МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ДУДАНЕЦЬ Роман Іванович

Алгоритм цифрового підпису на основі криптосистеми Нідеррейтера / Digital signature algorithm based on the Niederreiter cryptosystem

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
Р.І. Дуданець

Науковий керівник
к.т.н., доцент Н.Г. Яцків

Кваліфікаційну роботу
Допущено до захисту:

«____» _____ ___2024 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека та захист інформації
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
«____» _____ 2023 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ДУДАНЕЦЬ Роман Іванович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Алгоритм цифрового підпису на основі криптосистеми Нідеррейтера /
Digital signature algorithm based on the Niederreiter cryptosystem
керівник роботи к.т.н., доцент Н.Г. Яцків
затверджені наказом по університету від 12 грудня 2023 року № 753

2. Строк подання студентом закінченої кваліфікаційної роботи 12 грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– провести аналіз існуючих алгоритмів цифрового підпису та визначити їх переваги та недоліки в умовах класичних і квантових атак;

– вивчити математичний апарат теорії лінійних кодів та застосування криптосистем, побудованих на основі кодів;

– дослідити криптосистему Нідеррайтера та обґрунтувати її застосування для побудови алгоритму цифрового підпису;

– розробити алгоритм цифрового підпису на основі криптосистеми Нідеррейтера, здатний працювати з повідомленнями різної довжини;

– реалізувати алгоритм програмно, враховуючи оптимізацію продуктивності та стійкість до квантових атак;

– провести порівняння розробленого алгоритму з іншими постквантовими алгоритмами цифрового підпису за критеріями ефективності, швидкодії та стійкості.

5. Перелік графічного матеріалу у роботі:

– порівняння алгоритмів цифрового підпису на основі McEliece, Нідеррайтера та RSA;

– шифрування відкритим ключем на основі коду;

– порівняння алгоритму Нідеррайтера з іншими постквантовими алгоритмами;

– порівняння розмірів ключів, підписів і швидкості роботи алгоритму Нідеррайтера з іншими популярними постквантовими алгоритмами.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз алгоритмів цифрового підпису	12.2023 р. – 03.2024 р.	
2	Розробка та аналіз алгоритмів цифрового підпису на основі криптосистеми Нідеррайтера	03.2024 р. – 06.2024 р.	
3	Реалізація та дослідження алгоритму цифрового підпису Нідеррайтера	06.2024 р. – 11.2024 р.	

Студент _____ Дуданець Р.І.
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Алгоритм цифрового підпису на основі криптосистеми Нідеррейтера» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека та захист інформації» освітньо-професійної програми «Кібербезпека» написана обсягом 65 сторінок і містить 2 ілюстрації, 5 таблиць, 3 додатки та 32 джерел за переліком посилань.

Метою кваліфікаційної роботи є розробка, аналіз та дослідження алгоритму цифрового підпису на основі криптосистеми Нідеррейтера.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: теорію лінійних кодів, методи шифрування на основі кодів, порівняльний аналіз, теоретичний аналіз, алгоритмічний синтез, експериментальні методи.

Результати дослідження: Удосконалено алгоритм цифрового підпису на основі криптосистеми Нідеррайтера, який забезпечує роботи з різною довжиною повідомлень.

Розроблено алгоритм та програмну реалізацію брутфорс-атаки на алгоритм цифрового підпису Нідеррайтера.

У подальшому можливе розширення алгоритму для роботи з більш складними кодами та оптимізація його реалізації для зменшення обчислювальної складності. Крім того, важливим напрямом є дослідження гібридних рішень, що поєднують криптосистему Нідеррейтера з іншими постквантовими підходами для підвищення надійності захисту.

Ключові слова: ЦИФРОВІ ПІДПИСИ, ПОСТКВАНТОВІ АЛГОРИТМИ, ЦІЛІСНІСТЬ, АВТЕНТИЧНІСТЬ, КРИПТОСИСТЕМА НА ОСНОВІ КОДІВ.

ABSTRACT

This Master's thesis, titled "Digital signature algorithm based on the Niederreiter cryptosystem" explores methods to enhance the efficiency of digital signature algorithms within blockchain systems. The thesis, spanning 65 pages, incorporates 2 illustrations, 5 tables, 3 appendices, and references 32 sources.
The goal of the qualification work is the development, analysis, and investigation of a digital signature algorithm based on the Niederreiter cryptosystem.

Research Methods**.** To address the tasks set in this qualification work, the following were used: linear code theory, code-based encryption methods, comparative analysis, theoretical analysis, algorithmic synthesis, and experimental methods.

Research Results. An improved digital signature algorithm based on the Niederreiter cryptosystem was developed, enabling functionality with messages of varying lengths.

An algorithm and software implementation of a brute-force attack on the Niederreiter digital signature algorithm were developed.

In the future, the algorithm can be extended to work with more complex codes and optimized to reduce computational complexity. Additionally, an important direction is the study of hybrid solutions that combine the Niederreiter cryptosystem with other post-quantum approaches to enhance security reliability.

Keywords: DIGITAL SIGNATURES, POST-QUANTUM ALGORITHMS, INTEGRITY, AUTHENTICITY, CODE-BASED CRYPTOSYSTEM.

ЗМІСТ

ВСТУП

Актуальність роботи. Досягнення в галузі квантових обчислень ставлять під загрозу безпеку багатьох класичних криптографічних алгоритмів, що використовуються сьогодні. Традиційні алгоритми цифрового підпису, такі як RSA та ECDSA, виявляються вразливими до атак, які можуть бути реалізовані за допомогою квантових комп'ютерів. У зв’язку з цим дослідження та впровадження постквантових криптографічних рішень набуває особливої важливості. Алгоритми цифрового підпису на основі криптосистеми Нідеррейтера є перспективним напрямом завдяки їхній високій стійкості до квантових атак, заснованій на складності декодування кодів, що використовуються в теорії лінійних кодів [1, 2].

Криптографія на основі коду є однією з основних доступних постквантових методів разом із криптографією на основі решітки, багатоваріантною криптографією та криптографією на основі хешування.

Роберт МакЕліс створив першу криптосистему на основі коду в 1978 році. Вона належить до дуже вузького класу примітивів із відкритим ключем, які протистояли всім спробам криптоаналітики досі [3 – 5].

Актуальність дослідження обумовлена необхідністю розробки ефективних та стійких алгоритмів цифрового підпису, які можуть забезпечити захист даних у постквантовий період, а також їхньою потенційною інтеграцією у реальні криптографічні системи.

Мета і завдання дослідження. Метою роботи є розробка, аналіз та дослідження алгоритму цифрового підпису на основі криптосистеми Нідеррейтера, здатного забезпечити високий рівень безпеки у постквантовий період, а також оцінка його ефективності у порівнянні з іншими постквантовими алгоритмами.

Досягнення визначеної мети передбачає вирішення таких завдань:

– провести аналіз існуючих алгоритмів цифрового підпису та визначити їх переваги та недоліки в умовах класичних і квантових атак;

– вивчити математичний апарат теорії лінійних кодів та застосування криптосистем, побудованих на основі кодів;

– дослідити криптосистему Нідеррайтера та обґрунтувати її застосування для побудови алгоритму цифрового підпису;

– розробити алгоритм цифрового підпису на основі криптосистеми Нідеррейтера, здатний працювати з повідомленнями різної довжини;

– реалізувати алгоритм програмно, враховуючи оптимізацію продуктивності та стійкість до квантових атак;

– провести порівняння розробленого алгоритму з іншими постквантовими алгоритмами цифрового підпису за критеріями ефективності, швидкодії та стійкості;

– дослідити криптостійкість алгоритму до класичних і квантових атак та оцінити перспективи його використання у реальних умовах.

Об’єкт дослідження – процеси забезпечення цілісності, автентичності та незмінності.

Предмет дослідження – алгоритми та схеми цифрового підпису в на основі кодів.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: теорію лінійних кодів, методи шифрування на основі кодів, порівняльний аналіз, теоретичний аналіз, алгоритмічний синтез, експериментальні методи.

Наукова новизна одержаних результатів. Удосконалено алгоритм цифрового підпису на основі криптосистеми Нідеррайтера, який забезпечує роботи з різною довжиною повідомлень.

Практичне значення отриманих результатів. Розроблено алгоритм та програмну реалізацію брутфорс-атаки на алгоритм цифрового підпису Нідеррайтера.

Публікації та апробація КР.

Мачуляк М., Кондратюк В., Дуданець Р. Алгоритм цифрового підпису на основі криптосистеми Нідеррайтера. Матеріали науково-практична

конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2024), Тернопіль, 2024. – С. 98-99.

Кондратюк В., Сидорчук Р., Дуданець Р. Порівняння алгоритму Нідеррайтера з постквантовими. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2024. – С. 22-24.

1. АНАЛІЗ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ

1.1 Алгоритми цифрового підпису

До класичних алгоритмів цифрового підпису, які знайшли широке застосування належать: RSA, DSA, ECDSA, EdDSA [6, 7].

1. RSA (Rivest-Shamir-Adleman). Принцип роботи. RSA заснований на складності факторизації великих простих чисел. Підпис створюється за допомогою приватного ключа і перевіряється за допомогою відповідного публічного ключа.

До переваг необхідно віднести надійність і поширеність, підтримка в більшості криптографічних протоколів (наприклад, SSL/TLS).

Недоліки. Великі розміри ключів порівняно з іншими алгоритмами, що може вплинути на продуктивність.

2. DSA (Digital Signature Algorithm). Принцип роботи. DSA заснований на математичній проблемі дискретних логарифмів. Він є частиною стандарту цифрового підпису DSS (Digital Signature Standard).

До переваг можна віднести, те що добре підходить для цифрового підпису з низькими обчислювальними вимогами.

Недоліки. Порівняно з RSA, DSA може бути менш ефективним при використанні великих ключів.

3. ECDSA (Elliptic Curve Digital Signature Algorithm). Принцип роботи**:** ECDSA використовує криптографію на основі еліптичних кривих, що дозволяє створювати цифрові підписи зі значно меншими ключами при збереженні високого рівня безпеки.

До переваг необхідно віднести менші розміри ключів, що забезпечує швидшу генерацію і перевірку підписів, що робить його популярним для мобільних пристроїв і в середовищах з обмеженими ресурсами.

Недоліки: У складних системах реалізація може бути більш трудомісткою порівняно з RSA.

4. EdDSA (Edwards-curve Digital Signature Algorithm). Принцип роботи. EdDSA базується на еліптичних кривих Едвардса і розроблений для забезпечення високої швидкості та безпеки.

До переваг необхідно віднести високу швидкість генерації та перевірки підписів, стійкість до певних атак, зменшена ймовірність помилок у реалізації.

Недоліки. Відносно новий алгоритм, тому ще немає такої широкої підтримки, як у RSA або ECDSA.

1.1.1 Принцип роботи RSA

Алгоритм цифрового підпису RSA заснований на асиметричній криптографії та використовує пару ключів: відкритий і закритий. Принцип роботи алгоритму RSA для створення та перевірки цифрового підпису можна описати наступним чином [8-10].

1. Генерація ключів:

- вибираються два великих простих числа p і q;
- обчислюється їх добуток $n = p \cdot q$, який називається модулем;
- обчислюється функція Ейлера: $\phi(n) = (p-1) \cdot (q-1)$;
- вибирається число e, яке є взаємно простим із $\phi(n)$ (зазвичай e=65537);
- обчислюється d, де $d \cdot e \equiv 1 mod\ \phi(n)$. Це число є мультиплікативним оберненим до e за модулем $\phi(n)$.

В результаті отримаємо пари ключів:

- закритий ключ: (d, n);
- відкритий ключ: (e, n).

2. Створення цифрового підпису.

Підпис створюється з використанням закритого ключа відправника.

1. Хешування повідомлення. Оригінальне повідомлення M пропускається через криптографічну хеш-функцію (наприклад, SHA-256),

щоб отримати хеш-значення H(M). Хешування забезпечує компактне представлення даних.

2. Шифрування хеша. Хеш H(M) шифрується закритим ключем d за формулою: $S = H(M)d \ mod \ n$

S – це цифровий підпис, який додається до повідомлення.

3. Перевірка цифрового підпису. Перевірка підпису виконується з використанням відкритого ключа.

Розшифрування підпису. Отриманий підпис S розшифровується за допомогою відкритого ключа (e, n): $H'(M) = Se \ mod \ n$,

де H′(M) – це хеш, відновлений із підпису.

Хешування повідомлення. Повідомлення M знову хешується, щоб отримати H(M).

Порівняння хешів. Перевіряється, чи дорівнює H′(M) значенню H(M). Якщо вони співпадають, підпис вважається дійсним.

1.1.2 Принцип роботи DSA

Алгоритм цифрового підпису DSA складається з трьох основних етапів: генерація ключів, створення підпису та перевірка підпису.

1. Генерація ключів. Параметри алгоритму. Вибирається просте числоpp, яке є великим простим числом. Вибирається числоqq, яке є простим дільником числа $p-1$ (зазвичай q має довжину 160 біт).Вибирається число g, яке є генератором мультиплікативної групи модулю p, тобто g дорівнює $h(p-1)/q \bmod p$, де h – довільне число.

Приватний ключ. Власник генерує випадкове числоxx (приватний ключ), яке менше ніж q.

Публічний ключ. Обчислюється значення y, яке дорівнює $g \ x \ mod \ p$. Публічний ключ – це пара (p, q, g, y).

2. Створення підпису. Хешування повідомлення. Спочатку обчислюється хеш повідомленняmm за допомогою хеш-функції, наприклад, SHA-1. Отримане хеш-значення позначається як $H(m)$. Генерація

випадкового числа: Генерується випадкове число k, яке є секретним і обирається випадково для кожного підпису. Це число повинно бути меншеq.

Обчислення компонентів підпису. Обчислюється значення:

$$r = (g\ k\ (mod\ p\ ))mod\ q.$$

Обчислюється значення:

$$s = k - 1 \times (H(m) + x \times r)mod\ q,$$

де $k - 1$ це обернене до k за модулем q число.

Цифровий підпис. Цифровий підпис складається з двох значень (r, s), які додаються до повідомлення.

3. Перевірка підпису.

– Одержувач отримує підписане повідомлення і обчислює його хеш $H(m)$.

Одержувач використовує публічний ключ і виконує такі обчислення:

– Перевіряється, що r iss знаходяться в межах $0 < r < q$ і $0 < s < q$.

– Обчислюються значення $w = s - 1\ mod\ q,\ u1 = H(m) \times w\ mod\ q$ і $u2 = r \times w\ mod\ q$.

– Обчислюється значення $v = (gu1 \times yu2\ mod\ p)\ mod\ q$.

Якщо значення v дорівнює r, підпис вважається дійсним.

