Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

Дослідження несиметричних криптографічних алгоритмів в умовах використання квантових комп'ютерів. Дискретний логарифм
(тема)

Виконав: студент 2 курсу, групи _____ІПЗм-17-2_____
спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)
Освітньо-наукової програми
_____Інженерія програмного забезпечення_____
(повна назва освітньої програми)

_____Максутов Д.С._____
(прізвище, ініціали)
Керівник _____проф. Качко О.Г._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення
(код і повна назва)

Освітньо-професійна програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Максутову Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження несиметричних криптографічних алгоритмів в умовах використання квантових комп'ютерів. Дискретний логарифм

затверджена наказом по університету від "07" лютого 20 19 р № 546СТ
заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

3. Вихідні дані до роботи *алгоритми пошуку дискретного логарифму, пояснювальна записка, емулятор квантового комп'ютера, середовище об'єктно-орієнтованого проектування.*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, опис проблеми пошуку дискретного логарифму, використовувані методи та алгоритми, опис розробленої програмної системи.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) *мета роботи, обґрунтування доцільності розроблення, постановка задачі, об'єктна модель системи, базові моделі, результати тестування продуктивності програмної системи, демонстраційні матеріали, висновки*

6 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Снецчастина	проф. Качко О. Г.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1.	Аналіз предметної галузі	08.02.2019	
2.	Огляд існуючих методів	20.03.2019	
3.	Алгоритми пошуку дискретного логарфиму	16.04.2019	
4.	Підготовка пояснювальної записки	08.05.2019	
5.	Спецчастина	15.05.2019	
6.	Підготовка презентації та доповіді	25.05.2019	
7.	Попередній захист	28.05.2019	
8.	Нормоконтроль, рецензування		
9.	Занесення диплома в електронний архів		
10.	Допуск до захисту у зав. кафедри		
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ _____2019 р.

Студент _____
(підпис)

Керівник роботи _____ __проф. Качко О. Г._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 89 с., 23 рис., 4 додатки, 26 джерел.

КВАНТОВИЙ КОМПЬЮТЕР, КВАНТОВИЙ АЛГОРИТМ, ДИСКРЕТНИЙ АЛГОРИТМ ШОРА, КВАНТОВИЙ ПАРАЛЕЛІЗМ, Q#, JAVA, JMH, ДИСКРЕТНИЙ ЛОГАРИФМ, NP ЗАДАЧА.

Об'єктом є несиметричні криптографічні алгоритми в умовах використання квантових обчислень, квантовий алгоритм Шора пошуку дискретного логарифму.

Метою роботи є дослідження квантового алгоритму Шора для пошуку дискретного логарифму та його порівняння з не-квантовими аналогами.

Методи розробки базуються на мовах програмування C#, Java та спеціальній мові для квантових обчислень Q#. Також використовуються наступні методи розробки: паралельні обчислення, об'єктно-орієнтований підхід до розробки програмної системи, функціональний підхід до розробки системи.

У результаті роботи були досліджені переваги та недоліки квантових обчислень, дискретний квантовий алгоритм Шора для пошуку дискретного логарифму та його не квантові аналоги

QUANTUM COMPUTER, QUANTUM ALGORITHM, DISCRETE SHOR`S ALGORITHM, QUANTUM PARALLELISM, Q#, JAVA, JMH, DISCRETE LOGARITHM, NP COMPLEXITY CLASS.

The object of research are asymmetric cryptography algorithms in conditions of quantum computer usage, discrete quantum Shor’s algorithm.

The aim is development of application for research of discrete logarithm by discrete Shor`s quantum algorithm and it’s comparison to non-quantum analogs.

Methods of development are based on C# and Java languages and special language for quantum computing Q#. Also the following developing methods are used: parallel computing, object-oriented approach to software program system, functional approach.

As result of the research the pros and cons of quantum computations were investigated, as well as discrete Shor`s quantum algorithm and it’s non-quantum analogs.

ПЕРЕЛІК СКОРОЧЕНЬ

ECDLP	асинхронний генератор;
GF	поле Галуа (Galois field);
NP-клас	клас задач не поліноміальної складності;
P-клас	клас задач поліноміальної складності;
BQP-клас	клас задача квантової поліноміальної складності з обмеженою помилкою;
VRP	Задача маршрутизації транспортного засобу (Vehicle routing problem);
ECDH	протокол Діффі-Хелмана на еліптичних кривих (Elliptic-curve Diffie–Hellman);
ECDSA	алгоритм цифрового підпису на еліптичних кривих (Elliptic Curve Digital Signature Algorithm);
QCP	Quantum Computing Playground
MQM	Microsoft Quantum Machinery
QDK	Quantum Development Kit;

ЗМІСТ

ВСТУП

На початку ери дослідження квантової механіки та квантових обчислень, ніхто навіть і не мріяв про їх практичне застосування, яке б могло впливати на повсякденне життя оточуючих. Усі дослідження у цій області вважалися дуже суперечливими. Наукова спільнота, у своїй більшості, сприймала квантові ефекти, скоріше, як фантастику, а не реально існуючі фізичні явища. До того ж, дослідження ускладнювалися неможливістю пояснити природу спостережуваних явищ та результатів експериментів. Тим не менш, сьогодні на базі квантової механіки вчені будують, і навіть продають, квантові комп'ютери, а теорія квантових обчислень обзавелася несуперечливою математичною моделлю, на якій тепер базується ідея програмування квантових комп'ютерів.

