НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря Сікорського»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
Завідувач кафедри

_____М.В. Грайворонський
(підпис)

“___”_____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему:

Метод оцінки стійкості алгоритмів обміну ключами в алгоритмах постквантової криптографії SIDH і CSIDH

Виконав (-ла): студент (-ка) _4_ курсу, групи ___ФБ-51____
(шифр групи)

Власенко Андрій Валерійович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник к.ф.-м.н., доц. Орєхов О.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент к.ф.-м.н., ст. викладач Фесенко А.В. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Власенко Андрію Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод оцінки стійкості алгоритмів обміну ключами в алгоритмах постквантової криптографії SIDH і CSIDH,
науковий керівник роботи к.ф.-м.н., доц. Орєхов Олександр Арсенійович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 27 » травня 2019 р. № 1414-С

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи Метод оцінки алгоритмів обміну ключами, оцінка алгоритмів SIDH і CSIDH

4. Зміст роботи
1) Проаналізувати існуючи методи оцінки загроз в інформаційній безпеці;
2) розробити метод оцінки і класифікації алгоритмів обміну ключами;
3) проаналізувати SIDH і CSIDH дослідити їх стійкість, швидкодію, існуючи вразливості;
4) застосувати розроблений метод до досліджуваних алгоритмів.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____
_____
_____
_____

6. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка

Студент _____ А.В. Власенко
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ О.А. Орєхов
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Робота обсягом 65 сторінок містить 8 таблиць, 1 додаток та 37 літературних посилань.

Метою дипломної роботи є створення методу оцінки стійкості алгоритмів обміну ключами, дослідження існуючих постквантових алгоритмів систем обміну ключами, а саме SIDH і CSIDH; оцінка алгоритмів в контексті розробленого методу.

Об’єктом дослідження є алгоритми обміну ключами, базовані на ізогеній суперсингулярних елиптичних кривих, а саме SIDH і CSIDH.

Предметом дослідження є стійкість алгоритмів на основі ізогеній суперсингулярних елиптичних кривих, можливі атаки і методи захисту від них.

Результати роботи викладені у вигляді висновків щодо захищеності досліджуваних алгоритмів та таблиць.

Результати роботи можуть бути використані для пошуку кращого претенденту алгоритму для використання у постквантовій і класичної криптографії. Також метод оцінки стійкості може бути використаний для дослідження інших алгоритмів обміну ключами.

ПОСТКВАНТОВА КРИПТОГРАФІЯ, АЛГОРИТМИ ОБМІНУ КЛЮЧАМИ, ІЗОГЕНІЇ, СУПЕРСИНГУЛЯРНІ ЕЛІПТИЧНІ КРИВІ, АЛГОРИТМ SIDH, АЛГОРИТМ CSIDH, МЕТОД ОЦІНКИ СТІЙКОСТІ

ABSTRACT

The work includes 65 pages, 8 tables, 1 appendix and 37 literary references.

The purpose of the thesis is to create a method for assessing the resistance of key-exchange algorithms, the study of existing post-quantum algorithms of key exchange systems, namely SIDH and CSIDH; estimation of algorithms in the context of the developed method.

The object of researches is algorithms for key exchange, based on the isogeny of supersingular elliptic curves, namely SIDH and CSIDH.

The subject of researches is the resistance of algorithms based on isogeny of supersingular elliptic curves, possible attacks and methods of protection against them.

The results of the work are presented in the form of conclusions about the security of the studied algorithms and tables.

The results of the work can be used to find the best candidate for the algorithm for use in post-quantum and classical cryptography. Also, the security evaluation method can be used to research other key exchange algorithms.

POSTQUANTUM CRYPTOGRAPHY, KEY EXCHANGE ALGORITHM, IZOGENY, SUPERSINGULAR ELLIPTIC CURVES, ALGORITHM SIDH, ALGORITHM CSIDH, METHOD OF EVALUATION OF RESISTANCE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

SIDH (Supersingular isogeny Diffie–Hellman key exchange) – протокол Діффі-Геллмана з використанням суперсингулярної ізогенії

CSIDH (Commutative supersingular isogeny Diffie–Hellman key exchange) – комутативний протокол Діффі-Геллмана з використанням суперсингулярної ізогенії.

Кубіт – одиниця квантової інформації, квантовий аналог біта.

Факторизація або розкладання на множники – це декомпозиція об'єкту (наприклад, числа, многочлена або матриці) у добуток інших об'єктів, або множників, які після перемноження дадуть вихідний об'єкт.

Дискретне логарифмування – це задача обернення функції g^x в деякій кінцевої мультиплікативної групі G.

ECDLP (Elliptic Curve Discrete Logarithm Problem) – задача дискретного логарифмування в групі точок еліптичної кривої.

Аліса і Боб — імена, що зазвичай використовуються як імена заповнювачі, метазмінних для архетипічних символів у таких областях, як криптографія, комп'ютерна безпека і фізика. Представляють собою дві сторони обміну повідомленнями.

Єва – пасивний зловмисник. Вона може прослуховувати повідомлення між Алісою і Бобом, але вона не може впливати на них. У квантовій криптографії Єва може представляти навколишнє середовище.

Нечесний Боб – зловмисник, що лише прикидається справжнім Бобом.

NIST (Націона́льний інститу́т станда́ртів і технолóгії) – національний орган зі стандартизації у США.

ВСТУП

Актуальність дослідження. Розробка більш сучасних і досконалих криптографічних алгоритмів завжди була однією з головних завдань у сфері захисту інформації. У наш час технічний прогрес все сильніше прискорюється і перед алгоритмами постають нові завдання і нові загрози. Сучасна загроза усім класичним криптографічним алгоритмам – це квантовий комп'ютер, що може розв'язувати задачі, на складності розв'язання яких ґрунтуються асиметричні криптосистеми, а саме задачі факторизації цілих чисел і дискретного логарифмування. В рамках виставки CES 2019 підрозділ IBM Research анонсував першу в світі квантову систему, що придатна для комерційного застосування. Одним з напрямів постквантової криптографії є розробка алгоритмів обміну ключами на основі ізогеній суперсингулярнирних еліптичних кривих. Поки що було розроблено два алгоритма: SIDH і CSIDH. SIDH був розроблений у 2011 році, в 2016 році Microsoft реалізувала цей алгоритм. На основі SIDH було побудовано постквантовий алгоритм SIKE, що є одним із кандидатів конкурсу NIST для пошуку алгоритмів, ефективних проти квантових обчислень. CSIDH є більш новою розробкою, був опублікований 2018 року.

Метою дипломної роботи є створення методу оцінки захищеності алгоритмів обміну ключами, дослідження існуючих постквантових алгоритмів систем обміну ключами, а саме SIDH і CSIDH; оцінка алгоритмів в контексті розробленого методу. Для досягнення мети необхідно вирішити наступні завдання:

5) проаналізувати існуючи роботи за тематикою дослідження;
6) розробити метод оцінки і класифікації алгоритмів обміну ключами;
7) зробити аналіз SIDH і CSIDH, дослідити їх стійкість, швидкодію, існуючи вразливості;

Об'єктом дослідження є алгоритми обміну ключами, базовані на ізогеній суперсингулярних елиптичних кривих, а саме SIDH і CSIDH.

Предметом дослідження є стійкість алгоритмів на основі ізогеній суперсингулярних елиптичних кривих, можливі атаки і методи захисту від них.

Наукова новизна полягає у вдосконаленому аналізу існуючих алгоритмів обміну ключами для пошуку кращого претендента для подальшого використання в сучасних криптосистемах.

Практичне значення роботи полягає у методі оцінки захищеності криптографічних алгоритмів та в отриманих результатах захищеності алгоритмів обміну ключами, що можуть бути використані для подальшого дослідження у цій галузі.

1 НЕБЕЗПЕКА КВАНТОВИХ ОБЧИСЛЕНЬ. ПОСТКВАНТОВА КРИПТОГРАФІЯ

В розділі наведено теоретичне обґрунтування ефективності квантової криптографії і основи постквантової криптографії.

1.1 Квантові обчислення

Квантові обчислення нерозривно зв’язані з поняттям квантового комп’ютера. За визначенням, квантовий комп’ютер – це обчислювальний пристрій, який використовує явища квантової механіки (квантова суперпозиція, квантова заплутаність) для передачі і обробки даних. Квантовий комп'ютер (на відміну від звичайного) оперує не бітами (здатними приймати значення або 0, або 1), а кубітами, що знаходяться у суперпозиції і 0, і 1 [1]. Теоретично, це дозволяє обробляти всі можливі стани одночасно, досягаючи істотної переваги над звичайними комп'ютерами в ряді алгоритмів. Поки що створення універсального квантового комп’ютера неможливо, на даний момент реалізовані лише одиничні системи, що реалізують лише фіксовані алгоритми не високої складності.

Уперше концепція квантових обчислень була запропонована Юрієм Маніним в 1980 році [2]. Одна з перших моделей квантового комп’ютера була розроблена Річардом Фейнманом в 1981 році [3]. Стівен Візнер в 1983 році опублікував статтю з концепцією квантового комп’ютера незалежно від Фейнмана. 1994 рік став одним з найважливіших років для криптографії. Пітер Шор відкрив алгоритм, що дозволяє квантовим комп’ютерам швидко проводити факторизацію великих цілих чисел [4]. Цей алгоритм дозволяє розв’язати дві важливі задачі, на яких будується більшість класичних систем з відкритим ключем – проблему факторизації великих чисел і дискретного логарифмування.

Якщо застосовувати для розв'язання цих задач класичні комп'ютери, то їх складність досягає субекспоненції та експоненції. Існує багато класичних алгоритмів для вирішення цих задач. Так, найкращим алгоритмом факторизації великих чисел є загальний і спеціальний методи решета числового поля зі субекспоненційною складністю

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\log n)^{\frac{1}{3}}\,(\log\log n)^{\frac{2}{3}}\right)=\ L_n\left[\frac{1}{3},\sqrt[3]{\frac{64}{9}}\right] \qquad (1.1)$$

Але алгоритм Шора дозволяє розв'язати ці математичні задачі з поліноміальною складністю. Він здатен факторизувати число M приблизно за час $O(\log^3 M)$, використовуючи $O(\log M)$логічних кубітів. У 1996 році Лов Гровер винайшов квантовий алгоритм пошуку в базі даних [5], він дозволяє досягнути квадратичного приросту швидкості розрахунків у порівнянні з класичним комп'ютером. Він здатен знайти корінь рівняння $f(x)=1$, де f – це булева функція від n змінних, використовуючи $\frac{\pi}{4}\sqrt{N}$ звертань до функції f, з використанням $O(n)$ кубітів. Це не такий великий приріст у порівнянні з алгоритмом Шора, але алгоритм Гровера можна застосувати к більш великому спектру завдань. Зі створенням квантового комп'ютера, здатного застосовувати квантові алгоритми Шора або Гровера, усі сучасні асиметричні системи не будуть здатні забезпечити необхідну криптографічну стійкість для забезпечення захищеності інформації [6]. У 2001 році у дослідницькому центрі компанії IBM і в Стенфордському університеті вдалося виповнити алгоритм Шора для факторизації 15 на прості числа. Публікація цих двох алгоритмів надала потужного поштовху для досліджень у напрямі постквантової криптографії. Це галузь криптографії, яка буде стійка до криптоаналізу з використанням алгоритмів Шора і Гровера. Вже існують деякі класи криптосистем, стійких до квантових обчислень. З 2006 року проводиться конференція PQCrypto,

присвячена постквантовій криптографії [7], а у 2017 році NIST оголосив конкурс на постквантові криптосистеми [8]. У 2018 почався другий тур, криптосистема SIKE, що базується на SIDH, пройшла у нього.