Переваги DSA. DSA забезпечує високий рівень безпеки завдяки використанню дискретних логарифмів, які є обчислювально складними задачами.

Алгоритм забезпечує достатню стійкість до різних типів криптографічних атак, таких як атаки на основі факторизації або атак на колізії хеш-функцій.

Ефективність. DSA використовує менші ключі порівняно з RSA, що робить його привабливим для середовищ з обмеженими ресурсами.

Обчислення у DSA, зокрема підписання, можуть бути швидшими, ніж у деяких інших алгоритмів.

Захист від відтворення. Завдяки використанню випадкового числа k для кожного підпису, навіть для однакового повідомлення підпис завжди буде різним.

Недоліки DSA. Вимога до безпеки випадкового числа k.

Якщо k не генерується випадковим чином або повторюється для різних повідомлень, це може призвести до розкриття приватного ключа x. Це є критичним недоліком, оскільки випадковість k безпосередньо впливає на безпеку підпису.

Відносна складність реалізації. Реалізація DSA може бути складнішою порівняно з іншими алгоритмами, особливо в частині генерації ключів і обчислення оберненого числа за модулем q.

Перевірка підпису DSA може бути повільнішою порівняно з RSA, що може бути важливим у деяких застосуваннях.

Специфічність для підпису. DSA не призначений для шифрування або обміну ключами, він використовується виключно для цифрових підписів, що обмежує його універсальність.

Застосування DSA. DSA широко використовується для цифрових підписів в різних стандартних протоколах, таких як:

– Digital Signature Standard (DSS). Це стандарт, прийнятий урядом США, який визначає використання DSA для цифрових підписів;

– SSH (Secure Shell): DSA використовується для автентифікації в SSH;

– PGP (Pretty Good Privacy): DSA може бути використаний у системах PGP для підписання електронної пошти та файлів.

DSA є потужним і надійним алгоритмом цифрових підписів, який забезпечує високий рівень безпеки, але його ефективність і складність залежить від правильної реалізації та використання.писах.

ECDSA – це варіант алгоритму DSA, що використовує криптографію на еліптичних кривих для створення цифрових підписів. Алгоритм ECDSA

дозволяє створювати цифрові підписи меншого розміру і зі швидшою обробкою порівняно з традиційними алгоритмами, такими як RSA, завдяки властивостям еліптичних кривих.

Основні поняття еліптичних кривих. Еліптичні криві: В математиці еліптичні криві описуються рівнянням виду:

$$y^2 = x^3 + ax + b,$$

де a і b – константи, які визначають конкретну криву. Над цими кривими визначається операція додавання точок, що є основою криптографічних операцій.

Групові операції. Основна властивість еліптичних кривих полягає в тому, що для них можна визначити операцію додавання точок кривої. З допомогою цієї операції можна реалізувати множення точок кривої на скаляр, що є основою для побудови криптографічних алгоритмів.

Принцип роботи ECDSA. ECDSA використовує математичні властивості еліптичних кривих для створення пари ключів і процесу підписання та перевірки цифрових підписів.1. Генерація ключів:Вибір еліптичної кривої: Спочатку вибирається конкретна еліптична крива, яка визначається набором параметрів (a,b) та полем значень, наприклад, над простим числомpp.

Приватний ключ. Це випадкове число d, яке є секретним і менше ніж n — порядок еліптичної кривої (тобто кількість точок на кривій). Публічний ключ: Обчислюється як точка на кривій $Q = dG$, де G – фіксована точка на кривій (так звана "генераторна точка"), а d – приватний ключ.

2. Процес створення цифрового підпису. Хешування повідомлення: Спочатку обчислюється хеш повідомленняmm за допомогою криптографічної хеш-функції (зазвичай SHA-2).Генерація випадкового числа: Генерується випадкове число k, яке є секретним і обирається для кожного підпису. Це число повинно бути менше ніж n.

Схема Нідеррейтера має головну перевагу в стійкості до квантових атак, але великий розмір ключів і повільність роблять його менш практичним для сучасних комерційних систем.

Попри обмежене використання криптосистеми Нідеррейтера в реальних проєктах, вона залишається важливим кандидатом для майбутніх рішень постквантової криптографії. Вона приваблює увагу дослідників через свою стійкість до квантових атак і можливість забезпечити довготривалу безпеку. У найближчі роки можна очікувати, що криптографічні схеми на основі кодів, такі як Нідеррейтера, отримають більше уваги, особливо в контексті підготовки до ери квантових комп'ютерів (таблиця 1.1).

Таблиця 1.1 – Порівняння алгоритмів цифрового підпису

Характеристика	Нідеррейтера	RSA	DSA	ECDSA	EdDSA
Безпека (класична)	Висока	Висока	Висока	Висока	Висока
Безпека (постквантова)	Стійкий до квантових атак	Уразливий	Уразливий	Уразливий	Уразливий
Розмір ключів	Дуже великий (до кількох сотень кБ)	Великий (1024-4096 біт)	Великий (1024-3072 біт)	Малий (256-512 біт)	Малий (256-512 біт)
Швидкість підпису	Повільний	Помірний	Помірний	Швидкий	Швидкий
Швидкість перевірки	Помірний	Швидкий	Швидкий	Швидкий	Швидкий
Розмір підпису	Малий	Великий	Середній	Малий	Малий

Продовження таблиці 1.1

Характеристика	Нідеррейтера	RSA	DSA	ECDSA	EdDSA
Алгоритмічна основа	Теорія кодів (синдромне декодування)	Факторизація великих чисел	Дискретний логарифм	Дискретний логарифм на еліпт. кривих	Дискретний логарифм на еліпт. кривих
Стійкість до квантових атак	Висока	Низька	Низька	Низька	Низька
Основні недоліки	Великий розмір ключа, повільність	Великі ключі і підписи	Великі ключі, уразливий до квантових атак	Складна реалізація еліптичних кривих	Складність реалізації

RSA. Широко використовується, але уразливий до квантових атак. Має великі ключі й підписи, але є простим для реалізації.

DSA. Подібний до RSA, але використовує дискретний логарифм. Підписи та ключі можуть бути меншими, але теж не захищений від квантових атак.

ECDSA. Ефективніший завдяки використанню еліптичних кривих, що дозволяє мати менші ключі і швидші операції при високій безпеці. Проте, як і RSA та DSA, уразливий до квантових атак.

EdDSA. Удосконалення ECDSA, швидший і має деякі додаткові переваги в безпеці та продуктивності, але також уразливий до квантових атак.

Алгоритм Нідеррейтера є перспективним для постквантової ери, проте через свої недоліки у вигляді великих ключів і повільності поки що він рідко

використовується в реальних системах. ECDSA та EdDSA на сьогодні є більш популярними завдяки своїй ефективності та компактності, але в майбутньому їм загрожують квантові атаки.

1.2 Криптосистема Нідеррейтера

Криптосистема Нідеррейтера є різновидом криптографічної схеми, заснованої на теорії лінійних кодів. Вона була запропонована американським криптографом Йоганном Нідеррейтером (Johann Niderreiter) у 1986 році як варіант системи McEliece, але з деякими модифікаціями. Схема Нідеррейтера базується на використанні матричної теорії кодування та задачі синдромного декодування. Вона стала цікавою через свою стійкість до квантових атак, що є важливим аспектом у контексті розвитку квантових обчислень [11 – 13].

Криптосистема Нідеррейтера була розроблена як частина ширшого пошуку нових криптографічних алгоритмів, заснованих на важких задачах, відмінних від традиційних задач дискретного логарифму чи факторизації великих чисел. Ідея полягала в тому, щоб скористатися труднощами синдромного декодування, яке є NP - складною задачею, тобто складною для вирішення навіть за допомогою найшвидших класичних комп'ютерів.

Нідеррейтера надихнула інша криптографічна система – схема McEliece, яка використовувала схожу ідею, але базувалася на задачі декодування випадкових лінійних кодів. Основна різниця між системами полягає в тому, що McEliece використовує відкритий ключ у вигляді матриці генератора, тоді як Нідеррейтера – у вигляді перевіркової матриці.

Криптосистема Нідеррейтера базується на задачі синдромного декодування для лінійних кодів. У класичній схемі закритий ключ – це матриця H', яка є перевірковою матрицею для лінійного коду, разом з двома додатковими матрицями перетворень.

Відкритий ключ – це перетворена перевіркова матриця H, яку отримують шляхом множення H' на дві матриці перетворень.

Шифрування відбувається шляхом обчислення синдрому повідомлення (деформації помилки) за допомогою відкритого ключа H. Декодування здійснюється за допомогою закритого ключа, який дозволяє швидко вирішити задачу синдрому декодування. Застосування в цифровому підписі. Для створення схеми цифрового підпису криптосистема Нідеррейтера була адаптована, і процес включає [14, 15] :

1) використання закритого ключа для створення підпису на основі вирішення задачі синдромного декодування;

2) перевірка підпису здійснюється з використанням відкритого ключа.

Переваги криптосистеми Нідеррейтера.

1. Стійкість до квантових атак. Одна з найбільших переваг системи Нідеррейтера – це її стійкість до атак, заснованих на квантових комп'ютерах. Оскільки криптосистема базується на задачі синдромного декодування, вона не вразлива до атак Шора, які є загрозою для криптосистем на основі дискретного логарифму і факторизації.

2. Швидкість шифрування. У порівнянні з криптосистемою RSA, система Нідеррейтера може забезпечувати швидше шифрування, оскільки операції з матрицями зазвичай є ефективнішими.

3. Невеликий розмір шифротексту. Криптосистема Нідеррейтера генерує шифротекст, розмір якого є меншим, ніж у системі McEliece, що робить її більш зручною для передачі даних.

4. Стійкість до відомих криптографічних атак. Система Нідеррейтера добре зарекомендувала себе проти багатьох класичних криптографічних атак, таких як атаки на основі вибору відкритих текстів чи шифротекстів.

Недоліки криптосистеми Нідеррейтера.

1. Великий розмір ключа. Основний недолік криптосистеми Нідеррейтера (як і схеми McEliece) полягає в тому, що для забезпечення високого рівня безпеки необхідно використовувати великі матриці, що

призводить до великих розмірів відкритого ключа (до кількох сотень кілобайт або більше). Це може ускладнювати її застосування на пристроях з обмеженими ресурсами.

2. Складність управління ключами. Через великий розмір ключів виникає додаткова складність в їх зберіганні та передаванні, що робить цю систему менш практичною для деяких додатків у порівнянні з RSA або ECC (еліптичними кривими).

3. Неоптимальна продуктивність для підпису. Хоча шифрування може бути швидким, процес декодування та підпису (через необхідність вирішувати задачу синдромного декодування) є досить ресурсоємним, що може призвести до затримок у порівнянні з іншими криптосистемами.

Криптосистема Нідеррейтера – це цікава криптографічна схема, яка забезпечує стійкість до квантових атак і має низку переваг у сфері шифрування. Однак її недоліки, пов’язані з великими ключами та складністю обчислювальних процесів, обмежують її використання в багатьох практичних додатках. Тим не менше, вона є перспективною для застосувань, де безпека на основі кодів є важливою, особливо в контексті квантової криптографії.

На даний момент криптосистема Нідеррейтера та її варіанти, зокрема у вигляді алгоритмів цифрового підпису, мають обмежене застосування в реальних умовах, порівняно з такими популярними схемами, як RSA, DSA або алгоритми на основі еліптичних кривих (ECC). Однак криптографічні схеми, засновані на лінійних кодах, включаючи криптосистему Нідеррейтера, набувають все більшої уваги завдяки їх стійкості до квантових атак, що робить їх перспективними для постквантової криптографії.

Стійкість алгоритму Нідеррейтера до квантових атак досягається за рахунок використання математичної проблеми, яка залишається складною навіть для квантових комп'ютерів. Основна причина цього – основа алгоритму NP-складна задача синдромного декодування, що робить його захищеним від відомих квантових алгоритмів, таких як алгоритм Шора.

У порівнянні з алгоритмами, такими як RSA чи ECDSA, криптосистема Нідеррайтера забезпечує менший розмір підпису, але більший розмір публічного ключа. Це робить її зручною для сценаріїв, де важлива швидкість перевірки підпису, наприклад, у системах Інтернету речей (IoT) (таблиця 1.2).

Таблиця 1.2 – Порівняння алгоритмів цифрового підпису на основі McEliece, Нідеррайтера та RSA

Критерій	McEliece	Нідеррайтер	RSA
Основна ідея	Використання кодів корекції помилок (Goppa-коди).	Використання кодів корекції помилок (Goppa, Ріда-Соломона).	Використання факторизації великих чисел.
Стійкість до квантових атак	Висока. Постквантовий алгоритм.	Висока. Постквантовий алгоритм.	Низька. Уразливий до атак квантових комп'ютерів.
Розмір ключів	Дуже великий (кілька сотень кБ).	Дуже великий (кілька сотень кБ).	Невеликий (кілька кБ).
Розмір підпису	Великий (порядку сотень байтів).	Великий (порядку сотень байтів).	Невеликий (кілька сотень бітів).
Придатність для систем із обмеженими ресурсами	Обмежена через великий розмір ключів і підписів.	Обмежена через великий розмір ключів і підписів.	Придатний через невеликі розміри ключів і швидкість роботи.
Складність математичних операцій	Базується на складності декодування лінійних кодів.	Базується на складності декодування лінійних кодів.	Базується на складності факторизації цілих чисел.

Хоча алгоритм має переваги, він стикається з викликами, пов’язаними з великим розміром публічних ключів. Для оптимізації продуктивності пропонуються такі підходи: 1) використання скорочених лінійних кодів для зменшення розміру ключів; 2) апаратна реалізація операцій для підвищення швидкості декодування.

Алгоритм цифрового підпису на основі криптосистеми Нідеррайтера демонструє значний потенціал для використання у постквантових системах завдяки своїй математичній складності і стійкості до атак. Подальші дослідження спрямовані на зменшення розміру публічних ключів і оптимізацію обчислень, що відкриває нові перспективи її застосування у високонавантажених системах

1.2.1 Стійкість до квантових атак.

1. Задача синдромного декодування. Алгоритм Нідеррейтера базується на проблемі декодування випадкових лінійних кодів, а саме на задачі знаходження помилки за синдромом. Дана задача є NP-складною, тобто розв'язання її в загальному випадку вимагає експоненційних обчислень навіть для квантових комп'ютерів. Це означає, що квантові комп'ютери не можуть легко зламати цю криптосистему за допомогою ефективних квантових алгоритмів.

2. Відсутність алгоритму Шора. Алгоритм Шора, який здатний ефективно зламувати криптосистеми, засновані на задачах факторизації (як у RSA) і дискретного логарифму (як у DSA, ECDSA, EdDSA), не застосовується до задачі синдромного декодування. Це пов'язано з тим, що алгоритм Шора вирішує лише задачі, які можна звести до періодичної структури, тоді як задача синдромного декодування не має такої структури.

