпізнавши його.

Особливим типом ймовірнісного стану двох систем є стан, в якому системи є незалежними. Дві системи є незалежними, якщо знаходження класичного стану однієї з них не впливає на ймовірності, пов'язані з іншою. Тобто, знання класичного стану однієї з систем не дає жодної інформації про класичний стан іншої. Для визначення цього поняття припустимо, що X та Y - це системи з класичними множинами станів Σ та Γ відповідно. Відносно заданого ймовірнісного стану цих систем кажуть, що вони є незалежними, якщо виконується наступне: $\Pr((\mathrm{X},\mathrm{Y}) = (a,b)) = \Pr(\mathrm{X} = a)\Pr(\mathrm{Y} = b)$, де a ∈ Σ та b ∈ Γ. Щоб виразити цю умову в термінах векторів ймовірностей, припустимо, що заданий ймовірнісний стан (X,Y) описується вектором ймовірності, який записується в нотації Дірака як: $\sum_{(a,b)\in\Sigma\times\Gamma} p_{ab}|ab\rangle$, відповідно дану ймовірність можна представити, як комбінацію двох векторів ймовірності: $|\phi\rangle = \sum_{a\in\Sigma} q_a|a\rangle$ та $|\psi\rangle = \sum_{b\in\Gamma} r_b|b\rangle$, із цього получається $p_{ab}=q_a r_b$. Нехай $|\phi\rangle = \frac{1}{4}|0\rangle + \frac{3}{4}|1\rangle$ та $|\psi\rangle = \frac{2}{3}|0\rangle + \frac{1}{3}|1\rangle$, тоді $|\phi\rangle + |\psi\rangle = \frac{1}{6}|00\rangle + \frac{1}{12}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{4}|11\rangle$.

Умова незалежності, яку ми щойно описали, може бути більш стисло виражена через поняття тензорного добутку. Нехай дано два вектори $|\phi\rangle = \sum_{a\in\Sigma} \alpha_a|a\rangle$ та $|\psi\rangle = \sum_{b\in\Gamma} \beta_b|b\rangle$ , тоді тензорний добуток $|\pi\rangle=|\phi\rangle\otimes|\psi\rangle$ обчислюється як, $|\phi\rangle \otimes |\psi\rangle = \sum_{(a,b)\in\Sigma\times\Gamma} \alpha_a\beta_b|ab\rangle$ , $\langle ab|\pi\rangle=\langle a|\phi\rangle\langle b|\psi\rangle$.

Розглянемо, що таке квантовий вентиль – функціональна альтернатива класичного вентиля але який має багато відмінностей та особливостей. У моделі квантової схеми контакти представляють кубіти, а вентилі - операції, що діють над цими кубітами. Наразі ми зосередимось на операціях, з якими ми вже зустрічались, а саме на унітарних операціях та вимірюваннях у стандартному базисі. Коли ми дізнаємось про інші види квантових операцій та вимірювань, ми вдосконалимо нашу модель відповідним чином.

У цій схемі ми маємо один кубіт X, який зображено горизонтальною лінією, і послідовність вентилів, що представляють одиничні операції над цим кубітом. Потік інформації йде зліва направо - отже, перша виконана операція - це операція H(Адамара), друга - операція S(Фазовго зміщення), третя - ще одна операція Адамара, і остання операція - операція T(pi/8). Таким чином, при застосуванні всієї схеми застосовується композиція цих операцій,HSHT, до кубіту X.


Рисунок 2.4.1 – Схема HSHT для X

Наприклад, якщо ми застосуємо операцію HSHT до стану $|0\rangle$, то отримаємо стан $\frac{1+i}{2}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$.

Приведемо інший випадок:

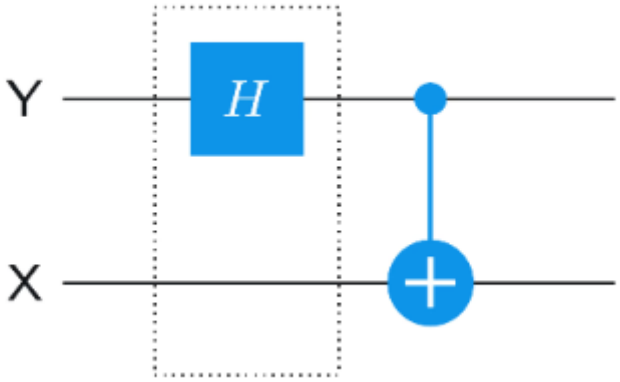


Рисунок 2.4.2 – Схема H та $\oplus$,

Вентиль, позначений H, відноситься до операції Адамара, в той час як другий вентиль - це вентиль з двох кубітів: це операція контрольоване-не контрольоване, де суцільне коло позначає керуючий кубіт, а коло, що нагадує символ $\oplus$, позначає цільовий кубіт. Таким чином, у наведеній вище схемі ми розглядаємо схему як операцію над двома кубітами (X,Y). Якщо на вхід подано $|\psi\rangle|\phi\rangle$, то нижній кубіт (X) починає роботу у стані $|\psi\rangle$, а верхній кубіт (Y) - у стані $|\phi\rangle$. Розглянемо поетапно два вентиля[11]:

Вентиль Адамара. При застосуванні вентиля до одного кубіта нічого не відбувається з іншими кубітами, що еквівалентно операції тотожності. У нашій схемі є лише один інший кубіт, X, $M_2$ – функція тотожності вказана вище, пунктирний прямокутник на рисунку вище представляє цю операцію:

$$M_2 \otimes H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \qquad (2.4.3)$$

Дія вентилю controlled-NOT у стандартних базових станах така:

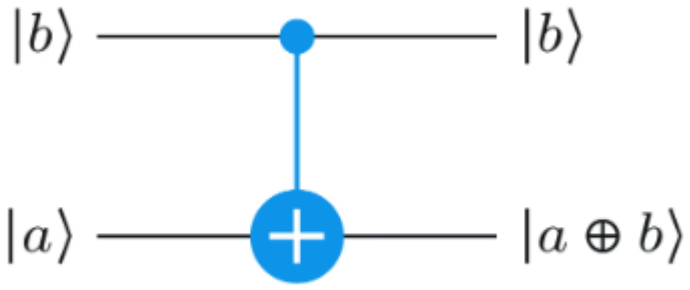


Рисунок 2.4.3 – Схема $\bigoplus$[11]

Матричний вигляд вентиля - $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$. Відповідно унітарна операція для цієї схеми обчислюється так:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \end{pmatrix}. \quad (2.4.4)$$

Вище вказану матрицю можна представити у формі Белла(формі квантово запутаних), ця форма представлення показує окрему унітарну матрицю для кожного можливого стану:

$$\begin{aligned} |\phi^+\rangle &= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \\ |\phi^-\rangle &= \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle \\ |\psi^+\rangle &= \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \\ |\psi^-\rangle &= \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \end{aligned} \quad (2.4.5)$$

знаходимо окремі унітарні матриці,

$$
\begin{aligned}
U|00\rangle &= |\phi^+\rangle \\
U|01\rangle &= |\phi^-\rangle \\
U|10\rangle &= |\psi^+\rangle \\
U|11\rangle &= -|\psi^-\rangle
\end{aligned}
\tag{2.4.6}
$$

Розглянемо одну із самих перспективних схем QKD. DI-QKD - метод захищеного зв'язку, який гарантує безпеку ключів навіть у випадку, коли квантові пристрої, що використовуються, не є надійними або навіть зловмисними. Безпека DI-QKD базується на перевірці порушення нерівностей Белла, які вимірюють ступінь квантового заплетення між частинками, що використовуються для генерації ключів. Якщо нерівності Белла не порушуються, це означає, що ключі не є випадковими або що їх може знати третя сторона. Якщо нерівності Белла порушуються, це означає, що ключі є випадковими і приватними, і що ніхто не може їх перехопити або підробити.

У цьому контексті DI-QKD можна вважати остаточним рішенням проблеми безпечної реалізації QKD, оскільки він не вимагає характеризувати внутрішнє функціонування будь-якого пристрою. Концептуально він базується на історичному протоколі Ekert 91, де центральне ненадійне джерело розподіляє заплутані пари фотонів між двома сторонами, скажімо, Алісою і Бобом - кожна з яких забезпечена одиницею вимірювання - і порушення нерівності Белла сигналізує про безпеку квантового каналу. Виконуючи локальні вимірювання падаючих фотонів, сторони можуть засвідчити наявність моногамних кореляцій між результатами своїх вимірювань, спираючись лише на свою статистику входів і виходів. Коли їхня статистика порушує нерівність Белла, то гарантовано, що їхні результати не є наслідком заздалегідь визначеної стратегії, яку приписують супротивнику, якого зазвичай називають Євою. Фактично даний спосіб є гібридом між квантовою криптографією та традиційною криптографією.

Експериментальна реалізація так званого «тесту Белла без лазівок»(loophole-free Bell test) є не простою задачею. Заплутаність має бути розподілена між віддаленими спостерігачами, які повинні бути здатні виконувати випадкові вимірювання на високій швидкості і з дуже високою ефективністю. В рамках технологічного турніру на витривалість за останні роки

було проведено кілька тестів Белла без лазівок. Слід зазначити, що для всіх досліджених протоколів можливість виконання DI-QKD є більш суворою, ніж сам тест Белла, оскільки він вимагає великого порушення Белла. Незважаючи на це, нещодавно з'явилися демонстрації правильної роботи DI-QKD.

Спочатку опишемо припущення на якому ґрунтується секретність DI-QCD:

- Відома квантова механіка (КМ) є коректною. Під цим мається на увазі, що, зокрема, статистика вимірювань приладів QKD підпорядковується правилу Борна для деяких квантових станів і деяких квантових вимірювань (які не обов'язково повинні бути відомі користувачам).
- Сторони попередньо обмінюються коротким секретним ключем для автентифікації класичних повідомлень.
- Сторони можуть добросовісно генерувати локальну випадковість. Примітно, що це дозволяє «вільний» вибір параметрів вимірювання в точному сенсі, що є достатнім для встановлення повноти QM для передбачення результатів вимірювання - припущення, на яке часто посилаються самі по собі - через виключення сумісних з ним більш прогностичних теорій.
- Сторони мають вірні класичні пристрої для післяобробки.
- Не відбувається небажаного витоку інформації поза межі локації сторін.

Принципи роботи протоколу DI-QKD. Найбільш популярною нерівністю Белла у двосторонньому сценарії є нерівність Клаузера-Хорна-Шімоні-Холта (CHSH), яка повністю характеризує набір локальних кореляцій в умовах бінарних входів і виходів.

Тест CHSH. Центральне джерело розподіляє квантові стани $\rho_{AB}$ між Алісою та Бобом, які взаємодіють зі своїми вимірювальними пристроями, подаючи двійкові входи (x та y) та записуючи двійкові виходи (a та b), щоб оцінити ймовірність виграшу в CHSH, ω.

Перша сторона $\rho_{AB}$, надсилає x та y до Аліси і Боба відповідно. Після чого Аліса і Боб вирішують яку дати відповідь на запитання $\rho_{AB}$, після чого $\rho_{AB}$ вирішує програш чи виграш за правилом $x \wedge y = a \oplus b$.

Таблиця 2.4.2 – Відображення x ^ y = a $\oplus$ b

| (x,y) | Виграш, якшо: |
|---|---|
| (0,0) | a=b |
| (0,1) | a=b |
| (1,0) | a=b |
| (1.1) | a≠b |

Із даної таблиці логічно можна зробити висновок, що найкраща стратегія для виграшу притримуватись відповіді 0, така стратегія дає ймовірність виграти 75%. Проте, є квантова стратегія набагато ефективніша.

Нехай $|\psi_\theta\rangle = \cos(\theta)|0\rangle + \sin(\theta)|1\rangle$ , тоді $\langle\psi_\alpha \mid \psi_\beta\rangle = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta) = \cos(\alpha - \beta)$. Шукаємо матрицю ймоврностей $|\phi^+\rangle$ тобто для значення при якому a=b = 0 та a=b=1, получаємо $\langle\psi_\alpha \otimes \psi_\beta \mid \phi^+\rangle = \frac{\cos(\alpha)\cos(\beta)+\sin(\alpha)\sin(\beta)}{\sqrt{2}}$. Далі шукаємо унітарну матрицю яка буде обчислювала відповідь, нехай $u_\theta = |0\rangle\langle\psi_\theta| + |1\rangle\langle\psi_{\theta+\pi/2}|$, тригонометричні властивості дають, що $\langle\psi_{\theta+\pi/2} \mid \psi_\theta\rangle = \cos(\pi/2) = 0$.

Для подальшої стратегії потрібно, щоб Аліса і Боб ділили два запутаних кубіта у стані $|\phi^+\rangle$ . Далі Аліса і Боб мають притримуватись вибору аргументу функції $u_\theta$ відповідно до таблиці:

Таблиця 2.4.3 – Залежність $\underline{\theta}$ від x,y

| Аліса | Боб |
|---|---|
| $\begin{cases} u_0 & \text{if } x = 0 \\ u_{\pi/4} & \text{if } x = 1 \end{cases}$ | $\begin{cases} u_{\pi/8} & \text{if } y = 0 \\ u_{-\pi/8} & \text{if } y = 1 \end{cases}$ |

Маючи це можна підрахувати, яку загальну точність будуть мати функції $u_\theta$, для Аліси і Боба на одному раунді.