Розробка квантового комп'ютера активно продовжується в наш час. По найоптимістичнішим прогнозам, у найближчі двадцять років вдасться створити повноцінний універсальний квантовий комп'ютер, що буде здатен застосовувати різні алгоритми для вирішення завдань великої складності. Дослідженнями у сфері квантового комп'ютера займаються багато різних компаній, дослідницьких центрів, наприклад Intel, IBM, Російський квантовий центр, Стенфордський університет та багато інших. Шлях до цього почався ще у 1981 році, коли Томмазо Тофоллі представив вентиль Тоффолі, який став популярним квантовим вентилем для побудови оборотних схем. У 1989 році було продемонстровано роботу двох-кубітного квантового комп'ютера. За наступні два десятиріччя було досягнуто великих успіхів у цьому напрямі і вже в 2017 році Microsoft представив мову квантового програмування Q#. Також у 2017 році в Intel розробили 17-кубітну мікросхему, а в 2018 році число кубітів в мікросхемі виросло до 49. Канадська компанія D-Wave Systems ще з 2007 року заявляла о створенні різних варіантів квантового комп'ютера: від 16-кубітного до 2000-кубітного [9]. Це найбільш вагомий результат у наш час. Але слід пам'ятати, що майже усі розроблені комп'ютери, процесори, мікросхеми поки не є універсальними і створені лише для вузькоспеціалізованих завдань, або навіть лише для однієї задачі. Зараз не є можливим реалізація алгоритмів , заснованих на квантових вентилях. Це означає, що алгоритм Гровера і Шора неможливо виконати на достатній складності для зламу сучасних криптосистем.

1.1.1 Алгоритм Шора

Алгоритм Шора можна поділити на дві частини: класичне зведення розкладання на множники до знаходження періоду деякої функції і квантове знаходження періоду цієї функції. Квантова використовує здатність кубітів приймати кілька значень одночасно і перебувати в стані «квантової заплутаності».

Нехай:

M – число, яке ми хочемо розкласти на множники (воно не повинно бути цілим степенем простого числа);

N – розмір регістра пам'яті, який використовується (без врахування додаткової пам'яті). Бітовий розмір цієї пам'яті $\mathrm{n} = \log_2 N$ приблизно в 2 рази більше розміру M, а саме $M^2 < N = 2^n < 2M^2$;

t – випадковий параметр такий, що $1 < t < M$ і $\gcd(t, M) = 1$, де gcd - найбільший спільний дільник.

t, N, M — фіксовані. В алгоритмі Шора використовується стандартний спосіб зведення задачі факторизації до задачі пошуку періоду r функції від випадково підібраного числа t.

Класична частина.

Мінімальне r таке, що $t^r \equiv 1 \; mod \; M$— це порядок t по модулю M.

Порядок r є періодом функції $f(x) = t^x mod \, M$, де $x = 0, 1, 2, \ldots, N-1$. Якщо можна ефективно обчислити r як функцію від t, то можна знайти власний дільник M за час, обмежений поліномом від $\log_2 M$ з ймовірністю $\geq 1 - M^{-m}$ для будь-якого фіксованого m.

Припустимо, що для даного t період r парний $r \equiv 0 \; mod \; 2$ і задовольняє умові $t^{\frac{r}{2}} \not\equiv -1 \; mod \; M$

Тоді $\gcd\left(t^{\frac{r}{2}} + 1, M\right)$ — власний дільник M. Функція gcd вирішується за поліноміальний час.

Ймовірність виконання цієї умови $\geq 1 - \frac{1}{2^{k-1}}$, де k — число різних непарних простих дільників M, отже, $\geq \frac{1}{2}$ в даному випадку. Тому хороше значення t з ймовірністю $\geq 1 - M^{-m}$ знайдеться за $O(\log M)$ спроб. Найдовше обчислення в одній спробі — обчислення $t^{\frac{r}{2}}$.

Квантова частина.

Для здійснення квантової частини алгоритму необхідна обчислювальна схема, що складається з 2-х квантових регістрів X і Y. Кожен з них складається із сукупності кубітів в нульовому булевому стан$|0\rangle$.і

Регістр X використовується для розміщення аргументів x функції $f(x)$. Регістр Y (допоміжний) використовується для розміщення значень функції $f(x)$ з періодом r, що підлягають обчисленню.

Квантова частина складається з 4 кроків:

- Перший крок. На першому кроці за допомогою операції Уолша – Адамара, яка здійснює перетворення кубіта за допомогою оператора

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.2}$$

первісний стан $|0\rangle$. регістра X перекладається в рівноймовірнісну суперпозицію всіх булевих станів X. Другий регістр Y залишається в стані $|0\rangle$. В результаті виходить наступний стан для системи двох регістрів:

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x, 0\rangle. \tag{1.3}$$

- Другий крок. Нехай U_f – унітарне перетворення, яке $|x,0\rangle$ переводить в$|x,f(x)\rangle$. На другому кроці застосовується унітарне перетворення до системи двох регістрів. Виходить наступний стан системи:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x,0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x,t^x \bmod M\rangle. \tag{1.4}$$

Тобто між станами обох регістрів утворюється певний зв'язок.

- Третій крок. Квантове Фур'є-перетворення є унітарним перетворенням стану квантового регістра, що описується N-мірним вектором стану виду

$$\sum_{x=0}^{N-1} f(x)|x\rangle, \tag{1.5}$$

в інший стан

$$\sum_{k=0}^{N-1} \tilde{f}(k)|k\rangle: \tag{1.6}$$

$$QFT_N: \sum_{x=0}^{N-1} f(x)|x\rangle \Rightarrow \sum_{k=0}^{N-1} \tilde{f}(k)|k\rangle, \tag{1.7}$$

де амплітуда перетворення Фур'є має вигляд.

У двовимірній x,k-площині перетворення Фур'є відповідає повороту осей координат на 90°, яке веде до перетворення шкали x в шкалу k. На третьому кроці станом першого регістра здійснюється перетворення Фур'є і виходить

$$\frac{1}{N}\sum_{x=0}^{N-1}\sum_{k=0}^{N-1}\exp\left(2\pi\, i\frac{kx}{N}\right)|k, t^x\ mod\ M\rangle. \quad (1.8)$$

- Четвертий крок. На четвертому кроці виконується вимірювання першого регістра X щодо ортогональної проекції виду: $|0,0\rangle \otimes I,\ |1,1\rangle \otimes I, \dots, |N-1, N-1\rangle \otimes I$, де I - тотожний оператор на гільбертовому просторі другого регістра Y. В результаті виходить $|k, t^k\ mod\ M\ \rangle$ з ймовірністю

$$\left|\frac{1}{N}\sum_{x:t^x \equiv t^k\ mod\ M}\exp\left(2\pi\, i\frac{kx}{N}\right)\right|^2. \quad (1.9)$$

- На той частині прогону, що залишилась, працює класичний комп'ютер:

  - Знаходиться найкраще наближення (знизу) до $\frac{k}{N}$ зі знаменником $r` < M < \sqrt{N}$:

$$\left|\frac{k}{N} - \frac{d`}{r`}\right| < \frac{1}{2N}. \quad (1.10)$$

  - Пробуємо $r`$ в ролі r:
    - Якщо $r` \equiv 0\ mod\ 2$, то слід обчислити $\gcd(t^{\frac{r`}{2}} \pm 1, M)$.
    - Якщо $r`$ непарне, або якщо $r`$ парне, але власний дільник M невиявлений, то слід повторити прогін $O(\log\log M)$ раз з тим же самим t. У разі невдачі, необхідно змінити t і почати новий прогін алгоритму.

Для визначення періоду r функції $f(x)$ не потрібно обчислювати всі значення. Нехай F – функція з невідомим періодом r:

$$F : |x,0\rangle \rightarrow |x,f(x)\rangle\, f: Z \rightarrow Z_{2^m}\, r < 2^n. \tag{1.11}$$

Щоб визначити період r, потрібні два регістра з розмірами $2n$ і m кубітів, які спочатку повинні бути в стані $|x,0\rangle$. На першому етапі виконується одностороння суперпозиція всіх базисних векторів першого регістра з використанням оператора U наступного вигляду:

$$U\,|0,0\rangle = \sum_{i=0}^{N-1} c_i|i,0\rangle \text{ , де } |c_i| = \frac{1}{\sqrt{N}} \text{ і } N = 2^{2n} \tag{1.12}$$

Тут використовується псевдо перетворення Адамара $H_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$. Застосувавши F до поточного стану, отримуємо:

$$|\psi\rangle = F\, H_1\, |0,0\rangle = F\frac{1}{2^n}\sum_{i=0}^{N-1}|i,f(i)\rangle. \tag{1.13}$$

Вимірювання другого регістра з результатом $k = f(s)$, де $s < r$, призводить стан до

$$|\psi`\rangle = \sum_{j=0}^{[N/r]-1} c`_j|rj+s,k\rangle, \qquad \text{де } c`_j = \left[\frac{N}{r}\right]^{-1/2} \tag{1.14}$$

Після вимірювання стану $|\psi`\rangle$ перший регістр складається тільки за базисних векторів виду $|rj+s\rangle$ таких, що $f(rj+s) = f(s)$ для всіх j. Тому він

має дискретний однорідний спектр. Неможливо прямо отримати період r або кратне йому число, вимірюючи перший регістр, тому що тут s – випадкова величина. Тут застосовується дискретне перетворення Фур'є виду

$$DFT : |x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i}{N} xy} |y\rangle \tag{1.15}$$

на регістр, так як ймовірність спектра в перетвореному стані інваріантна щодо зміщення (перетворюються тільки фази, а не абсолютні значення амплітуд).

$$|\psi``\rangle = DFT\ |\psi`\rangle = \sum_{i=0}^{N-1} c``_i\ |i, k\rangle. \tag{1.16}$$

$$c``_i = \frac{\sqrt{r}}{N} \sum_{j=0}^{p-1} \exp\left(\frac{2\pi i}{N} i(\,jr + s)\right) = \frac{\sqrt{r}}{N} e^{\varphi_i} \sum_{j=0}^{p-1} \exp\left(\frac{2\pi i}{N} ijr\right) \tag{1.17}$$

де $\varphi_i = 2\,\pi i \frac{is}{N}$ і $p = \left[\frac{N}{r}\right]$.

Якщо $N = 2^{2n}$ кратне r, тоді $c``_i = \frac{e^{\varphi_i}}{\sqrt{r}}$, якщо $i$ кратне $\frac{N}{r}$, і $c``_i = 0$ в іншому випадку. Навіть якщо r не є ступенем числа 2, то спектр $|\psi``\rangle$ показує окремі піки з періодом $\frac{N}{r}$, бо

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} e^{2\pi i k\alpha} = \begin{cases} 1, & \alpha \in Z \\ 0, & \alpha \notin Z \end{cases} \tag{1.18}$$

Для першого регістра використовується $2n$ кубітів, коли $r < 2^n$, бо це гарантує принаймні 2^n елементів в наведеній сумі, і таким чином ширина піків буде порядку $O(1)$. Якщо тепер обчислити перший регістр, то вийде значення c,

близьке до $\frac{\lambda N}{r}$, де $\lambda \epsilon Z_r$. Воно може бути записано як $\frac{c}{N} = c * 2^{-2n} \approx \frac{\lambda}{r}$. Це зводиться до знаходження апроксимації $\frac{a}{b}$, де $a, b < 2^{-2n}$, для конкретного числа двійкової крапки $c * 2^{-2n}$. Для вирішення цієї задачі використовуються ланцюгові дроби. Оскільки форма раціонального числа не єдина в своєму роді, то λ і r визначаються як $\frac{a}{b} = \frac{\lambda}{r}$, якщо $\gcd(\lambda, r) = 1$. Імовірність того, що обидва числа λ і r прості, більше ніж $\frac{1}{\ln r}$, тому, для наближення ймовірності успіху до одиниці необхідно лише спроб $O(n)$.

1.1.2 Емулятор квантових обчислень

Хоча на сьогоднішній день доступ для обчислень за допомогою існуючих реалізацій квантових комп'ютерів доступний лише для дослідників цієї області, було створено достатньо велику кількість емуляторів квантових обчислень, які можуть симулювати квантові обчислення. При цьому вони реалізовані на класичних комп'ютерах. Можна навести точний перелік цих емуляторів:

- Microsoft Quantum Development Kit;
- Microsoft LIQUi>;
- IBM Quantum Experience;
- Qiskit;
- Quantum Computing Playground (Google);
- Rigetti;
- Forest;
- ProjectQ;
- QuTiP;
- OpenFermion;
- Qbsolv;
- ScaffCC;
- Raytheon;

- BBN;
- Quirk Simulator;

Для демонстрації квантових обчислень було обрано емулятор Quantum Computing Playground від Google [37]. Це перший браузерний емулятор квантових обчислень, може емулювати до 22 кубітів, використовуючи можливості WebGL і GPU, програми працюють на мові QScript, що не сильно відрізняється від існуючих мов квантового програмування. Емулятор працює у браузерах Chrome і Firefox.