3. Складність квантових алгоритмів для решіток і кодів. Криптосистеми, засновані на теорії кодів (включно з Нідеррейтером), використовують лінійні перетворення і задачі, такі як знаходження коротких векторів у решітках або декодування кодів з помилками. Відомі квантові

алгоритми, як наприклад алгоритм Гровера, можуть лише частково зменшити час пошуку (лінійне скорочення), але це не робить криптосистему Нідеррейтера вразливою, оскільки для її зламу все одно потрібні експоненційні ресурси.

Квантові комп'ютери неефективні для зламу. Квантові комп'ютери, використовуючи алгоритм Шора, можуть розв'язувати задачі факторизації і дискретного логарифму за поліноміальний час, що робить сучасні криптосистеми на основі RSA, DSA і ECDSA вразливими. Проте задача декодування за синдромом є зовсім іншою математичною проблемою, яка не підпадає під категорії задач, що їх може ефективно вирішувати квантовий комп'ютер.

Алгоритм Гровера, який використовується для загального пошуку в невпорядкованих базах, дає лише квадратичне прискорення, що недостатньо для зламу NP-складних задач.

Стійкість до квантових атак алгоритму Нідеррейтера досягається завдяки використанню математично складної задачі синдромного декодування, яку не можуть ефективно вирішити квантові алгоритми. Це робить алгоритм Нідеррейтера однією з перспективних схем для постквантової криптографії.

Алгоритм Нідеррейтера, завдяки своїй стійкості до квантових атак, найкраще підходить для застосування в сферах, де безпека і конфіденційність мають критичне значення на довгострокову перспективу, і де захист від квантових загроз стає важливим.

1.3 Область застосування цифрового підпису на основі алгоритм Нідеррейтера

1. Військова та урядова криптографія.

– Захист секретної інформації. Військові та урядові організації мають потребу в криптографії, яка може забезпечити захист на десятиліття. Алгоритм Нідеррейтера підходить для шифрування і підпису даних, які повинні залишатися конфіденційними навіть після появи квантових комп'ютерів.

2. Довготривала конфіденційність. Важлива інформація, така як стратегічні плани або дипломатичні переговори, може бути захищена алгоритмом Нідеррейтера, оскільки вона залишається безпечною навіть після появи нових обчислювальних технологій.

2. Фінансовий сектор.

– Безпечні фінансові транзакції. Криптосистема Нідеррейтера може бути використана для захисту фінансових транзакцій, особливо в умовах, де важливий захист від квантових загроз. Це може включати захист транзакцій у банках, біржах та фінансових установах.

– Цифровий підпис для смарт-контрактів. У блокчейн-системах або смарт-контрактах, які потребують довготривалої безпеки, цифровий підпис на основі Нідеррейтера може стати альтернативою нинішнім криптосистемам.

3. Інфраструктура критичної інформації (ICS/SCADA).

– Захист інфраструктурних мереж. Критичні інфраструктури, такі як електромережі, водопостачання, транспортні системи і телекомунікації, потребують надійного захисту. Алгоритм Нідеррейтера може бути використаний для захисту комунікацій та управління, забезпечуючи стійкість до атак майбутнього.

– Захист промислового Інтернету речей (IIoT). IIoT-пристрої, що керують виробничими процесами і працюють у мережах SCADA, можуть

використовувати Нідеррейтера для забезпечення безпеки даних на рівні пристроїв.

4. Довгострокове зберігання конфіденційних даних.

– Архіви з високим рівнем конфіденційності: Дані, які мають залишатися конфіденційними протягом десятиліть (медичні записи, юридичні документи, історичні дані), можуть бути зашифровані за допомогою алгоритму Нідеррейтера для захисту від майбутніх загроз з боку квантових комп'ютерів.

– Захист медичних даних. В охороні здоров'я, де конфіденційність і захист даних пацієнтів є пріоритетом, алгоритм Нідеррейтера може використовуватися для захисту медичних записів та іншої важливої інформації, яка має зберігатися довгий час.

5. Блокчейн і криптовалюти.

– Постквантові блокчейн-технології. Алгоритм Нідеррейтера може бути застосований для захисту цифрових підписів у блокчейн-системах, які є основою криптовалют і смарт-контрактів. Це забезпечить захист від атак, що можуть з'явитися з розвитком квантових обчислень.

– Довготривалий захист активів: Власники цифрових активів у блокчейнах (криптовалюти, токени) можуть скористатися алгоритмом для захисту своїх активів від майбутніх квантових атак.

6. Комунікаційні системи.

– Безпечний обмін повідомленнями: Алгоритм Нідеррейтера може бути застосований у системах захищених комунікацій (наприклад, дипломатичних або військових) для забезпечення стійкості до квантових атак.

– Захист протоколів зв'язку. Протоколи, які потребують високого рівня безпеки, як-от TLS/SSL або VPN-з'єднання, можуть використовувати алгоритм Нідеррейтера для забезпечення надійного захисту у майбутньому.

7. Електронний уряд і електронне голосування.

– Електронне голосування. Для забезпечення конфіденційності та достовірності результатів виборів у системах електронного голосування

можна використовувати алгоритм Нідеррейтера для шифрування даних виборців і цифрового підпису бюлетенів.

– Електронні документи і послуги. Урядові системи, які працюють з електронними документами, можуть використовувати алгоритм Нідеррейтера для захисту важливої інформації, яка має залишатися конфіденційною на довгий термін.

8. Розробка постквантових криптографічних стандартів.

– Наукові дослідження і стандартизація. Алгоритм Нідеррейтера є кандидатом для постквантової криптографії. Його можна використовувати для експериментів і розробки нових стандартів у рамках ініціатив, подібних до конкурсу постквантової криптографії від NIST.

Алгоритм Нідеррейтера має найбільший потенціал для застосування в сферах, де довготривалий захист даних від квантових комп'ютерів є критичним. Найкращі області його використання — це урядова, військова та фінансова сфери, а також системи, що потребують безпеки для зберігання даних і комунікацій на тривалі періоди часу.

Алгоритм Нідеррейтера є частиною класу криптосистем на основі кодів, які є перспективними для постквантової криптографії. Однак існують інші криптографічні алгоритми, також стійкі до квантових атак, що можна вважати його альтернативами або конкурентами в постквантовій криптографії.

2. РОЗРОБКА ТА АНАЛІЗ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ НА ОСНОВІ КРИПТОСИСТЕМИ НІДЕРРАЙТЕРА

2.1 Математичний апарат теорії лінійних кодів для криптографічних застосувань

2.1.1 Коди Хеммінга

Коди Хеммінга – це клас лінійних блокових кодів, які використовуються для виявлення та виправлення помилок у цифрових даних. Ці коди є одним із найбільш простих і широко застосовуваних способів забезпечення надійності передачі та зберігання інформації [14].

Коди Хеммінга базуються на лінійній алгебрі та працюють у двійковому полі GF(2). Основний принцип цих кодів – додавання надлишкових (контрольних) бітів до інформаційного повідомлення для забезпечення можливості перевірки й виправлення помилок.

1. Структура кодового слова.

Кодове слово в кодах Хеммінга складається з:

– інформаційних бітів (k): це корисна частина повідомлення.

– контрольних бітів (r): це біти, які додаються для забезпечення можливості виявлення й виправлення помилок.

Загальна довжина кодового слова:

$$n = k + r.$$

Кількість контрольних бітів r визначається таким чином, щоб задовольнити умову:

$$2^r \geq n + 1.$$

2. Мінімальна відстань коду.

Мінімальна відстань коду Хеммінга дорівнює $d_{min} = 3$, що дозволяє:

– виявляти до 2-х помилок;

– виправляти 1 помилку.

2. Математичні основи коду Хеммінга.

2.1. Перевіркова матриця H.

Для кодування й декодування кодів Хеммінга використовується перевіркова матриця H. Ця матриця має розмір $r \times n$ і визначається як (для прикладу коду Хеммінга (7,4)):

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

2.2. Синдром. Синдром – це вектор, який обчислюється для перевірки помилок:

$$Syndrome = H \cdot r^T \ (mod\ 2),$$

де r – отримане кодове слово.

3. Кодування.

1. До початкового повідомлення додаються контрольні біти.

2. Контрольні біти обчислюються так, щоб виконувалась умова:

$$H \cdot c^T = 0,$$

де c – кодове слово.

4. Декодування.

1. Отримане кодове слово множиться на *H*, щоб обчислити синдром.

2. Синдром вказує на позицію біта з помилкою (або відсутність помилок).

Якщо синдром вказує на помилку, відповідний біт виправляється.

5. Приклад коду Хеммінга (7,4).

Параметри

– загальна довжина кодового слова, $n = 7$;

– кількість інформаційних бітів, $k = 4$;

– кількість контрольних бітів, $r = 3$.

Наприклад, для повідомлення $m = [1, 0, 1, 1]$.

Контрольні біти обчислюються за допомогою перевіркової матриці H.

Кодове слово c може виглядати так: [1, 0, 1, 1, 0, 1, 0].

Коди Хеммінга – це базовий інструмент для забезпечення надійності передачі даних. Їх простота та ефективність роблять їх незамінними в багатьох технологічних системах, а також в основі більш складних алгоритмів кодування.

2.1.2 Коди Ріда-Соломона

Коди Ріда-Соломона є важливим класом лінійних коригувальних кодів, які широко використовуються в телекомунікаціях, зберіганні даних та криптографії. Їхньою основною характеристикою є здатність коригувати помилки, що виникають під час передачі чи зберігання інформації [15].

Код Ріда-Соломона є блоковим кодом, що належить до класу циклічних кодів. Він працює над скінченним полем $GF(q)$, де $q = 2^m$.

Код визначається параметрами (n, k, d),

де:

n – довжина кодового слова (кількість символів);

k – кількість інформаційних символів;

$d = n - k + 1$ – мінімальна відстань коду, яка визначає кількість помилок, які можна виправити.

Основна ідея полягає у представленні інформації у вигляді поліномів та оцінюванні цих поліномів у різних точках над скінченним полем.

Конструкція коду Ріда-Соломона.

1. Скінченне поле $GF(q)$. Всі обчислення виконуються в полі $GF(q)$, яке забезпечує обмежену кількість символів (зазвичай $q = 2^m$).

2. Поліноми. Інформаційні символи представлені у вигляді полінома $P(x)$ ступеня $k-1$:

$$P(x) = a_0 + a_1\, x + a_2\, x^2 + \cdots + a_{k-1}\, x^{k-1},$$

де a_i – коефіцієнти з $GF(q)$.

3. Оцінювання. Поліном $P(x)$ оцінюється у n різних точках $\alpha_1, \alpha_2, \ldots, \alpha_n$ з $GF(q)$, що дає закодоване слово:

$$C = \{P(\alpha_1), P(\alpha_2), \ldots, P(\alpha_n)\}.$$

4. Кодові слова. Набір оцінок полінома $P(x)$ у вибраних точках утворює кодове слово, яке передається або зберігається.

Основні властивості коду Ріда-Соломона.

– Висока коригувальна здатність. Код може виправляти до $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ помилок, де d – мінімальна відстань.

– Оптимальність. Коди Ріда-Соломона досягають межі Сінглтона:

$$d = n - k + 1.$$

– Стійкість до стирань. Код може виправити $n-k$ стирань, тобто втрату відомих позицій символів.

– Гнучкість. Коди Ріда-Соломона можуть бути адаптовані для різних застосувань завдяки вибору параметрів n, k, q.

Алгоритм декодування. Процес декодування полягає у відновленні початкового полінома $P(x)$ за отриманими кодовими словами, які можуть містити помилки. Основні етапи декодування:

1. Визначення помилок:

– виявлення позицій помилок за допомогою синдрому коду;

– обчислення корекційного полінома;

– використовується алгоритм Берлекемпа-Мессі для знаходження полінома локаторів помилок.

2. Виправлення помилок. Виправлення помилкових символів за допомогою локаторів помилок.

Використання в криптографії [16].

1. Криптосистема МакЕліса. Коди Ріда-Соломона використовуються в криптосистемах на основі кодування як базові коди завдяки їхній здатності ефективно коригувати помилки.

2. Постквантова криптографія. Коди Ріда-Соломона знаходять застосування в побудові стійких до квантових атак криптосистем, таких як варіанти алгоритмів Нідеррайтера або МакЕліса.

3. Електронний підпис. Завдяки ефективності у виправленні помилок коди Ріда-Соломона можуть бути адаптовані для використання в схемах цифрового підпису.

Коди Ріда-Соломона є потужним інструментом для захисту даних у різних областях, таких як телекомунікації, сховища даних та криптографія. Завдяки високій коригувальній здатності та стійкості до помилок, ці коди залишаються важливим компонентом сучасної теорії кодування та криптографічних алгоритмів.

2.1.3 Лінійні коди Гоппа

Лінійні коди Гоппа є основою для багатьох криптосистем, включаючи криптосистему МакЕліса та Нідеррайтера. Вони становлять важливий клас кодів у теорії кодування завдяки їхнім математичним властивостям, таким як висока корекційна здатність і стійкість до різноманітних атак.

Код Гоппа є підкласом циклічних кодів, побудованих над скінченним полем $GF(2^m)$. Основою для побудови такого коду є перетворювальна функція Гоппа, що гарантує лінійність і хорошу мінімальну відстань коду.

Лінійні коди Гоппа визначаються через поліном Гоппа, що має специфічні властивості, такі як відсутність спільних коренів із похідною.

Для побудови коду Гоппа необхідні такі елементи [17, 18]:

– скінченне поле $GF(2^m)$;

– усі операції виконуються в розширенні поля GF(2), де m є ступенем розширення;

– поліном Гоппа g(x) – це поліном над $GF(2^m)$ зі специфічними властивостями. Поліном g(x) має степінь t і не має спільних коренів із похідною $g'(x)$;

– перевіркова матриця H розмірності $(n-k) \times n$, що відповідає визначенню коду. Рядки цієї матриці утворюються за допомогою значень $g(x)$ для елементів поля $GF(2^m)$.

Усі лінійні комбінації рядків матриці H утворюють підпростір кодових слів.

Алгоритм побудови коду Гоппа складається з наступних кроків:

1) визначаємо ступінь розширення m і обчисліть кількість елементів поля $GF(2^m)$, що дорівнює $n = 2^m - 1$;

2) обираємо поліном Гоппа $g(x)$ зі ступенем t, який задовольняє властивість $gcd(g(x), g'(x)) = 1$;

3) будуємо перевіркову матрицю H розміром $(n-k) \times n$, використовуючи значення $g(x)$;

4) Використовуйте матрицю H для генерування та перевірки кодових слів.

Властивості кодів Гоппа.

1. Висока мінімальна відстань. Мінімальна відстань коду Гоппа d задовольняє нерівність $d \geq 2t + 1$. Це означає, що код може виправляти до t помилок.