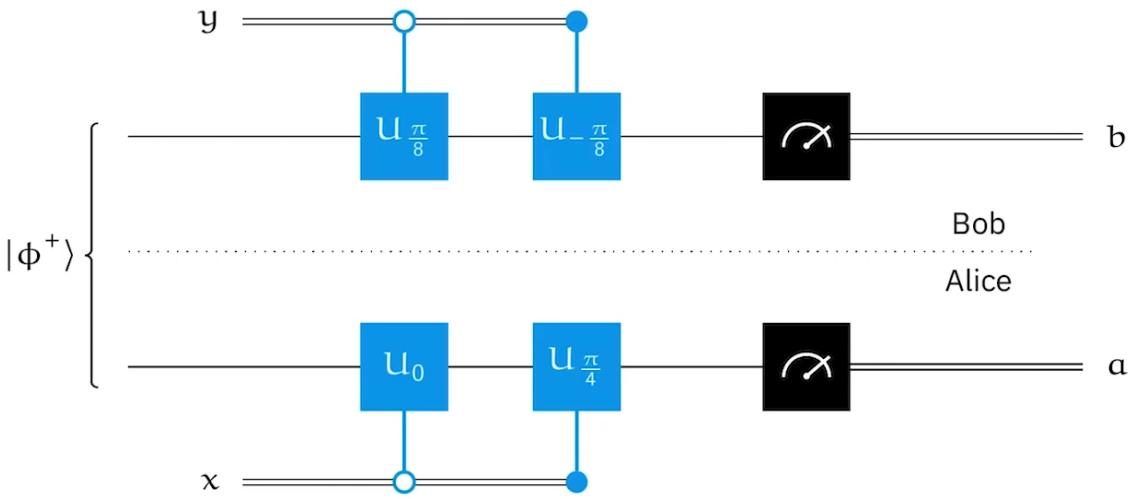


Рисунок 2.4.4 – Схема оптимальної стратегії[11]

Якщо взяти (x,y) = (0,0) тоді

$$\left(U_0 \otimes U_{\frac{\pi}{8}}\right)|\phi^+\rangle = \frac{\cos\left(-\frac{\pi}{8}\right)|00\rangle+\cos\left(-\frac{5\pi}{8}\right)|01\rangle+\cos\left(\frac{3\pi}{8}\right)|10\rangle+\cos\left(-\frac{\pi}{8}\right)|11\rangle}{\sqrt{2}} \qquad (2.4.7)$$

получаєм таблицю, ймовірностей для (x,y) = (0,0):

Таблиця 2.4.4 – Наглядний результат функції, $U_0 \otimes U_{\frac{\pi}{8}}$.

| (a, b) | Ймовірність | Значення |
|---|---|---|
| (0,0) | $\frac{1}{2}\cos^2\left(-\frac{\pi}{8}\right)$ | $\frac{2+\sqrt{2}}{8}$ |
| (0,1) | $\frac{1}{2}\cos^2\left(-\frac{5\pi}{8}\right)$ | $\frac{2-\sqrt{2}}{8}$ |
| (1,0) | $\frac{1}{2}\cos^2\left(\frac{3\pi}{8}\right)$ | $\frac{2-\sqrt{2}}{8}$ |
| (1,1) | $\frac{1}{2}\cos^2\left(-\frac{\pi}{8}\right)$ | $\frac{2+\sqrt{2}}{8}$ |

Рахуємо загальну ймовірність перемоги: $\Pr(a=b)=\frac{2+\sqrt{2}}{4}\approx 0.85$ $\Pr(a\neq b)=\frac{2-\sqrt{2}}{4}\approx 0.15$, шанс виграти використовуючи у цьому випадку(x,y) функцію $u_0 \approx 85\%$ або іншими словами така точність функції $u_0$. Якщо підрахувати так для (0,1) та (1,0), то буде відповідно такий самий результат, але якщо взяти (1,1), то результат буде протилежний: $\Pr(a=b)=\frac{2-\sqrt{2}}{4}\approx 0.15$ $\Pr(a\neq b)=\frac{2+\sqrt{2}}{4}\approx 0.85$. Але це не проблема тому, що у цьому випадку a≠b. Тому ймовірність на всі множення (x,y) ≈ 85% і це набагато кращий результат ніж стратегія завжди відповідати 0.

Протокол DI-QKD на основі CHSH виконується послідовно, випадковим чином чергуючи ключові раунди (де сторони вимірюють свої частки у фіксованих кореляційних базисах) і тестові раунди (де вони виконують CHSH, щоб кількісно оцінити інформацію Єви про результати ключових раундів). Ключовим базисом обрано один з тестових базисів Аліси, а пристрій Боба має додаткове налаштування для роботи у цьому ж базисі. Як завжди, нерелевантні пари вибору базису відкидаються апостеріорі.

Розглянемо типову модель з обмеженою ефективністю, де кожен фотон незалежно втрачається з фіксованою ймовірністю $\eta_{det}$ , яка визначає загальну ефективність виявлення системи. Далі, припускаючи ідеальну підготовку станів Белла, позитивна асимптотична швидкість ключа (тобто, вважаючи, що сторони виконують нескінченну кількість раундів протоколу) вимагає, щоб ефективність фотодетекторів задовольняла умовам $\eta_{det} > 92.4\%$ ,якщо мають місце лише втрати при виявленні (але не втрати каналу). Цей показник можна зменшити до 90,9%, якщо Боб скасує призначення невиявлених сигналів для цілей IK66. Крім того, якщо виникають лише канальні втрати (але не втрати при виявленні), позитивна ключова швидкість вимагає, щоб відстань між користувачами не перевищувала L < 3,5 км, враховуючи типове оптичне волокно з коефіцієнтом ослаблення 0,2 дБ/км на телекомунікаційній довжині хвилі.

Також потрібно механізм сповіщення який буде вказував на точність. Механізм сповіщення - це інструмент, який інформує сторони про прихід фотона

або успішний розподіл заплутаності між ними. Таким чином, можна відокремити втрати в каналі від налаштувань вимірювання, просто відклавши вибір останніх до того моменту, коли відбудеться сповіщення. Це дозволяє відкинути сигнали, втрачені в каналі, не відкриваючи лазівки для виявлення. Рішенням є так звані «квантові підсилювачі» (QA). Якщо залишити осторонь технічні деталі, то QA, по суті, є телепортаційними воротами, розташованими на місці однієї зі сторін, так що успішна телепортація локально попереджає цю сторону про прибуття фотона. Одна з можливостей полягає в тому, щоб розмістити джерело заплутування $\rho_{AB}$ на ділянці Аліси так, щоб тільки Боб відчував втрату каналу, і, таким чином, тільки він був оснащений QA. Крім того, можна розмістити $\rho_{AB}$ у каналі і забезпечити як Алісу, так і Боба QA[12]. Наприклад, QA може бути побудовано з допоміжним джерелом заплутаності $\rho_{BC}$ та вимірюванням стану Белла (BSM). У BSM фотон, що рухається від ρAB, інтерферує з фотоном від $\rho_{BC}$. Після успішного виконання BSM Боб отримує попередження про прибуття фотона - що символізується прапорцем у QA - і заплутаність міняється місцями з крайніми фотонами, які потрапляють у вимірювальні пристрої Аліси та Боба. Лише у цьому успішному випадку Боб обирає своє налаштування вимірювання. Таким чином, неперехоплені сигнали можуть бути вилучені без відкриття лазівки для виявлення. Для генерації ключа пристрій Боба допускає третє вхідне значення (y = 2), яке відповідає одній з тестових баз Аліси.

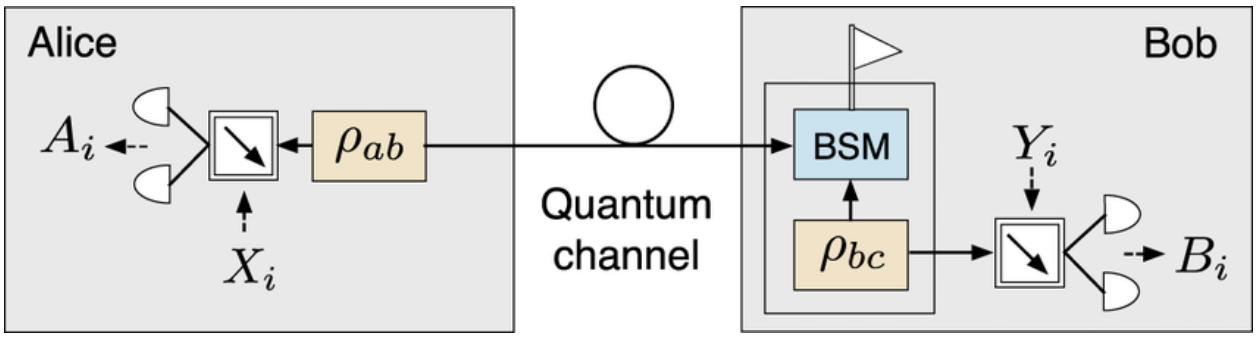


Рисунок 2.4.5 – Схематичне зображення DI-QKD

Науковий прорив в механізмах сповіщення є обов'язковим для збільшення відстані, яку потенційно може подолати DI-QKD. Однак, для їхнього застосування все ще необхідно вдосконалити різні аспекти. Зокрема, загальним

вузьким місцем схем сповіщення є те, що за нинішніх технологій потрібні дуже довгі сеанси DI-QKD для збору блоків даних, необхідних для забезпечення позитивної скінченної довжини ключа на відповідних відстанях. Крім того, в повністю фотонній реалізації продуктивність схеми сповіщення обмежена, якщо розглядати практичні джерела заплутування, які іноді випромінюють вакуумні імпульси або кілька пар фотонів, як, наприклад, ті, що базуються на спонтанному параметричному перетворенні вниз (SPDC) (зауважте, що в умовах без сповіщення були отримані фундаментальні обмеження на максимальне порушення CHSH, досяжне з джерелами SPDC).

## 2.5 Незалежний від вимірювального пристрою квантовий розподіл ключів(MDI-QKD)

MDI-QKD - споріднений підхід, більш симетричний по відношенню до обох сторін, полягає у створенні заплутаності через обмін заплутаними станами. У цьому випадку локальні квантові системи на обох сайтах відповідно заплутуються фотонними станами, які надсилаються до центрального вузла, де відбувається обмін. Звідти успішний розподіл заплутаності повідомляється назад обом сторонам. Цей підхід забезпечує основу для нещодавніх експериментів з DI-QKD на основі пам'яті, де передбачувана заплутаність встановлюється між довгостроковою квантовою пам’яттю. Протокол MDI-QKD вимагає, щоб Аліса і Боб незалежно генерували одиночні фотони у вигляді нерозрізнених слабких когерентних імпульсів (WCPs), які надсилаються Чарлі і проектуються на один з чотирьох станів Белла.

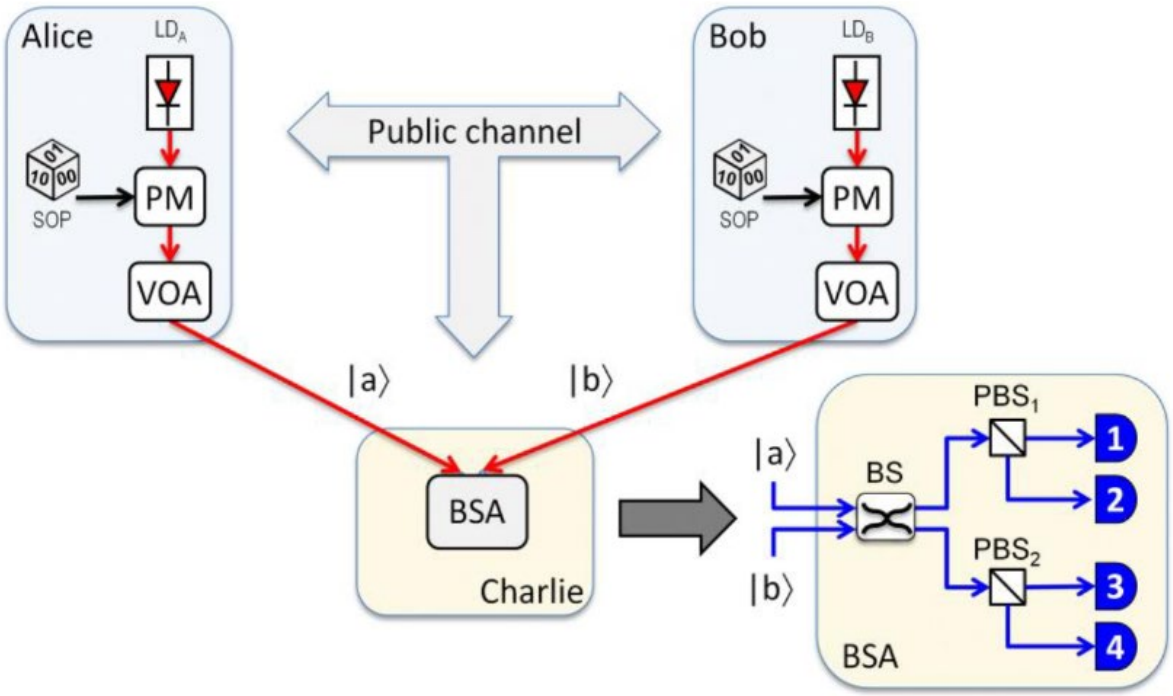


Рисунок 2.5.1 – Схема MDI-QKD

SOP слабких імпульсів випадковим чином і незалежно кодуються в стан один із двох взаємно незміщених базисів у двовимірному гільбертовому просторі. Зазвичай обирають прямолінійний базис (⊕), що складається з горизонтальних і вертикальних SOP, і діагональний базис (⊗), що визначається SOP +45° і -45°, подібно до традиційних протоколів QKD, таких як BB84[13]. Таким чином, вихідні сигнали лазерних діодів (LD) Аліси і Боба випадковим чином поляризаційно модулюються, відповідно до вибору базису і біта, і надсилаються Чарлі, після проходження через змінні оптичні атенюатори (VOA) для встановлення бажаної середньої кількості фотонів на імпульс. Поляризаційний BSA на основі лінійної оптики виконує вимірювання вхідних фотонів, а результати публічно повідомляються Алісі і Бобу, які потім переходять до узгодження базису, виправлення помилок і посилення конфіденційності, як в традиційних протоколах QKD.

Можливими результатами BSA є поодинокі виявлення в будь-якому з чотирьох детекторів, спричинені одним або кількома фотонами, спрямованими на один і той самий SPD (що є непереконливими подіями), а також підрахунки збігів між парами однофотонних детекторів. Всі одиночні детектування

ігноруються протоколом, оскільки наявні в даний час SPD на основі APD зазвичай не мають роздільної здатності за кількістю фотонів, щоб відрізнити їх від однофотонних подій (в деяких часових вікнах комбінація чисел фотонів, відмінних від одиниці, від Аліси і Боба, може досягти BSA). Про збіг подій Чарлі публічно оголошує як остаточний результат БСА. На етапі узгодження базисів всі випадки, коли Аліса і Боб посилали стан з несумісних базисів ($\oplus$ з $\otimes$ та навпаки), також відкидаються, як і в традиційних протоколах на основі BB84. Сумісні за базисом достовірні результати BSA підсумовано в таблиці I нижче для ідеального випадку однофотонних станів Фока, одночасно посланих Алісою і Бобом, а також для слабких когерентних імпульсів. Варто зазначити, що модифікація станів-приманок дозволяє Алісі та Бобу отримати інформацію про статистику однофотонних імпульсів, коли використовуються когерентні стани, які, зрештою, є корисними подіями для генерації ключа.