Було обрано алгоритм Шора (додаток А) і за його допомогою факторизовано три числа. Це 15, 21 і 55. Для роботи алгоритму необхідно $O(\log M)$ логічних кубітів за час $O(\log^3 M)$, де M – число. Тобто потужності емулятора достатньо. Також треба враховувати ймовірнісний характер роботи квантових алгоритмів. Для кожного числа було застосовано алгоритм 1000 разів. У наступній таблиці наведено кількість успішних завершень роботи алгоритму.

Таблиця 1.1 – Емуляція алгоритму Шора

Число	Кількість успіхів	Кількість кубітів
15	875	4
21	927	5
55	971	6

Як можна побачити з отриманих даних, кількість успіхів зростає з числом використовуваних кубітів. Застосування більшого числа кубітів було обмежено можливостями використовуваного GPU.

1.1.3 Порівняння складності класичних і квантових алгоритмів

Проведемо порівняння складності розв'язання задач факторизації і дискретного логарифмування на еліптичній кривій за допомогою класичних алгоритмів і квантових алгоритмів. При обчисленні складності задачі факторизації для класичного комп'ютера прийнята складніть загального методу числового поля, для квантового – модифікація алгоритму Шора. Складність дискретного логарифмування на еліптичній кривій: для класичного комп'ютера - алгоритм ρ-методу Полларда, для квантового – модифікація схеми Борегарду. Дані взяти з [10].

Таблиця 1.2 – Порівняння складності задачі факторизації

Квантовий комп'ютер			Класичний комп'ютер
Квантовий алгоритм факторизації, модифікація алгоритму Шора			Загальний метод решета числового поля (GNFS)
довжина ключа, біт	розмір регістру, кубіт	кількість квантових операцій	кількість класичних операцій
l	$2l$	$4l^3$	$L_n\left[\frac{1}{3}; \sqrt[3]{\frac{64}{9}}\right]$
512	1024	$0{,}54 * 10^9$	$2{,}96 * 10^{11}$
1024	2048	$4{,}3 * 10^9$	$5{,}61 * 10^{15}$
2048	4096	$34 * 10^9$	$2{,}58 * 10^{21}$
3072	6144	$120 * 10^9$	$3{,}40 * 10^{23}$
15360	30720	$1{,}5 * 10^{13}$	$1{,}87 * 10^{50}$

Таблиця 1.3 – Порівняння складності задачі дискретного логарифмування

Квантовий комп'ютер			Класичний комп'ютер
Квантовий алгоритм ECDLP, модифікація схеми Борегарду			Алгоритм ρ-методу Полларда
ключ, біт	розмір регістру, кубіт	кількість квантових операцій	кількість класичних операцій
l	$5l + 8l^{\frac{1}{2}} + 2\log_2 l + 10$	$360\ l^3$	$(\pi\ 2^l)^{\frac{1}{2}}$
110	657,47	$0{,}48 * 10^9$	$6{,}39 * 10^{16}$
163	941,84	$1{,}56 * 10^9$	$6{,}06 * 10^{24}$
224	1256	$4{,}05 * 10^9$	$9{,}20 * 10^{33}$
256	1434	$6{,}04 * 10^9$	$6{,}03 * 10^{38}$
512	2769	$48{,}32 * 10^9$	$2{,}05 * 10^{77}$

Порівняння табл. 1 та 2 свідчить про те, що для еквівалентних по складності для класичного комп'ютеру задач факторизації та ECDLP, квантове рішення задачі ECDLP потребує менших ресурсів (як кількості кубітів, так й квантового часу), у порівнянні з рішенням задачі факторизації. Різниця об'ємів потрібних ресурсів зростає в залежності від збільшення класичної складності. Криптосистеми, побудовані на задачі факторизації і електронні цифрові підписи на основі ECDLP, будуть зламані за поліноміальний час.

1.1.4 Напрямки розробки у сфері постквантової криптографії

Постквантова криптографія – це частина криптографії, що залишиться актуальною при появі квантового комп'ютера і квантових атак. Криптоаналітики

почали пошук стійких систем серед існуючих і розробку абсолютно нових методів побудов криптосистем. Початком роботи у цьому напрямі можна вважати публікацію алгоритма Шора, але зазвичай ще приймають конференцію PostQuantumCrypto у 2006 році [7], цей захід проводиться щорічно для представлення нових розробок у постквантовій криптографії.

Сучасна постквантова криптографія заснована на п’яти різних підходах, захищаючих від квантових атак [10]:

1) Криптографія на основі хеш-функцій

Класичним прикладом є підпис Меркла з відкритим ключем на основі хеш-дерева [14], був запропонований ще у 1979, але не використовувався через обмеження на кількість підписів.

2) Криптографія на основі кодів виправлення помилок

Система McElice [15], не отримала використання через ряд недоліків, що ускладнює практичну реалізацію і використання.

3) Криптографія на основі решіток

Перший алгоритм протидії квантовим атакам на основі решіток NTRU [16] був представлений ще у 1996 році, існують і більш сучасні алгоритми.

4) Криптографія на основі багатовимірних квадратичних систем

Однією з найцікавіших схем є підпис з відкритим ключем Жака Патаріна HFE, запропонована ним у 1996 році, як узагальнення пропозицій Matsumoto і Imai [18].

5) Симетрична криптографія

Прикладом стійкого к квантовим атакам алгоритму є шифр Rijndael, запропонований в 1998 році, згодом перейменований в AES [17].

6) Шифрування з використанням суперсингулярних ізогеній

Це аналог протоколу Діффі-Геллмана [36], заснований на блуканні в суперсінгулярном графі ізогеній, що дозволяє двом і більш сторонам отримати загальний секретний ключ, використовуючи незахищений від прослуховування канал зв'язку.

В даній роботі ми зупиняємося на криптографії, заснованій на суперсингулярних ізогеній, дослідимо переваги і недоліки існуючих алгоритмів, можливі атаки і методи захисту від них.

1.2 Еліптична криптографія

Дослідження еліптичних кривих почались ще у III сторіччі до н.е. грецьким вченим Діофантом. З того часу у цій галузі математики працювали такі дослідники, як І. Ньютон, К. Вейєрштрас, К. Гаус. Лише в XIX сторіччі еліптичні криві знаходять своє застосування в теорії елиптичних функцій, які, в свою чергу пов’язані з еліптичними інтегралами. А використання еліптичних кривих для створення криптосистем було запропоновано Нілом Кобліцем і Віктором Міллером у 1985 році, незалежно один від одного. Так і виник новий розділ криптографії – еліптична криптографія, що вивчає асиметрічні криптосистеми, засновані на еліптичних кривих над скінченими полями [7-11]. Основна перевага еліптичної криптографії полягає в тому, що на сьогоднішній день не існує субекспоненціальних алгоритмів розв'язання задачі дискретного логарифмування.

1.2.1 Еліптичні криві. Суперсингулярні еліптичні криві. Ізоморфізм еліптичних кривих. Кільця

Введемо загальне визначення еліптичної кривої.

Визначення 1.1. Еліптичною кривою E називається множина точок (x, y), що задовольняють рівнянню:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1.19)$$

Це рівняння можна розглядати над довільними полями, але в криптографії особий інтерес представляють рівняння над скінченими полями, а саме над простими полями непарної характеристики ($\mathbb{Z}_p$, де $p > 3$ просте число) і полями характеристики 2 ($GF(2^m)$). Необхідно визначити такі криві.

Визначення 1.2. Еліптичною кривою E в формі Вейєрштраса над скінченим полем F_q , $q = p^n$, $p > 3$ ми називаємо множину точок (x, y); $x, y \in F_q$, що задовольняють рівняння:

$$E: y^2 = x^3 + ax + b;\ a, b \in F_q \tag{1.20}$$

У даному випадку рівняння залежить лише від двох параметрів, це більш вузький вигляд рівняння (1.18).

Також, визначення еліптичної кривої потребує, щоб крива не мала особливих точок. Геометрично це означає, що графік не повинен мати самоперетинів. Криві, що мають самоперетини, називаються сингулярними і не мають практичного застосування в криптографії. Несингулярними є криві без самоперетинів. Визначити вид кривої можна за допомогою дискримінанта кривої $\Delta = -16(4a^3 + 27b^2)$. Якщо дискримінант дорівнює 0, то ця крива є сингулярному, у іншому випадку – несингулярною.

На множині точок еліптичної кривої в формі Вейєрштраса над полем F_q можна ввести груповий закон з операцією додавання в афінних координатах. Нехай: $P = (x_1, y_1), Q = (x_2, y_2), x_1 \neq x_2, s = \frac{y_1 - y_2}{x_1 - x_2}$. s суворо визначено, тому що F_q – поле. Нейтральний елемент групи точок еліптичної кривої в формі Вейєрштраса - точка в нескінченності: O. Точка є винятковою, оскільки її немає на афінній площині над полем F_q , але її можна зобразити точкою на проективній площині. Операцію додавання можна визначити як

$$R = (x_3, y_3) = P + Q \tag{1.21}$$

$$x_3 = s^2 - x_1 - x_2, \tag{1.22}$$

$$y_3 = -y_1 + s(x_1 - x_3). \tag{1.23}$$

Якщо $x_1 = x_2$, то є два варіанта. Якщо $y_1 = -y_2$, то сума визначається як 0; зворотну точку до будь-якої точки на кривій можна знайти, відобразив її відносно осі Ox. Якщо $y_1 = y_2 \neq 0$, то $R = P + P = 2P = (x_3, y_3)$ визначається так:

$$s = \frac{3x_1^2 + a}{2y_p}, \tag{1.24}$$

$$x_3 = s^2 - 2x_1, \tag{1.25}$$

$$y_3 = -y_1 + s(x_1 - x_3). \tag{1.26}$$

У випадку $y_1 = y_2 = 0 \Rightarrow P + P = O$.

Зворотний елемент до точки P, визначаєм як $-P$, такий, що $P + (-P) = 0$. Якщо координата y_1 точки $P = (x_1, y_1)$ не дорівнює 0, то $-P = (x_1, -y_1)$. У випадку $y_1 = 0 \Rightarrow -P = (x_1, y_1) = P = (x_1, y_1)$. Коли $P = O$ – точка на нескінченності, то і $-P = O$.

Скалярний добуток $Q = nP$, де n ціле, визначається як $Q = \sum_n P$ (за умови $(n > 0)$. Якщо $n < 0$, то Q це зворотний елемент до $|n|P$. Якщо $n = 0$, то $Q = 0 * P = O$.

Одна з проблем, що виникає при використанні формул групового закону як при великій, так і при парній характеристиці поля, пов'язана з необхідністю ділення. Ділення в скінченому поле вважається дорогою операцією, так як включає в себе варіант розширеного алгоритму Евкліда, який хоча і має приблизно ту ж складність, що і множення, проте зазвичай не може бути реалізований досить ефективно. Щоб уникнути операції ділення застосовують проективні координати.

Рівняння кривої в проективних координатах записується через три координати (X, Y, Z) замість двох (x, y) і має вигляд:

$$E:\ Y^2Z = X^3 + aXZ^2 + bXZ^2 \tag{1.27}$$

Точка на нескінченності має координати $(0, 1, 0)$. Між проективними і афінними координатами існує зв'язок:

$$x = \frac{X}{Z}, \qquad y = \frac{Y}{Z}. \tag{1.28}$$

Структура групи точок еліптичної кривої E над F_q – алгебраїчним замиканням поля визначається згідно з теоремою.