2. Ефективне декодування. Існують алгоритми, які дозволяють ефективно декодувати коди Гоппа, виправляючи до t помилок.

3. Стійкість до атак. Завдяки складній структурі полінома Гоппа, коди мають високий рівень стійкості до структурних атак.

4. Велика кількість кодів. Поліноми Гоппа можуть бути обрані різними способами, що дає змогу створювати різноманітні коди.

Використання кодів Гоппа в криптографії. Коди Гоппа використовуються у постквантових криптосистемах, таких як: криптосистема МакЕліса. Використовує генераторну матрицю коду Гоппа для шифрування та перевіркову матрицю для декодування.

Завдяки стійкості до атак на основі квантових обчислень, коди Гоппа є перспективними для застосування в майбутніх криптосистемах.

2.2 Криптосистеми на основі кодів

Схема шифрування з відкритим ключем McEliece була запропонована майже 40 років тому, і з тих пір вона практично не зазнавала злому. Початкова ідея МакЕліса полягала в тому, щоб використати як зашифрований текст слово ретельно підібраного лінійного коду з виправленням помилок, у даному прикладі – двійкового коду Гоппи – до якого були додані випадкові помилки. Довільна основа коду – генераторна матриця – є відкритим ключем, що дозволяє будь-кому шифрувати (рисунок 2.1).

Рисунок 2.1 – Шифрування відкритим ключем на основі коду

Зашифрований текст у криптосистемі на основі кодів виглядає як кодове слово з помилками, які може виправити лише той, хто володіє спеціальними знаннями. Це дозволяє відновити початковий відкритий текст. Щоб відновити це повідомлення, потрібен секретний ключ – ефективний алгоритм декодування. Для всіх інших, включно зі зловмисниками, завдання декодування є надзвичайно складним, навіть якщо використовувати квантові комп'ютери.

Безпека цієї криптосистеми базується на двох важливих припущеннях:

1. Завдання загального декодування є обчислювально складним у середньому.

2. Відкритий ключ (наприклад, генераторна матриця) складно відрізнити від випадкової матриці.

Обчислювально складним вважається таке завдання, яке неможливо розв'язати швидко або ефективно. Збільшення параметрів системи робить задачу ще складнішою, і, як результат, захищає систему від зломів з обмеженими обчислювальними ресурсами.

Подвійне обґрунтування безпеки системи було зрозумілим ще з моменту її створення, але вперше це було чітко сформульовано в роботах Ніколя Куртуа та його колег. Перша проблема – завдання загального декодування – належить до класу NP-повних задач. Це означає, що навіть у середньому така задача є важкою для розв’язання. Хоча прогрес у розв’язанні можливий, це зазвичай вимагатиме збільшення параметрів системи. Але ймовірність серйозного прориву залишається дуже низькою.

Аналогічно до проблем розкладання чисел на множники або обчислення дискретного логарифма у криптосистемах на основі теорії чисел, безпека системи Нідеррайтера також залежить від постійних досліджень. Це допомагає не лише підтримувати довіру до системи, але й вчасно коригувати її параметри, якщо з’являються нові загрози.

Друга проблема – неможливість відрізнити відкритий ключ від випадкової матриці – поки що залишається менш дослідженою. Її

формулювання залежить від конкретної реалізації системи. Наприклад, МакЕліс запропонував використовувати бінарні коди Гоппа, для яких припущення про невідмінність ключа залишається справедливим (окрім ситуацій, коли швидкість коду наближається до 1, але це не актуально для шифрування).

Інші типи кодів, такі як коди Ріда-Соломона, конкатеновані коди чи низькодвільні коди, не відповідають цьому припущенню, тому системи, побудовані на їх основі, вважаються менш надійними. Пошук таких кодів, які відповідають цим умовам, залишається важливим завданням у криптографії, що базується на кодах.

Криптографічні примітиви, що базуються на теорії кодування, дозволяють реалізувати важливі функції, такі як шифрування, аутентифікація та цифрові підписи. Наприклад, шифрування з відкритим ключем можна здійснити за допомогою схеми Нідеррайтера, яка вважається еквівалентною схемі МакЕліса з точки зору безпеки. Крім цього, коди дозволяють реалізувати протоколи автентифікації з нульовим знанням і створювати цифрові підписи.

Протокол автентифікації без знань. Перший такий протокол був запропонований Жаком Стерном у 1993 році. Ідея протоколу полягає в тому, що одна сторона обирає кодове слово xx, тримає його в секреті, а потім публікує його модифіковану версію $y = x + e$, де e – це вектор помилок з невеликою вагою. Під час взаємодії ця сторона може довести іншій, що знає кодове слово, близьке до yy, не розкриваючи сам xx. Таким чином, протокол забезпечує автентифікацію, не надаючи стороннім додаткової інформації.

Цифрові підписи. На основі протоколів з нульовим знанням можна створювати схеми цифрових підписів за допомогою парадигми Фіата-Шаміра. Наприклад, протокол Стерна можна модифікувати для отримання схеми цифрового підпису. Проте, щоб забезпечити стійкість до квантових атак, конструкція вимагає певних змін. Ця схема відрізняється простотою

реалізації та невеликими розмірами ключів (кілька сотень байтів), хоча підписи виходять досить великими (від сотень до тисяч кілобайт).

Іншим підходом до створення цифрових підписів є парадигма "хеш і підпис". У цьому випадку користувач генерує дайджест повідомлення, який вважається підписаним як зашифрований текст, а відкритий ключ використовується для перевірки дійсності підпису. На жаль, стандартний примітив шифрування МакЕліса, який підходить для шифрування, не підходить для реалізації підписів.

Цифрові підписи, хоча й можливі, потребують використання бінарних кодів Гоппа зі швидкістю, що наближається до 1. Це саме той підклас кодів Гоппа, який може бути відрізнений від випадкових кодів. Хоча здатність розрізняти такі коди не дозволяє здійснити ефективну атаку, вона робить математичний доказ безпеки системи недійсним. Окрім того, для таких підписів необхідні великі відкриті ключі, алгоритм підпису має високу обчислювальну складність і погано масштабується.

Безпека шифрування в схемі McEliece. Схема шифрування McEliece є дуже гнучкою. Наприклад, додавання випадкового кодового слова до зашифрованого тексту призводить до утворення нового зашифрованого тексту, який відрізняється від оригінального. Якщо відкритий текст зашифрувати двічі з використанням одного і того ж ключа, то результати також будуть різними. Через це для забезпечення безпеки необхідно використовувати семантично захищене перетворення.

Цікавий побічний ефект такого підходу полягає в тому, що відкритий ключ у систематичній формі стає безпечним, тоді як у початковій схемі це вважалося небажаним. Перехід до систематичної форми відкритого ключа дозволяє суттєво зменшити його розмір, що є важливою перевагою.

На сьогодні криптографічні продукти, які використовують примітиви на основі кодів, ще не набули широкого розповсюдження. Однак дослідники активно працюють у цьому напрямку та пропонують різні ефективні реалізації таких систем як у програмному забезпеченні, так і для вбудованих

пристроїв. Одним із сучасних програмних рішень є McBits. Це програмне забезпечення повністю захищене від атак, що використовують аналіз часу виконання, і демонструє кращу продуктивність порівняно з іншими асиметричними схемами.

Щоб досягти «класичної» безпеки 128 біт (найвідоміші атаки вимагають щонайменше 2^{128} елементарних операцій) з оригінальною схемою McEliece з використанням двійкових кодів Гоппи, коди повинні мати розмірність $k = 3376$, ймовірно, і довжину $n = 4096$ з коригуванням $t = 60$ помилок. Розмір відкритого ключа становить 303 840 байт (з матрицею в систематичній формі), а розширення між відкритим і зашифрованим текстом становить приблизно 20 відсотків. Для такої самої безпеки від квантових супротивників розмір блоку має збільшитися у 2 рази, а розмір ключа – у 4 рази.

Переваги схеми McEliece та її варіантів включають наступне [19, 20]:

– безпека добре зрозуміла;

– він витримав 40 років ретельної перевірки;

– обчислювально ефективний у шифруванні та дешифруванні.

До мінусів можна віднести наступне:

– немає дійсно практичної схеми цифрового підпису;

– великий розмір відкритого ключа (порядку одного мегабайта для довготривалої безпеки);

– має бути припущення про нерозрізнення коду.

Напрямки розвитку методів шифрування на основі кодів.

1. Компактні ключі з квазіциклічних кодів. Вибравши належне сімейство кодів, можна обмежити вибір загальнодоступної генераторної матриці G блочно-циркуляційними матрицями (рисунок 3), дозволяючи суттєво зменшити вимоги до зберігання відкритого ключа.

2. Код, побудований блочно-циркуляційною матрицею, є квазіциклічним. Філіп Габорі вперше запропонував такі конструкції [13].

Система безпечна, доки:

– загальне декодування в квазіциклічному коді є важким;

– відкритий ключ неможливо відрізнити від випадкової блочно-циркуляційної матриці.

Подібно до циклічних варіантів ґратчастих жорстких задач, існує консенсус, що загальне декодування для квазіциклічних кодів залишається складним. З іншого боку, квазіциклічні коди є більш структурованими, і сімейство кодів потрібно вибирати ретельно. Зауважте також, що компактні ключі можна отримати за допомогою інших конструкцій, наприклад блочно-діадичних матриць, з подібним впливом на розмір ключа та безпеку.

3. Блоково-циркуляційні матриці.

Циркулянтна матриця – це квадратна матриця, в якій кожен рядок є поворотом на один елемент праворуч від попереднього рядка. Його повністю визначає перший рядок.

Блочно-циркуляційна матриця формується з циркулянтних квадратних блоків. Індекс блочно-циркуляційної матриці містить кількість блоків циркулянту в кожному рядку. Його порядок - це розмір циркулянтних блоків.

Для представлення матриці блок-циркулянт потрібен лише перший рядок кожного блоку циркулянта. Коли індекс невеликий, це призводить до значно менших відкритих ключів.

Циркулянтні двійкові матриці розміру $p \times p$ утворюють комутативне кільце, ізоморфне кільцю факторів полінома $F_2\,[x]/(x^p - 1)$. З циркулянтною матрицею, перший рядок якої $(a_0, a_1, \dots, a_{p-1})$, я зв'язую поліном $a(x) = a_0 + a_1 x + \dots + a_{p-1}x^{p-1}$. Матричне додавання, множення та інверсія відповідають одній поліноміальній операції за модулем $x^p - 1$. Вага Хеммінга для $a(x)$ є вагою Хеммінга двійкового слова $(a_0, a_1, \dots, a_{p-1})$.

Алгебраїчні квазіциклічні коди. Алгебраїчні коди мають спеціальну математичну структуру, яка дозволяє описувати їх за допомогою алгебраїчних засобів і використовувати швидкі алгоритми декодування. У теорії кодування багато напрямків, але для криптографії можна зосередитися

на підкласі альтернативних кодів, до яких належать, наприклад, двійкові коди Гоппа, що використовуються у схемі McEliece.

Якщо в криптосистемі McEliece використовується сімейство альтернативних кодів, можна з відкритого ключа створити систему багатоваріантних поліноміальних рівнянь над скінченним полем. Така система має розв'язок, якщо прихований код є альтернативним, але знайти його в загальному випадку дуже складно.

Проте, якщо швидкість коду (відношення k/n) близька до 1, система стає сильно перевизначеною. У такому випадку підсистема, створена з рівнянь низького степеня, хоча й не має розв'язку, може мати властивості, які дозволяють провести розрізнення (визначення коду). Якщо ж код є квазіциклічним, кількість невідомих значно зменшується, особливо за малого індексу квазіциклічності. Це може зробити розв'язання системи більш реалістичним, як показано в дослідженнях [21, 22].

Таким чином, поєднання алгебраїчної та квазіциклічної структур робить можливим спрощення атак. Проте "спрощення" не означає, що атака стає здійсненною на практиці. Більшість варіантів McEliece на основі альтернативних кодів, які використовують компактні ключі, залишаються безпечними. Однак дослідження в цьому напрямі тривають, і важливо розуміти межі можливостей сучасних алгоритмів розв'язання таких поліноміальних систем.

Квазіциклічні коди MDPC. Іншим підходом до побудови варіантів схеми McEliece є використання кодів MDPC (коди з помірною щільністю перевірок парності), запропоноване Рафаелем Місоцкі та його колегами. У таких кодах матриця перевірки парності є розрідженою: вона має рядки довжиною nn, вага Хеммінга яких пропорційна n.

Головна перевага полягає в тому, що знання розрідженої матриці перевірки парності дозволяє ефективно декодувати помилки ваги, яка також пропорційна n. Це робить коди MDPC перспективними для створення

криптосистем із компактними ключами, зберігаючи їхню надійність та безпеку.

Попри те, що такі підходи до зменшення розміру ключів активно досліджуються, вони вимагають додаткових оцінок щодо стійкості до атак, зокрема для нових класів криптографічних застосувань.

Відкритий ключ - це будь-яка матриця-генератор цього коду. Така матриця загалом щільна і нічого не розкриває про секретну розріджену матрицю перевірки парності.

Розріджена матриця перевірки парності може бути блочною циркуляцією, у цьому випадку загальнодоступна матриця генератора також є блочною циркулянтом, а відповідний код MDPC є квазіциклічним. Тоді всю схему можна описати в термінах поліномів у кільці факторів $\mathcal{R}_p = F_2[x]/(x^{p-1})$.

Вибір параметрів. Для 128 біт безпеки, с Коди QC-MDPC індексу 2, розміри параметрів є $p = 9857$, $w = 142$ і $t = 134$, що призводить до відкритого ключа розміром 9857 біт (близько 1,2 Кбайт). Проти квантового противників, розмір відкритого ключа має збільшитися до 32 771 біт, приблизно 4 Кбайт, замість 1 Мбайт для оригіналу Гоппа-МакЕліс.

Декодування кодів QC-MDPC складається з вирішення наступної проблеми.

Перша проблема (декодування QC-MDPC): задано s, h_0, h_1 у $\mathcal{R}_p$ з $wt(h_0) = wt(h_1) = w/2$, знайдіть e_0, e_1 у $\mathcal{R}_p$ так, щоб $wt(e_0) + wt(e_1) \leq t$ і $e_0\, h_0 + e_i\, h_j = s$.

Легко можна переконатися, що будь-хто, хто зможе розв’язати цю задачу, може виконати дешифрування.

Перша проблема відносно проста, доки w і t малі порівняно з довжиною коду. Насправді добуток t на w має бути такої ж величини, як і код довжини $n = 2\,p$. Різні відомі алгоритми можуть вирішити цю проблему;

точна продуктивність декодування залежатиме від обраного алгоритму, а параметри повинні бути перевірені за допомогою моделювання або аналізу.