Імовірнісний відгук BSA чарлі за протоколом MDI-QKD для однофотонних імпульсів(Singe photons) та WCP при узгоджених базах. Прямолінійний базис використовується для генерації ключів, в той час як діагональний базис використовується для безпеки .

Таблиця 2.5.1 − Імовірнісний відгук BSA.

| ⊕ базис | | BSA | | | |
|---|---|---|---|---|---|
| SOP | | Однофотонний | | WCP | |
| Alice | Bob | $\|\psi+\rangle$ | $\|\psi-\rangle$ | $\|\psi+\rangle$ | $\|\psi-\rangle$ |
| $\|H\rangle$ | $\|H\rangle$ | 0 | 0 | 0 | 0 |
| $\|V\rangle$ | $\|V\rangle$ | 0 | 0 | 0 | 0 |
| $\|H\rangle$ | $\|V\rangle$ | 0.5 | 0.5 | 0.5 | 0.5 |
| $\|V\rangle$ | $\|H\rangle$ | 0.5 | 0.5 | 0.5 | 0.5 |
| $\|+45\rangle$ | $\|+45\rangle$ | 1 | 0 | 0.75 | 0.25 |
| $\|-45\rangle$ | $\|-45\rangle$ | 1 | 0 | 0.75 | 0.25 |
| $\|+45\rangle$ | $\|-45\rangle$ | 0 | 1 | 0.25 | 0.75 |
| $\|-45\rangle$ | $\|+45\rangle$ | 0 | 1 | 0.25 | 0.75 |

Оскільки використовується частковий BSA на основі лінійної оптики, то лише $|\Psi^{+}\rangle$ та $|\Psi^{-}\rangle$ стани Белла, які відповідають виявленням збігів типу $C_{12}$ або

$C_{34}$ та $C_{14}$ або $C_{23}$ відповідно, де $C_{ij}$ - збіги між детекторами i та j (рис. 2.4.6, синій півкруг), можна однозначно відрізнити від інших станів Белла $|\phi^{+}\rangle$ та $|\phi^{-}\rangle$, причому останні випадки відповідають двом фотонам в одній просторовій моді (один і той самий детектор). Коли Аліса і Боб надсилають однакові SOP в базисі ⊕, BSA ніколи не видає дійсної події виявлення збігу ($C_{13}$ і $C_{24}$ не пов'язані з проекцією Белла. Коли ортогональні поляризації в базисі ⊕ надходять до Чарлі, BSA вихід буде $|\Psi^{+}\rangle$ або $|\Psi^{-}\rangle$ також Аліса або Боб повинні виконати перестановку бітів, щоб співвіднести їхні біти. Нарешті, коли використовується базис ⊗, BSA вихід буде $|\Psi^{+}\rangle$ або $|\Psi^{-}\rangle$ в залежності від комбінації надісланих станів згідно таблиці , і перекидання бітів потрібне лише коли $|\Psi^{-}\rangle$ генерується в BSA. Біти, отримані з базису ⊕, використовуються для формування спільного секретного ключа між Алісою та Бобом, тоді як інший базис використовується для тестування частоти помилок та пропускної здатності каналу за допомогою станів-приманок. Це полягає у випадковому надсиланні імпульсів з різною середньою кількістю фотонів для тестування проти селективного підслуховування багатофотонних імпульсів, оскільки вихід (ймовірність того, що Боб виявить однофотонний імпульс) і можна отримати квантовий коефіцієнт бітової помилки (QBER) для однофотонних імпульсів.

Оскільки слабкі когерентні імпульси мають розподіл пуассона для статистики кількості фотонів, ймовірність того, що Аліса випромінює два фотони, а Боб посилає вакуумний імпульс (або навпаки), вдвічі менша за ймовірність одночасного випромінювання одного фотона кожною зі сторін. Ця особливість вносить шумовий зсув до діагонального базису, що призводить до 25% $|\Psi^{-}\rangle$ подій, коли SOP Аліси і Боба рівні 25% $|\Psi^{+}\rangle$ подій (проти 75% $|\Psi^{+}\rangle$) , коли надсилаються ортогональні стани (проти 75% $|\Psi^{-}\rangle$). Крім того, трапляються збіги $C_{13}$ і $C_{24}$, але вони відкидаються, оскільки не відіграють жодної ролі в протоколі. Незважаючи на хибні збіги, протокол залишається стійким, коли його реалізовано за допомогою WCP, оскільки вони не впливають на прямолінійний базис. Крім того, статистику подій, згенерованих парою одиночних фотонів у

діагональному базисі можна отримати, застосувавши метод приманних станів, як і було спочатку.

## 2.6 Спутниковий QKD(SatQKD)

Супутниковий QKD або інколи  усуває обмеження відстані, притаманні QKD через наземне оптичне волокно, які є бар'єром для квантово-безпечного зв'язку в континентальному, міжконтинентальному і глобальному масштабах. Дослідження і розробки, проведені командами по всьому світу, призвели до нещодавніх демонстрацій технічної можливості супутникового QKD,  і вони лежать в основі низки сучасних міжнародних програм. Тому варта розглянути дану концепція.

Перші демонстрації квантового зв'язку на супутнику Micius вагою ~650 кг показали, що SatQKD і розподіл заплутаності можливі в рекордних масштабах. Спираючись на ці результати, задачі малих супутників (<100 кг) є привабливими завдяки меншим витратам на розробку і швидшому часу розробки порівняно з традиційними великими супутниками. Однак обмежені розміри, вага і потужність малих супутників і обмежені можливості ставлять їх у невигідне становище порівняно з більшими супутниками, такими як Micius. Незважаючи на це, техніко-економічне обґрунтування QKD на базі малих супутників і демонстраційні місії з пошуку траєкторій на базі CubeSat(супутник, 10 см., 2 кг.) є багатообіцяючими[14]. Для супутників на низькій навколоземній орбіті (LEO) особливим викликом є обмежений часовий проміжок для роботи квантового каналу з оптичною наземною станцією (OGS). Це обмеження непропорційно обмежує обсяг захищених ключів, які можуть бути згенеровані через виражений вплив статистичної невизначеності оцінюваних параметрів. Разом з обмеженими можливостями SWaP(Size, Weight and Power) малі супутникові місії працюють в умовах обмеженого обсягу квантової інформації. Розумінню впливу цих обмежень на SatQKD приділялося мало уваги, хоча воно має як безпосереднє, так і практичне значення для майбутніх супутникових місій. Тут ми заповнюємо

цю прогалину, встановлюючи практичні межі продуктивності роботи SatQKD при репрезентативному наборі фізичних ресурсів.

Одна з проблем, з якою стикається супутниковий QKD (SatQKD), виникає через ефекти кінцевого розміру ключа, які стають значними через відносно короткі рядки ключів, що генеруються в супутниковому шляхопроводі. Щоб забезпечити відповідність і безпеку ключів на квантовому передавачі (QTx) і приймачі, застосовуються процеси, відомі як корекція помилок (EC) і посилення конфіденційності (PA). Після цих процесів вихідний ключовий рядок буде зменшено на коефіцієнт, який залежить від частоти помилок і параметрів системи. У наземних каналах зв'язку з QKD, де робота є безперервною, довгі ключі можуть подаватися на вхід PA, і вихідний захищений ключ може наближатися до асимптотичної межі. Це теоретична максимальна довжина вихідного ключа, яка виводиться з припущення, що вхідні ключі мають нескінченну довжину. Однак, для дуже коротких рядків, що подаються на вхід PA, ключ скорочується на більшу величину через статистичну невизначеність. Для особливо коротких ключових рядків процес PA може призвести до відсутності вихідного ключа взагалі. Щоб подолати цю проблему, системи SatQKD повинні бути розроблені таким чином, щоб максимізувати ефективність передачі між передавачем і приймачем. Це дорого, оскільки ціна телескопів зростає приблизно експоненціально з розміром апертури. Для більш бюджетних і комерційно життєздатних розмірів апертури можна замість цього спробувати збільшити кількість кубітів, що передаються в межах заданого часового вікна. Цього можна досягти, збільшивши тактову частоту системи з сотень мегагерц до гігагерц. Однак це також збільшує обсяг класичних даних, які необхідно передавати між супутником і наземною станцією. Для радіозв'язку це є надзвичайно складним завданням через велику розбіжність променів радіочастотного зв'язку і низьку потужність антени на невеликих супутникових платформах, таких як CubeSat або Microsat. Лазерний зв'язок (lasercomms), однак, може легко забезпечити пропускну здатність, необхідну для передачі всієї релевантної інформації в межах часового вікна естакади.

Система інтегрує передавач QTx з високошвидкісним квантовим генератором випадкових чисел (QRNG), квантовий приймач (QRx), класичний зв'язок через лазерний зв'язок, а також наведення і стеження (PAT) за допомогою швидкодіючих дзеркал (FSM) у замкнутому контурі зворотного зв'язку з використанням маячкових лазерів. У наступних розділах ми визначимо деталі емуляції шляхопроводу, опишемо апаратне забезпечення системи і покажемо продуктивність нашої системи впродовж всього шляхопроводу.

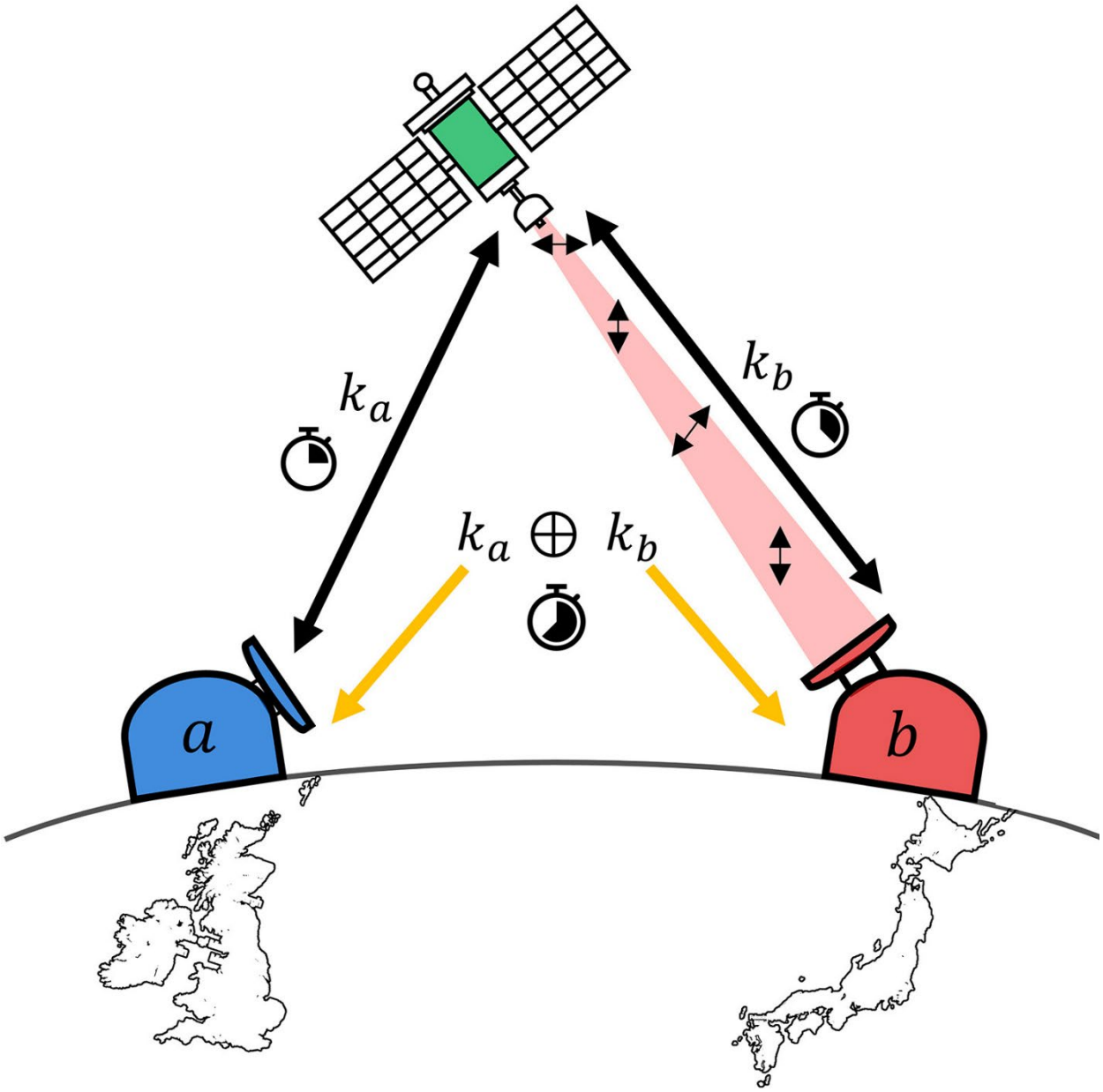


Рисунок 2.6.1 – Принцип SatQKD

Метод, який забезпечує квантовий безпечний зв'язок між двома наземними вузлами, опосередкований супутниковим зв'язком на низькій орбіті, показаний на рис. 2.4.7. Супутник, на якому знаходиться QTx, виконує сеанс QKD з першим користувачем a, який отримує сигнал QKD за допомогою QRx і генерує безпечний квантовий ключ ka. Супутник завершує цей сеанс QKD, і новий сеанс виконується з користувачем b, формуючи другий унікальний квантовий ключ kb. Після цього супутник може публічно транслювати операцію виключного АБО (XOR) двох розподілених ключів ka ⊕ kb. Виконуючи операцію XOR ключа зі своєї унікальної сесії з ключем публічно трансльованого шифрованого тексту, два користувачі, a і b, можуть отримати ключ іншого користувача, тобто kb = ka ⊕ (ka ⊕ kb). У цій роботі ми розглянемо, як наша система працює під час типового сеансу SatQKD, тобто для розподілу ключів від супутника до наземного користувача. Для цього ми спочатку повинні оцінити очікувані втрати.