Теорема 1.1. Якщо m не ділиться на характеристику поля $p = char(F_q)$, група точок порядку m має наступну структуру:

$$E\ [m] =\ Z_m \times Z_m \tag{1.29}$$

Кількість F_q раціональних точок над еліптичною кривою скінченна. Визначимо його $\#E(F_q) =\ N_E$. Очікуване число точок привої близько к $q + 1$. У сучасній еліптичній криптографії важливу роль грають еліптичні криві майже простого порядку $N_E = kp$, де p – просте, а k – малий кофактор, тому обчислення порядку групи раціональних точок кривої є важливою задачею. Майже простий порядок кривої при побудові криптосистеми на еліптичних кривих дозволяє базуватись на задачі дискретного логарифмування в циклічній групі точок. Найкращий класичний алгоритм ρ-методу Полларда для дискретного логарифмування має складність $(\pi\ 2^l)^{\frac{1}{2}}$. Для визначення порядку точок еліптичної кривої користуються границею Гассе.

Теорема 1.2. Границя Гассе. Порядок групи точок еліптичної кривої $N_E = \#E(F_q)$ над полем F_q задовольняє наступну нерівність:

$$|q + 1 - N_E| \leq 2\sqrt{q} \tag{1.30}$$

Також це значення можна прийняти як $N_E + q + 1 = t$. Параметр t називається слідом відображення Фробеніуса над полем. Обчислення сліду Фробеніуса є ключовою задачею для визначення однозначного значення порядку кривої над полем. У 1985 вчений Рене Шуф опублікував перший поліноміальний алгоритм підрахунку кількості точок еліптичної кривої над скінченим полем. Використання швидких операцій з многочленами і арифметики цілих чисел дозволило скоротити складність алгоритму до $O(\log^5 q)$. В 1990-х роках Елкіс і Аткін модифікували алгоритм Шуфа і презентували алгоритм Шуфа-Елкіса-Аткіна, що зменшив складність обчислення до $O(\log^4 q)$.

Визначення 1.3. Крива $E(F_q)$ називається суперсингулярною, якщо характеристика p поля F_q ділить слід відображення Фробеніуса t.

При $p = q$ суперсингулярна крива нараховує $p + 1$ точок, оскільки $t = 0$ в такому разі. Якщо ж $q = p^f$, то t у суперсингулярних кривих може приймати значення

при непарному f: $t = 0, t^2 = 2q$ і $t^2 = 3q$;

при парному f: $t^2 = 4q, t^2 = q$, якщо $p = 1\ mod\ 3$; і $t = 0$, якщо $p \neq 1\ mod 4$.

Зазвичай в класичній криптографії суперсингулярні криві не використовуються, оскільки вважаються більш слабим класом кривих, менш стійких до так званих MOV-атак, але для постквантової криптографії вони є корисними, оскільки кільця ендоморфізмів в них мають більш складну структуру і є некомутативними, це робить їх менш вразливими до квантових атак.

Над еліптичними кривими (як і над іншими алгебраїчними структурами) можна ввести визначення ізоморфізму.

Визначення 1.4. Еліптична крива E_1 називається ізоморфною над полем F_q кривій E_2, якщо існує деяке раціональне перетворення координат φ', що дозволяє перетворити рівняння кривої E_1 до рівняння E_2.

Визначення ізоморфізму тісно пов'язане з поняттям j-інваріанта.

$$j = 1728\,\frac{4a^3}{4a^3 + 27b^2} \tag{1.31}$$

Ізоморфізм еліптичних кривих є класом еквівалентності, j-інваріант поділяє класи еквівалентності цього відношення над алгебраїчним замиканням поля. Ізоморфні над полем F_q криві мають той самий j–інваріант. З іншого боку, будь-які криві з співпадаючими j–інваріантами ізоморфні над алгебраїчним замиканням поля. Але криві не обов'язково можуть бути ізоморфні над основним полем.

Якщо при множенні точки P кривої E на число n виходить точка на нескінченності, то така точка називається точкою $n-$кручення. Підгруппа $n-$кручення $E[n]$:

$$E[n] = \{P \in E : n * P = O\} \tag{1.32}$$

Ця підгруппа складається з точки на нескінченності і усіх точок кручення. Якщо порядок точки ділить без залишку, то така точка буде точкою кручення.

Задаймо ідеал порядку $\mathcal{O}$. Нехай $K = F_{p^2}$, $\mathcal{O} \subseteq K$. Норма ідеалу $\mathcal{O}$ визначається як $N(\alpha) = |\mathcal{O}/\alpha|$, де $\alpha \subseteq \mathcal{O}$. У нормі виконується властивість мультиплікативності. Підмодуль-$\mathcal{O}$ в K виду $\mathbb{a}\alpha$ (де $\mathbb{a} \in K^*, \alpha$ - ідеал) називається дробовим ідеалом. Якщо існує β таке, що $\alpha\beta = \mathcal{O}$, то дробовий ідеал є зворотним. Усі головні дробові ідеали є зворотними, множина головних дробових ідеалів $P(\mathcal{O})$. Множина зворотних дробових ідеалів $I(\mathcal{O})$ утворює абелеву групу над ідеалом множення. Визначемо клас ідеалів

$$cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O}) \tag{1.33}$$

1.2.2 Ізогенії еліптичних кривих

Визначення 1.5. Многочленом на еліптичній кривій E/F_q називається елемент кільця $F_q[X,Y]/(Y^2 - X^3 - aX - b)$.

Визначення 1.6. Раціональною функцією на еліптичній кривій E над полем F_q називається функція виду $\frac{f}{g}$, де f і g многочлени на кривій E/F_q.

Визначення 1.5. Ізогенією еліптичної кривої E_1 на еліптичну криву E_2 називається пара функцій (r,s), раціональних на E і задовольняючих умовам:

$$s^2 = r^3 + a'r + b', \tag{1.34}$$

$$(r(O)\,, s(O)\,) = O'. \tag{1.35}$$

Ізогенія ψ: $E_1 \rightarrow E_2$ задає гомоморфізм груп точок $E_1(F_q) \rightarrow E_2(F_q)$.

Ізогенія, що відображає всі точки еліптичної кривої E в нескінченно віддалену точку O_2 кривої E_2, називається нульовою. У подальшому, під ізогенією будемо розуміти ненульову ізогенію.

Можемо ввести відношення еквівалентності під назвою ізогенність на множині всіх еліптичних кривих Ell_p над полем F_q. Дві криві є ізогенними, якщо між ними існує ізогенія. Всі еліптичні криві над полем можна поділити на класи ізогенних кривих. Важливою характеристикою класу можна назвати кардинальність раціональних груп точок його представників. Певний клас ізогенних кривих можна зобразити у вигляді неорієнтованого мультиграфу, вершинами якого будуть класи ізоморфних еліптичних кривих, а ребрами - ізогенії між ними. Представлення класу ізогенних кривих як графу ізогеній

малого порядку важливо для алгоритмів SIDH і CSIDH. На задачі пошуку шляху між двома вершинами, або пошуку ядра ці алгоритми і будуть базуватися.

Теорема 1.3.(Тате) Еліптичні криві E_1/F_q і E_2/F_q ізогенні тоді і лише тоді, коли мають однакове число раціональних точок $\#E_1(F_q) = \#E_2(F_q)$.

Теорема 1.4. Нехай $\psi: E_1 \rightarrow E_2$ – ізогенія. Тоді існує дуальна до неї ізогенія $\psi': E_2 \rightarrow E$, така, що

$$\psi \cdot \psi' = l, \tag{1.36}$$

де $l \in N$. Це означає, що композиція відображень $\psi'(\psi(P))$ еквівалентна множенню точки P на число l. Параметр l будемо називати порядком ізогенії ψ.

Ізогенія, що переходить сама у себе є ендоморфною. Для еліптичної кривої E позначимо $End(E)$, для безлічі ендоморфізмів $\varphi: E \rightarrow E$, включаючи нульовий морфізм. На ендоморфізмі можливо визначити операції додавання $(\varphi + \phi)(P) = \varphi(P) + \phi(P)$ і множення $(\varphi * \phi)(P) = \varphi(P) * \phi(P)$, це і надає кільцеву структуру. Ендоморфізмом Фробеніуса π називається ендоморфізм комутативного кільця простої характеристики, задається формулою $x \rightarrow x^p$.

Суперсингулярні криві однакового порядку мають масивні кільця ендоморфізмів, що робить їх корисними для використання в криптографічних цілях. Для суперсингулярних кривих з порядком існує граф ізогеній з порядком i, який є $(i+1)$-регулярним, тобто таким, у якому кожна вершина має $(l+1)$ суміжне ребро. Це полегшує алгоритми на графах ізогеніях і ускладнює криптоаналіз.

Визначення 1.6. Ядром $\ker\psi$ ізогенії ψ називається множина точок, що відображуються ізогенією ψ в нескінченно віддалену точку O' кривої.

Теорема 1.5. $\#\ker\psi = l$.

Задачу пошуку ізогенних кривих і відображень в них вирішив математик Велу в 1971 році. На вході він приймає криву $E_1: y^2 = x^3 + ax + b$ і одну з її

підгруп C. На виході отримуємо криву $E_2: y^2 = x^3 + a'x + b'$, що є ізогенна E_1і раціональне відображення $\left(\frac{f_1(x,y)}{f_{2(x,y)}}, \frac{g_1(x,y)}{g_{2(x,y)}}\right)$. Складність такого алгоритму складає $O(\#C)$.

В алгоритмі Велу є деякі складності: він працює в циклі, поки не пройдеться по усім точкам з множини $S = C_2 \cup R_+$, де C_2 – множина точок парного порядку з C. Це означає, що кількість кроків буде більше половини порядку ядра ізогенії $\ker \psi$ і менше порядку C. Алгоритм може використовувати тільки відносно маленькі підгрупи, щоб можна було в прийнятний час пройти всі кроки. Це ускладнює криптографічне застосування ізогеній. Але є метод, що дозволяє швидко обчислювати ізогенії, навіть якщо група C має великий порядок $\#C$, за умови якщо ступінь $\#C$ невеликого числа l, тобто l^e. В такому разі алгоритм складається з e кроків, на кожному з яких обчислюється ізогенія ступеню l і один скалярний добуток точки на число.

Формули Велу говорять про те, що ізогенії визначаються виключно її ядром. У випадку Аліси існує $3 * 2^{n-1}$ вибору ядер, і загальна кількість варіантів для (a_1, a_2) становить близько 2^{2n}, так що будуть приватні ключі, які відповідають таким же відкритим ключам. Визначимо відношення еквівалентності приватних ключів $(a_1, a_2) \sim (a'_1, a'_2)$, якщо два ключі ведуть до однієї підгрупи для всіх можливих точок входу. Співвідношення виконується $(a'_1, a'_2) = (\theta_{a1}, \theta_{a2})$ для будь $\theta \in \mathbb{Z}_{2^n}^*$, і тому клас еквівалентності є точкою в проективному просторі над кільцем. Визначимо цей клас еквівалентності за допомогою нормалізації. Нехай $P, Q \in E[2^n]$ є лінійно незалежними генераторами $E[2^n]$. Тоді для деяких $(a_1, a_2) \in \mathbb{Z}^2$ виконується $(a_1, a_2) \sim (1, \alpha)$ або $(a_1, a_2) \sim (\alpha, 1)$ для деякого $\alpha \in \mathbb{Z}$.

Висновки до розділу 1

В розділі розглянуто небезпеку квантової криптографії для сучасної криптографії, зроблено порівняння класичних і квантових алгоритмів для вирішення задач факторизації і ECDLP. Також уведено і розглянуто основні математичні поняття для побудови постквантових алгоритмів обміну ключами SIDH і CSIDH.

2 ПОБУДОВА МЕТОДУ ОЦІНКИ СТІЙКОСТІ АЛГОРИТМУ ОБМІНУ КЛЮЧАМИ

У цьому розділі опис методу оцінки стійкості алгоритмів обміну ключами на основі загальних характеристик алгоритму і ступеню небезпеки існуючих атак.