2.3. Ефективність алгоритму в умовах класичних і квантових атак

Алгоритм Нідеррайтера є однією з криптографічних схем на основі кодування, яка пропонує високу стійкість до криптоаналітичних атак. Завдяки використанню теорії лінійних кодів, таких як коди Гоппа, алгоритм має потенціал протидіяти як класичним, так і квантовим загрозам. У цьому пункті проведемо аналіз ефективність алгоритму Нідеррайтера, зокрема його стійкість до атак, продуктивність і придатність до постквантових умов.

2.3.1 Класичні атаки

Атака на структуру коду Гоппа. Основу алгоритму Нідеррайтера складає задача декодування загального лінійного коду (General Linear Code Problem, GLCP), яка відома своєю NP-складністю. Для криптографічних цілей використовуються коди Гоппа, які мають регулярну структуру.

Атаки на структуру коду Гоппа зазвичай базуються на наступних підходах [23, 24]:

1) Аналіз регулярності перевіркової матриці. Навіть після маскування (застосування перестановок і скремблера) структура коду Гоппа може зберігати певні закономірності. Ці закономірності можна виявити через лінійні залежності або інші властивості матриці;

2) використання алгебраїчних властивостей. Код Гоппа будується на основі поліномів над кінцевими полями. Зловмисник може спробувати відновити параметри коду, такі як поліном Гоппа або поле, в якому він побудований;

3) алгоритм Міньона. Це метод для пошуку лінійних залежностей у кодах, який застосовується для виявлення структури. Він ефективний для кодів із низькою ентропією ключів;

4) атаки через розширені обчислювальні можливості. Квантові комп'ютери, що можуть ефективно вирішувати лінійні системи, потенційно становлять загрозу для кодів із визначеною регулярністю.

Складність атаки на структуру коду. На практиці ефективні атаки, такі як алгоритм Міньона або адаптивні лінійні атаки, вимагають експоненційних обчислень, що робить їх малопридатними для розв'язання задачі для великих розмірів ключів.

Брутфорс-атака на приватний ключ. Приватний ключ складається з перестановочної матриці, перевіркової матриці та скремблера. Перебір усіх можливих ключів є обчислювально складним завданням.

Етапи брутфорс-атаки:

1) відновлення матриці H. Брутфорс-атака починається з пошуку вихідної структури матриці H. У випадку коду Гоппа матриця H має регулярну структуру, зумовлену поліномом Гоппа. Зловмисник може:

– перебирати можливі поліноми Гоппа, які відповідають заданим параметрам;

– аналізувати масковану матрицю H' на наявність залишків регулярної структури.

2. Перебір перестановочної матриці. Перестановочна матриця P є бінарною матрицею розміром $n \times n$, яка виконує перестановку стовпців. Загальна кількість перестановок для n -вимірного коду дорівнює n!n!. Зловмисник перебирає всі можливі перестановки, намагаючись знайти таку, яка відновить вихідну структуру H;

3) перебір скремблера. Скремблер S є квадратною матрицею розміром $k \times k$, де k – кількість інформаційних символів. Зловмисник може перебирати всі можливі невироджені матриці S (матриці, які мають обернену).

Кількість таких матриць визначається як:

$$| GL(k, F2) | = (2k-1)(2k-2)\ldots(2k-2k-1),$$

де GL – позначає загальну лінійну групу.

4) перевірка правильності ключа: кожна пара S, P і відновленої структури H перевіряється на те, чи відтворює вона публічний ключ H':

$$H' = S \cdot H \cdot P.$$

Якщо це рівняння виконується, зловмисник отримує приватний ключ.

При оцінці складності брутфорс-атаки необхідно враховувати:

1) перестановочна матриця P. Для матриці розміру $n \times n$ кількість можливих варіантів дорівнює $n!$. Наприклад, для коду Гоппа з $n = 1024$:

$$1024! \approx 102568.1024! \approx 10^{2568};$$

2) Скремблер S: для $k = 524$ (типовий розмір для кодів Гоппа):

$$| GL(k, F_2) | \approx 21^{36136};$$

3) структура H: кількість можливих поліномів Гоппа для заданих n і t також є великою.

Сукупна складність атаки становить добуток цих факторів, що робить брутфорс практично неможливим навіть для квантових комп'ютерів у разі правильно вибраних параметрів.

Брутфорс-атака на приватний ключ алгоритму Нідеррайтера теоретично можлива, але складність перебору компонент приватного ключа – матриць S, P і структури H – настільки висока, що робить цю атаку непрактичною для реальних параметрів. Правильний вибір параметрів і достатнє маскування забезпечують стійкість алгоритму навіть до потужних обчислювальних засобів.

Оцінка складності. Ентропія ключів із сучасними параметрами (наприклад, 256-бітна стійкість) захищає від брутфорс-атак навіть на потужних комп'ютерах.

Атака розкриття структури матриці. Застосування методів аналізу матриць може дозволити зловмиснику відновити приватний ключ.

Коди Гоппа мають випадкову структуру, яка приховує їхні властивості, забезпечуючи захист від таких атак.

2.3.2 Квантові атаки

Атака за алгоритмом Гровер**а**. Квантовий алгоритм Гровера дозволяє прискорити перебір ключів із класичної складності $2n$ до $2n/2$. Це ставить під загрозу криптосистеми з малими розмірами ключів. Збільшення довжини ключа до 512 біт або більше може забезпечити захист від цієї атаки.

Атака з використанням алгоритму Шора. Алгоритм Шора ефективний проти схем, заснованих на задачах факторизації чи дискретного логарифма, але не застосовується безпосередньо до задачі GLCP.

Стійкість. Алгоритм Нідеррайтера не вразливий до цієї атаки, що робить його перспективним у постквантовій криптографії.

Атака на специфіку кодування. Для квантових атак може знадобитися спеціалізований квантовий алгоритм для розв'язання задачі декодування.

Прогнозована складність. Відсутність ефективних квантових алгоритмів для GLCP надає алгоритму Нідеррайтера значні переваги.

2.3.3 Ефективність алгоритму

Розмір ключів. Алгоритм Нідеррайтера вимагає великих ключів, оскільки перевіркова матриця має розмір $n \times (n-k)$. Проте це створює проблеми для сховищ із обмеженими ресурсами. Однак, великі ключі забезпечують високу стійкість до атак.

Швидкість виконання. Операції кодування та декодування є ефективними, оскільки базуються на матричних множеннях і додаваннях.

Алгоритм перевершує інші схеми постквантових підписів за швидкістю виконання, наприклад, SPHINCS+.

Придатність до впровадження. Алгоритм добре підходить для використання в обчислювальних пристроях із високою пропускною здатністю. Підтримка апаратних оптимізацій покращує ефективність.

В таблиці 2.1 приведено Порівняння з іншими постквантовими алгоритмами.

Алгоритм цифрового підпису Нідеррайтера є перспективним варіантом для постквантових умов завдяки своїй високій стійкості до класичних і квантових атак.

Таблиця 2.1 – Порівняння алгоритму Нідеррайтера з іншими постквантовими алгоритмами

Характеристика	Нідеррайтер	SPHINCS+	CRYSTALS-Dilithium
Стійкість до квантових атак	Висока	Висока	Висока
Розмір ключів	Великий	Середній	Малий
Швидкість підпису	Висока	Низька	Висока
Швидкість перевірки	Висока	Низька	Висока
Область застосування	Загальна криптографія	Критичні системи	Загальна криптографія

Хоча його основним недоліком є великий розмір ключів, це компенсується ефективністю підпису та перевірки. У майбутньому дослідження, спрямовані на оптимізацію розмірів ключів і підвищення продуктивності, дозволять забезпечити ще ширше впровадження цього алгоритму.

2.4 Тенденції розвитку постквантових алгоритмів шифрування

Криптосистеми на основі решіток. Алгоритми на основі решіток є одними з перспективних для постквантової криптографії, оскільки також мають NP-складні задачі, які не можуть бути вирішені квантовими комп'ютерами. Деякі з них мають кращу продуктивність і менші розміри ключів порівняно з Нідеррейтера [25].

Kyber (для шифрування). Цей алгоритм, заснований на задачах з решітками, був обраний для стандартизації як постквантова криптосистема в конкурсі NIST. Він пропонує високий рівень безпеки та порівняно менші розміри ключів.

FrodoKEM. Також використовує математичні структури на основі решіток і пропонує квантову стійкість. Серед переваг, менші розміри ключів і підписів порівняно з Нідеррейтером, висока ефективність у перевірці підписів.

Однак, у деяких випадках можуть бути повільнішими для створення підпису або шифрування.

2. Криптосистеми на основі многочленних карт. Різновиди алгоритмів на основі многочленних карт також стійкі до квантових атак і можуть бути застосовані для шифрування та цифрового підпису.

Rainbow. Це алгоритм цифрового підпису на основі многочленних карт. Він був кандидатом на стандартизацію в конкурсі NIST. Математична проблема, на якій він базується, є складною як для класичних, так і для квантових комп'ютерів.

UOV (Unbalanced Oil and Vinegar**)**. Один із варіантів схеми на основі многочленних карт для цифрових підписів. Серед переваг, висока швидкість генерування підписів і перевірки. Однак, більший розмір підпису, що може бути проблемою для деяких застосувань.

3. Криптосистеми на основі ізогеній еліптичних кривих. Алгоритми на основі ізогеній еліптичних кривих **(**постквантові алгоритми на еліптичних кривих) також пропонують високу стійкість до квантових атак.

SIKE (Supersingular Isogeny Key Encapsulation**)**. Це криптосистема на основі ізогеній еліптичних кривих, яка використовує важкообчислювані ізогенії для забезпечення квантової стійкості.

При цьому, дуже малі ключі порівняно з іншими постквантовими системами. Однак, високі обчислювальні витрати, повільніші операції.

4. Криптосистеми на основі хешування. Ці системи використовують лише криптографічні хеш-функції для створення стійких до квантових атак цифрових підписів. Вони можуть бути ефективними й простими для реалізації.

SPHINCS+.Це алгоритм цифрового підпису на основі хешування, який забезпечує постквантову безпеку і був обраний для стандартизації в конкурсі NIST. Серед переваг проста реалізація, стійкість до квантових атак. До недоліки необхідно віднести великі розміри підписів і відносно повільне створення підпису.

5. Криптосистеми на основі решіток Лерта.

NTRU. Один із найстаріших алгоритмів, стійких до квантових атак, заснований на решітках. Він був розроблений у 1996 році й далі вдосконалюється для використання в постквантових системах. Забезпечує високу ефективність і компактні ключі. Даний алгоритм є менш досліджений порівняно з іншими алгоритмами.

Гідними аналогами алгоритму Нідеррейтера є криптосистеми на основі решіток, многочленних карт, ізогеній еліптичних кривих і хешування. Серед них особливо перспективними для реального застосування є Kyber і SIKE, які пропонують менші розміри ключів і високий рівень безпеки. Алгоритм Нідеррейтера залишається конкурентним у галузі постквантової криптографії, але його великі ключі роблять його менш зручним у багатьох сучасних додатках, де компактність має значення.

3. РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ ЦИФРОВОГО ПІДПИСУ НІДЕРРАЙТЕРА

3.1 Алгоритм Нідеррайтера для підпису повідомлень різної довжини

Алгоритм Нідеррайтера для підпису повідомлень різної довжини складається з наступних кроків [26, 27].

Крок 1. Генерація ключів.

1.1. Вибрати код Гоппа параметрів $n, k, t,$

де n – довжина коду;

k – кількість інформаційних символів;

t – здатність коду виправляти помилки.

1.2. Побудова перевірочної матриці H для обраного коду Гоппа.

1.3. Вибір випадкових параметрів матриць:

– скремблер S розміру $t \times t$ (невироджена матриця);

– перестановочна матриця P розміру $n \times n$.

1.4. Створення публічного ключа:

$$H' = S \cdot H \cdot P.$$

Приватний ключ: (S, H, P).

Публічний ключ: H'.

Крок 2. Формування підпису.

2.1. Перевірити довжину повідомлення m. Якщо довжина $| m | \neq k$, доповнити m нулями до довжини k.

2.2. Згенерувати випадковий вектор помилок e розмірності n, який має вагу t (тобто кількість одиниць у векторі дорівнює t).

2.3. Створити кодове слово s:

$$s = H' \cdot mT + e,$$

де m^{T} – транспонована версія m.

Крок 3. Перевірка підпису

3.1. Верифікатор отримує (m,s) та публічний ключ H′.

3.2. Обчислити синдром S_s повідомлення s:

$$S_s = H' \cdot s^T.$$

3.3. Перевірити, чи відповідає синдром очікуваному значенню (чи можливо виправити помилки). Якщо S_s коректний, підпис вважається дійсним.

Розроблений алгоритм ефективний для захисту інформації в умовах класичних і квантових загроз, але потребує додаткових оптимізацій для використання в середовищах з обмеженими ресурсами.

3.2 Порівняння алгоритму нідеррайтера з постквантовими алгоритмами

Цифровий підпис на основі алгоритму Нідеррайтера є одним із перспективних рішень у постквантовій криптографії. Він використовує переваги задачі декодування випадкових лінійних кодів, яка є NP-складною, для забезпечення безпеки. Такий підхід робить алгоритм стійким до атак квантових комп'ютерів, що є однією з головних загроз сучасній криптографії.

Алгоритм цифрового підпису Нідеррайтера складається з наступних кроків [28, 29]:

1) генерація ключів:

– приватний ключ включає перевіркову матрицю H лінійного коду, матрицю перестановки P, та матрицю скремблера S.

– публічний ключ створюється шляхом множення матриць:

$$H' = S \cdot H \cdot P,$$

2) процес підписання:

– генерується випадковий вектор помилок e із заданою кількістю одиниць t;

– вектор повідомлення m (або його хеш) кодується за допомогою публічного ключа:

$$S = m \cdot H' + e;$$

– пара (e, S) є підписом;

3) процес перевірки:

– отримане повідомлення m та підпис (e, S) перевіряються шляхом обчислення:

$$S - m \cdot H';$$

– якщо результат є вектором із t – помилками, підпис вважається дійсним.

Розроблена програма, яка реалізує алгоритм цифрового підпису Нідеррайтера. Ми використовуємо бібліотеки *numpy* та *galois* для роботи з полем $GF(2\hat{}m)$, а також код Ріда-Соломона для забезпечення коригувальних властивостей. Програма забезпечує генерацію ключів, підписання повідомлень і перевірку підпису.

Алгоритм Нідеррайтера є одним з перспективних постквантових алгоритмів цифрового підпису, який базується на теорії кодування. Однак, він не є єдиним у своєму роді. Тому необхідно провести порівняємо його з іншими відомими постквантовими алгоритмами: Lattice-based, Code-based та Multivariate-quadratic (таблиця 3.1) [30, 31].