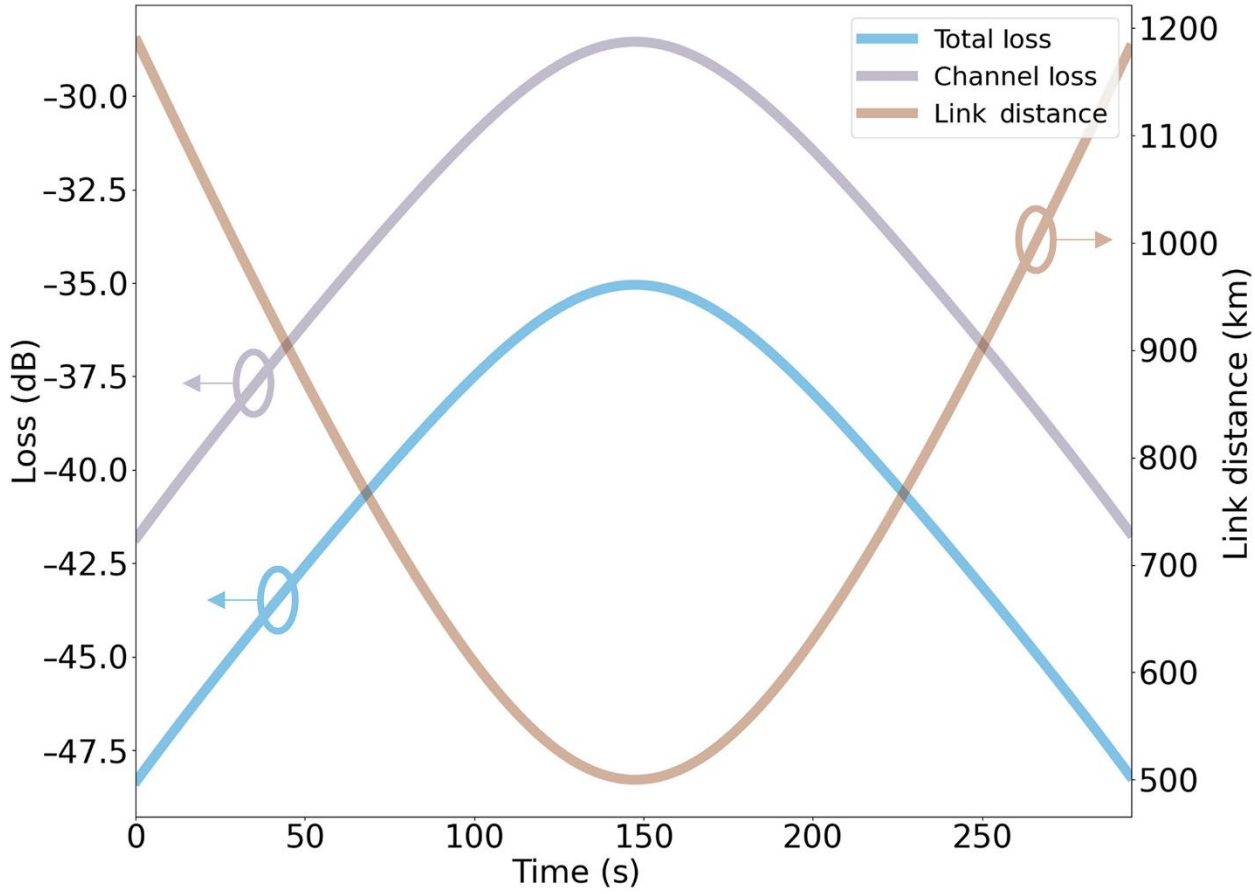


Рисунок 2.6.2 – Графік ефективності SayQKD

Моделюючи очікувану ефективність передачі через супутниковий обхід, оцінивши втрати в каналі через низку різних процесів. До них відносяться дифракція, поглинання в атмосфері, похибка наведення, ефекти турбулентності, ефективність оптичних компонентів наземної станції та ефективність однофотонного детектування. Ефективність передачі за одиницю часу проходження супутника використовується для визначення втрат в каналі в наших лабораторних експериментах. Ми досліджували функціональність нашої системи для багатьох різних проходжень, де максимальна висота над рівнем моря варіюється. Ми починаємо з розгляду "оптимального" прольоту, коли супутник проходить безпосередньо над землею з висотою при найближчому зближенні 90°, і вважаємо, що система QKD працює в діапазоні кутів підйому від 20° до 160°, в результаті чого загальний час прольоту $t_{pass} \approx 294$ с.[15]. Коли всі втрати розглядаються разом, ми оцінюємо середні втрати в каналі (внаслідок дифракції, атмосферного поглинання і турбулентності) для квантової низхідної лінії зв'язку на рівні 34,00 дБ при оптимальному проходженні над рівнем моря. Втрати в каналі, сумарні втрати (які також включають ефективність виявлення і втрати, зумовлені оптикою збору) і відстань до лінії зв'язку показані як функція часу ($t_{pass}$) на рис. 1B. Детальні параметри переходу, включаючи втрати у висхідному і низхідному каналах для допоміжних каналів, які враховуються в наших експериментах, можна знайти в Матеріалах і методах та Додаткових матеріалах. Для досягнення очікуваних втрат досліджувані в цій роботі переходи передбачають роботу в нічний час, а параметри переходів, що працюють вдень, будуть предметом подальших досліджень.

## 2.7 Висновок до другого розділу

Були розглянуті 3 PQC шифри, із 4 сертифікованих NIST. Беручи до уваги розмір ключів, підпису та швидкодії, шифри які базуються на ґратках не мають конкуренції. Для прикладу,  Kyber має відчутно більший розмір ключів та підпису але він є у 2 рази швидшим ніж алгоритми шифрування на еліптичних кривих. Тому використання шифрів Kyber та Falcon є економічно доцільними.

Проте не варто забувати про SPHINCS+, цей алгоритм підпису є одним серед найповільніших і найбільших за розміром ключів і підпису. Однак SPHINCS+ можна рахувати найбільш стійким серед PQC алгоритмів, Постквантова криптографія, відносно, нова галузь тому до кінця не зрозуміло як себе поведуть дані шифри. Но із ситуація інша SPHINCS+ має добре вивчений вектор атак, тому його доцільно використовувати для підписання дуже важливої інформації.

Із методів квантового захисту інформації були розглянуті DI-QKD на конкретному прикладі MDI-QKD та супутниковій QKD. Обидва методи є перспективними і на даний час важкі в реалізації проте вже є експериментальні установки які справно працюють. Методи квантової передачі інформації, є найбільш стійкими тому, що область пізнання у квантовій механіці незрівнянно менша ніж у математиці. Через то, ще не швидко варта очкувати появи ефективних атак на ці системи.

Неправильно розглядати ці системи як окремі, вони можуть бути ефективно поєднані або навіть гібридизовані. Насправді, теоретично можливо створити єдину систему використовуючи всі засоби шифрування та підпису які були розглянуті у цьому розділі. Однак зараз реалізувати таку ідею буде дуже складно.

# 3 РОЗРОБКА PQC OPENSSL BUILD'A ТА ЙОГО ПРАКТИЧНЕ ЗАСТОСУВАННЯ

## 3.1 Створення версії openssl, яка містить PQC алгоритми

Для того щоб стоврити openssl який підтримує PQC алгоритми, я створюю віртуальну машину Ubuntu на VirtualBox. Для тестування буде використана Openssl бібліотека в яку ми добавимо oqs-provider які містить постквантові алгоритми захисту інформації:

– Створимо теку PQC в якій будуть відбувались всі операції. Також встановим додаткові, потрібні нам застосунки. Та налаштуємо скорочені змінні для терміналу через команду export:

```
1) mkdir PQC
2) sudo apt update
3) sudo apt -y install git build-essential perl cmake autoconf
   libtool zlib1g-dev
4) export WORKSPACE=~/quantumsafe
5) export BUILD_DIR=$WORKSPACE/build
6) mkdir -p $BUILD_DIR/lib64
7) ln -s $BUILD_DIR/lib64 $BUILD_DIR/lib
```

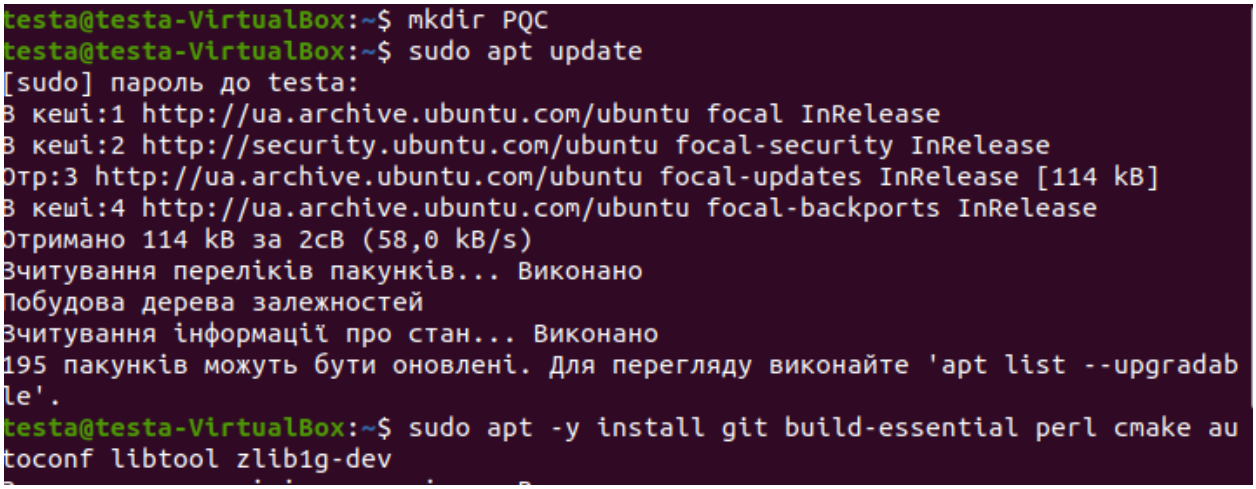


Рисунок 3.1.1 – apt update

– Завантажуємо репозиторій opnessl, він нам буде потрібен для генерації ключів.

git clone https://github.com/openssl/openssl.git

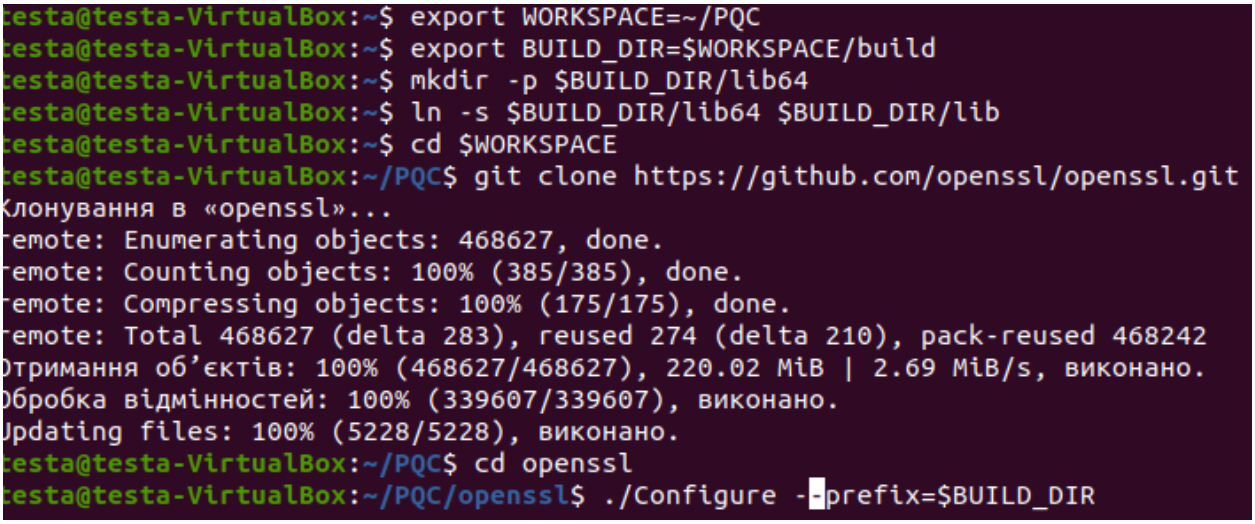


Рисунок 3.1.2 – git clone

Також, потрібно зробити конфігурацію, щоб можна було за допомогою cmake скласти потрібний сценарій.

Після того як встановимо нам потрібно встановити liboqs, який містить бібліотеки із шифрами для PQC шифрування на C. Підчас складання задаємо параметри які будуть нам потррібні для роботи із, для цього використаємо команду:

```
cmake \
      -DCMAKE_INSTALL_PREFIX=$BUILD_DIR \
      -DBUILD_SHARED_LIBS=ON \
      -DOQS_USE_OPENSSL=OFF \
      -DCMAKE_BUILD_TYPE=Release \
      -DOQS_BUILD_ONLY_LIB=ON \
      -DOQS_DIST_BUILD=ON \
        ..
```

– Для спільної роботи Openssl та liboqs нам потрібен oqs-provider, конфігуруємо за допомогою команди[17]:

```
liboqs_DIR=$BUILD_DIR cmake \
      -DCMAKE_INSTALL_PREFIX=$WORKSPACE/oqs-provider \
      -DOPENSSL_ROOT_DIR=$BUILD_DIR \
      -DCMAKE_BUILD_TYPE=Release \
      -S . \
      -B _build
```

cmake --build _build

```
testa@testa-VirtualBox:~/PQC/oqs-provider$ cmake --build _build
Scanning dependencies of target oqsprovider
[  3%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqsprov.c.o
[  7%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqsprov_capabilitie
s.c.o
[ 10%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqsprov_keys.c.o
[ 14%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqs_kmgmt.c.o
[ 17%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqs_sig.c.o
[ 21%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqs_kem.c.o
[ 25%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqs_encode_key2any.
c.o
[ 28%] Building C object oqsprov/CMakeFiles/oqsprovider.dir/oqs_endecoder_commo
```

Рисунок 3.1.3 – Build

Після виконання cmake, появиться помилка тому переносимо бібліотеку всередину openssl вручну:

```
cp _build/lib/* $BUILD_DIR/lib/
```

– Перевіряємо чи включений провайдер oqs-provider в openssl.

```
$BUILD_DIR/bin/openssl list -providers -verbose -provider
oqsprovider
```

```
testa@testa-VirtualBox:~/PQC/oqs-provider$ $BUILD_DIR/bin/openssl list -provide
rs -verbose -provider oqsprovider
Providers:
  default
    name: OpenSSL Default Provider
    version: 3.3.0
    status: active
    build info: 3.3.0-dev
    gettable provider parameters:
      name: pointer to a UTF8 encoded string (arbitrary size)
      version: pointer to a UTF8 encoded string (arbitrary size)
      buildinfo: pointer to a UTF8 encoded string (arbitrary size)
      status: integer (arbitrary size)
  oqsprovider
    name: OpenSSL OQS Provider
    version: 0.5.3-dev
    status: active
    build info: OQS Provider v.0.5.3-dev (f205f11) based on liboqs v.0.10.0-dev
    gettable provider parameters:
      name: pointer to a UTF8 encoded string (arbitrary size)
      version: pointer to a UTF8 encoded string (arbitrary size)
      buildinfo: pointer to a UTF8 encoded string (arbitrary size)
      status: integer (arbitrary size)
```

Рисунок 3.1.4 – oqs-provider

На зображені видно, що окрім стандартного провайдера також входить oqsprovider.