За будь якого часу, інформація мала велику ціну у світі. Ще з давніх давен, певну, важливу, інформацію намагалися захистити різними способами, основним з яких було шифрування. Проте за давніх часів інформація передавалася за допомогою гінців, або ж спеціально навчених птиць. Інформацію, відправлену таким чином, було досить важко перехопити. Сьогодні ж, інформація пронизує усе життя суспільства. Інформаційні канали, такі як оптоволоконні кабелі з високою пропускною здатністю, або ж радіосигнали різних частот охоплюють усю планету. Перехід до такої моделі розповсюдження інформації призвів до того, що тепер, перш ніж дійти до адресата, інформаційний пакет проходить через десятки вузлів – пристроїв, які до адресата не мають ніякого відношення. Це значно збільшує ризик перехоплення повідомлення на одному з таких вузлів. Для захисту інформації, що подорожує, через відкриті, вразливі, канали зв'язку сьогодні застосовуються методи асиметричної криптографії. За допомогою цих методів, інформаційні повідомлення надійно шифруються і відправник повідомлення може бути впевнений, що зміст його повідомлення не буде розкрито. Така система задовольняла усіх користувачів, аж доки квантові комп'ютери

не стали набирати популярність. Справа у тому, що протоколи асиметричної криптографії, базуються на складності вирішення деяких математичних проблем. І ця складність дійсно зберігалася, аж доки не виявилося, що ці проблеми можуть бути вирішені у прийнятний час за допомогою можливостей для обчислень, що надають квантові комп’ютери.

Такі обчислювальні можливості досягаються квантовими комп’ютерами завдяки тому, що їх робота заснована на використанні різноманітних квантових ефектів, таких як невизначеність, зчеплення та суперпозиція кубітів. Кубіті – це саме те, що, в першу чергу, відрізняє квантові комп’ютери від класичних. Кубіт, або ж квантовий біт, виділяється тим, що на відміну від звичайного біту, його стан виражається ймовірнісним полем. Тобто його стан – це не певна задача величина (1 або 0), а набір ймовірностей величин.

Завдяки цьому, обчислення над кубітами також набувають ймовірнісного характеру і це, разом з можливістю впливу на ймовірності, дозволяє значно прискорити вирішення таких задач, як: моделювання поведінки мікрочасток (молекул і т.і.), секвенування ДНК, пророкування біржових котирувань і підбір криптографічних ключів, оптимізація маршрутів транспорту.

В наші дні, розробка квантових комп’ютерів та написання програм під них набувають все більшого поширення IT-гіганти такі, як Google, Intel, Microsoft, International Business Machines (IBM), D-Wave Systems [1] вступили у гонку по розробці першого потужного квантового комп’ютера, який буде придатний для вирішення багатьох реальних проблем. На сьогодні уже існують квантові комп’ютери, що придатні для рішення незначних задач, або проведення певних досліджень в областях фізики, біології, хімії і криптографії. В силу цього, останнім часом, досить велика, увага приділяється пост-квантовим алгоритмів криптографії. Таким чином, можна спрогнозувати, що існує велика вірогідність побачити потужні квантові комп'ютери у недалекому майбутньому.

Метою даної роботи є дослідження алгоритмів квантових обчислень, а саме дискретній версії квантового алгоритму Шора для знаходження дискретного логарифму у циклічній групі чи на еліптичній кривій, яка уявляє собою окремий випадок циклічної групи та порівняння цього алгоритму з його не квантовими аналогами. Методи розробки базуються на мовах програмування C#, Java та Q#. Також використовуються наступні методи: квантові обчислення, об'єктно-орієнтований підхід до розробки програмної системи. Окрім цього будуть застосовані готові бібліотеки чи інструменти для квантових обчилсень, які містять реалізовані квантові примітиви, що допоможуть значно спростити та пришвидшити розробку та сконцентрувати розробку та дослідження саме на квантових алгоритмах, а саме, квантовому алгоритму Шора для пошуку дискртеного логарифму у звичайних групах та на еліптичних кривих.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

Криптографія має на меті проаналізувати, які проблеми можна зробити простими в той же час роблячи пов’язані з ними проблеми важкими. На практиці, якщо я хочу відправити своєму другові повідомлення, яке ніхто інший не зможе прочитати, я хочу, щоб мені було легко закодувати це повідомлення, моєму другові було просто його розшифрувати і третій особі було важко його розшифрувати. Одна з проблема, які вважаються важкою, і на базі якої побудовано багато криптографічних систем, це проблема дискретного логарифму. Наразі немає швидких методів для вирішення цієї проблеми, хоча цілком можливо, що вони коли-небудь будуть знайдені. Проблема дискретного логарифму більшою мірою задається з використанням еліптичних кривих, ніж кінцевих областей.