2.1 Складові методу

Для оцінки захищеності і стійкості алгоритму необхідно скласти чітку методологію, по якій можна порівняти між собою алгоритми. На сьогоднішній день не існує чітко спеціалізованої методології оцінювання загального рівню надійності і захищеності алгоритмів обміну ключами. Рівні криптографічної стійкості NIST враховують лише загальний рівень стійкості, не приймаючи до уваги наявні вразливості алгоритмів. Також існує двох класова класифікація стійкості алгоритмів: абсолютно стійкі алгоритми і практично стійкі алгоритми. До першого класу відносяться криптосистеми, які не можуть бути зламані ні теоретично, ні практично, за умови необмежених великих обчислювальних можливостях. Зараз існує лише один шифр, що підтримує стійкість цього рівню – це шифр Вернама. Усі інші алгоритми відносяться до класу практично стійких, це означає, що потенційна можливість зламати шифр існує. Для оцінки стійкості таких алгоритмів рекомендується враховувати такі фактори: доказова стійкість алгоритму, обчислювальна складність повного перебору і відомі на даний момент вразливості.

Для покращення методу оцінки стійкості були враховані існуючі класифікації загроз безпеки, а саме OWASP, CWE, CAPEC і WASC [18-22]. У тому вигляді, у якому зараз з себе представляють ці методології, неможливо застосувати їх для класифікації і оцінки стійкості криптографічного алгоритму:
- OWASP і WASC оцінює загрози для веб-застосунків, CWE і CAPEC

представляє собою більш загальну класифікацію загроз. У контексті алгоритмів обміну ключами необхідно врахувати особливості, що властиві саме для них. У загальному вигляді метод класифікації і оцінки алгоритму можна поділити на дві частини: критерії оцінювання існуючих атак на алгоритм і загальні характеристики алгоритму.

2.2 Критерії класифікації і оцінки існуючих атак.

Усі атаки на алгоритм потрібно розглядати незалежно один від одної. Може виникнути питання, навіщо це робити, якщо у однієї атаки може бути ефективність більша, ніж у інших. Щоб відповісти на це питання, необхідно прийняти до уваги всі критерії, за якими можна оцінити атаку. Одні атаки можуть більш ефективними, ніж інші, але вимагатимуть особливих умов або будуть спрямовані на одну з вразливостей алгоритму. Ця вразливість може бути пізніше виправлена і атака стане не ефективною на практиці.

При розробці і виборі критерії класифікації і оцінки існуючих атак необхідно відмітити такі особливості: не за всім критеріям можна дати кількісну оцінку, кількісні оцінки критеріїв можуть нести приблизний характер, через те, що не існує чіткої існуючої загальної градації для таких критеріїв. Був розроблений так перелік для класифікації і оцінки атак:

- Клас атак, опис атаки (якісна оцінка);
- Успіх атаки (кількісна оцінка S);
- Залежність успіху атаки від використовуваних параметрів алгоритму (кількісна оцінка O_1);
- Необхідні ресурси (якісна оцінка);
- Проведення додаткових дій для реалізації атаки (якісна оцінка);
- Ознаки, за якими можна встановити факт проведення атаки (кількісна оцінка O_2);
- Група зловмисника (якісна оцінка);

- Існуючі методи захисту, їх ефективність (кількісна оцінка E).

Для кількісної оцінки ступеню рівню небезпеки від атаки G використовується формула:

$$G = E * S * \sum_{i=1}^{2} O_i. \quad (2.1)$$

Також, з урахуванням специфіки теми, компетентність зловмисника за замовчуванням є високою, зловмисник добре володіє навичками і знаннями, необхідними для проведення атаки.

2.2.1 Клас атак, опис атаки

Цей критерій відповідає за класифікації атаки до існуючих класів атак, також у ньому наводиться короткий опис, алгоритм атаки. Необхідно відмітити, чому можлива така атака, на яку вразливість алгоритму вона направлена. Також необхідно зазначити, на який з ключових принципів захищеності інформації може бути направлена атака (конфіденційність, доступність або цілісність), якщо це можливо, оцінити ймовірні наслідки при успіху атаки.

2.2.2 Ймовірність успіху атаки

Цей критерій відповідає за оцінку ймовірності успіху атаки. За замовчуванням при розрахунку ймовірності вважається, що атака здійснюється на алгоритм, що використовує стандартні параметри схеми, тобто такі, які рекомендували або використовували розробники алгоритму. На цей критерій також впливає складність проведення додаткових дій, тобто оцінка виражає рівень ймовірності успіху атаки.

Для частини алгоритмів можна привести точне значення ймовірності. Але була також розроблена кількісна градації оцінки $O_1 \in \{0, 1, 2, 3, 4\}$. Кожному значенню відповідає умова.

0 – Ймовірність успіху атаки дуже низька; успіх атаки дуже сильно залежить від успіху проведення додаткових дій або їх практичне застосування майже неможливо.

1 – Ймовірність успіху низька; успіх атаки сильно залежить від успіху проведення додаткових дій, їх практичне застосування є можливим, але з дуже великими складнощами або витратами.

2 – Середня ймовірність успіху; успіх атаки не сильно залежить від успіху проведення додаткових дій, їх практичне застосування ускладнює проведення атаки.

3 – Висока ймовірність успіху, успіх атаки слабо залежить від успіху проведення додаткових дій, їх практичне застосування є тривіальним.

4 – Дуже висока ймовірність успіху; успіх атаки не залежить від успіху проведення додаткових дій, успішне завершення атаки не потребує додаткових дій.

2.2.3 Залежність успіху атаки від використовуваних параметрів алгоритму

Для точнішого оцінювання ступеню небезпеки атаки необхідно врахувати можливу залежність успіху атаки від використовуваних параметрів алгоритму. Для цього параметра також існує кількісна оцінка $O_2 \in \{0, 1, 2, 3, 4\}$.

0 – Успіх атаки дуже сильно залежить від використання проблемних параметрів алгоритму, при зміні параметрів неможливо застосувати атаку; дуже мало проблемних параметрів або їх зміна не впливає на працездатність алгоритму.

1 – Успіх атаки сильно залежить від використання проблемних параметрів, зміна параметрів сильно збільшує необхідні обчислювальні або часові ресурси, застосування атаки не вигідне з практичної точки зору; мала кількість проблемних параметрів, їх зміна призводить до незначного ускладнення практичного використання алгоритму.

2 – Між успіхом атаки и використанням проблемних параметрів є пряма залежність, зміна параметрів призводить до значного ускладнення атаки; існує достатня кількість проблемних параметрів, їх зміна призводить до помітного уповільнення роботи алгоритму.

3 – Успіх атаки не сильно залежить від використання проблемних параметрів, зміна параметру призводить до незначного ускладнення процесу атаки; більша частина можливих параметрів є проблемними, їх зміна призводить до значного уповільнення роботи алгоритму.

4 – Успіх атаки не залежить або майже не залежить від використання проблемних параметрів, зміна параметру не привносить помітного ускладнення для атаки; майже усі параметри є проблемними, робота алгоритму з іншими параметрами неможлива або практично не вигідна.

2.2.4 Передумови для проведення атаки

Цей критерій відповідає за додаткові, спеціальні апаратні, людські, тощо ресурси, які необхідні для проведення атаки. Без використання цих ресурсів атака є практично не може бути застосована. Також описується виконання технічних умов, які роблять проведення атаки можливим.

2.2.5 Ознаки, за якими можна встановити факт проведення атаки

Цей критерій відповідає за наявність ознак, за якими можна визначити, що на алгоритм проводиться атака. Цьому критерію відповідає кількісне значення $O_2 \in \{0, 1, 2, 3, 4\}$.

0 – Атака майже завжди може бути виявлена.

1 – Атака може бути виявлена при використанні додаткових методів виявлення атак, їх робота майже не впливає на практичну реалізацію алгоритму.

2 – Атака виявляє себе приблизно у половині випадків.

3 – Атаку можливо виявити лише при використанні додаткових методів виявлення атак, їх використання значно ускладнює можливість практичного застосування алгоритму; може бути виявлена при виконанні деяких рідких умов.

4 – Атака ніяк не виявляє себе, не існує ефективних методів виявлення атаки.

2.2.6 Група зловмисника

Цей критерій відповідає за необхідну належність зловмисника до однієї з існуючих класифікацій для проведення успішної атаки. У контексті можливості проведення атаки необхідно відмітити такі існуючі групи серед усіх:

- Зовнішній зловмисник

Не має доступу до коректних параметрів ключів і до фізичного середовища іншої сторони

- Авторизований зловмисник

Має доступ до коректних параметрів ключів, але не має доступу до фізичного середовища іншої сторони

- Внутрішній зловмисник

Має доступ до коректних параметрів ключів і до фізичного середовища іншої сторони.

За замовчуванням приймається належність зловмисника до першої групи.

2.2.7 Існуючі методи захисту, їх ефективність

Цей критерій відповідає за можливі методи захисту від атак. Якщо це можливо, необхідно привести опис цих методів захисту, їх складність, можливість і реальність практичного застосування. Цьому критерії відповідає кількісне значення $E \in [0, 1]$, яке є загальним коефіцієнтом ступеню небезпеки від атаки. Коефіцієнт виставляється у залежності від ефективності методу захисту, його практичної цінності, складності, впливу на роботу алгоритму. Значення коефіцієнту 1 означає, що не існує ефективних методів захисту або їх практично неможливо реалізувати, 0 означає максимальний рівень захисту від такої атаки.

2.3 Загальні характеристики алгоритму

Для більш якісного і точного оцінювання стійкості алгоритму необхідно розглянути його загальні характеристики і провести деяку класифікацію. Кількісні характеристики алгоритму беруться відповідно до рівню криптостійкості NIST 1. Був складений перелік з таких характеристик і властивостей:

- Довжина ключа, біт

  Довжина ключа є однією з основних характеристик криптостійкості алгоритмів. Для асиметричних систем, до яких і належать досліджувані алгоритми, ключем є так звана ключова пара: приватний і публічний ключ. Для сучасних асиметричних систем, побудованих на задачах факторизації

і дискретного логарифмування, мінімальною безпечною довжиною ключа вважається 1024 біт. Для алгоритмів на еліптичних кривих мінімальними безпечними вважаються ключі від 163 біт.

- Час, необхідний для успішного завершення роботи алгоритму

  Для ефективного використання алгоритму на практиці необхідно щоб у нього була достатньо висока швидкість роботи. Повільна робота алгоритму ускладнює його застосування для великого обсягу даних

- Класична і квантова криптостійкість

  У контексті роботи необхідно розділяти поняття класичної і квантової криптостійкості. Ці характеристики є оцінками ресурсів для зламу задачі, на складності яких вони базуються. Класична стійкість показує рівень захищеності від обчислень на класичному комп’ютері, квантова – від обчислень на квантовому комп’ютері.

- Структура публічних ключів

  У контексті асиметричних криптосистем необхідно розглядати, з яких параметрів складається публічний ключ. Ключ може нести в собі зайву інформацію, яка може бути використана зловмисником для своїх цілей.

- Режим встановлення спільного секрету.

  Існує два режими встановлення спільного секрету: інтерактивний і не інтерактивний. Перший передбачає взаємний обмін повідомленнями для встановлення спільного секрету, другий передбачає передачу інформації лише від однієї сторони до другої. На основі цього можливі деякі класи атак.

Також необхідно зазначити, що алгоритми, що розглядаються відповідають принципам Керкгоффза, рівень стійкості забезпечується не збереженням системи у таємниці, зловмисник знає все про алгоритм, крім приватних ключів.

2.4 Загальний підсумок

Результати аналізу і оцінювання можна представити у вигляді двох таблиць, у першій привести загальні характеристики, у другій підсумок небезпеки атак по окремим критеріям.

Таблиця 2.1 – Загальні характеристики

Назва характеристики, можлива розмірність	Значення характеристики, властивості
Довжина ключа, біт	…
Швидкодія, мс	…
Класична стійкість	…
Квантова стійкість	…
Режим встановлення спільного секрету	…

Таблиця 2.2 – Оцінка атак

	Критерій оцінювання				
	S	O_1	O_2	E	G
Назва атаки №1	…	…	…	…	…
…	…	…	…	…	…
Назва атаки № n	…	…	…	…	…
Загальна сума	$\sum_{i=1}^{n} S_i$	$\sum_{i=1}^{n} O_1^i$	$\sum_{i=1}^{n} O_2^i$	$\sum_{i=1}^{n} E_i$	$\sum_{i=1}^{n} G_i$

За результатами дослідження можна зробити висновок о загальній криптостійкості алгоритмів, існуючих вразливостях, направлених на них атаках, можливих методах захисту, тенденціях залежності ступеню небезпеки атак від визначених критеріїв.