Встановимо додатковий програмне забезпечення, яке дозволить працювати із передачою даних у URL форматі:

```
git clone https://github.com/curl/curl.git
```

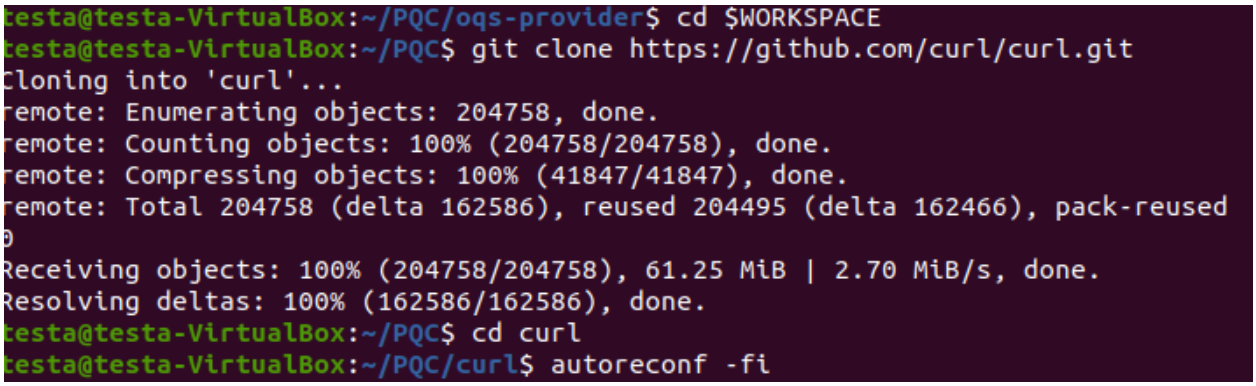


Рисунок 3.1.5 – curl

Налаштування конфігурації знову необіхдно.

–За допомогою сайту https://test.openquantumsafe.org/ тестуємо працездатність алгоритмів. Використовуємо команди:

1) `$BUILD_DIR/bin/curl -vk https://test.openquantumsafe.org/CA.crt --output $BUILD_DIR/ca.cert`

2) `$BUILD_DIR/bin/curl -v --curves p521_kyber1024 --cacert $BUILD_DIR/ca.cert https://test.openquantumsafe.org:6130/`

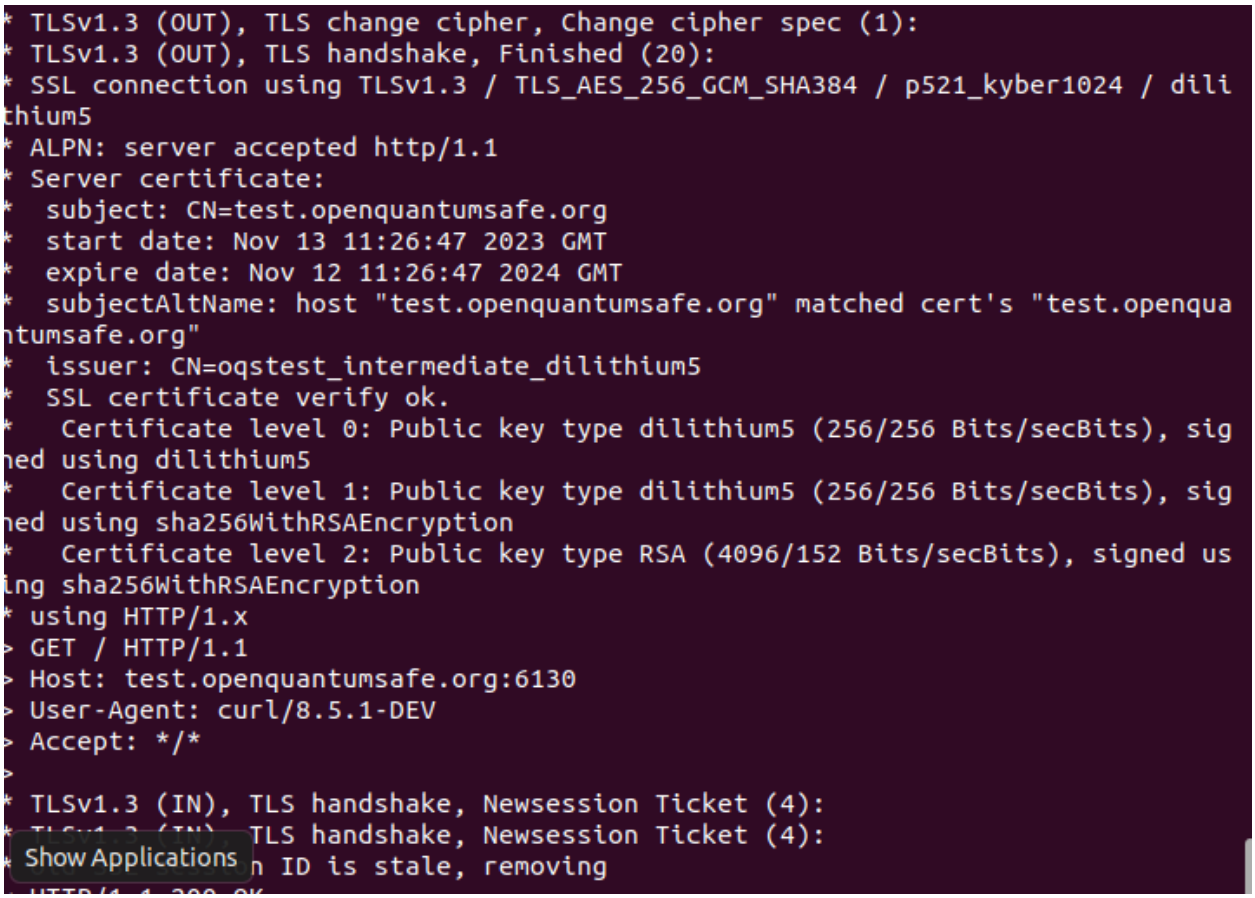


Рисунок 3.1.6 – Перевірка oqs-provider

Як видно із (рис 3.1.5) все чудово працює.

Згенеровану версію openssl можна використовувати для PQC шифрування та підписання. Це ситуативне рішення, в подальшому  не зручно кожного разу включати oqsprovider у openssl. Тому потрібно добавити PQC алгоритми в бібліотеки офіційного релізу, щоб кожен міг ними користуватись.

## 3.2 Аналіз швидкодії постквантових алгоритмів.

Для того щоб розглянути швидкодії PQC алгоритмів встановимо docker та скористаємось зображенням openquantumsafe/curl, яке дозволяє як і тестувати так і здійснювати PQC шифрування та підписання.

Щоб протестувати алгоритми була створена VM ubuntu якій було виділено 4 ядра та 4095 Мб оперативної пам’яті, процесор host - Intel Core i7-2600 3.40 GHz.

Для початку перевірим продуктивність TLS рукостискання для підпису FALCON-512 та алгоритму шифрування Kyber-768, :

```
docker run -e TEST_TIME=5 -e KEM_ALG=kyber768 -e SIG_ALG=falcon512 -it openquantumsafe/curl perftest.sh
```

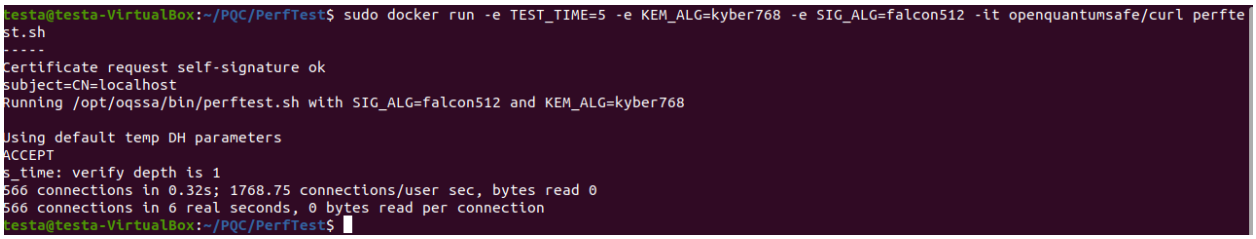


Рисунок 3.2.1 – FALCON-512 та Kyber768

FALCON-512 та Kyber768 дає можливість утворювати ≈1700 з’єднань за секунду.

Використаєм повільний підпис SPHINCS+ (sphincsshake128fsimpl) та Kyber768.

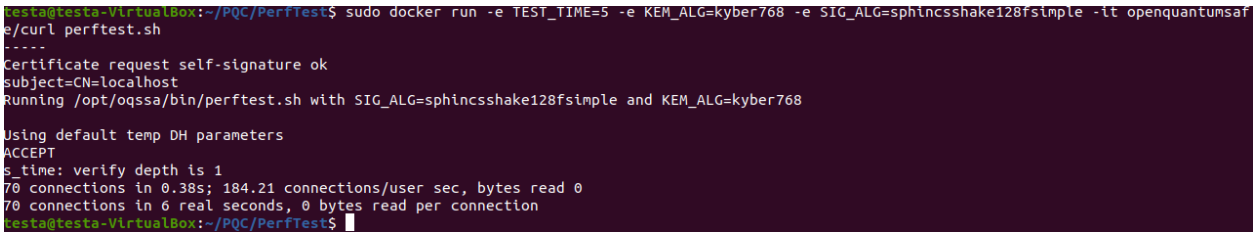


Рисунок 3.2.2 - SHAKE – 128 та Kyber768

Результат: ≈180 з’єднань за секунду, що є приблизно у 10 раз повільніше ніж FALCON-512.

Тепер потрібно складність машиного виконання кожного окремого алгоритму, для цього використаєм команду:

```
docker run -it openquantumsafe/curl openssl speed -seconds 2 falcon512
```

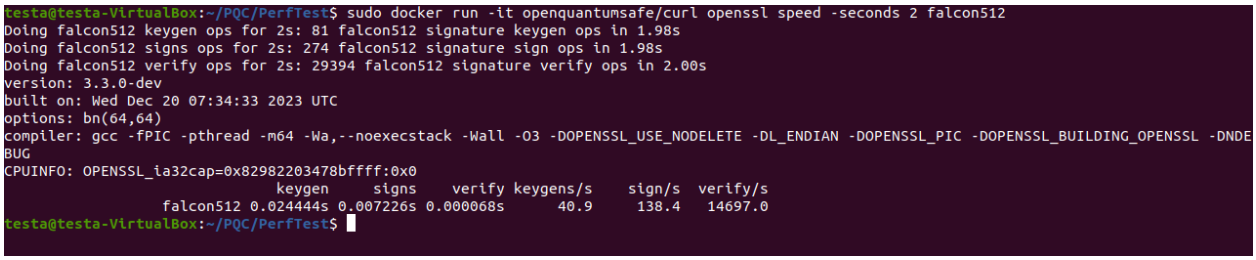


Рисунок 3.2.3 – Продуктивність Falcon 512

Як вказано на (рис. 3.1.9) алгоритм FALCON-512 повільно генерує ключі та підписує але швидко перевіряє сертифікат. FALCON-512 здатен перевірти ≈15 000 сертифікатів за сек., для порівняння ED25519(один із найшвидших алгоритмів) – 7000.

Протестуємо SPHINCS+:

```
testa@testa-VirtualBox:~/PQC/PerfTest$ sudo docker run -it openquantumsafe/curl openssl speed -seconds 2 sphincsshake128fsimple
Doing sphincsshake128fsimple keygen ops for 2s: 629 sphincsshake128fsimple signature keygen ops in 1.96s
Doing sphincsshake128fsimple signs ops for 2s: 26 sphincsshake128fsimple signature sign ops in 2.00s
Doing sphincsshake128fsimple verify ops for 2s: 453 sphincsshake128fsimple signature verify ops in 1.99s
version: 3.3.0-dev
built on: Wed Dec 20 07:34:33 2023 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -O3 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDE
BUG
CPUINFO: OPENSSL_ia32cap=0x82982203478bffff:0x0
                              keygen     signs    verify keygens/s    sign/s  verify/s
     sphincsshake128fsimple 0.003116s 0.076923s 0.004393s     320.9      13.0     227.6
testa@testa-VirtualBox:~/PQC/PerfTest$
```

Рисунок 3.2.4 - Продуктивність SPHINCS+-128

SPHINCS+ є дуже повільним, і здатен, у даному випадку підписати, не більше 15 повідомлень за секунду. Що навіть у порівняні із FALCON-512 є дуже повільно, проте теоретично вже створеними блоками шифру можна підписувати інші повідомлення набагато швидше. Також, теоретично, цей алгоритм підпису є найнадійніший.