Так сталося, що ключ шифрування одного і того ж розміру забезпечить більшу безпеку, якщо ми використовуємо ECC. Іншими словами, ми можемо використовувати менший ключ, щоб отримати той саме обсяг безпеки, що прискорює розрахунки, які ми хочемо прискорити.

У 1985 році Н. Кобліц і В. Міллер незалежно запропонували використовувати групу точок на еліптичній кривій визначеній над кінцевим полем у криптосистемі дискретного логарифму. Це зовсім інший шлях у вирішенні криптографічних проблем. З ним можна використовувати групу еліптичних кривих, яка менша за розміром ніж звичайна кінцева група і при цьому надає той же рівень безпеки. У багатьох випадках це призводить до використання меншого за розміром ключа, економії трафіку і більш продуктивної імплементації, що особливо важливо для смарт карток і смартфонів.

У 2005 році американське агентство національної безпеки США опублікувало роботу, в якій вони рекомендували "взяти" до уваги результати останніх 30 років

дослідження алгоритмів з публічним ключем і перейти від алгоритмів першого покоління (RSA) до алгоритмів на еліптичних кривих.

Класична проблема дискретного логарифму полягає в наступному: нехай, існує таке ціле число k, що $a^k \equiv b \ (mod\ p)$, де p являється простим числом, і треба знайти k. Зазвичай, для знаходження k використовується операція обчислення логарифму. В рамках задачі пошуку саме дискретного логарифму k є цілим числом. Так як за визначенням a^k повинно ділити p-1, k може бути визначено у кільці за модулем (mod p-1).

У 1997 році Certicom анонсував декілька призів за вирішення проблеми дискретного логарифму на еліптичних кривих (ECDLP). Задача була поставлена наступним чином: розрахувати приватні ключі ECC маючі список публічних ключів і асоційованих з ними параметрів. Загалом, це проблема з якою стикається кожен, хто хоче повністю зламати криптосистему базовану на еліптичних кривих.

Таблиця 1.1 – Основні числа ECDLP Break Challenge

Число	Двійкових чисел	Винагорода	Зламано
ECCp-79	79	book	1997 Baisley and Harley
ECC2-79	79	book	1997 Harley et al.
ECCp-89	89	book	1998 Harley et al.
ECC2-89	89	book	1998 Harley et al.
ECCp-97	97	$5,000	1998 Harley et al. (1288 комп’ютерів)
ECC2K-95	95	$5,000	1998 Harley et al. (200 комп’ютерів)
ECC2-97	97	$5,000	1999 Harley et al. (740 комп’ютерів)

Кінець таблиці 1.1

Число	Двійкових чисел	Винагорода	Зламано
ECC2K-108	108	$10,000	2000 Harley et al. (9500 комп’ютерів)
ECCp-109	109	$10,000	2002 Monico et al. (10000 комп’ютерів)
ECC2-109	109	$10,000	2004 Monico et al. (2600 комп’ютерів)
ECC2K-130	130	$20,000	-
ECCp-131	131	$20,000	-
ECC2-131	131	$20,000	-
ECC2K-163	163	$30,000	-
ECCp-163	163	$30,000	-
ECCp-191	191	$40,000	-
ECC2K-238	238	$50,000	-
ECC2-238	238	$50,000	-
ECCp-359	359	$100,000	-

У таблиці 1.1 наведено дані про характеристики еліптичних кривих які були представлені на змаганні, а також результати їх злому.

Як видно з таблиці, спроби злому з використанням звичайних комп’ютерів не дали позитивних результатів навіть для ключів порівняно невеликого розміру. Тим не менш, згідно з дослідженнями останніх років, квантові комп’ютери та алгоритм Шора є великою загрозою для криптографії на еліптичних кривих. У 2015 році NASA опублікувало роботу у який закликає до розробки нових, так званих, пост-квантових алгоритмів криптографії і відмови від алгоритмів на базі еліптичних кривих і RSA [2].

1.1 Асиметричні алгоритми криптографії. Односторонні функції та їх застосування.

У своїй відомій роботі "New Directions in Cryptography" Діффі і Хеллман описали абсолютно новий і надійний спосіб встановлення каналу для обміну секретною інформацією. Новизна способу виражалась у тому, що зникала необхідність попереднього обміну секретними ключами. Необхідні ключи могли вільно передаватися по відкритим каналам зв'язку. За цю властивість, цей спосіб отримав назву шифрування з відкритим ключем. Представлений спосіб, або ж протокол, передбачає використання декількох ключів. Одного відкритого ключа і одного закритого ключа. Плюс декількох відкритих змінних. При чому знання одного з цих ключів не дозволяю тут же визначити другий. Як наслідок цієї властивості, один з ключів (ключ шифрування) може бути відкритим без ризику злому шифрограм, інший же ключ (ключ розшифрування) все ще повинен триматися в таємниці. В результаті такої організації схеми шифрування, системи в яких використовується шифрування з відкритим ключем також називають асиметричними (несиметричними) криптосистемами [3].