Висновок до розділу 2

Був побудований метод оцінки алгоритмів обміну ключами з урахуванням важливих для цієї галузі характеристик, критеріїв і властивостей. Цей метод може бути застосований до оцінювання і класифікації будь-якого алгоритму обміну ключами, не лише SIDH і CSIDH.

3 ПРОТОКОЛ ДІФФІ-ГЕЛЛМАНА З ВИКОРИСТАННЯМ СУПЕРСИНГУЛЯРНОЇ ІЗОГЕНІЇ

У цьому розділі буде проведено огляд історії розвитку алгоритму SIDH, короткий опис принципу роботи, оцінка стійкості і класифікація в рамках розробленої методології.

3.1 Історія

Ідея створення криптосистеми, базованої на ізогеніях еліптичних кривих уперше з’явилась в роботах Кувейна, Ростовцева і Столбунова [23]. Але їх алгоритми базувались на ординарних кривих, і через свою властивість комутативності вони були вразливі до атаки Чайлдса-Яо-Сухарева [24]. Розробники алгоритму SIDH позбулися цієї проблеми перенісши схему на суперсингулярні криві і використовуючи некомутативні кільця ендоморфізмів. Також частина ідеї взята з ідеї хеш-функції Чарльза, Лаутера і Горена.

Протокол Діффі-Гелмана з використанням суперсингулярної ізогенії – це постквантовий криптографічний алгоритм, що дозволяє двом сторонам отримати спільний секретний ключ, використовуючи незахищений від прослуховування канал зв’язку. Був створений Де Фео, Яо і Плутом в 2011 році [25]. Базується на блуканні в суперсингулярному ізогенному графі, що дозволяє протистояти криптоаналітичній атаці зловмисника, який використовує квантовий комп’ютер. Підтримує цілковито пряму таємність, це означає, що сеансові ключи, які генеруються з використанням довготривалих ключів, не будуть скомпрометовані, якщо один або декілька з цих довготривалих ключів будуть скомпрометовані. У 2016 році було створено програмну реалізацію алгоритму. На базі SIDH можливі схеми розділення таємниці, передачі шифрованих повідомлень, доведення з нульовим розголошенням та незаперечного підпису. Також у на основі цього примітиву було побудовано криптографічний протокол

SIKE, що є одним з кандидатів у конкурсі NIST у якості криптосистеми, стійкої до квантових обчислень.

3.2 Схема алгоритму

На початку сесії у двох сторін обміну є доменні параметри, що відомі першій стороні, другій стороні і зловмиснику. Це початкова суперсингулярна крива E над полем F_{p^2}, де $p = {l^{e_a}}_a * {l^{e_b}}_b * f \pm 1$. f – невелике число таке, щоб p було простим, l_a, l_a – невеликі прості числа. Порядок кривої $E: \#E = ({l^{e_a}}_a * {l^{e_b}}_b * f)^2$. Друга ступінь розширення поля вибрана через те, що всі суперсингулярні криві над полем мають інваріант в полі, тому всі суперсингулярні криві над скінченим полем можуть бути визначені над F_{p^2} і кількість кривих з точністю до ізоморфізму є скінченою. $E[{l^{e_a}}]$ містить ${l_a}^{e_a-1}(l_a + 1)$ циклічних підгруп порядку ${l_a}^{e_a}$. e_a і $e_и$ підбираються такі, щоб ${l^{e_a}}_a = m$ і ${l^{e_b}}_b = n$ були приблизно рівні. Далі обираються точки P_a, Q_a– базис для $E[m]$, тобто за допомогою комбінації $x * P_a + y * Q_a$ можна отримати будь-яку точку $E[n]$, і точки P_b, Q_b.

При обміні кожна сторона повинна згенерувати ізогенію із спільної еліптичної кривої E. Це робиться за допомоги генерації випадкової точки (R_a і R_b), в якій буде ядро їх ізогеній. Базисними векторами будуть точки P_a, Q_a, P_b, Q_b. Використання різних пар точок гарантує, що сторони сгенерували різні, некоммутуючі ізогенії. Випадкова точка в ядрі ізогеній генерується як випадкова лінійна комбінація точок P_a, Q_a і точок P_b, Q_b. Для цього спочатку Аліса генерує випадкові $a_1, a_2: 0 < a_1, a_2 < 2^m$ (a_1, a_2 не кратні 2). a_1, a_2 – це закритий ключ Аліси. Потім обчислює точку в ядрі ізогеній $R_A = a_1 * P_a + a_2 * Q_a$. Використовуючи формулу Велу отримує ізогенію $\varphi A: E \rightarrow E_a = E/\langle R_A \rangle$, після чого перетворює за допомогою ізогенії φA базис Боба на криву $E_a: \varphi A(P_b), \varphi A(Q_b)$. Аналогічні кроки робить Боб.

Отже, у результаті відкритими параметрами є: криві E_a, E_b і точки $\varphi A(P_b), \varphi A(Q_b), \varphi B(P_a), \varphi B(Q_a)$. Закритими параметрами є a_1, a_2, b_1, b_2.

Аліса і Боб обмінюються відкритими ключами. Вони користуються точками, отриманими від іншої сторони $\varphi A(P_b), \varphi A(Q_b), \varphi B(P_a), \varphi B(P_a)$, в якості базиса для ядра їх нової ізогенії. Використовуючи ті ж лінійні коефіцієнти, які брались раніше для генерації випадкової точки a_1, a_2, b_1, b_2, кожен з них генерує точку в ядрі ізогенії $S_{ab} = b_1 * \varphi A(P_b) + b_2 * \varphi A(Q_b)$ і $S_{ba} = a_1 * \varphi B(P_a) + a_2 * \varphi B(Q_a)$, яку вони хочуть отримати. Далі, використовуючи формулу Велу отримують нові ізогенії $E_{ba} = E_a|S_{ba}$, $E_{ab} = E_b|S_{ab}$. Для отримання спільного секрету Аліса обчислює j–інваріант від E_{ab}, а Боб від E_{ba}. j–інваріанти повинні бути однакові.

3.3 Загальні характеристики

Довжина ключа складає 378 біт [25], це відповідає рівню криптостійкості NIST 1 128-біт. Як можна побачити, це приблизно в 2.7 разів менше, ніж ключ такої ж стійкості популярного асиметричного алгоритму RSA. При цьому можлива компресія ключа до 222 біт [26], час компресії складає приблизно 15 мс.

Згідно з представлених розробниками даних, час роботи алгоритму складає приблизно 10 мс. Це хороший результат, цю криптосистему можна віднести до класу швидких криптосистем з відкритим ключем, навіть якщо застосувати операцію компресії ключа.

Найкращим класичним алгоритмом, що розв’язує задачу пошуку ізогеній між кривими, на якій базується стійкість криптосистеми, є алгоритм Дельфс і Гальбрайта [27]. Алгоритм базується на покращеному ρ-методі Полларда і методі зустрічі посередині. Складність алгоритма складає $O\left(p^{\frac{1}{4}}\right)$ бітових

операцій. При стандартному параметру p, який пропонували розробники, складність є експоненційною величиною.

Найкращим квантовим алгоритмом, що вирішує задачу пошуку ізогенії між двома суперсингулярними еліптичними кривими є алгоритм метод пошукового кігтя (Claw-finding) [28]. Цей алгоритм будувався для криптографічного протоколу SIKE, але може застосовуватись і до SIDH. Він використовує властивості квантового обчислення, може застосовуватись разом з такими допоміжними квантовими алгоритмами, як алгоритм Гровера, van Oorschot-Wiener і Tani. Складність алгоритму складає $O\left(p^{\frac{1}{6}}\right)$ операцій.

Розглядаючи публічний ключ алгоритму, необхідно відмітити, що він складається з трьох параметрів: еліптичної кривої і двох точок. Еліптична крива не розкриває ніякої зайвої таємної інформації. Точки несуть в собі зайву інформацію, що може бути корисною для зловмисника і додають зайву вразливість алгоритму, на них може бути направлена атака.

Алгоритм підтримує лише інтерактивний режим встановлення спільного секрету, це надає додаткову вразливість для алгоритму, під час обміну повідомленнями зловмисник може дізнатись таємну інформацію про таємний ключ і може спрямувати на це атаку.

3.4 Існуючі атаки

На алгоритм SIDH існують такі класи атак:

- Активні атаки;
- Адаптивні атаки;
- Атаки на основі вирішення проблеми ізогенності при відомому кільці ендоморфізмів;
- Атаки з ін’єкцією помилок;
- Ізогенна проблема прихованого числа.

3.4.1 Активні атаки

Активні атаки є давно відомим методом атак на криптосистеми із статичним приватним ключем. Нечесний Боб надсилає Алісі свій публічний ключ (E_b, P_b, Q_b). Аліса обчислить ізогенію $\varphi: E \rightarrow E'$ з ядром $\ker\langle [a1] * P_b + [a2] * Q_b \rangle$. Використовуючи знання E', Боб розкриває деяку інформацію про приватний ключ Аліси. Можливими наслідками успішної атаки може бути часткове розкриття конфіденційності. Можливість такої атаки розглядали у [29-30], але без розкриття детальної інформації.

На основі відомої інформації, можна присвоїти критерію успішності атаки значення (для успіху атаки необхідно обчислення корисної інформації від знання E', Задача не є тривіальною) $S = 2$.

Проведення атаки не залежить від можливих проблемних параметрів алгоритму обміну ключами, $O_1 = 4$.

Для атаки не потрібно володіти якимись особливими технічними чи часовими ресурсами. Необхідно, щоб Аліса користувалась статичним приватним ключем.. Атаку можливо ефективно виявити через те, що параметри, які передає нечестний Боб, є не коректними, існують ефективні методи виявлення атаки. $O_2 = 1$.

Для захисту алгоритму від цієї атаки був створений метод перевірки Кірквуда [29]. Він полягає у верифікації правильності параметрів, що надсилає Боб. Після завершення алгоритму отримання спільного ключа проводяться додаткові дії перевірки честності Боба. Перевірка Кірквуда завжди визначає, чи не належить Боб до зловмисників, але вимагає розкриття Бобом свого таємного ключа і проводиться вже після отримання спільного секрету сторонами. $E = 0{,}4$.

Загальна оцінка небезпеки атаки: $G = 0{,}4 * 2 * (4 + 1) = 4$.

Якщо зловмисник належить до групи авторизованих користувачів, степінь небезпеки атаки різко зростає, значення критерію O_2 зростає до 4, методи захисту стають не ефективні. $G = 1 * 2 * (4 + 4) = 16$.

3.4.2 Адаптивні атаки

У контексті цієї атаки Алісу можна представити у моделі оракула, що повертає значення 0 або 1 у відповідь на запит до неї. При кожному запиті зловмисник дізнається лише один біт корисної інформації. Атака направлена на конфіденційність, у випадку успіху Аліса розкриває свій приватний ключ.

Ймовірність успіху основної частини майже стовідсоткова, отримання приватного ключа математично не складне, при стандартних параметрах успішна майже у всіх випадках. Кількість звертань до Аліси складає менше, ніж $n \approx \frac{1}{2}\log_2(p)$ [31]. $S = 4$.

Цей клас атак спрямований на вразливий параметр l в схемі. Він повинен бути простим малим числом, при збільшенні l атака стає не ефективною на практиці, але враховуючи, що підвищення цього параметру ускладнює роботу алгоритму SIDH, можна поставити оцінку критерію $O_1 = 2$.

Необхідно, щоб Аліса користувалася статичним таємним ключем. Для атаки необхідні достатні ресурси для проведення брутфорсу. Можна відмітити, що кількість запитів до Аліси можна зменшити, за умови більш сильного і довгого брутфорсу.