Таблиця 3.1 – Порівняння постквантових алгоритмів цифрового підпису

Характеристика	Алгоритм Нідеррайтера	Lattice-based	Code-based	Multivariate-quadratic
Математична основа	Теорія кодування	Теорія грат	Теорія кодування	Мультиваріатна алгебра
Стійкість до квантових атак	Висока	Висока	Висока	Висока
Швидкість	Середня	Залежить від параметрів	Середня	Залежить від параметрів
Розмір ключа	Відносно невеликий	Може бути великим	Може бути великим	Відносно невеликий
Довжина підпису	Відносно невелика	Може бути великою	Може бути великою	Відносно невелика
Складність реалізації	Середня	Висока	Середня	Висока

Для порівняння алгоритму Нідеррайтера вибрано наступні постквантові алгоритмами: CRYSTALS-Dilithium**,** FALCON **і** SPHINCS+ [31].

Алгоритм цифрового підпису CRYSTALS-Dilithium – постквантовий алгоритм цифрового підпису, заснований на проблемах із теорії решіток, таких як модулярна задача найкоротшого вектора (SVP) *та* модулярна задача найближчого вектора (CVP). Цей алгоритм входить до фінального раунду конкурсу NIST щодо стандартизації постквантової криптографії і є одним із найбільш перспективних підходів до захисту даних від атак квантових комп'ютерів.

FALCON (Fast Fourier Lattice-based Compact Signatures over NTRU) є постквантовим алгоритмом підпису. Він заснований на решіткових проблемах, таких як вектор найближчого сусіда (CVP) **і** короткий вектор у

решітці (SVP), та використовує структуру NTRU-решіток. FALCON відомий своєю високою ефективністю та компактним розміром підписів.

SPHINCS+ (Stateless Practical Hash-based Incredibly Nice and Compact Signature Scheme) – це постквантовий алгоритм цифрового підпису, заснований на хешуваннях. Він вирізняється стійкістю до атак квантових комп'ютерів, є статичним (stateless), не потребує збереження стану та гарантує високий рівень безпеки [31, 32].

В таблиці 3.2 приведено порівняння розмірів ключів, підписів і швидкості роботи алгоритму Нідеррайтера з іншими популярними постквантовими алгоритмами, такими як CRYSTALS-Dilithium, FALCON і SPHINCS+, для рівнів безпеки, еквівалентних AES-128 та AES-256. Наведені значення є загальними і можуть відрізнятися залежно від конкретних реалізацій і налаштувань.

Таблиця 3.2 – Порівняння розмірів ключів, підписів і швидкості роботи алгоритму Нідеррайтера з іншими популярними постквантовими алгоритмами

Алгоритм	Рівень безпеки	Розмір публічного ключа	Розмір приватного ключа	Розмір підпису	Швидкість підпису	Швидкість перевірки
Нідеррайтера	AES-128	50-150 КБ	1-2 КБ	0.5-1 КБ	Швидко	Повільно
	AES-256	200-500 КБ	2-3 КБ	1-2 КБ	Швидко	Повільно

Продовження таблиці 3.2

Алгоритм	Рівень безпеки	Розмір публічного ключа	Розмір приватного ключа	Розмір підпису	Швидкість підпису	Швидкість перевірки
CRYSTALS-Dilithium	AES-128	1.3 КБ	2.5 КБ	2.7 КБ	Дуже швидко	Дуже швидко
	AES-256	2.6 КБ	5 КБ	4.6 КБ	Дуже швидко	Дуже швидко
FALCON	AES-128	0.9 КБ	1.3 КБ	0.7 КБ	Швидко	Швидко
	AES-256	1.7 КБ	2.7 КБ	1.2 КБ	Швидко	Швидко
SPHINCS+	AES-128	1 КБ	1 КБ	16-17 КБ	Повільно	Повільно
	AES-256	1 КБ	1 КБ	41 КБ	Повільно	Повільно

Аналіз таблиці 2 Нідеррайтер має великий розмір публічного ключа (50-500 КБ), але підпис порівняно короткий (0.5-2 КБ). Це забезпечує високу квантову стійкість, проте великий розмір ключів може впливати на обсяг пам'яті та передачі даних.

CRYSTALS-Dilithium і FALCO мають менші розміри ключів і підписів, демонструючи кращі показники ефективності при більш компактних параметрах. Обидва алгоритми підтримують високу швидкість генерації та перевірки підписів.

SPHINCS+ гарантує квантову стійкість, проте має значно більший розмір підпису і працює повільніше через використання хеш-дерев.

Таким чином, алгоритм Нідеррайтера є перспективним для захищених середовищ, де пріоритет надається високій стійкості, навіть якщо це потребує великих розмірів ключів і низької швидкості перевірки підписів.

3.3 Програмна реалізація алгоритму цифрового підпису Нідеррайтера

Код програми, яка реалізує алгоритм цифрового підпису Нідеррайтера приведений в додатку А. Даний код реалізує базову версію алгоритму цифрового підпису на основі коду Хемінга (12, 8). Код ілюструє, як генерується підпис і перевіряється його автентичність, використовуючи концепції камуфльованої перевіркової матриці та випадкового вектора помилок.

Алгоритм цифрового підпису складається з наступних кроків.

1. Перевіркова матриця H:

```
H = np.array([
[1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0],
[1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0],
[0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0],
[0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1],
[0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1],
[0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1],
[1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1]
], dtype=int)
```

Це перевіркова матриця для коду Хемінга (12, 8), яка використовується для перевірки кодових слів. Вона дозволяє виявляти та виправляти однобітові помилки в повідомленнях.

2. Матриця скремблера S:

```
S = np.eye(H.shape[0], dtype=int)
```

3. Перестановочна матриця P.

Матриця P виконує перестановку стовпців матриці H, щоб зробити її вигляд менш очевидним для криптоаналітика. Це підвищує стійкість системи.

```
P = np.array([
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
], dtype=int)
```

4. Камуфльована матриця H′:

```
H_prime = np.mod(S @ H @ P, 2).
```

Камуфльована перевіркова матриця H′ обчислюється шляхом множення матриць S, H, і P. Вона виступає в ролі публічного ключа і використовується для створення та перевірки підпису.

5. Повідомлення m:

```
m = np.array([1, 0, 1, 1, 0, 0, 1, 0], dtype=int)
```

Це повідомлення довжиною 8 біт, яке підписується.

6. Розширене повідомлення $m_{extended}$

```
m_extended = np.zeros(H.shape[1], dtype=int)
m_extended[:len(m)] = m
```

Повідомлення m розширюється до розміру $n = 12$ (довжина кодового слова для коду Хемінга). Додаткові біти заповнюються нулями.

7. Генерація випадкового вектора помилок e:

```
e = np.zeros(H_prime.shape[0], dtype=int)
e[np.random.randint(0, H_prime.shape[0])] = 1
```

Вектор e моделює помилку. У цьому випадку генерується одинична помилка (єдиний ненульовий елемент), розташування якої обирається випадково.

8. Генерація підпису s:

```
s = np.mod(H_prime @ m_extended + e, 2)
```

Підпис обчислюється як сума $H' \cdot m_{extended} + e$. Це кодове слово, яке включає закамуфльовану перевіркову інформацію та помилку e.

9. Перевірка підпису:

```
reconstructed_e = np.mod(s - H_prime @ m_extended, 2)
reconstructed_e = np.mod(s - H_prime @ m_extended, 2)
print("\nПеревірка підпису (повинно дорівнювати e):", reconstructed_e)
```

На стороні перевірки отримане кодове слово s використовується для обчислення синдрому, з якого можна відновити e. Якщо відновлений e збігається з вихідним, підпис вважається дійсним.

10. Виведення результатів.

Програма виводить:

- матриці H, S, P, H'.
- оригінальне повідомлення m та розширене $m_{extended}$.
- генерований вектор помилок e.
- обчислений підпис s.
- результат перевірки підпису.

Результати роботи програми:

```
Перевіркова матриця H:
[[1 1 0 1 0 1 0 1 0 0 0 0]
 [1 0 1 1 0 0 1 1 0 0 0 0]
 [0 1 1 1 1 0 0 0 1 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 0 0]
 [0 0 0 0 0 0 1 1 1 0 1 1]
 [0 0 0 0 0 1 0 1 0 1 1 1]
 [0 0 0 0 1 0 1 0 1 1 1 1]
 [1 1 1 0 0 0 0 0 1 1 1 1]]

Матриця скремблера S:
[[1 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0]
 [0 0 0 1 0 0 0 0]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 1]]
```

```
Перестановочна матриця P:
[[0 0 0 0 0 0 0 0 1 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 1]]

Камуфльована матриця H' (публічний ключ):
[[1 0 1 0 1 0 1 0 1 0 0 0]
 [0 1 1 0 0 1 1 0 1 0 0 0]
 [1 1 1 1 0 0 0 1 0 0 0 0]
 [0 0 0 1 1 1 1 1 0 1 0 0]
 [0 0 0 0 0 1 1 1 0 0 1 1]
 [0 0 0 0 1 0 1 0 0 1 1 1]
 [0 0 0 1 0 1 0 1 0 1 1 1]
 [1 1 0 0 0 0 0 1 1 1 1 1]]

Повідомлення m: [1 0 1 0 0 0 1 1]

Розширене повідомлення m_extended: [1 0 1 0 0 0 1 1 0 0 0 0]

Вектор помилок e: [0 0 0 1 0 0 0 0]

Підпис (кодове слово) s: [1 0 1 1 0 1 1 0]

Перевірка підпису (повинно дорівнювати e): [0 0 0 1 0 0 0 0]
```

Програма генерує нову перевіркову матрицю, розширює повідомлення, додає помилку та формує підпис для коду Хеммінга (12, 8). Перевірити підпис також можена, відновлюючи вектор помилок e.

3.4 Дослідження криптостійкості алгоритму цифрового підпису Нідеррайтера

Розглянемо брутфорс-атаку на алгоритм цифрового підпису Нідеррайтера на прикладі коду Хеммінга (12, 8)

Алгоритм цифрового підпису Нідеррайтера використовує основні властивості лінійних кодів, а саме здатність до виправлення помилок. Код Хеммінга (12, 8) є прикладом лінійного блочного коду, що має такі параметри:

– довжина коду: $n = 12$;

– розмір інформаційного блоку: $k = 8$;

– число перевірок парності: $r = 4$ (тобто $n - k = 4$).

Кількість виправлених помилок: $t = 1$ (код виправляє одну помилку).

У контексті цифрового підпису Нідеррайтера, код Хеммінга використовується як основа криптографічного алгоритму. Захист алгоритму базується на складності задачі пошуку найкоротшого вектора помилок e із заданою вагою $w(e) = 1$, яка відповідає зашифрованому повідомленню.

Брутфорс-атака є методом "грубої сили", який передбачає перебір усіх можливих варіантів до знаходження правильного рішення.

Принцип роботи брутфорс-атаки.

1. Основна задача атаки. Брутфорс-атака на алгоритм Нідеррайтера спрямована на знаходження вектора помилок e, який відповідає зашифрованому тексту (або хешу).

Мета атакувальника*:*

– відновити початковий вектор повідомлення m, підписаний відправником;

– встановити відповідність між відкритим ключем (перетвореною матрицею G′) і початковою інформацією про код.

2. Модель алгоритму.

Цифровий підпис ґрунтується на таких перетвореннях:

– приватне повідомлення m перетворюється на кодове слово $c = m \cdot G$, де G – генераторна матриця;

– додається вектор помилок e із вагою $w(e) = t = 1$, $y = c + e$;

– результат y є зашифрованим текстом (або підписом).

3. Основні кроки брутфорс-атаки.

Брутфорс-атака передбачає послідовний перебір усіх можливих векторів помилок e, щоб знайти такий, який задовольняє рівняння:

$$y - e = c,$$

де $c \in$ кодовий простір.

1. Перебір векторів помилок. Усі можливі вектори e мають довжину $n = 12$ і вагу $w(e) = 1$. Це означає, що атакувальник повинен перебрати всі 12 можливих позицій помилки.

Кількість векторів e обчислюється як кількість комбінацій $C(n, w) = C(12, 1) = 12$.

Перевірка правильності вектора:

– для кожного вектора e обчислюється $y - e$;

– перевіряється, чи належить результат c до кодового простору, визначеного матрицею перевірки парності H:

$$H \cdot c^{\mathrm{T}} = 0.$$

Завершення атаки. Якщо вектор e правильний, то результат $c = y - e$ буде належати кодовому простору, і рівняння перевірки буде виконано.

Знаючи c, атакувальник може відновити початкове повідомлення m, оскільки: $m = c \cdot G^{-1}$.

4. Обчислювальна складність атаки.

У випадку коду Хеммінга (12, 8), вага $w(e) = 1$, а кількість можливих векторів помилок обмежується $C(12, 1) = 12$. Таким чином, атака для цього коду є відносно простою.

Однак для реальних криптографічних реалізацій, таких як коди з довжиною n=2048 і t=50, кількість можливих векторів зростає експоненційно:

$$C(n,t) = \frac{n!}{t!(n-t)!}.$$

Так як для прикладу, розглянуто код Хеммінга (12, 8), відповідно він має слабкі місця:

– мала довжина коду. Невеликий значення $n = 12$ робить брутфорс-атаку легко здійсненною, оскільки кількість можливих векторів e мала;

– мала вага помилок. Кількість можливих позицій для додавання помилок обмежена ($t = 1$), що також зменшує обчислювальну складність атаки;

– відсутність складної структури. Код Хеммінга простий для математичного аналізу, що полегшує атаку.

Брутфорс-атака на алгоритм цифрового підпису Нідеррайтера, що базується на коді Хеммінга (12, 8), можлива через низьку довжину коду, просту структуру і малу вагу помилок. У реальних системах використовуються більш складні коди (наприклад, двійкові коди Гоппи з більшими параметрами n і t), щоб зробити атаку грубою силою практично неможливою через експоненційне зростання кількості можливих векторів помилок.

Приклад коду на мові Python для брутфорс-атаки на алгоритм цифрового підпису Нідеррайтера, заснованого на коді Хеммінга (12, 8) приведено в додатку Б. Цей код ілюструє принцип атаки "грубої сили", де перебираються всі можливі вектори помилок, щоб знайти правильний.

Пояснення коду. Матриця перевірки парності застосовується для перевірки, чи належить вектор кодовому простору. У коді це робить функція *is_codeword*.

За допомогою бібліотеки *itertools.combinations* перебираються всі можливі вектори помилок із вагою $t = 1$.

Перевірка кодового слова. Для кожного вектора помилок обчислюється потенційне кодове слово $y - e$ і перевіряється його належність кодовому простору.