Шифрування з відкритим ключем базується на такому такій математичній сутності, як одностороння функція (one-way function) [4]. Основною властивістю такої функції є те, що її значення $F(x)$, легко обчислити для заданого аргументу $x \in X$, в той час, як інверсію, тобто пошук значення x за значенням $F(x)$, зробити досить важко. Під важко, тут розуміється відсутність алгоритму для пошуку інверсу за поліноміальний час роботи. Теоретично, завжди можна знайти x по відомим значенням $F(x)$ перебравши усі можливі варіанти з множини X і підставляючи їх у функцію $F(x)$, тобто використати метод простого перебору. Тим не менш, з практичної точки зору, такий підхід не є прийнятним при великій розмірності множини X, так як перебір займе занадто багато часу.

Доказ існування таких односторонніх функцій такого роду і до сих пір являється відкритою проблемою. Доказ їх існування автоматично доводить твердження про нерівність класів складності P та NP. Уся сучасна асиметрична криптографія базується на тому, що гіпотеза про існування односторонніх функцій все ж вірна.

З математичної точки зору, односторонніми називають такі функції $F(x): X \to Z, x \in X$, що володіють двома наступними властивостями:

– існує поліноміальний алгоритм обчислення значень $z = F(x)$;

– не існує поліноміального алгоритму для знаходження інверсу функції, або ж функції інверсу $F(z) = x$.

Варто зауважити, що існування односторонніх функцій все ще не було математично доведено, тому використання їх як основи для асиметричних алгоритмів шифрування може мати місце до тих пір, доки не знайдені ефективні алгоритми, для пошуку інверсу односторонніх функцій за поліноміальний час.

Одним з прикладів можливо односторонньої функції є зведення у ступінь по модулю (або ж у кільці) (1.1):

$$F(x) \equiv a^x mod\ p, \tag{1.1}$$

де a – примітивний елемент поля $GF(p)$ (генератор);

p – велике просте число.

Досить легко можна продемонструвати, що така функція може бути розрахована за ефективний час навіть для параметрів великої розрядності (сотні знаків). Візьмемо за приклад: a^{30} можна обчислити за допомогою шести операцій множення (множенням вважається і зведення в квадрат).Число 30 в двійковій системі числення записується як 11110, тому $30 = 2^4 + 2^3 + 2^2 + 2$ згідно з формулою (1.2).

$$a^{30} mod\ p \equiv (a^{16}a^8a^4a^2) mod\ p \equiv ((((a^2a)^2a)^2a)^2) mod\ p, \tag{1.2}$$

де a – примітивний елемент поля $GF(p)$;

p – велике просте число.

Завдання пошуку функції виду (1.3), називається завданням пошуку дискретного логарифму.

$$\log_a A = b \Leftrightarrow a^b = A \; mod \; p, \tag{1.3}$$

де a – примітивний елемент поля $GF(p)$;

p – велике просте число.

На даний момент не було знайдено будь-яких ефективних алгоритмів для обчислення дискретних логарифмів у полях великої розмірності.

Сама по собі, будь-яка одностороння функція не придатна для використання у якості функції шифрування, так як повідомлення x зашифроване односторонньою функцією $F(x)$ – не можливо розшифрувати згідно з визначенням односторонньої функції. Для вирішення цієї проблеми можна використовувати односторонню функцію з секретом (one-way trapdoor function). Роздивимось приклад: Дана функція $E_k: X \to Y$, яка має обернену функцію $D_k: Y \to X$, проте знайти обернену функцію використовуючи тільки інформацію про E_k (без знання секрету k) - неможливо. Тож функція $E_k: X \to Y$, результат виконання якої залежить від параметра k називається односторонньою функцією з секретом k. Односторонній функції з секретом має наступні три властивості:

– існує поліноміальний алгоритм розрахунку значень $E_k(x)$, для будь-якого k;

– поліноміального алгоритму інвертування E_k не може бути знайдений якщо значення k не відоме;

– поліноміальний алгоритм інвертування E_k існує і може бути знайдений, якщо значення k відоме.

Функцію E_k придатна для шифрування інформаційних пакетів. Інформацію зашифровану функцією E_k можна розшифрувати використовуючи обернену їй функцію D_k, так як для всіх $x \in X$ виконується $D_k(E_k(x)) = x$.

Варто звернути увагу на головну властивість цієї комбінації функцій, а саме, той хто шифрує інформацію за допомогою функції E_k не обов'язково повинен мати змогу її розшифрувати. Існування односторонніх функцій з секретом, так само як і існування односторонніх функцій, не було математично доведено.