Для визначення, що атака була проведена, потрібні додаткові алгоритми перевірки, вони аналогічні, що й для активних атак. Оцінка критерія $O_2 = 1$.

Методи захисту також співпадають з методами захисту від активних атак, це перевірка Кірквуда. $E = 0{,}4$.

Загальний степінь небезпеки атаки складає: $G = 0{,}4 * 4 * (2 + 1) = 4{,}8$.

При належності зловмисника до другої групи і вище повторюється ситуація з активними атаками. $G = 1 * 4 * (2 + 4) = 24$.

3.4.3 Атаки на основі вирішення проблеми ізогенності при відомому кільці ендоморфізмів (скорочення ПІ)

Ця атака спрямована на пошук коректної ізогенії між еліптичними кривими. Для зламу алгоритму недостатньо знайти будь-яку ізогенію між кривими, необхідно обчислити ізогенію, дуальну ізогенії Аліси. Лише з її допомогою можна коректно завершити алгоритм. В разі успіху атаки зловмисник отримує можливість проводити обмін ключами, видаючи себе за чесного Боба.

Між параметрами алгоритму і ймовірністю успіху є залежність, яку можна виразити формулою $P \approx \max\left(0, 1 - \frac{90}{\pi^2}\frac{l_a^{2n}}{p}\right)$ [31]. При використанні вразливих параметрів ймовірність досягає 100%. При стандартних значеннях параметрів вона складає приблизно 50%. Отже, $S = 2$.

При зміні параметрів проведення атаки значно ускладняється, але ефективність алгоритму також падає, критерій можна оцінити в $O_1 = 2$.

Для проведення атаки не потрібні ніякі додаткові ресурси, усі кроки мають поліноміальний час обчислення. За зовнішніми ознаками неможливо виявити факт проведення атаки. $O_2 = 4$.

Для захисту від атаки рекомендується використовувати невразливі параметри в роботі алгоритму. При цьому збільшується час, необхідний для роботи алгоритму. $E = 0{,}6$.

Загальний степінь небезпеки складає $G = 0{,}6 * 2 * (2 + 4) = 7{,}2$.

Не зовнішньому зловмиснику немає сенсу проводити атаку, так як він вже має можливість коректно проводити алгоритм з Алісою.

3.4.4 Атака з ін’єкцією помилок

Атака з ін’єкцією помилок є розповсюдженим класом атак, який заснований на можливості втручатись в виконання Алісою протоколу, змушуючи її робити помилки в свої розрахунках. У контексті цього алгоритму, під час останньої ітерації обчислення ізогенії вноситься помилка і це змушує Алісу переривати цикл обчислення. Аліса знову починає алгоритм спочатку. Перша сторона знову моделюється, як оракул, що повертає одне з двох можливих значень. С кожного запиту зловмисник відтворює один біт приватного ключа Аліси. Для повного відтворення потрібно зробити близько $2n/\mu = \log_2 p/\mu$, де μ – ймовірність успішної ін’єкції помилки [32]. Для існуючої програмної реалізації SIDH час останньої ітерації є чітко визначеним і ін’єкція помилки не є складною задачею. Складність виражається формулою $O(n)$. $S = 3$.

Успіх атаки майже не залежить від параметрів алгоритму, атака не направлена на вразливі значення параметрів. $O_1 = 4$.

Для успіху атаки потрібен фізичний доступ до апаратного забезпечення Аліси. Тобто зловмиснику необхідно відноситись до групи внутрішніх користувачів. Також потрібно, щоб у Аліси був статичний приватний ключ.

На даний момент поки не створено методів виявлення цієї атаки, до можливих ознак можна віднести подовжену роботу алгоритму. $O_2 = 3$.

Від цієї атаки існує простий метод захисту, він полягає в простій перевірці кількості ітерацій, що зробив алгоритм. Щоб протистояти цим контр мірам, атаку можна вдосконалити, застосовуючи меншу кількість запитів і збільшивши час проведення атаки. Це значно ускладнює ефективність атаки. $E = 0{,}2$.

Загальний степінь небезпеки від атаки складає $G = 0{,}2 * 3 * (4 + 3) = 4{,}2$.

3.4.5 Ізогенна проблема прихованого числа (скорочення ППЧ)

Ця атака відноситься до класу атак по стороннім каналам, коли третя сторона намагається дізнатися спільний секрет, створений Алісою і Бобом. У випадку успіху Єва отримує спільний секрет з Алісою і Бобом і має можливість підмінювати інформацію, якою обмінюються сторони. Аліса модулюється як оракул. Для отримання необхідної часткової інформації для обчислень необхідно лише $O(r)$ звертань, де r це мале, взаємно ціле просте з l число [31]. Але подальші обчислення є не тривіальним завданням. Ймовірність успіху можна оцінити у $S = 1$.

Успіх атаки залежить від вибраного параметру r, який в свою чергу залежить від l. Зміна l дуже сильно вплине на ефективність практичного застосування атаки, але, як вже розглядалось, цей параметр сильно впливає на роботу алгоритму. Тому $O_1 = 2$.

Для проведення цієї атаки не потрібні додаткові ресурси. Методи визначення ідентичні з активною і адаптивною атакою, $O_2 = 1$.

Перевірка Кірквуда захищає від цієї атаки так само, як від активної і адаптивної атак. $E = 0{,}4$.

Загальний степінь небезпеки атаки: $G = 0{,}4 * 1 * (2 + 1) = 1{,}2$.

Якщо зловмисник авторизований користувач, то $G = 1 * 1 * (2 + 4) = 6$

3.5 Загальні підсумки

Таблиця 3.1 – Загальні характеристики SIDH

Назва характеристики, можлива розмірність	Значення характеристики, властивості
Довжина ключа, біт	378 (з компресією 222)
Швидкодія, мс	≈ 10 мс (+ ≈ 15 компресія)
Класична стійкість	$O\left(p^{\frac{1}{4}}\right)$
Квантова стійкість	$O\left(p^{\frac{1}{6}}\right)$
Режим встановлення спільного секрету	Інтерактивний

Таблиця 3.2 – Оцінка атак на SIDH

	Критерій оцінювання				
	S	O_1	O_2	E	G
Активна	2	4	$1/4^*$	$0{,}4/1^*$	$4/16^*$
Адаптивна	4	2	$1/4^*$	$0{,}4/1^*$	$4{,}8/24^*$
ПІ	2	1	4	0,6	7,2
З ін’єкцією помилок	3	1	3	0,2	4,2
ППЧ	1	2	$1/4^*$	$0{,}4/1^*$	$1{,}2/6^*$
Загальна сума (max)	12 (20)	10 (20)	$10/19^*(20)$	$2/3{,}8^*(5)$	21,4/57,4 (160)

З цих даних можна зробити такий висновок: алгоритм може бути ефективно застосований у якості кандидату асиметричної системи через свої характеристики швидкодії і довжини ключа. Він відповідає вимогам криптостійкості за класичним і квантовим показникам. Розглядаючи алгоритм у контексті існуючих атак, можна відмітити, що більшість з них має достатній рівень успішності, алгоритм має сильну тенденцію до появи вразливостей через проблемні параметри, методи виявлення атак працюють на середньому рівні. Майже для усіх атак існує ефективні методи захисту, що сильно знижують степінь небезпеки від них. Але за умови участі внутрішнього зловмисника для трьох з п’яти атак різко зростає рівень небезпеки. Необхідно враховувати такий випадок і розробляти ефективні методи протидії.

Висновок до розділу 3

Було розглянуто алгоритм SIDH в контексті розроблених методів оцінки і класифікації алгоритмів і зроблено висновки щодо загального рівню захищеності.

4 КОМУТАТИВНИЙ ПРОТОКОЛ ДІФФІ-ГЕЛЛМАНА З ВИКОРИСТАННЯМ СУПЕРСИНГУЛЯРНОЇ ІЗОГЕНІЇ

У цьому розділі буде проведено огляд історії розвитку алгоритму CSIDH, короткий опис принципу роботи, оцінка стійкості і класифікація в рамках розробленої методології.

4.1 Історія

Як і у SIDH, першим кроком до створення CSIDH була модель криптосистеми на ізогеніях запропонована Кувейном у 1997 році. Але вона не набула популярності і у 2004 році Ростовцев і Столбунов заново запропонували цю криптосистему. Ідея полягає у неінтерактивному обміні ключами за допомогою комутативності у групі класу ідеалів $cl(\mathcal{O})$. У 2010 році Чайлдсом, Яао та Сухаревим було доведено [24], що вразливість у схемі Кувейна-Ростовцева-Столбунова зводить проблему складності алгоритма до задачі вирішення проблеми абелової прихованої зміни, для якої існує квантовий алгоритм зі складністю $L_q\left[\frac{1}{2}\right]$. Вразливість пов'язана з комутативністю $cl(\mathcal{O})$. Розробники алгоритму SIDH модифікували схему, використовуючи суперсингулярні еліптичні криві, з використанням некомутативності кільця ендоморфізмів. Але у 2018 році був розроблений алгоритм CSIDH [33], який пропонує використовувати підкільце F_p-раціональних ендоморфізмів замість повного кільця ендоморфізмів. Це дозволяє визначити операції над простим полем і повернутися до комутативності. Як і SIDH, CSIDH базується на задачі блукання у графі ізогеній, але не підтримує квантову стійкість через те, що не усуває вразливість Чайдлса-Яо-Сухарева. Більшість існуючих класів атак поки не застосовні на CSIDH, на відміну від SIDH. У 2019 році [34] було вперше запропонована практична реалізація алгоритму, яка завершує роботу за

константний час. На основі алгоритму CSIDH можлива схема розділення таємниці, реалізація передачі шифрованих повідомлень.

4.2 Схема алгоритму

У алгоритмі розглядаються дві ізоморфні криві E, що визначені над простим полем F_p. Визначимо $End_p(E)$ як підкільце кільця ендоморфізмів $End(E)$. У полі F_p підкільце є ізоморфним до порядку $\mathcal{O}$. Множина усіх еліптичних кривих над простим полем визначається як $Ell_p(\mathcal{O}, \pi)$, де $\pi \in \mathcal{O}$, $End_p(E) \cong \mathcal{O}$.

Параметрами у схемі обміну ключами є: просте число p виду $p = 4 \cdot l_1 \cdots l_n - 1$, де l_i малі, відмінні один від одного непарні прості числа, суперсингулярна крива $E_0: y^2 = x^3 + x$ над F_p з кільцем ендоморфізмів $\mathcal{O} = \mathbb{Z}[\pi]$.

Генерація ключів. Приватним ключем сторони є кортеж $(e_1, \dots, e_n)$ з n цілих чисел. e_i вибирається випадково з діапазону $\{-m, \dots, m\}$, де m таке, що $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$. Ці цілі числа представляють клас ідеалів $[\alpha] = \left[\mathfrak{J}_1^{e_1} \dots \mathfrak{J}_n^{e_n}\right] \in cl(\mathcal{O})$, де $\mathfrak{J}_i = (l_i, \pi - 1)$. Публічним ключем є $A \in F_p$, коефіцієнт у рівнянні еліптичної кривої $[\alpha]E_0: y^2 = x^3 + Ax^2 + x$, отриманої шляхом застосування операції $[\alpha]$ до кривої E_0.

Обмін ключами. На початку у Аліси і Боба є пари ключів $([\alpha], A)$ і $([\beta], B)$ відповідно. Боб відправляє свій публічний ключ $B \in F_p \backslash \{\pm 2\}$, Аліса перевіряє, чи дійсно крива $E_b: y^2 = x^3 + Bx^2 + x$ належить множині $Ell_p(\mathcal{O}, \pi)$. Потім вона застосовує свій ключ $[\alpha]$ до E_b для отримання кривої $[\alpha]E_b = [\alpha][\beta]E_0$. Боб аналогічним чином, використовуючи свій приватний ключ $[\beta]$, отримує криву $[\beta]E_a = [\beta][\alpha]E_0$. Спільним секретом є коефіцієнт S отриманої кривої $[\alpha][\beta]E_0 = [\beta][\alpha]E_0: y^2 = x^3 + Sx^2 + x$, яка однакова для Аліси і Боба завдяки комутативності $cl(\mathcal{O})$.