Також порівняти алгоритми шифрування повідомлення. Протестуємо RSA512:

```
testa@testa-VirtualBox:~/PQC/PerfTest$ sudo docker run -it openquantumsafe/curl openssl speed -seconds 2 rsa512
Doing 512 bits private rsa sign ops for 2s: 35578 512 bits private RSA sign ops in 1.96s
Doing 512 bits public rsa verify ops for 2s: 475192 512 bits public RSA verify ops in 1.97s
Doing 512 bits private rsa encrypt ops for 2s: 276609 512 bits public RSA encrypt ops in 1.88s
Doing 512 bits private rsa decrypt ops for 2s: 27438 512 bits private RSA decrypt ops in 1.99s
Doing rsa512 keygen ops for 2s: 401 rsa512 KEM keygen ops in 1.95s
Doing rsa512 encaps ops for 2s: 315774 rsa512 KEM encaps ops in 2.70s
Doing rsa512 decaps ops for 2s: 37475 rsa512 KEM decaps ops in 1.81s
Doing rsa512 keygen ops for 2s: 386 rsa512 signature keygen ops in 1.89s
Doing rsa512 signs ops for 2s: 37034 rsa512 signature sign ops in 2.00s
Doing rsa512 verify ops for 2s: 493121 rsa512 signature verify ops in 2.00s
version: 3.3.0-dev
built on: Wed Dec 20 07:34:33 2023 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -O3 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDE
BUG
CPUINFO: OPENSSL_ia32cap=0x82982203478bffff:0x0
                  sign    verify    encrypt   decrypt   sign/s verify/s  encr./s  decr./s
rsa   512 bits 0.000055s 0.000004s 0.000007s 0.000073s  18152.0 241214.2 147132.4  13787.9
                              keygen     encaps    decaps keygens/s  encaps/s  decaps/s
                      rsa512 0.004863s 0.000009s 0.000048s     205.6  116953.3   20704.4
                              keygen     signs    verify keygens/s    sign/s  verify/s
                      rsa512 0.004896s 0.000054s 0.000004s     204.2   18517.0  246560.5
testa@testa-VirtualBox:~/PQC/PerfTest$
```

Рисунок 3.2.5 - Продуктивність RSA512

RSA512 здатен за згенерувати тільки 200 пар ключів за сек. але зашифровує ≈$15\times10^4$ повідомлень за секунду та розшифровує ≈$19\times10^3$ повідомлень, що є швидко. Порівняєм із Kyber512:

```
testa@testa-VirtualBox:~/PQC/PerfTest$ sudo docker run -it openquantumsafe/curl openssl speed -seconds 2 kyber512
Doing kyber512 keygen ops for 2s: 31944 kyber512 KEM keygen ops in 1.48s
Doing kyber512 encaps ops for 2s: 22509 kyber512 KEM encaps ops in 0.92s
Doing kyber512 decaps ops for 2s: 25600 kyber512 KEM decaps ops in 1.97s
version: 3.3.0-dev
built on: Wed Dec 20 07:34:33 2023 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -O3 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDE
BUG
CPUINFO: OPENSSL_ia32cap=0x82982203478bffff:0x0
                              keygen     encaps     decaps keygens/s  encaps/s  decaps/s
                 kyber512 0.000046s 0.000041s 0.000077s   21583.8   24466.3   12994.9
testa@testa-VirtualBox:~/PQC/PerfTest$
```

Рисунок 3.2.6 - Продуктивність Kyber512

Kyber512 генерує ключі набгати швидше ≈$15\times10^3$ за сек. та шифрує і розшифровує із швидкістю $10\times10^3$ за сек., що в реальності є набагато практичніше ніж RSA512. При тому, RSA512 із таким розміром ключа рахується не стійким, як мінімум йому потрібен ключ розміру 2048 біт, що також вплине на його продуктивність:

```
testa@testa-VirtualBox:~/PQC/PerfTest$ sudo docker run -it openquantumsafe/curl openssl speed -seconds 2 rsa2048
Doing 2048 bits private rsa sign ops for 2s: 1597 2048 bits private RSA sign ops in 1.95s
Doing 2048 bits public rsa verify ops for 2s: 55144 2048 bits public RSA verify ops in 1.97s
Doing 2048 bits private rsa encrypt ops for 2s: 47479 2048 bits public RSA encrypt ops in 1.95s
Doing 2048 bits private rsa decrypt ops for 2s: 1637 2048 bits private RSA decrypt ops in 2.00s
Doing rsa2048 keygen ops for 2s: 22 rsa2048 KEM keygen ops in 2.23s
Doing rsa2048 encaps ops for 2s: 46141 rsa2048 KEM encaps ops in 1.96s
Doing rsa2048 decaps ops for 2s: 1689 rsa2048 KEM decaps ops in 2.01s
Doing rsa2048 keygen ops for 2s: 25 rsa2048 signature keygen ops in 2.09s
Doing rsa2048 signs ops for 2s: 1679 rsa2048 signature sign ops in 1.99s
Doing rsa2048 verify ops for 2s: 55324 rsa2048 signature verify ops in 2.00s
version: 3.3.0-dev
built on: Wed Dec 20 07:34:33 2023 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -O3 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDE
BUG
CPUINFO: OPENSSL_ia32cap=0x82982203478bffff:0x0
                  sign    verify    encrypt   decrypt   sign/s verify/s  encr./s  decr./s
rsa  2048 bits 0.001221s 0.000036s 0.000041s 0.001222s    819.0  27991.9  24348.2    818.5
                              keygen     encaps     decaps keygens/s  encaps/s  decaps/s
                    rsa2048 0.101364s 0.000042s 0.001190s       9.9   23541.3     840.3
                              keygen      signs     verify keygens/s    sign/s  verify/s
                    rsa2048 0.083600s 0.001185s 0.000036s      12.0     843.7   27662.0
testa@testa-VirtualBox:~/PQC/PerfTest$
```

Рисунок 3.2.7 - Продуктивність RSA2048

При збільшені ключа у алгоритмі RSA продуктивність зменшилась більше ніж у 10 раз. Тому навіть без загрози квантових комп’ютерів Kyber є кращим варіантом.

## 3.3 Висновок до третього розділу

У цьому розділі за допомогою oqs-provider була інтегрована бібліотека liboqs у openssl, що дало можливість встановлювати PQC зв'язки. Проте openssl в офіційному релізі, ще не добавив PQC алгоритми тому таке шифрування, через безпеку і незручність, є доцільним тільки у експериментальних цілях.

Була проаналізована продуктивність алгоритмів розглянутих у 2 розділі. Деякі із них показують себе як не надто продуктивні, як от SPHINCS+ але можливості даного алгоритму до кінця не розкриті, є можливість частково використовували вже згенерований код для підписання іншої інформації. Швидкодія FALCON та Kyber на рівні із найшвидшими ECC, проте мають також недолік, як от великий розмір підпису та ключів.

Навіть враховуючи, що liboqs є відносно новою, проти деякі PQC алгоритми, при виконанні через openssl, є набагато швидші ніж традиційні асиметричні шифри.

# 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

## 4.1 Охорона праці

У сучасному світі технічний прогрес та технологічні інновації виробництва неухильно ведуть до постійної автоматизації та удосконалення виробничих процедур. Наразі важко собі уявити будь-яку компанію, яка б не використовувала комп'ютерну техніку у своїй діяльності, особливо з огляду на розповсюдженість робіт, що виконуються за допомогою комп'ютерів. В Україні, враховуючи масштабність цього явища, законодавство строго регулює умови використання комп'ютерної техніки на робочих місцях, включаючи аспекти охорони праці під час роботи з комп'ютерами.

Розглянемо ключові нормативи, які визначають умови для створення належного робочого середовища в офісі, згідно з державними стандартами. Інструкція з охорони праці є важливим інструментом: для працівника вона стає засобом захисту прав, а для власника бізнесу - це спосіб дотримання законодавчих вимог, збереження фінансів та уникнення штрафів від інспекторів. Основа документації включає перелік визначених державних санітарних стандартів.

Згідно з частиною 1 статті 13 Закону про охорону праці, керівник державного підприємства, власник бізнесу або роботодавець має обов'язок забезпечити своїм працівникам відповідні умови праці та також особисто пройти відповідне навчання.[18]

Є такі вимоги до приміщення та місця роботи: у офісі повинні бути комфортні меблі, розміщені з урахуванням важливих принципів. При встановленні комп'ютерів необхідно враховувати площу, яка, згідно з ДСанПіН 3.3.2.007-98, призначена для кожного робочого місця. Необхідно забезпечити здорові умови роботи в офісних приміщеннях, що включає дотримання стандартів для меблів, підлоги та стелі. Меблі повинні відповідати гігієнічним вимогам і бути комфортними для працівників.

Основні принципи роботи з комп'ютером включають:

- Необхідно розробити робочі місця так, щоб працівники могли змінювати своє положення і рухатися, враховуючи достатні розміри.
- Розміщення та елементи робочого місця мають відповідати ергономічним стандартам, а також вимогам антропології та психофізіології, враховуючи специфіку роботи.
- Освітлення має забезпечувати належний контраст між екраном та оточенням, відповідно до характеру роботи і стандартів ДСанПіН 3.3.2.007-98.[19]
- Мікроклімат у виробничих приміщеннях має відповідати ДСН 3.3.6.042-99[20] і підтримуватися стабільним.
- Робоча поверхня та стіл повинні бути достатнього розміру із низьким рівнем відблисків, з можливістю гнучкого розміщення обладнання, включаючи екран, клавіатуру та документи.
- Робоче крісло має бути стійким, забезпечувати легкість переміщення та комфорт. Сидіння та спинка крісла повинні регулюватися за висотою та нахилом. Для додаткового комфорту може бути використана підніжка.
- Перед початком робочого дня слід очищати екрани від пилу та інших забруднень.
- Не слід працювати на пристроях з несправними екранами, які видають незвичні звуки, мають нестабільне зображення чи інші дефекти.

Робота з електроприладами в офісному середовищі вимагає суворого дотримання правил безпеки, щоб запобігти електричним травмам, короткому замиканню чи пожежам. Перш за все, слід переконатися, що всі електричні прилади сертифіковані та перевірені на відповідність встановленим стандартам безпеки. Приміщення офісу має бути обладнане відповідними засобами протипожежної безпеки, включаючи вогнегасники та детектори диму. Важливо також регулярно перевіряти стан електропроводки та уникати перевантаження електричних мереж, особливо у місцях з великою кількістю електроніки, таких як серверні кімнати чи робочі станції.

Для забезпечення безпеки працівників, кожен офіс повинен мати чіткі інструкції з користування електроприладами, а також проводити регулярні

навчання з охорони праці. Це включає правильне використання розеток, уникнення використання подовжувачів та розгалужувачів у великій кількості та знання основних заходів першої допомоги у випадку електричної травми. Також важливо встановити правила для періодичного відключення електроприладів для проведення технічного обслуговування та перевірки їх стану. Дотримання цих простих, але ефективних правил може значно знизити ризик нещасних випадків та забезпечити безпечне та здорове робоче місце.

Вимоги до рівня шуму та вібрацій: документально зафіксовані рівні звукового тиску та стандарти вібрації, особливо в приміщеннях з технічним обладнанням, включаючи комп'ютерне, повинні бути дотримані всіма членами колективу та керівництвом під час організації робочих просторів. Ці вимоги визначені у ДСанПіН 3.3.2.007-98. Ефективна організація робочого середовища передбачає встановлення звукопоглинаючих пристроїв та використання ізоляційних матеріалів, таких як поролон або гума, що сприяє створенню кращих умов для роботи, мінімізації шуму та забезпеченню зосередженості під час виконання робочих завдань.[20]

Вимоги до освітлення та видимості: усі приміщення повинні бути належно освітлені, щоб забезпечити здоров'я та комфорт працівників. Норми освітлення визначені в ДБН В.2.5-28-2006 «Природне і штучне освітлення», затвердженому Мінрегіоном 15.05.2006 р. під номером 168. В залежності від функціонального призначення приміщення, освітлення може бути загальним, місцевим або комбінованим, вибір якого залежить від технічних характеристик приміщення та його розмірів. Важливо, щоб світло ефективно освітлювало робочу зону і не осліплювало, зазвичай розміщуючись з лівого боку[21]. За потреби, у приміщеннях з високочастотним обладнанням рекомендується використовувати захисні окуляри зі світлофільтрами, а для роботи за комп'ютером - спеціальні окуляри або екранні фільтри для захисту очей від випромінювання монітора. Ці засоби захисту повинні мати сертифікати від акредитованих лабораторій та щорічно перевірятися, як це передбачено пунктом 4.19 ДСанПіН 3.3.2.007-98. При роботі з документами можна використовувати комбіноване освітлення. Люмінесцентні лампи, металогалогенні лампи на 250 Вт, та лампи розжарювання

часто використовуються у світильниках для місцевого освітлення. Також непоганим варіантом є використання світлодіодних лам, через свою ефективність, довговічнісь та великий вибір на ринку.

Вимоги до стану повітря: у ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування» для України встановлено конкретні норми щодо умов мікроклімату на робочому місці. Ці норми включають в себе вимоги до вологості, температури та швидкості повітряного потоку. Вологість повітря в офісних приміщеннях повинна утримуватися в межах 40-60%, щоб забезпечити комфорт та уникнути здоров'язних проблем, пов'язаних зі збереженням сухого або надто вологого повітря. Щодо температури, ідеальним стандартом вважається діапазон 22-24°C у теплий період та 20-22°C у холодний, забезпечуючи оптимальний комфорт працівників і ефективність роботи. Швидкість повітряного потоку має бути достатньою для забезпечення свіжості повітря, але не вищою за 0.1 м/с, щоб уникнути відчуття протягів та дискомфорту. Всі ці параметри повинні строго контролюватися для забезпечення здорового та продуктивного робочого середовища[22].

Вимоги підчас епідемії: особлива увага приділяється заходам щодо запобігання поширенню інфекцій у робочих просторах. Необхідно забезпечити регулярну дезінфекцію поверхонь та обладнання, особливо тих, які часто торкаються (дверніручки, перила, клавіатури, миші). Провітрювання приміщень має відбуватися значно частіше, а системи вентиляції повинні бути перевірені на ефективність. Розміщення антисептиків для рук у доступних місцях і встановлення бар'єрів з пластику або скла між робочими місцями допоможуть зменшити прямий контакт та ризик передачі вірусів. Крім того, слід заохочувати працівників до дотримання соціальної дистанції, а також, при можливості, впроваджувати гнучкі графіки роботи або роботу з дому.

Вимоги підчас війни:  вимоги до робочих місць адаптуються для забезпечення максимальної безпеки та продовження роботи в умовах конфлікту. Важливим є забезпечення надійних шляхів евакуації та доступу до укриттів, які повинні бути чітко позначені та обладнані необхідними запасами (вода, перша медична допомога, запасні батареї). Також, важливо використовувати системи

раннього оповіщення про небезпеку та плани евакуації. Роботодавці повинні надавати працівникам психологічну підтримку та гнучкість у графіку роботи, оскільки стрес та турбота про особисту безпеку можуть значно впливати на працездатність. Крім того, розробка планів для дистанційної роботи або роботи у зміненому режимі може бути необхідною для забезпечення безперервності бізнес-процесів.

Інструктажі з охорони праці є ключовим елементом системи безпеки на робочому місці. Вони мають на меті ознайомлення працівників з потенційними ризиками, правилами безпеки, обладнанням для особистого захисту, а також процедурами дій у випадку надзвичайних ситуацій. Існують різні типи інструктажів: вступний (проводиться перед початком роботи), первинний на робочому місці (перед початком виконання обов'язків), повторний (періодично протягом трудової діяльності), позаплановий (у випадку зміни технологій, обладнання або після травматизму на виробництві) та цільовий (для тимчасових або сезонних працівників).