Існує декілька функцій, які ймовірно являються односторонніми функціями з секретом і використовуються у практичній криптографії. Друга властивість односторонніх функцій з секретом, для цих функцій не доведена. Тим не менш, про них відомо, що пошук інверсу цих функцій еквівалентний вирішенню важкої (затратної по часу) математичної задачі.

Першою і досить важливою односторонньою функцією з секретом, що використовується у практичній криптографії для протоколів з відкритим ключем, можна назвати функцію факторизація цілих чисел. Іншою, не менш важливою і досить широко використовуваною функцію, являється функція пошуку дискретного логарифму.

Задіяння цих функцій в криптографії дозволяє досягти наступних переваг:

– прибрати необхідність підтримки секретних каналів зв'язку, які раніше були необхідні для обміну ключами симетричного шифрування, і використовувати для обміну інформацією тільки відкриті канали зв'язку;

– підвищити стійкість шифрування і легко змінювати стійкість шляхом зміни довжини ключа;

– застосовувати криптографію для вирішення нових класів завдань, таких як електронний цифровий підпис та ін.

Тож безпека та стійкість до злому переважної більшості сучасних асиметричних протоколів шифрування залежить від складності вирішення двох складно обчислювальних математичних проблем:

– факторизація великих цілих чисел;

– пошуку дискретного логарифму на кінцевих полях.

Ці дві математичні задачі широко засновуються в асиметричних криптографічних протоколах, так як досі не було знайдено ефективних алгоритмів для вирішення цих задач або ж такі алгоритми були знайдені для певних виключних параметрів таких задач.

В заключення, можна констатувати, що односторонні функції являються фундаментальним компонентом асиметричної криптографії, протоколів ідентифікації та автентифікації особи та інших областей пов’язаних з захистом інформації. Не зважаючи на те, що їх односторонність все ще математично не доведена існує декілька функцій, що піддалися десятиліттям безперервного аналізу і активно застосовуються в системах, які стали невід’ємною частиною повсякденного життя. Таких, як інтернет, телефонний зв’язок та ін.

1.2 Класифікація задач за обчислювальною складністю. Задачі класу NP

Якщо звернутися до класичної теорії складності алгоритмів, алгоритми поділяються за обчислювальною складності на наступні категорії: NP-складні, NP, BQP та P. Задачі, що відносяться до класу складності P мають обчислювальну складність (за часом), що виміряється поліномом залежним від обсягу вихідних даних і можуть бути вирішені за поліноміальний час на детермінованої обчислювальної машини (машині Т'юрінга). Задачі, що відносяться до класу BQP з певною ймовірністю можуть бути вирішні на квантовому комп’ютері за поліноміальний час. До NP[5] класу відносяться задачі, які мають поліноміальну складність при їх вирішенні з використанням недетермінованої обчислювальної машини, особливістю якої є те, що кожен її наступний стан не завжди однозначно залежить від

попереднього. Тобто на відміну від класичного детермінованої машини, обчислення на такій машині будуть розгалужуватися (виконуватися паралельно) при кожному виникненні неоднозначності: Іншою особливістю недетермінованої машини, являється те, що обчислення на ній можуть ніколи не зупинитися, тобто рішення задачі може бути не знайдено. Тому задача запущена на такий машині вважається вирішеною, тільки в тому випадку, коли хоча б одна гілка розгалуженого процесу обчислень процесу привела до якоїсь відповіді. Тобто, хоча б одна з гілок завершилася. Належність до NP класу складності можна також визначити по іншому критерію. Якщо вирішення задачі може бути перевірене за поліноміальний час з використанням додаткової інформації поліноміальної довжини (що була надана ззовні), то таку задачу можна віднести до NP класу. Таким чином до NP класу також відносяться усі задачі, вирішення яких можна перевірити за поліноміальний час. Варто зауважити, що NP клас включає в себе P клас. Прикладом задач NP класу можуть служити задача про комівояжера, або її узагальнений варіант VRP.

Так як клас NP включає в себе клас P, неможливо гарантовано довести належність задачі до класу NP. Тому відношення задач до того чи іншого класу зазвичай відображає лише нагальний рівень знань про способи вирішення того чи іншого завдання і не є постійною величиною. Загалом, у нас не має підстав вважати, що відношення здачі до NP класу є справедливим і кінцевим, так як на даний момент відсутність рішень складності P для NP задач не доведена. Це питання, а саме, питання еквівалентності P і NP класів (існування P-рішення для будь-якої NP-задачі) вважається однією з основних проблем теорії складності алгоритмів. Доказів того чи іншого не було представлено до сих пір. Саме ж питання еквівалентність P і NP класів виникає через існування класу NP-повних задач. Множина NP-повних задач, як і множина задач P класу) є підмножиною задач NP класу і вирізняється тим, що усі завдання NP класу так чи інакше можуть бути зведені до цього класу NP-повних задач. А це означає, що якщо для NP-повної задачі буде знайдено рішення складності класу

P, то таке рішення буде знайдено для всіх завдань NP класу. Одним з прикладів NP-повної задачі є задача про кон’юнктивну форму.