Якщо знайдено кодове слово, атака завершується, повертаючи знайдене слово та відповідний вектор помилок. Цей код демонструє основні ідеї брутфорс-атаки, але на практиці реальні системи використовують коди з набагато більшими параметрами n та t, що робить атаку такого типу обчислювально неможливою.

1. Генерація матриці перевірки парності.

Функція generate_parity_check_matrix() створює матрицю H для коду Хеммінга (12, 8).

Матриця H розміром (4×12) використовується для перевірки, чи належить отриманий вектор у кодовому простору.

Якщо $H \cdot y^{\mathrm{T}} mod\ 2 = 0$, то y є кодовим словом.

Структура H. Рядки матриці H задають контрольні рівняння для коду Хеммінга, які визначають, чи дотримується паритет кожної групи бітів.

Приклад:

H =

[[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0],

[1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0],

[1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0],

[0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1]]

2. Функція is_codeword (y, H). Ця функція перевіряє, чи є вектор y кодовим словом для заданого коду.

Алгоритм виконання функції наступний:

– обчислюється добуток $H \cdot y^{\mathrm{T}}\ mod\ 2$;

– якщо результат – нульовий вектор (всі елементи дорівнюють 0), то y є кодовим словом.

Кодове слово задовольняє всі рівняння перевірки парності, визначені матрицею H.

Функція brute_force_attack(received_vector, H).

Це основна функція, яка виконує атаку грубої сили на отриманий вектор received_vectorreceived_vector.

Алгоритм виконання функції наступний:

– перебирає всі можливі вектори помилок з вагою до $t = 1$;

– відновлює можливе кодове слово, виправляючи помилк;

– перевіряє, чи належить відновлений вектор кодовому простору за допомогою is_codeword.

3. Генерація векторів помилок. За допомогою itertools.combinations створюються всі можливі вектори помилок із точно однією одиницею (максимальна вага помилки $t = 1$).

Відновлення кодового слова. Кожен вектор помилок e віднімається від отриманого вектора received_vectorreceived_vector за модулем 2: candidate_codeword=(received_vector−e) mod 2 .

Перевірка кодового слова. Якщо candidate_codeword є дійсним кодовим словом, функція повертає його разом із вектором помилок.

4. Основний блок. Цей блок демонструє, як використовуються наведені функції.

Кроки алгоритму:

– вихідне повідомлення. Оригінальне повідомлення original_message має розмір $k = 8$.

Обчислення кодового слова. Кодове слово обчислюється як добуток

$$original\_message \cdot G \ (mod\ 2),$$

де G – генераторна матриця.

Матриця G визначає, як повідомлення перетворюється в кодове слово.

Додавання помилки. Вектор помилок error_vector додається до кодового слова, створюючи вектор received_vector.

Виконання атаки. Викликається brute_force_attack, щоб відновити кодове слово та вектор помилок.

Приклад виконання атаки.

1. Оригінальне повідомлення:
   original_message = [1, 0, 1, 1, 0, 1, 0, 1]
2. Згенероване кодове слово:
   codeword = [1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1]
3. Доданий вектор помилок:
   error_vector = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
4. Отриманий вектор:
   received_vector = [1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1]
5. Результат брутфорс-атаки:
   Знайдене кодове слово:
   recovered_codeword = [1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1]
   Вектор помилок:

recovered_error = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0].

Код демонструє базовий приклад для $t = 1$. У реальних атаках використовують вищі значення t, що значно збільшує обчислювальну складність.

Для великих розмірів коду (наприклад, $n = 2048$) цей підхід стає непридатним через експоненційне зростання кількості комбінацій.

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу підвищення ефективності цифрових підписів основі криптосистеми Нідеррейтера. При цьому отримано наступні результати.

1. Проведено аналіз сучасних алгоритмів цифрового підпису, що використовуються для захисту інформації, зокрема їхніх переваг і недоліків в умовах класичних і квантових атак. Встановлено, що традиційні алгоритми, такі як RSA та ECDSA, вразливі до квантових обчислень, що підкреслює актуальність розробки постквантових алгоритмів. У межах дослідження детально розглянуто криптосистему Нідеррейтера та обґрунтовано її застосування для створення стійкого цифрового підпису.

2. Досліджено математичний апарат теорії лінійних кодів, зокрема кодів Хеммінга, що лежить в основі криптосистеми Нідеррейтера.

3. Розроблено алгоритм цифрового підпису, що враховує ефективність в умовах класичних та квантових атак. Продемонстровано, що використання теорії лінійних кодів забезпечує високу стійкість до атак на основі квантових обчислень завдяки складності задачі декодування.

4. Реалізовано алгоритм цифрового підпису для роботи з повідомленнями різної довжини. У ході експериментів підтверджено його коректність і практичну застосовність.

5. Проведено порівняння алгоритму Нідеррейтера з іншими постквантовими алгоритмами за критеріями продуктивності, швидкодії та стійкості до атак. Результати засвідчили, що алгоритм Нідеррейтера є конкурентоспроможним у багатьох аспектах.

6. Вивчено криптостійкість алгоритму до класичних і квантових атак. Дослідження підтвердили високий рівень безпеки завдяки складності розв'язання задачі декодування, що є основою криптосистеми Нідеррейтера.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Baldi, M. Post-quantum cryptographic schemes based on codes. In: Proceedings of the International Conference on High Performance Computing & Simulation, 2017, pp. 908–910.

2. Rao, T.N.R., Nam, K.H. A private key algebraic coded cryptosystem. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO' 86. Lecture Notes in Computer Science, 1986, vol. 263, pp. 35–48.

3. Hooshmand, R., Aref, M.R. Efficient secure channel coding scheme based on low-density lattice codes. IET Commun. 2016, 10(11), pp. 1365–1373

4. Rao, T.N.R.: Joint encryption and error correction schemes. In: Proceedings of the Annual International Symposium on Computer Architecture, Association for Computing Machinery, New York, NY, 1984, pp. 240–241.

5. Sendrier, N.: Code based cryptography: State of art and perspectives. IEEE Secur. Privacy 2017, 15(4), pp. 44–50.

6. Cayrel, P.L., Meziani, M.: Post-quantum cryptography: Code-based signatures. In: Kim, Th., Adeli, H. (eds.), Advances in Computer Science and Information Technology. Lecture Notes in Computer Science, 2010, vol. 6059, pp. 82–99.

7. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Des. Codes Cryptogr. 2008, 49(1-3), pp. 289 –305.

8. Cayrel, P.L., Gaborit, P., Girault, M.: Identity based identification and signature schemes using correcting codes. In: Proceedings of International Workshop on Coding and Cryptograph, 2007, pp. 69–78.

9. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theory 1986, 15(2), pp. 159–166.

10. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology –

ASIACRYPT 2001. Lecture Notes in Computer Science, 2001, vol. 2248, pp. 157–174.

11. Xu, S.B., Doumen, J.M., Van Tilborg, H.C.A.: On the security of digital signature scheme based on error correcting codes. Des. Codes Cryptogr. 2003, 28(2), pp. 187–199.

12. K., Kapoor, S., Bhatia, R. (eds.) Intelligent Systems and Applications. Advances in Intelligent Systems and Computing, 2018, vol. 868, pp. 1251–1258.

13. Zhang, W., Wu, Q., Qin, B., Han, T., Zhang, Y., Chen, X., Li, N. TTP-free fair exchange of digital signatures with Bitcoin. In: Liu, J., Samarati, P. (eds.) Information Security Practice and Experience. Lecture Notes in Computer Science, 2017, vol. 10701, pp. 62–81.

14. Xinmei, W. Digital signature scheme based on error correcting codes. Electron. Lett. 1990, 26(13), pp. 898–899.

15. Finiasz, M. Parallel-CFS: Strengthening the CFS McEliece-based signature scheme. In: Proceedings of the International Conference on Selected Areas in Cryptography, 2011, pp. 159–170.

16. Chou, T., Niederhagen, R., Ran, L., & Samardjiska, S. Reducing Signature Size of Matrix-Code-Based Signature Schemes. In *International Conference on Post-Quantum Cryptography*. 2024, June, pp. 107-134.

17. Panthi, S., & Bhuyan, B. Quantum-Resistant Hash-Based Digital Signature Schemes: A Review. In *International Conference on Frontiers in Computing and Systems*. 2023, October, pp. 637-655.

18. Li, B., Wang, P., Zhang, X., & Wang, Z. Digital Signature Technology of Mobile Phone Verification Code based on Biometrics. In *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2023, September, pp. 1-5. IEEE.

19. Prabowo, Theo Fanuela, and Chik How Tan. "Attack on a code-based signature scheme from QC-LDPC codes." *International Conference on Codes, Cryptology, and Information Security*. Cham: Springer Nature Switzerland. 2023, pp. 136-149.

20. Kichna, A., & Farchane, A. Secure and efficient code-based cryptography for multi-party computation and digital signatures. In *Computer Sciences & Mathematics Forum.* 2023, May, Vol. 6, No. 1, p. 1.

21. Alahmadi, A., Çalkavur, S., Solé, P., Khan, A. N., Raza, M. A., & Aggarwal, V. A new code based signature scheme for blockchain technology. Mathematics. 2023, 11(5), 1177.

22. Moody, D., Perlner, R., Regenscheid, A., Robinson, A., & Cooper, D. (2024). Transition to Post-Quantum *Cryptography Standards* (No. NIST Internal or Interagency Report (NISTIR) 8547 (Draft)). National Institute of Standards and Technology.

23. Vambol, A., Kharchenko, V., Potii, O., & Bardis, N. Post-Quantum Network Security: McEliece and Niederreiter Cryptosystems Analysis and Education Issues. *WSEAS Transactions on Systems and Control*, 2020, *15*, 627-634.

24. Циганенко О. С. Development of digital signature algorithm based on the Niederriter crypto-code system. *Системи обробки інформації*, 2020, 3 (162): 86-94.

25. Погасій С. С., Мілевський С. В., Жученко, О. С., Томашевський Б. П., Рагімова, І. Р., & Сергієв, С. В. Development of Niederriter crypto-code design models on LDPC-codes. *Системи обробки інформації*, 2021, 4 (167): 58-68.

26. Pöppelmann, T., Ducas, L., Güneysu, T.: Enhanced lattice-based signatures on reconfigurable hardware. In: Batina, L., Robshaw, M., (eds.) Cryptographic Hardware and Embedded Systems – CHES 2014, Lecture Notes in Computer Science, 2014, vol. 8731, pp. 353–370. Springer, Berlin.

27. Howe, J., Pöppelmann, T., O'Neill, M., O'Sullivan, E., Güneysu, T. Practical lattice-based digital signature schemes. ACM Trans. Embedded Comput. Syst. 2015, 14(3), pp. 1–24

28. Das, D., Hoffstein, J., Pipher, J., Whyte, W., Zhang, Z. Modular lattice signatures, revisited. Des. Codes Cryptogr. 2019, 88(3), pp. 505–532.

29. Diemert, D., Gellert, K., Jager, T., Lyu, L. More efficient digital signatures with tight multi-user security. In: Proceedings Public-Key Cryptography. Lecture Notes in Computer Science, 2021, vol. 12711, pp. 1–31, Springer, Berlin.

30. Goldwasser, S., Micali, S., Rivest, R. A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 1988, 17(2), pp. 281–308.

31. Мачуляк М., Кондратюк В., Дуданець Р. Алгоритм цифрового підпису на основі криптосистеми Нідеррайтера. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2024. – С. 98-99.

32. Кондратюк В., Сидорчук Р., Дуданець Р. Порівняння алгоритму нідеррайтера з постквантовими. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2024. – С. 30-32.

ДОДАТОК А

Код програми цифрового підпису Нідеррайтера

```
import numpy as np

# Перевіркова матриця для коду Хеммінга (12, 8)
H = np.array([
    [1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0],
    [1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0],
    [0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1],
    [0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1],
    [0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1],
    [1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1]
], dtype=int)

# Матриця скремблера (одинична матриця)
S = np.eye(H.shape[0], dtype=int)

# Перестановочна матриця
P = np.array([
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
], dtype=int)

# Камуфльована матриця H' (публічний ключ)
H_prime = np.mod(S @ H @ P, 2)
```

```
# Повідомлення довжиною 8 біт
m = np.array([1, 0, 1, 1, 0, 0, 1, 0], dtype=int)

# Розширене повідомлення (довжина 12)
m_extended = np.zeros(H.shape[1], dtype=int)
m_extended[:len(m)] = m  # Заповнення перших 8 бітів

# Генерація випадкового вектора помилок e
e = np.zeros(H_prime.shape[0], dtype=int)  # Довжина e повинна відповідати кількості рядків H'
e[np.random.randint(0, H_prime.shape[0])] = 1

# Формування підпису: s = H' @ m_extended + e
s = np.mod(H_prime @ m_extended + e, 2)

# Виведення результатів
print("Перевіркова матриця H:")
print(H)
print("\nМатриця скремблера S:")
print(S)
print("\nПерестановочна матриця P:")
print(P)
print("\nКамуфльована матриця H' (публічний ключ):")
print(H_prime)
print("\nПовідомлення m:", m)
print("\nРозширене повідомлення m_extended:", m_extended)
print("\nВектор помилок e:", e)
print("\nПідпис (кодове слово) s:", s)

# Перевірка підпису
reconstructed_e = np.mod(s - H_prime @ m_extended, 2)
print("\nПеревірка підпису (повинно дорівнювати e):", reconstructed_e)
```

ДОДАТОК Б

Код програми атаки брутфорс-атаки на алгоритм цифрового підпису Нідеррайтера

```
import numpy as np
from itertools import combinations
def generate_parity_check_matrix():
    H = np.array([
        [1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0],
        [1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0],
        [1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0],
        [0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1]
    ])
    return H
def is_codeword(y, H):
    return np.all(np.dot(H, y.T) % 2 == 0)
def brute_force_attack(received_vector, H):
    n = len(received_vector)
    t = 1  # Максимальна вага помилки
    for error_positions in combinations(range(n), t):
        # Створення вектора помилок
        e = np.zeros(n, dtype=int)
        for pos in error_positions:
            e[pos] = 1
        candidate_codeword = (received_vector - e) % 2
        if is_codeword(candidate_codeword, H):
            return candidate_codeword, e
    return None, None
# Приклад роботи
if __name__ == "__main__":
```

```
original_message = np.array([1, 0, 1, 1, 0, 1, 0, 1])
G = np.array([
    [1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1],
    [0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0],
    [0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1],
    [0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1],
    [0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1]
])
codeword = np.dot(original_message, G) % 2
error_vector = np.zeros(12, dtype=int)
error_vector[3] = 1  # Помилка у 3-й позиції
received_vector = (codeword + error_vector) % 2
H = generate_parity_check_matrix()
recovered_codeword, recovered_error = brute_force_attack(received_vector, H)
print("Зашифроване слово:", received_vector)
print("Знайдене кодове слово:", recovered_codeword)
print("Вектор помилок:", recovered_error)
```

Вивід результатів:

```
Зашифроване слово: [1 0 1 0 0 1 0 1 0 0 1 0]
Знайдене кодове слово: [1 0 1 0 0 1 1 1 0 0 1 0]
Вектор помилок: [0 0 0 0 0 0 1 0 0 0 0 0]
>>>
```

ДОДАТОК В
Копії публікацій



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В'ЯЧЕСЛАВА ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2024)

науково-практична конференція
молодих вчених, аспірантів та студентів

26–28 серпня 2024
Тернопіль

БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ

КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

УДК 004.056.5

Михайло МАЧУЛЯК, Владислав КОНДРАТЮК, Роман ДУДАНЕЦЬ

Західноукраїнський національний університет

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ НА ОСНОВІ КРИПТОСИСТЕМИ НІДЕРРАЙТЕРА

Вступ. Розвиток квантових обчислень створює серйозну загрозу сучасним криптографічним алгоритмам, зокрема схемам цифрового підпису на основі RSA, DSA та ECC. Квантові алгоритми, такі як алгоритм Шора, здатні ефективно розв'язувати задачі факторизації та дискретного логарифма, що робить традиційні криптосистеми вразливими. У зв'язку з цим постає гостра потреба у впровадженні постквантових криптографічних методів, які забезпечують високий рівень безпеки навіть за наявності квантових обчислювальних потужностей [1]. Таким чином, дослідження алгоритму цифрового підпису на основі криптосистеми Нідеррайтера є актуальним, оскільки воно сприяє розвитку постквантової криптографії, забезпечує захист інформації у нових технологічних умовах.