Проведення інструктажів має бути систематичним та документально оформленим. Інструктаж виконується відповідальною особою, яка має відповідні знання та кваліфікацію. Важливо, що кожен інструктаж має бути адаптований до специфіки роботи, ризиків, пов'язаних з конкретним робочим місцем, та індивідуальних особливостей працівника. Всі учасники інструктажу повинні підтвердити свою участь підписом у відповідних журналах інструктажів. Особлива увага приділяється аналізу потенційних небезпек та навчанню правильним реакціям на аварійні ситуації. Також, необхідно регулярно переглядати та оновлювати інструкції з охорони праці, щоб вони відповідали актуальним стандартам безпеки та змінам у законодавстві.

Робота із квантовими комп’ютерами та лабораторними установками: вимагає строгого дотримання вимог безпеки, оскільки ці технології можуть включати роботу з високонапруговим обладнанням, лазерами, магнітними полями та кріогенними речовинами. Необхідно забезпечити наявність і використання засобів індивідуального захисту (ЗІЗ), включаючи захисні окуляри, рукавиці, спеціальний одяг та взуття. Робоче місце має бути добре

вентильованим, а доступ до обладнання обмежений для неавторизованих осіб. Важливо також забезпечити належне навчання та інструктажі для всіх працівників, задіяних у роботі з квантовими комп'ютерами та лабораторними установками.

При роботі з квантовими комп'ютерами важливо враховувати ризики, пов'язані з випромінюванням, високочастотними електромагнітними полями, та надто низькими температурами. Необхідно використовувати захисне обладнання для шкідливого випромінювання та забезпечити наявність аварійних вимикачів та систем швидкої евакуації. При роботі з кріогенними матеріалами потрібно використовувати спеціальні контейнери для безпечного зберігання та транспортування. Інструкції з безпеки мають включати детальні процедури поводження з цими матеріалами, а також заходи першої допомоги у випадку надзвичайних ситуацій.

## 4.2 Безпека в надзвичайних ситуаціях

Основні види надзвичайних ситуацій, які можуть виникнути в офісному середовищі, охоплюють широкий спектр подій, кожна з яких має свої специфічні ознаки. Пожежа, наприклад, може проявлятися через задимлення, високу температуру та появу вогню, тоді як землетрус характеризується раптовими тремтіннями землі та поштовхами, що можуть призвести до падіння предметів і пошкодження будівель. Терористичний акт може бути ідентифікований за наявності підозрілих пакетів, поведінки людей або неочікуваних вибухів. Хімічна аварія в офісі, у свою чергу, може статися в результаті витоку небезпечних речовин, що часто супроводжується різким запахом або подразненням дихальних шляхів. У кожному випадку необхідно мати чіткий план дій та бути обізнаним про заходи безпеки, щоб максимально зменшити ризики для здоров'я та безпеки працівників.

Під час надзвичайних ситуацій у офісі, таких як пожежі, землетруси чи інші аварії, необхідно дотримуватися спокою та організованості. При виявленні ознак небезпеки, негайно сповіщайте про це відповідні служби, використовуючи

телефон або систему екстреного оповіщення. При евакуації слідуйте чітко вказаним маршрутам, уникаючи паніки та дотримуючись інструкцій офісної безпеки. У разі потреби, надавайте першу медичну допомогу постраждалим, використовуючи аптечку першої допомоги, і залишайтеся з ними до прибуття кваліфікованої медичної допомоги. Важливо також вести облік всіх працівників під час евакуації, щоб забезпечити безпеку кожного.

Кожен працівник має бути ознайомлений з розташуванням усіх евакуаційних виходів, які повинні бути чітко позначені та вільні від перешкод. Основні маршрути евакуації ведуть до найближчих безпечних виходів, вказаних на схемах, розміщених у кожному коридорі. Збірні пункти знаходяться за межами будівлі, в безпечній та відкритій зоні, де всі працівники повинні зібратися після виходу з офісу. Кожен відділ має призначеного відповідального за евакуацію, який перевіряє наявність всіх членів свого відділу на збірному пункті та підтримує зв'язок зі службою безпеки за допомогою мобільного телефону або рації. Цей план евакуації має бути розміщений на видному місці в кожному офісі та регулярно перевіряти на актуальність.

Для забезпечення безпеки в офісі необхідно мати наступне обладнання та інвентар:

- Вогнегасники: розміщені у доступних місцях на кожному поверсі, особливо біля виходів та у зонах із високим ризиком виникнення пожежі. Наприклад, в кухонній зоні та біля серверної кімнати.
- Аптечки першої допомоги: розташовані в центральних місцях на кожному поверсі, легко доступні для всіх працівників. Важливо, щоб аптечка містила необхідні засоби для надання першої допомоги, включаючи бинти, антисептики, пластирі.
- Світильники аварійного освітлення: встановлені у коридорах та при виходах, щоб забезпечити візуальну орієнтацію під час відключення електроенергії.
- Рації для зв'язку: зберігаються у відповідальних за безпеку працівників або на охоронному посту. Вони дозволяють підтримувати зв'язок між персоналом під час евакуації та інших надзвичайних ситуацій.

Кожен елемент цього списку має бути регулярно перевіряти на наявність та справність. Наприклад, вогнегасники потребують періодичного технічного обслуговування, а аптечки - поповнення витратних матеріалів. Ознайомлення працівників з місцезнаходженням цих предметів та інструкціями щодо їх використання є ключовим аспектом підтримки безпеки в офісі.

Перелік телефонів та адрес служб, які можуть надати допомогу в надзвичайних ситуаціях, є надзвичайно важливим для забезпечення швидкого реагування та ефективної допомоги. Серед основних служб, до яких можна звернутися, слід включити:

- 101 - пожежна охорона та служба порятунку;
- 102 – поліція;
- 103 - швидка медична допомога;
- 104 - аварійна газова служба;
- 112 - єдиний номер для виклику всіх екстрених служб.

4 .2 .1 Основні поняття в БЖД

*Безпека життєдіяльності (БЖД)* — наука, що вивчає вплив на людину зовнішніх та внутрішніх факторів у всіх сферах її життєдіяльності.

*Об'єктом вивчення БЖД* є людина у всіх аспектах її діяльності (фізичному, психологічному, духовному, суспільному).

*Предметом вивчення БЖД* є вплив на життєдіяльність та здоров'я людини зовнішніх і внутрішніх факторів.

*ш* виявлення умов позитивного і негативного впливу на життєдіяльність та здоров'я людини зовнішніх і внутрішніх факторів, обґрунтування оптимальних умов і принципів життя.

*Дисципліна "Безпека життєдіяльності" має* світоглядно професійних характер. Вона використовує досягнення та методи фундаментальних і прикладних наук, зокрема: 1) гуманітарні науки (філософія, теологія, лінгвістика); 2) природознавчі науки (математика, фізика, хімія, біологія); 3) інженерні науки (технічне конструювання, опір матеріалів, інженерна справа, електроніка й електротехніка); 4) науки про людину (медицина, психологія,

ергономіка, педагогіка); 5) науки про суспільство (соціологія, економіка, право). За європейською концепцією науки з безпеки мають дві частини: спеціальну, яка вивчає події, випадки, інциденти, і спільну, яка складається з компонентів гуманітарних, природничих, інженерних наук, наук про людину, суспільство і яка формує світогляд людини. Адже проблема безпеки життєдіяльності — це, перш за все, проблема нового світогляду щодо управління безпекою на основі ризик орієнтованого підходу.

*Життя* — це вища, порівняно з фізичною та хімічною, форма існування матерії, яку відрізняє від інших здатність до самовідтворення (розмноження), росту, розвитку, активної регуляції свого складу та функцій, різних форм руху, можливість пристосування до навколишнього середовища, реакції на подразнення, наявність специфічного упорядкованого обміну речовин і енергії (керованих біохімічних реакцій), самовідновлення, систем управління, фізичної і функціональної дискретності живих істот і їх суспільних конгломератів. Вона виникає лише при певних умовах навколишнього середовища. Суттєвим моментом життя є постійний обмін окремого суб'єкта або певної системи речовиною, енергією та інформацією з оточуючим його середовищем, з наступним їх перетворенням чи розсіюванням в організмі суб'єкта або в системі при передачі від однієї ланки до іншої. Найголовніша відмінність між людиною і тваринним світом полягає у способі життя. Тваринне життя здійснюється природним чином, як існування, людське — суспільним, соціальним, як життєдіяльність. Все що є в суспільстві, як і саме суспільство, — результат людської діяльності.

Діяльність — це специфічна форма активного відношення людини до навколишнього середовища, зміст якої полягає у доцільній зміні та перетворенні його в інтересах людей, що включає в себе мету, засоби, результат і сам процес створення людиною умов для свого існування та розвитку. В цей процес включається природне і соціальне середовище відповідно до індивідуальних потреб людини.

Діяльність — це активна взаємодія людини з навколишнім середовищем, завдяки чому вона досягає свідомо поставленої мети, яка виникла внаслідок

прояву у неї певної потреби. Діяльність вводить людину в складну систему відносин і зв’язків з умовами навколишнього середовища.

Діяльність необхідно розглядати не як односторонній вплив людини на навколишнє середовище, а як складну взаємодію, що має зворотній зв’язок та вплив на життєдіяльність.

Під життєдіяльністю можна розуміти:

- властивість людини не просто діяти в життєвому середовищі, яке її оточує, а процес збалансованого існування та самореалізації індивіда, групи людей і людства загалом в єдності їхніх життєвих потреб і можливостей;
- складний біологічний процес, що відбувається в організмі людини та дозволяє зберігати їй здоров’я і працездатність;
- регульований стан навколишнього середовища при якому, згідно з чинним законодавством нормами та нормативами, забезпечується комфортна і безпечна взаємодія людини з його компонентами, запобігання погіршення екологічної обстановки, умов і охорони праці, виникнення небезпеки та дій в умовах надзвичайних ситуацій;
- складну систему, яка здатна забезпечити і підтримати в середовищі буття певні умови життя та всі види діяльності людей.
- До основних принципів забезпечення життєдіяльності відносяться:
- безперервне забезпечення фізіологічних процесів організму людини (для цього потрібні повітря, питна вода, продукти харчування, світло, тепло, одяг, взуття);
- принцип взаємозв’язку і взаємозалежності з навколишнім середовищем — навколишнє середовище забезпечує життєдіяльність параметрами споживання, енергоресурсами, корисними копалинами, продуктами харчування, елементами штучного середовища та іншими матеріальними благами. В свою чергу життєдіяльність впливає на середовище буття змінюючи параметри споживання (виснажує енергоресурси, корисні копалини, змінює клімат, рослинний та тваринний світ, забруднює навколишнє середовище);

- принцип раціональної організації праці за ціллю, часом, місцем і нормами. Грамотна організація праці включає управління, принципи організації, цілі і завдання, засоби праці, виробничу діяльність і результати праці;
- принцип матеріального заохочення при організації життєдіяльності, що безпосередньо пов'язаний з продуктивністю праці, яка визначається людським фактором (способом матеріального заохочення), працездатністю виробничого персоналу, ступенем підготовленості до праці (професійним, фізіологічним, психологічним);
- принцип захисту здоров'я, меж і умов життєдіяльності. Для реалізації цього принципу людство створило спеціальні інститути — медичного забезпечення, оборони, екологічного захисту, моралі та ін. Окремі інститути як структурні частини життєдіяльності можуть створюватись для захисту людей і народного господарства в особливих (надзвичайних) ситуаціях. До них можна віднести Державну службу України з надзвичайних ситуацій, комісії з питань техногенноекологічної безпеки та надзвичайних ситуацій тощо;
- принцип ліквідації негативних наслідків життєдіяльності. Для більшості людей відчуття небезпеки пов'язане з буденними проблемами і повсякчасними клопотами, а не ґрунтується на побоюванні глобальних катастроф чи міжнародних конфліктів. Захист житла, робочого місця, достатку, здоров'я, довкілля — основні проблеми безпечного самовідчуття людини. Звідси, власне, широкий діапазон потреби в безпеці: від усунення хуліганства і злочинності до захисту від непродуманих політичних дій та неефективних управлінських рішень. Ось чому відчуття безпеки має глибоко індивідуальний відтінок, який залежить, з одного боку, від рівня соціального і духовного розвитку особистості, з іншого — від культурної ситуації і суспільного устрою, які позитивно чи негативно впливають на світовідчуття громадянина. Більшість людей інтуїтивно розуміє значення безпеки. Це і запобігання хворобам, і порушення усталеного способу життя у сім'ї, трудовому колективі чи природному середовищі, і захист від

хуліганства та злочинності, так само як і захист держави. Тому простіше визначити відсутність безпеки, ніж її наявність.

Основні визначення поняття "безпека": 1) це збалансований, за експертною оцінкою, стан людини, соціуму, держави, природних та антропогенний систем; 2) стан захищеності особи та суспільства від ризику зазнати шкоди; 3) прийнятний рівень ризику; 4) такий стан будь-якого об’єкту, за якого йому не загрожує небезпека.

Визначення безпеки життєдіяльності як збалансованої взаємодії людини і середовища її соціально-культурного життя підкреслює методологічну універсальність та світоглядний зміст цієї категорії, яка стосується не стільки політичної, економічної чи військової сфер суспільної діяльності, скільки особистого сприймання і внутрішнього відчуття безпеки окремої людини. Відтак вона, формуючи загальне культурне уявлення певної соціальної групи про безпеку, характеризує якість людського життя, його гідність і самоефективність.

Безпека людини — це поняття, що відображає саму суть людського життя, її ментальні, соціальні і духовні надбання. Вона є невід’ємною складовою характеристикою стратегічного напряму людства, що визначений ООН як "сталий людський розвиток" (Sustainable Human Development) — такий розвиток, який веде не тільки до економічного, а й до соціального, культурного, духовного зростання, що сприяє гуманізації національного менталітету і збагаченню позитивного загальнолюдського досвіду. Основною ознакою, що відрізняє сталий розвиток від усіх інших форм соціального руху і видозміни, є відновлення природного і культурного довкілля, коли не тільки не знищується життєвий потенціал, а й підвищується соціальна відповідальність людей, гуманізуються взаємини, ставлення, реакції. Тому сталий розвиток — це розвиток для людей і природи, для збільшення робочих місць і досягання нових рубежів безпеки у побуті, виробництві і наодинці самого індивіда з собою.