Історично склалося, що дослідження складності алгоритмів відкрило нові ракурси при погляді багато класичних математичних задач і способи їх вирішення. Це дозволило знайти рішення, які є менш ресурсно-затратними порівняно з традиційними, для таких завдань, як: множення многочленів і матриць, рішення систем лінійних рівнянь і т.і.

У теорії складності, множину алгоритмів, складність яких суттєво залежить від розміру вхідних даних; і при цьому знижується до поліноміальної, якщо надати алгоритму певні додаткові дані (так звані свідки рішення), називають множиною недетермінованих поліноміальних алгоритмів.

Теоретично, NP клас складається з задач, для яких за поліноміальний час , використовуючи недетерміновану машину Т’юрінка можна перевірити потенційне рішення задачі. Іншими словами задачі з множини NP класу можуть бути вирішені за поліноміальний час на недетермінованої машині Т'юрінга.

NP клас можна визначити для множини мов, які уявляють собою множини слів над кінцевим алфавітом. Мова Len можна назвати належною до класу NP, коли існує певний двомісний предикат R (x, y) з P класу і многочлен n^c такі, що постулат "x належить Len" рівноцінний постулату "існує у довжиною менше n^c(де n - довжина слова x) таке, що предикат R (x, y) виконується" для будь-якого слова x. У такому випадку слово y називають свідком належності x до мови Len. Тобто, якщо у нас є слово, що належить мові, і слово-свідок обмеженої довжини, то можна швидко переконатися в тому що x дійсно належить Len.

Клас складності NP можна визначити і іншим чином, вводячи поняття недетермінованої машини Т'юрінга, яка уявляє собою машину Т'юрінга, у програмі якої можуть існувати різні рядки з однаковою лівою частиною. У разі появи "розвилки", тобто неоднозначність в програмі, під час виконання обчислення може піти різними шляхами. Недетерміновану машину Т'юрінга можна представити

предикат R (x), який вважається рівним одиниці в тому випадку, коли існує хоча б один варіант, або ж шлях, обчислення, результатом якого є 1. В іншому випадку цей предикат дорівнює 0. У разі, коли довжина шляху обчислення, що призводить до одиниці, не перевищує деякого многочлена від довжини x, предикат називають таким, що належить класу NP. Коли у певній мові є предикат що належить NP класу і розпізнає його, мова називається такою, що належить класу NP. Наведене визначення еквівалентно визначенню, що приведено вище. Якщо в якості свідка взяти номери необхідних гілок на розвилках в обчисленні. Беручі до уваги, що x належить мові, довжина усього шляху обчислення не буде перевищувати многочлен від довжини x, тому довжина свідка також буде обмежена многочленом від довжини x.

Можна зауважити, що будь-яку задачу NP класу, можна вирішити за експоненціальний час методом перебору усіх можливих свідків довжини яких менше n^c.

З вищенаведеного можна побачити, що множина мов з класу NP не є замкненою щодо доповнення. Клас мов, доповнення яких належить класу NP, називається co-NP класом.

Далі наведено задачі, належність яких класу P не визначена, але точно відома їх належність до класу NP[6]:

– VRP-задача: полягає в знаходженні оптимального маршруту на графі для декількох агентів і уявляє собою узагальнення задачі про комівояжера;

– задача доказу існування цілочисельного рішення системи лінійних нерівностей. Свідком в даному випадку виступає знайдене рішення;

– задача пошуку клік (повних підграфів) певного розміру у даному графі. Свідок виступають номери вершин, з яких складається знайдена кліка;

– задача про комівояжера, яка полягає у пошуку найбільш ефективного шляху по усіх вершинах графа і є більш наближений до реальності варіантом здачі наведеної вище;

– задача пошуку Гамільтонова циклу на графі. Свідком в даному випадку виступає послідовність вершин, з яких будується Гамільтонів цикл.

Окрім цього до задач NP класу відносять задачу факторизації цілих чисел і задачу пошуку дискретного логарифма. Проте ці задачі не відносяться класу NP-повних задач, тому алгоритму P класу для пошуку дискретного логарифма не буде означати еквівалентності P та NP класів.

Як згадувалося раніше, багато сучасних криптосистем побудовано на базі асиметричних протоколів. Серед таких протоколів можна виділити DH, стійкість якого базується на складності пошуку дискретного логарифму, і популярний в останній час ECDH, стійкість якого базується на складності задачі пошуку дискретного логарифму на еліптичних кривих.