Метою роботи є дослідити алгоритм цифрового підпису на основі криптосистеми Нідеррайтера для забезпечення високого рівня безпеки у постквантових обчислювальних середовищах.

1. Основи криптосистеми Нідеррайтера

Алгоритм цифрового підпису на основі криптосистеми Нідеррайтера є одним із перспективних рішень у галузі постквантової криптографії. Цей підхід базується на складності декодування лінійних кодів, зокрема, кодів Ріда-Соломона або кодів Гоппа, що є NP-складною задачею навіть для квантових комп'ютерів. Завдяки цьому, алгоритм Нідеррайтера має високий рівень криптографічної стійкості та підходить для використання в довготривалих системах захисту інформації. Для алгоритму цифрового підпису на основі криптосистеми Нідеррайтера обчислення базуються на генераторній матриці та векторах синдрому. Підпис утворюється шляхом створення вектору помилок і перетворень з використанням приватного ключа. Алгоритм цифрового підпису Нідеррайтера виконується в три основні етапи [2, 3].

1. Генерація ключів: Публічний ключ G' приховує структуру базового коду, шляхом множення на матрицю скремблера S та матрицю перестановок P. Приватний ключ включає інформацію про S, P і початковий код.

2. Процес підпису:

- хешування повідомлення для отримання вектору-образу в просторі синдрому;
- декодування цього вектору за допомогою приватного ключа для отримання вектору помилок, який слугує підписом.

3. Перевірка підпису:

- підпис разом із повідомленням хешується для порівняння з відповідним вектором синдрому;
- якщо отримані результати збігаються, підпис вважається дійсним.

У порівнянні з алгоритмами, такими як RSA чи ECDSA, криптосистема

Нідеррайтера забезпечує менший розмір підпису, але більший розмір публічного ключа. Це робить її зручною для сценаріїв, де важлива швидкість перевірки підпису, наприклад, у системах Інтернету речей (IoT) (таблиця 1).

Таблиця 1 - Порівняння алгоритмів цифрового підпису на основі McEliece, Нідеррайтера та RSA

Критерій	McEliece	Нідеррайтер	RSA
Основна ідея	Використання кодів корекції помилок (Goppa-коди).	Використання кодів корекції помилок (Goppa, Ріда-Соломона).	Використання факторизації великих чисел.
Стійкість до квантових атак	Висока. Постквантовий алгоритм.	Висока. Постквантовий алгоритм.	Низька. Уразливий до атак квантових комп'ютерів.
Розмір ключів	Дуже великий (кілька сотень кілобайт).	Дуже великий (кілька сотень кілобайт).	Невеликий (кілька кілобайт).
Розмір підпису	Великий (порядку сотень байтів).	Великий (порядку сотень байтів).	Невеликий (кілька сотень бітів).
Придатність для систем із обмеженими ресурсами	Обмежена через великий розмір ключів і підписів.	Обмежена через великий розмір ключів і підписів.	Придатний через невеликі розміри ключів і швидкість роботи.
Складність математичних операцій	Базується на складності декодування лінійних кодів.	Базується на складності декодування лінійних кодів.	Базується на складності факторизації цілих чисел.

Хоча алгоритм має переваги, він стикається з викликами, пов'язаними з великим розміром публічних ключів. Для оптимізації продуктивності пропонуються такі підходи: 1) використання скорочених лінійних кодів для зменшення розміру ключів; 2) апаратна реалізація операцій для підвищення швидкості декодування.

Висновок. Алгоритм цифрового підпису на основі криптосистеми Нідеррайтера демонструє значний потенціал для використання у постквантових системах завдяки своїй математичній складності і стійкості до атак. Подальші дослідження спрямовані на зменшення розміру публічних ключів і оптимізацію обчислень, що відкриває нові перспективи її застосування у високонавантажених системах.

Перелік використаних джерел.

1. Vambol, A., Kharchenko, V., Potii, O., Bardis, N. Post-Quantum Network Security: McEliece and Niederreiter Cryptosystems Analysis and Education Issues. WSEAS Transactions on Systems and Control, 2020, *15*, 627-634.

2. Циганенко О. С. Development of digital signature algorithm based on the Niederriter crypto-code system. Системи обробки інформації, 2020, 3 (162): 86-94.

3. Погасій С.С., Мілевський С.В., Жученко, О.С., Томашевський Б.П., Рагімова, І.Р., Сергієв, С.В. Development of Niederriter crypto-code design models on LDPC-codes. Системи обробки інформації, 2021, 4 (167): 58-68.

ГРОМАДСЬКА ОРГАНІЗАЦІЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2024
Тернопіль

УДК 004.056.5

Владислав КОНДРАТЮК, Роман СИДОРЧУК, Роман ДУДАНЕЦЬ

Західноукраїнський національний університет

ПОРІВНЯННЯ АЛГОРИТМУ НІДЕРРАЙТЕРА З ПОСТКВАНТОВИМИ АЛГОРИТМАМИ

Вступ. Розвиток квантових технологій ставить під загрозу безпеку традиційних криптографічних алгоритмів, таких як RSA, DSA та алгоритми на основі еліптичних кривих. Постквантова криптографія покликана забезпечити стійкість до атак, реалізованих квантовими комп'ютерами, і є одним із ключових напрямів досліджень у галузі інформаційної безпеки. Алгоритм цифрового підпису Нідеррайтера, заснований на використанні коригуючих кодів, є перспективним рішенням, що забезпечує високу криптографічну стійкість і задовольняє вимоги до постквантових систем. Водночас існують й інші підходи, такі як алгоритми на основі ґраток, хеш-функцій і мультіваріантних поліномів. Кожен із цих методів має свої переваги та обмеження, які важливо враховувати при виборі оптимального рішення для конкретних задач. Актуальність порівняння алгоритму цифрового підпису Нідеррайтера з іншими постквантовими алгоритмами полягає в необхідності оцінки його ефективності та придатності до реальних умов застосування. Це допоможе визначити сильні та слабкі сторони алгоритму Нідеррайтера, а також виявити сценарії, в яких він є найкращим вибором [1, 2].

Метою роботи є проведення комплексного порівняльного аналізу алгоритму цифрового підпису Нідеррайтера з іншими постквантовими алгоритмами цифрового підпису.

1. Алгоритм цифрового підпису Нідеррайтера

Цифровий підпис на основі алгоритму Нідеррайтера є одним із перспективних рішень у постквантовій криптографії. Він використовує переваги задачі декодування випадкових лінійних кодів, яка є NP-складною, для забезпечення безпеки. Такий підхід робить алгоритм стійким до атак квантових комп'ютерів, що є однією з головних загроз сучасній криптографії. Алгоритм цифрового підпису Нідеррайтера складається з наступних кроків [2]:

1) генерація ключів:

- приватний ключ включає перевіркову матрицю H лінійного коду, матрицю перестановки P, та матрицю скремблера S.
- публічний ключ створюється шляхом множення матриць: $H' = S \cdot H \cdot P$.

2) процес підписання**:**

- генерується випадковий вектор помилок e із заданою кількістю одиниць t;
- вектор повідомлення m (або його хеш) кодується за допомогою публічного ключа:

$$S = m \cdot H' + e;$$

- пара (e, S) є підписом;

3) процес перевірки**:**

- отримане повідомлення m та підпис (e, S) перевіряються шляхом обчислення:

$$S - m \cdot H';$$

- якщо результат є вектором із t - помилками, підпис вважається дійсним.

Розроблена програма, яка реалізує алгоритм цифрового підпису Нідеррайтера. Ми використовуємо бібліотеки numpy та galois для роботи з полем $GF(2\hat{}m)$, а також код Ріда-Соломона для забезпечення коригувальних властивостей. Програма забезпечує генерацію ключів, підписання повідомлень і перевірку підпису.

Алгоритм Нідеррайтера є одним з перспективних постквантових алгоритмів цифрового підпису, який базується на теорії кодування. Однак, він не є єдиним у своєму роді. Тому необхідно провести порівняємо його з іншими відомими постквантовими алгоритмами: Lattice-based, Code-based та Multivariate-quadratic (таблиця 1).

Таблиця 1 - Порівняння постквантових алгоритмів цифрового підпису

Характеристика	**Алгоритм Нідеррайтера**	**Lattice-based**	**Code-based**	**Multivariate-quadratic**
Математична основа	Теорія кодування	Теорія грат	Теорія кодування	Мультиваріатна алгебра
Стійкість до квантових атак	Висока	Висока	Висока	Висока
Швидкість	Середня	Залежить від параметрів	Середня	Залежить від параметрів
Розмір ключа	Відносно невеликий	Може бути великим	Може бути великим	Відносно невеликий
Довжина підпису	Відносно невелика	Може бути великою	Може бути великою	Відносно невелика
Складність реалізації	Середня	Висока	Середня	Висока

Для порівняння алгоритму Нідеррайтера вибрано наступні постквантові алгоритмами: CRYSTALS-Dilithium, FALCON і SPHINCS+ [3].

Алгоритм цифрового підпису CRYSTALS-Dilithium - постквантовий алгоритм цифрового підпису, заснований на проблемах із теорії решіток, таких як модулярна задача найкоротшого вектора (SVP) та модулярна задача найближчого вектора (CVP). Цей алгоритм входить до фінального раунду конкурсу NIST щодо стандартизації постквантової криптографії і є одним із найбільш перспективних підходів до захисту даних від атак квантових комп'ютерів.

FALCON (Fast Fourier Lattice-based Compact Signatures over NTRU) є постквантовим алгоритмом підпису. Він заснований на решіткових проблемах, таких як вектор найближчого сусіда (CVP) і короткий вектор у решітці (SVP), та використовує структуру NTRU-решіток. FALCON відомий своєю високою ефективністю та компактним розміром підписів.

SPHINCS+ (Stateless Practical Hash-based Incredibly Nice and Compact Signature Scheme) - це постквантовий алгоритм цифрового підпису, заснований на хешуваннях. Він вирізняється стійкістю до атак квантових комп'ютерів, є статичним (stateless), не потребує збереження стану та гарантує високий рівень безпеки.

В таблиці 2 приведено порівняння розмірів ключів, підписів і швидкості роботи

алгоритму Нідеррайтера з іншими популярними постквантовими алгоритмами, такими як CRYSTALS-Dilithium, FALCON і SPHINCS+, для рівнів безпеки, еквівалентних AES-128 та AES-256. Наведені значення є загальними і можуть відрізнятися залежно від конкретних реалізацій і налаштувань.

Таблиця 2 - Порівняння розмірів ключів, підписів і швидкості роботи алгоритму Нідеррайтера з іншими популярними постквантовими алгоритмами

Алгоритм	**Рівень безпеки**	**Розмір публічного ключа**	**Розмір приватного ключа**	**Розмір підпису**	**Швидкість підпису**	**Швидкість перевірки**
Нідеррайтера	AES-128	50-150 КБ	1-2 КБ	0.5-1 КБ	Швидко	Повільно
	AES-256	200-500 КБ	2-3 КБ	1-2 КБ	Швидко	Повільно
CRYSTALS-Dilithium	AES-128	1.3 КБ	2.5 КБ	2.7 КБ	Дуже швидко	Дуже швидко
	AES-256	2.6 КБ	5 КБ	4.6 КБ	Дуже швидко	Дуже швидко
FALCON	AES-128	0.9 КБ	1.3 КБ	0.7 КБ	Швидко	Швидко
	AES-256	1.7 КБ	2.7 КБ	1.2 КБ	Швидко	Швидко
SPHINCS+	AES-128	1 КБ	1 КБ	16-17 КБ	Повільно	Повільно
	AES-256	1 КБ	1 КБ	41 КБ	Повільно	Повільно

Аналіз таблиці 2 Нідеррайтер має великий розмір публічного ключа (50-500 КБ), але підпис порівняно короткий (0.5-2 КБ). Це забезпечує високу квантову стійкість, проте великий розмір ключів може впливати на обсяг пам'яті та передачі даних.

CRYSTALS-Dilithium і FALCO мають менші розміри ключів і підписів, демонструючи кращі показники ефективності при більш компактних параметрах. Обидва алгоритми підтримують високу швидкість генерації та перевірки підписів.

SPHINCS+ гарантує квантову стійкість, проте має значно більший розмір підпису і працює повільніше через використання хеш-дерев.

Висновок. Таким чином, алгоритм Нідеррайтера є перспективним для захищених середовищ, де пріоритет надається високій стійкості, навіть якщо це потребує великих розмірів ключів і низької швидкості перевірки підписів.

Перелік використаних джерел.

1. Kichna, A., & Farchane, A. Secure and efficient code-based cryptography for multi-party computation and digital signatures. In Computer Sciences & Mathematics Forum. 2023, May, Vol. 6, No. 1, p. 1.

2. Alahmadi, A., Çalkavur, S., Solé, P., Khan, A. N., Raza, M. A., & Aggarwal, V. A new code based signature scheme for blockchain technology. Mathematics, 2023, 11(5), 1177.

3. Moody, D., Perlner, R., Regenscheid, A., Robinson, A., & Cooper, D. (2024). Transition to Post-Quantum Cryptography Standards (No. NIST Internal or Interagency Report (NISTIR) 8547 (Draft)). National Institute of Standards and Technology.