Парадигма людського розвитку — це не підхід до людини тільки як до людського капіталу. Хоч парадигма й визнає стрижневу роль цього капіталу щодо зростання продуктивності праці, все ж за мету визначає створення такого економічного і соціального середовища, яке б забезпечувало примноження

можливостей кожного громадянина. Концепція людського розвитку передбачає розкриття духовних потенцій особи, які виходять за межі економічного добробуту чи матеріального достатку. Тоді метою буде високий життєвий тонус культурної особистості. Сталий розвиток це також моральне зобов'язання одного покоління перед прийдешніми. Звідси неприпустиме зростання економічних боргів, зменшення дотацій на освіту та охорону здоров'я, виснаження природних ресурсів. Загалом борги (економічні, соціальні, екологічні) — це кредит під заставу сталості, порушення її законів. Тому стратегія сталого гуманітарного розвитку зорієнтована на збільшення різноманітного капіталу — фізичного, людського, природного, а не накопичення боргів чи кредитів[23].

### 4.2.2 Соціально-політичні небезпеки

Соціально-політичні небезпеки досить часто виникають при соціальнополітичних конфліктах. Існує досить багато визначень конфліктів. Так, у політологічних словниках найпоширенішим є таке трактування конфлікту: зіткнення двох чи більше різноспрямованих сил з метою реалізації їхніх інтересів за умов протидії.

Джерелами конфлікту є:

- соціальна нерівність, яка існує в суспільстві;
- система поділу таких цінностей, як влада, соціальний престиж, матеріальні блага, освіта.

Конфлікт — це зіткнення протилежних інтересів, поглядів, гостра суперечка, ускладнення, боротьба ворогуючих сторін різного рівня та складу учасників.

Конфлікт передбачає усвідомлення протиріччя і суб'єктивну реакцію на нього.

Якщо конфлікт виникає в суспільстві, то це суспільний конфлікт.

Будь-який соціальний конфлікт, набуваючи значних масштабів, об'єктивно стає соціально-політичним. Політичні інститути, організації,

політичні (конфліктують політичні системи) соціальні (конфліктують соціальні системи) економічні (конфліктують економічні системи (наприклад; корпорації)) рухи, втягуючись у конфлікт, активно обстоюють певні соціально-економічні інтереси.

Конфлікти, що відбуваються в різних сферах, набувають політичної значущості, якщо вони зачіпають міжнародні, класові, міжетнічні, міжнаціональні, релігійні, демографічні та інші відносини. Існує дві форми перебігу конфліктів:

- відкрита — відверте протистояння, зіткнення, боротьба;
- закрита, або латентна, коли відвертого протистояння немає, але точиться невидима боротьба.

### 4.2.3 Війна

Війна — це збройна боротьба між державами (їх коаліціями) або соціальними, етнічними та іншими спільнотами; у переносному розумінні слова — крайня ступінь політичної боротьби, ворожих відносин між певними політичними силами.

Найбільша кількість жертв через політичні причини є наслідком війни. Так, за час другої світової війни в СРСР (1941 — 1945) загинуло близько 55 млн. осіб, було повністю знищено 1710 міст та 70 тисяч селищ.

Під час в'єтнамської війни в 1960-ті роки було вбито близько 7 млн. місцевих мешканців і 57 тисяч американців. Окрім загибелі людей і великих руйнувань, військові дії завдають величезних збитків навколишньому середовищу.

Учені підрахували, що за більш як чотири тисячоліття відомої нам історії лише близько трьохсот років були абсолютно мирними. Війни на планеті забрали вже понад 4 млрд. людських життів. Кількість загиблих різко зростала з розвитком засобів знищення людей та розширенням масштабів військових дій.

Найбільшу потенційну небезпеку для людства та природного середовища становить ядерна зброя. Про це свідчать результати атомного бомбардування в

серпні 1945 року міст Хіросіма та Нагасакі в Японії. Окрім смертельного опромінення, сталося радіоактивне зараження грунту, рослин, повітря, будівель. Кількість убитих становила 273 тисячі осіб, під смертельне радіоактивне опромінення потрапило 195 тисяч осіб.

Велику небезпеку становлять хімічна та бактеріологічна зброя. Перше досить ефективне застосування хімічної зброї у великих масштабах було здійснене німцями 22 квітня 1915року на північ від Іпру в Бельгії. Цей хімічний напад зазвичай прийнято вважати початком хімічної війни в сучасному її розумінні. Внаслідок першої газобалонної атаки на Західному фронті було отруєно 15 тисяч осіб, з них 5 тисяч загинуло. 31 травня 1915року німці здійснили першу газобалонну атаку на Східному фронті в районі Болинова біля Волі Шиловської. Російські війська втратили отруєними понад 9 тисяч осіб, з них померло 1200. У XX ст. військові дії проводились доволі активно. За приблизними даними, з часу закінчення Другої світової війни в локальних військових конфліктах загинуло 22-25 мільйонів осіб.

Наведемо приклади локальних військових конфліктів середини та кінця XX ст. Це війна у В'єтнамі, воєнні дії в Афганістані, вторгнення Іраку в Кувейт, війна в Руанді, військовий конфлікт в Югославії, війна в Чечні та низка інших «малих» війн. Кожна з них принесла людські втрати, біль та страждання тисячам і тисячам сімей, окрім того супроводжувалась глибоким руйнуванням біосферних структур.

Сучасний світ дуже малий і вразливий для війни. Врятувати і зберегти його неможливо, якщо не покінчити з думками та діями, які століттями будувалися на прийнятності та припустимості війн і збройних конфліктів.

## 4.3 Висновок до четвертого розділу

У 4 розділі розглянуто вимоги до робочого місця в офісі, організацію виробничого процесу та як проводиться інструктаж. Також було проаналізовано яку небеспку може нести робота із квантовим комп’ютером та квантовими системами.

Також розглянуто основні поняття БЖД, найбільш прості загрози які можуть бути при роботі в офісі. Розглянуто питання соціально-політичної небезпеки та війни.

# ВИСНОВОК

Обрана дана тема була через зростаючий розвиток технологій та математики: спочатку поява алгоритму Шора, запуск першого квантового комп’ютера та вдала факторизація, бурний розвиток штучного інтелекту, загроза появи штучного супер інтелекту. Все це більше і більше привертає увагу до проблем криптографії, які просто в один день можуть бути реалізовані. Факт того, шо в теперішньому світі обмін інформаціє є дуже важливим аспектом життя, від якого на пряму залежить благополуччя людини. Це все робить дану проблему загрозою і для людського добробуту.

Детально розглянуті постквантові алгоритми цифрового підпису та шифрування. Досліджено залежність стійкості алгоритму від розміру ключа та параметрів , проаналізовані можливі атаки на ці криптосистеми. Розглянуто перспективні квантові системи розподілу ключа, їх реалізація та принцип роботи та перспективи у майбутньому

Проведений аналіз продуктивності квантових комп’ютерів для атак на розглянуті криптографічні системи. Розглянуто основну математичну проблему криптографії та квантові схеми, які моделюють роботу квантових комп’ютерів.

На практиці було інтегровано PQC алгоритми у openssl та перевірено їхню працездатність. За допомогою даного інструменту також були проведені тести ефективності окремих алгоритмів та у парах для TLS рукостискання.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Eliezer Yudkowsky on the Dangers of AI 5/8/23 [Електронний ресурс] – Режим доступу до ресурсу: https://www.youtube.com/watch?v=fZlZQCTqIEo&ab_channel=EconTalk

2. Post-Quantum Cryptography PQC [Електронний ресурс] – Режим доступу до ресурсу: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

3. SPHINCS: practical stateless hash-based signatures [Електронний ресурс] – Режим доступу до ресурсу: https://sphincs.cr.yp.to/

4. An Overview of Hash Based Signatures [Електронний ресурс] – Режим доступу до ресурсу: https://eprint.iacr.org/2023/411.pdf

5. FPGA-based SPHINCS+ Implementations: Mind the Glitch [Електронний ресурс] – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/9217834

6. Falcon Takes Off - A Hardware Implementation of the Falcon Signature Scheme [Електронний ресурс] – Режим доступу до ресурсу: https://eprint.iacr.org/2023/1885.pdf

7. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU Specification v1.2 — 01/10/2020 [Електронний ресурс] – Режим доступу до ресурсу: https://falcon-sign.info/falcon.pdf

8. Kyber itzmeanjan/**kyber** [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/itzmeanjan/kyber

9. The Feasibility of the CRYSTALS-Kyber Scheme for Smart Metering Systems [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/366324775_The_Feasibility_of_the_CRYSTALS-Kyber_Scheme_for_Smart_Metering_Systems

10. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.01) [Електронний ресурс] – Режим доступу до ресурсу: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf

11. Basics of quantum information IBM[Електронний ресурс] – Режим доступу до ресурсу: https://learning.quantum.ibm.com/course/basics-of-quantum-information/quantum-circuits#quantum-circuits

12. Long-distance device-independent quantum key distribution [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/figure/a-Working-principle-of-an-heralded-qubit-amplifier-based-on-teleportation-34-36-A_fig1_337603425

13. Proof-of-principle demonstration of measurement-device-independent quantum key distribution using polarization qubits [Електронний ресурс] – Режим доступу до ресурсу: https://www.semanticscholar.org/reader/86eff824c6806b157bb9fa57d0d621bbbac0657f

14. Real-time gigahertz free-space quantum key distribution within an emulated satellite overpass [Електронний ресурс] – Режим доступу до ресурсу: https://www.science.org/doi/10.1126/sciadv.adj5873?utm_campaign=SciAdv&utm_source=Twitter&utm_medium=ownedSocial

15. Satellite Communication Basics [Електронний ресурс] – Режим доступу до ресурсу: https://www.youtube.com/watch?v=ig2QeQ_Xzwo&ab_channel=Dr.AVARUVENKATARAMANA

16. oqsprovider - Open Quantum Safe provider for OpenSSL (3.x) [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/open-quantum-safe/oqs-provider

17. OPEN QUANTUM SAFE [Електронний ресурс] – Режим доступу до ресурсу: https://openquantumsafe.org/applications/tls.html

18. Закон України від 14.10.1992 № 2694-XII [Електронний ресурс] – Режим доступу до ресурсу: https://ips.ligazakon.net/document/T269400

19. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний

ресурс] – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/v0007282-98

20. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс] – Режим доступу до ресурсу:https://zakon.rada.gov.ua/rada/show/va042282-99

21. ДБН В.2.5-28-2006 Природне і штучне освітлення [Електронний ресурс] – Режим доступу до ресурсу:https://dbn.co.ua/load/normativy/dbn/1-1-0-394

22. Про внесення змін до наказів Мінрегіону від 08.04.2013 № 134 [Електронний ресурс] – Режим доступу до ресурсу:https://zakon.rada.gov.ua/rada/show/v0410858-13

Додаток А

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

**М А Т Е Р І А Л И**

**XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

# «ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»


**13-14 грудня 2023 року**

**ТЕРНОПІЛЬ**
**2023**

**УДК 004.056**
**С. Пащак**
(Тернопільський національний технічний університет імені Івана Пулюя)

## ПЕРСПЕКТИВНІ МЕТОДИ ТА ЗАСОБИ ПОСТКВАНТОВОГО ТА КВАНТОВОГО ЗАХИСТУ ІНФОРМАЦІЇ

**S. Pashchak**

## PROMISING METHODS AND TOOLS FOR POST-QUANTUM AND QUANTUM INFORMATION SECURITY

Важко уявити сучасний світ без ІКС, які в свою чергу спираються на засоби криптографії. Проте, існує низка загроз які ставлять під сумнів стійкість поширених криптографічних систем, тому поява постквантових і квантових систем ставить собі мету розробку систем які не мають такого недоліку.

Постквантова криптографія (Post-quantum cryptography, PQC) є криптографічними способами шифрування та підписання інформації, які є стійкими до атак на базі квантових алгоритмів. Amazon, IBM та Microsoft пропонує своїм користувачам використовувати постквантові алгоритми для захисту даних[1], і не дивно, бо ще у 2021 році у Китаї розробили квантовий комп'ютер який виконує деякі задачі у $10^{24}$-рази швидше ніж звичайний комп'ютер[2].

У випадку із квантовою криптографією(Quantum cryptography, QC), безпека базується не на математичних властивостях, а на законах квантової механіки. Якщо постквантову систему можна реалізувати просто змінивши код, для реалізації квантового шифрування потрібні технології рівня провідних лабораторій світу та великого бюджету[3]. Тому сьогодні цей інструмент доступний хіба, що науковій та військовій галузі.

Квантова криптографія, наразі володіє тільки інструментом квантового розподілу ключа(QKD), різняться тільки методи.

В свою чергу зараз перевірені та сертифіковані алгоритм постквантового криптографії такі:

- шифрування з відкритим ключем та встановлення ключів;
- генерація цифрового підпису[4].

Взаємне застосування постаквантових та квантових алгоритмів захисту, теоретично, призводить до існування найбільш захищених ІКС. Наразі, невідомо таких гібридних систем але деякі сервіси вже частково перейшли на системи PQC, дехто це робить в цілях реклами, а дехто заради безпеки. У деяких конкретних випадках, це не несе додаткових витрат, як-от алгоритм Kyber при однаковому розмірі ключа є швидшим за буд-який ECC алгоритм[5], але мають більший розмір ключів та шифру Проте, NIST вже оголосив нових кандидатів, які можливо розв'яжуть дану проблему. Тому подальші дослідженні і впровадження, є дуже перспективним, як у академічному так і комерційному плані.

**Література**

1. Amazon, IBM Move Swiftly on Post-Quantum Cryptographic Algorithms Selected by NIST URL: https://www.darkreading.com/cyber-risk/amazon-ibm-move-swiftly-on-post-quantum-cryptographic-algorithms-selected-by-nist
2. Two of World's Biggest Quantum Computers Made in China URL: https://spectrum.ieee.org/quantum-computing-china
3. Measurement-Device-Independent Quantum Key Distribution URL: https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.108.130503
4. Post-Quantum Cryptography URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
5. Kyber and Post-Quantum Crypto - How does it work? URL: https://pretalx.c3voc.de/media/rc3-2021-cwtv/submissions/UGZY8B/resources/Kyber_and_Post-Quantum_Crypto_-_CCC_pR5OKou.pdf