До теперішнього часу ніяких крипто атак небезпечних для протоколів ECDH та ECDSA виявлено не було. Основними атаками є атака заснована на MOV алгоритмі та атака заснована на алгоритмі Гельфонда - Шенкса. Безпека алгоритму ECDH заснована на складності задачі пошуку дискретного логарифму на еліптичних кривих (пошук порядку у циклічній групі). У рамках цього протоколу відкритий та закритий ключі уявляють собою точки на еліптичній кривій, кожна координата яких має розмір 50-150 розрядів або більше і є простим числом. При чому, розшифрування зашифрованої відкритим ключем інформації еквівалентно знаходженню дискретного логарифма на еліптичній кривій. Існує достатньо багато алгоритмів для пошуку дискретного логарифма, але жоден з них не може знайти дискретний логарифм на стандартизованій еліптичній кривій за поліноміальний часу (тобто не входить до P класу). Для того щоб забезпечити безпеку, ECDH вимагає, щоб модуль був принаймні 160 бітів завдовжки.. Навіть при використанні потужного і найшвидшого комп'ютера, доступного на сьогодні, пошук дискретного логарифму у групі такого прядку вимагає нездійсненно великого часу. Це означає, що ECDH безпечний, поки не буде знайдений ефективний алгоритм пошуку дискретного алгоритму на еліптичних кривих.

Для пошуку простого дискретного логарифму та для інших задач з класу NP на сьогодні також не знайдено рішень класу P, і усі запропоновані класичні алгоритми для вирішення цих задач мають експоненційну складність. Для груп великого порядку, робота таких алгоритмів може займати роки. В наслідок чого атаки на криптосистеми, в основі яких лежать ці алгоритми, стають не доцільними і не ефективними, з точки зору час витрат та актуальності здобутої інформації.

Таким чином, можна заявити, що на даний момент існує проблема рішення задач класу NP, і використання для їх рішення класичних комп’ютерів будь-якої потужності не є раціональним і не переведе до їх рішення у задовільний час. Виходом можуть послужити квантові комп’ютери та алгоритми для них. Далі більш детально будуть розглянуті особливості квантового алгоритму для пошуку дискретного логарифму та його реалізація.

1.3 Сьогоднішній стан розробки квантових комп’ютерів

Квантові комп'ютери, придатні для вирішення актуальних задач, мають усі шанси з’явитися у найближчий час, і їх поява змінить світ. З їх допомогою науковці можуть зробити нову революцію у медицині, фізиці, хімії, біології, інформаційній безпеці, машинному навчанні і багатьох інших областях науки.

Мова про квантові комп'ютери, що будуть здатні швидко зламувати криптосистеми іде уже декілька десятиліть. Проте до недавніх років, усе це було лише красивою картинкою у головах математиків, фізиків та криптоаналітиків.

З недавнього часу усі ці мрії почали переходити у реальні реалізації. Про це свідчать рекомендації американського випущені в серпні 2016 року. Агентства національної безпеки США захищає державні таємниці і виступає радником американського державного та приватного секторів у питаннях актуальних

алгоритмів комп'ютерного захисту, і криптографії в тому числі. В рамках оновлення рекомендацій було оновлено набір рекомендованих криптографічних алгоритмів (SUITE B).

З оглядом на швидкий прогрес фізики і техніки в області дослідження та розробки квантових комп’ютерів, АНБ рекомендує готуватися до швидкої появи квантових комп'ютерів придатних для практичного використання у сфері криптографії. Поява цих комп’ютерів призведе до можливості злому найпопулярніші сьогодні асиметричних криптографічних протоколів та цифрового підпису.

Багато великих гравців взялося за створення власних квантових комп’ютерів намагаючись виграти у конкурентній боротьбі за потенційні ринки.

Наприклад Google анонсував та випустив наступні продукти пов’язані з квантовими обчисленнями:

– 7 липня 2016 року Google почав експеримент пов’язаний з пост-квантовою криптографією який проходив на базі їх браузера Chrome. Експеремент полягав у тому, що в Chrome Canary (версії браузера для розробників у якій випускаються різні функції з раннім доступом) для певних користувачів була ввімкнена нова система шифрування з’єднання до серверів Google що базувалася на новому алгоритмі. Базою для розробки системи послужила наукова робота у якій йшлося про впровадженню пост-квантового алгоритму шифрування в протокол TLS. Назва алгоритму ("Нова надія") була такою-собі відсилкою до поп-культури; За словами розробників головною метою було дати дослідникам можливість проведення криптоаналізу практичної реалізації пост-квантового алгоритму у продакшені. Окрім цього розробники хотіли оцінити швидкість і надійність з'єднання при використанні цього алгоритму. Google не збирається нав'язувати цей алгоритм, як новий крипто стандарт, чи просувати його іншим чином;

– QCP. Своєрідний ігровий майданчик для тих, хто хоче спробувати свої сили в написанні алгоритмів для квантового комп’ютера, або ознайомитися ближче з квантовими обчисленнями. Додаток представлений у вигляді веб-додатка Chrome, і

використовує WebGL, для імітації кубітів (до 22) на GPU. Додаток включає в себе просте середовище розробки, для написання, компіляції та запуску коду. Окрім цього додаток надає можливість проекспериментувати з відомими реалізованими квантовими алгоритмами Гровера та Шора, має зручний інструментарій для дебагу та 3D візуалізації квантових станів, що наглядно демонструє результат застосування квантових операцій (гейтів) на кубіти. Для написання програм використовується мова QScript. 22-х кубітів не достатньо для практичного застосування, тож QCP варто сприймати, як навчальну платформу.