4.3 Загальні характеристики

Довжина ключа CSIDH складає лише 64 біта при забезпечуваному рівні криптостійкості [33]. Це одне з найменших відомих значень довжини ключа у сучасній криптографії. Але швидкість алгоритму складає приблизно 100 мс, що є у 10 раз менше, ніж у SIDH. Але навіть при такому результаті цю криптосистему можна віднести до класу швидких криптосистем з відкритим ключем.

Як і для SIDH, найкращим класичним алгоритмом для розв'язання задачі, на якій базується стійкість схеми обміну ключами, є метод Дельфс і Гальбрайта. Складність також складає $O\left(p^{\frac{1}{4}}\right)$. Це достатній рівень класичної стійкості.

Набагато гірше ситуація склалася для квантової стійкості алгоритму. Розробники стверджують, що вибір правильного параметру !! дозволяє алгоритму мати достатню квантову стійкість проти методу пошукового кігтя. Але на алгоритм склався достатньо великий клас квантових алгоритмів розв'язання його задачі стійкості. До них можна віднести алгоритми на основі проблеми абелового прихованого зсуву, алгоритм Чайлдса-Яо-Сухарева [35]. Вони мають субекспоненційний час роботи. Найкращим квантовим алгоритмом для CSIDH вважається модифікований алгоритм Куперберга зі складністю $O\left(2^{1{,}8\sqrt{\log(O(\sqrt{p})}}\right)$. Це оцінюється у субекспоненційну складність.

Цей алгоритм підтримує не інтерактивний режим отримання спільного секрету. Це запобігає використанню на нього значного пласту атак.

4.4 Існуючі атаки

Алгоритм існує відносно малий проміжок часу і проти нього поки не відомо можливих атак. Розглянемо можливість застосування проти нього атак,

що відомі для SIDH. Усі атаки для CSIDH передбачають можливість ініціювати нечестним Бобом або Євою обмін повідомленнями для отримання часткової корисної інформації для проведення атаки. Але це не працює у контексті CSIDH, тому що він підтримує не інтерактивний режим отримання спільного секрету, він не повертає ніякої інформації і має ефективний механізм перевірки істинності іншої сторони. Тому існуючі атаки не можливо застосувати до CSIDH.

4.5 Загальні підсумки

Таблиця 4.1 – Загальні характеристики CSIDH

Назва характеристики, можлива розмірність	Значення характеристики, властивості
Довжина ключа, біт	64
Швидкодія, мс	≈ 100 мс
Класична стійкість	$O\left(p^{\frac{1}{4}}\right)$
Квантова стійкість	$O\left(2^{1,8\sqrt{\log(O(\sqrt{p})}}\right)$
Режим встановлення спільного секрету	Не інтерактивний

З цих даних можна зробити такий висновок: алгоритм може бути ефективно застосований для реалізації у вбудованих системах через свою довжину ключа і швидкість. Підтримує достатній рівень класичної безпеки, але вразливий до квантових обчислень. На сьогоднішній день поки не відомі ефективні атаки на CSIDH, це можна пояснити відносною новизною алгоритму і його не інтерактивним режимом роботи, що запобігає застосуванню великого пласту атак.

Висновок до розділу 4

Було розглянуто алгоритм CSIDH в контексті розроблених методів оцінки і класифікації алгоритмів і зроблено висновки щодо загального рівню захищеності.

ВИСНОВКИ

Результатом цієї роботи є метод оцінки алгоритмів обміну ключами для визначення кращого практичного використання алгоритмів. Він враховує загальні характеристики, що впливають на захищеність алгоритмів і існуючи атаки, направлені на цей алгоритм. Представлені критерії методу були вибрані з урахуванням специфіки галузі для найкращої оцінки алгоритмів. За допомогою цього методу були оцінені два криптографічні алгоритми обміну ключами: SIDH і CSIDH. Вони базуються на основі ізогеній суперсингулярних кривих і є можливими кандидатами для використання у криптографії. Обидва алгоритми є підтримують достатній рівень класичної криптостійкості, CSIDH, на відміну від SIDH, є вразливим до квантових обчислень, продовжуються дослідження для підвищення його квантової криптостійкості. На SIDH існує велика кількість класів атаки, більшість не складають небезпеки для алгоритму за умови не авторизованого зловмисника. На CSIDH на сьогоднішній день не існує можливих атак. Обидва алгоритми є перспективними кандидатами для використання в криптографії, можливе застосування для різних цілей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Nannicini G. An Introduction to Quantum Computing, Without the Physics [Текст] / Giacomo Nannicini // 2017.

2. Манін Ю. І. Вычислимое и невычислимое [Текст] / Юрій Іванович Манін. – Москва: «Советское радио», 1980. – (Редакция кибернетической литературы).

3. Feyman R. P. Simulating physics with computers [Текст] /R. P. Feyman // International Journal of Theoretical Physics. – 1982. – №21.

4. Shor P. Algorithms for Quantum Computation: Discrete Logarithms and Factoring [Текст] / P. Shor. // IEEE. – 1984. – №35.

5. Grover L. K. A fast quantum mechanics algorithm for database se arch [Текст] / L. K. Grover // Proceeding of the 28th ACM Symposium on Theory of Computation, New York: ACM Press. - 1996. - P. 212-219

6. The Impact of Quantum Computing on Present Cryptography. [Текст] // International Journal of Advanced Computer Science and Applications. – 2018. – №9

7. Main page of PQCrypto 2014 [Електронний ресурс] / University of Waterloo, Ontario, Canada Сайт конференції PQCrypto 2014 Режим доступа : URL:- https://pqcrypto2014.uwaterloo.ca. - 21.08.2014 p .

8. Post-Quantum Cryptography [Електронний ресурс] // NIST. – 2017. – Режим доступу до ресурсу: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

9. The D-Wave 2000Q™ System [Електронний ресурс] – Режим доступу до ресурсу: https://www.dwavesys.com/d-wave-two-system

10. Бурковський В. С. Огляд можливостей квантового криптоаналізу та криптографічних платформ, що є стійкими до нього та можуть бути основою для систем електронного цифрового підпису [Текст] / В. С. Бурковський. // Наука і техніка Повітряних Сил Збройних Сил України. – 2016. – №3.

11. Василенко О. Н. Об алгоритмах построения изогений эллиптических кривых над конечными полями и их приложениях [Текст] / О. Н. Василенко. – Москва, 2010. – (Лаборатория ТВП). – (Матем. вопр. криптогр.; т. 1).

12. Долгов В. І. Эллиптические кривые в криптографии [Текст] / В. І. Долгов. // Системи обробки інформації. – 2008. – №6.

13. Прохоров Ю. Г. Эллиптические кривые и криптография [Текст]/ Ю. Г. Прохоров. – Москва, 2007. – (Механико-математический факультет МГУ).

14. *Merkle, Ralph Charles.* Secrecy, authentication, and public key systems [Текст]. — Citeseer, 1979. - №1

15. *R. J. McEliece.* A Public-Key Cryptosystem Based On Algebraic Coding Theory [Текст]// DSN Progress Report 42-44. — 1978

16. Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman. NTRU: A Ring Based Public Key Cryptosystem. In Algorithmic Number Theory (ANTS III) [Текст], Portland, OR, June 1998, J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer-Verlag, Berlin, 1998, 267—288

17. Federal Information Processing Standards Publication Specification for the ADVANCED ENCRYPTION STANDARD (AES) [Текст], 2001.

18. Patarin J. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new Families of Asymmetric Algorithms [Текст]/ Jacques Patarin., 1996.

19. OWASP, Application Security Verification Standard 4.0, 2019.

20. Common Weakness Enumeration [Електронний ресурс] – Режим доступу до ресурсу: https://cwe.mitre.org/

21. Common Attack Pattern Enumeration and Classification [Електронний ресурс] – Режим доступу до ресурсу: https://capec.mitre.org/.

22. WASC THREAT CLASSIFICATION VERSION 2.00 [Текст], 2010.

23. Rostovtsev A. Public-key cryptosystem based on isogenies [Текст]/ A. Rostovtsev, A. Stolbunov /. – 2006.

24. Childs A. Constructing elliptic curve isogenies in quantum subexponential time [Текст]/ A. Childs, D. Jao, V. Soukharev // J. Mathematical Cryptology, 8(1):1–29. – 2014

25. De Feo L. , Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies [Текст]/ L. De Feo, D. Jao, J. Plut // – 2011.

26. Efficient compression of SIDH public keys [Текст]/ [C. Costello, D. Jao, P. Longa та ін.], 2016.

27. Delfs C. Computing isogenies between supersingular elliptic curves over Fp [Текст]/ C. Delfs, S. Galbraith., 2013.

28. Jaques S. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE [Текст]/ S. Jaques, J. Schanck., 2019.

29. Failure is not an option: Standardization issues for post-quantum key agreement [Текст] / [D. Kirkwood, B. Lackey, J. McVey та ін.]., 2015. – (Workshop on Cybersecurity in a Post-Quantum World).

30. Costello C. Efficient algorithms for supersingular isogeny Diffie-Hellman [Текст]/ C. Costello, P. Longa, M. Naehrig., 2016.

31. On the security of supersingular isogeny cryptosystems [Текст]/ S.Galbraith, C. Petit, B. Shani, Y. Bo Ti //, 2016.

32. G´elin A. Loop-abort faults on supersingular isogeny cryptosystems [Текст]/ A. G´elin, B. Wesolowsk., 2017.

33. CSIDH: An Efficient Post-Quantum Commutative Group Action [Текст] / [W. Castryck, T. Lange, C. Martindale та ін.] , 2018.

34. Towards Optimized and Constant-Time CSIDH on Embedded Devices [Текст] / A.Jalali, R. Azarderakhsh, M. M. Kermani, D. Jao., 2019.

35. Bonnetain X. Quantum Security Analysis of CSIDH and Ordinary Isogeny-based Schemes [Текст] / X. Bonnetain, A. Schrottenloher // 2018.

36. Smith B. Pre- and post-quantum Diffie–Hellman from groups, actions, and isogenies [Текст]/ Benjamin Smith., 2018.

37. Quantum Computing Playground [Електронний ресурс] – Режим доступу до ресурсу: http://www.quantumplayground.net.

ДОДАТОК А

```
// Based on C++ code from libquantum library.

proc FindFactors N
  x = 0

  if N < 15
    Print "Invalid number!"
    Breakpoint
  endif

  width = QMath.getWidth(N)
  twidth = 2 * width + 3

  for x; (QMath.gcd(N, x) > 1) || (x < 2); x
    x = Math.floor(Math.random() * 10000) % N
  endfor

  Print "Random seed: " + x

  for i = 0; i < twidth; i++
    Hadamard i
  endfor

  ExpModN x, N, twidth

  for i = 0; i < width; i++
    MeasureBit twidth + i
  endfor

  InvQFT 0, twidth

  for i = 0; i < twidth / 2; i++
    Swap i, twidth - i - 1
  endfor

  for trycnt = 100; trycnt >= 0; trycnt--
    Measure
    c = measured_value

    if c == 0
      Print "Measured zero, try again."
      continue
    endif

    q = 1 << width

    Print "Measured " + c + " (" + c / q + ")"

    tmp = QMath.fracApprox(c, q, width)

    c = tmp[0];
    q = tmp[1];

    Print "fractional approximation is " + c + "/" + q

    if (q % 2 == 1) && (2 * q < (1 << width))
      Print "Odd denominator, trying to expand by 2."
```

```
      q *= 2
    endif

    if q % 2 == 1
      Print "Odd period, try again."
      continue
    endif

    Print "Possible period is " + q

    a = QMath.ipow(x, q / 2) + 1 % N
    b = QMath.ipow(x, q / 2) - 1 % N

    a = QMath.gcd(N, a)
    b = QMath.gcd(N, b)

    if a > b
      factor = a
    else
      factor = b
    endif

    if (factor < N) && (factor > 1)
      Display "<h2>Success: " + factor + " " + N / factor
      Breakpoint
    else
      Print "Unable to determine factors, try again."
      continue
    endif
  endfor
endproc
```