Далі детально розглянемо успіхи інших гравців на рику продуктів для квантових обчислень:

– IBM представила IBM Q, в якості ініціативи направленої на розвиток квантових обчислень та створення доступних та універсальних квантових комп’ютерів придатних для використання в науці і бізнесі. Ця ініціатива стала першою у своєму роді. На даний момент IBM вже має прототип комерційного квантового комп’ютеру, на основу якого далі будуть будуватися інші системи. Співпрацюючи зі своїми ключовими партнерами по галузі, IBM збирається вагомо збільшити обчислювальну потужність квантових комп’ютерів у майбутньому і показати перевагу в ефективності порівняно з класичними обчислювальними системами. IBM збирається відкрити комерційне API для хмарних квантових обчислень та перший комерційний центр квантових обчислень в 2019 році. В даний час найбільший квантовий комп’ютер доступний в рамках IBM Q має тільки 20 кубітів, проте IBM обіцяє надати комп’ютер з 50ю кубітами до кінця 2019 року;

– D-Wave Systems. Невелика компанія в порівнянні з гігантами індустрії, яка була заснована спеціально для розробки квантових комп’ютерів. Обравши для розробки дещо інший підхід ніж лідери ринку. У листопаді 2017 року D-Wave Systems анонсувала вихід свого нового квантового комп'ютера під назвою D-Wave 2000Q. Анонсований комп’ютер містить в два рази більше кубітів ніж його попередник D-Wave X2. Вже 23 січня 2017 D-Wave Systems почала продаж анонсованих D-Wave

2000Q. D-Wave 2000Q, як і попередні комп’ютери компанії уявляє собою адіабатичний комп'ютер, який працює за принципом квантового відпалу. D-Wave 2000Q складається з великої кількості компонентів і має багато контрольованих параметрів. Для функціонування D-Wave 2000Q, компоненти у ньому охолоджують до дуже низьких температур (комп'ютер попередньої моделі функціонував при температурі близькій до абсолютного нуля - близько -273 ° C). Після охолодження і досягнення системою мінімальної енергії, задані параметри повільно змінюються з використанням законів квантової механіки для переводу системи з початкового стану в новий стан мінімальної енергії за рахунок квантового тунелювання. Не заважаючи на велику кількість кубітів, комп'ютери компанії D-Wave System придатні для виконання лише одного типу завдань, а саме - завдань оптимізації. Беручи це до уваги, D-Wave System не варто порівнювати з іншими компаніями розробниками квантових систем, а їх комп’ютери вважати повноцінними квантовим комп’ютерами[7];

– Google. Google представи 20 кубітний квантовий комп’ютер ще в 2017 році. За словами інженерів кубіти в цьому комп’ютері були розташовані в довгі рядки. Пізніше, наприкінці 2017 року, Google анонсував вихід нового 49-кубітного комп’ютера, кубіти в якому розташовані у сітці 7 на 7. Невдовзі після цього було анонсовано 72-кубітний комп’ютер з сіткою 6 на 12, який за словами інженерів буде використано для демонстрації так званої «квантової переваги», тобто для показу переваги ефективності обчислень на квантовому комп’ютері в порівнянні з класичним. Таким чином Google поки що виграє квантову гонку з найближчім конкурентом у вигляді IBM і їх 50-кубітного квантового комп’ютера;

– Microsoft. MQM – ініціатива, представлена Microsoft наприкінці 2017 року і направлена залучення до дослідження квантових обчислень усіх бажаючих. Нові можливості для дослідження стають можливі завдяки QDK що складається з нової мови програмування Q# [8] та емулятора квантового комп’ютера, який можна запустити, як локально, так і використовуючи Microsoft Azure. Окрім цього, ініціатива включає розробку квантових комп’ютерів з підтримкою розподілених

квантових обчислень, що в найближчій час плануються до виходу у загальний доступ на платформі Azure.

Зважаючи на стрімкий розвиток квантових комп’ютерів, варто згадати про способи захисту своїх даних у пост-квантову еру:

– збільшити розмір ключів в алгоритмах шифрування. Навіть враховуючи те, що до появи потужного квантового комп'ютера можуть пройти десятиліття, не можна виключати появу інших варіантів криптоатак. Заважаючи на це, використання ключів RSA-8192, P-384, або навіть P-521 цілком себе виправдовує, особливо якщо справа стосується конфіденційних документів;

– почати використовувати, або хоча б впроваджувати, пост-квантові криптографічні алгоритми. Різні пост-квантові алгоритми лежать у широкому спектрі практичності і зручності використання. Окрім того, зважаючи на досить короткий час в використання криптостійкість ряду з них залишається під питанням. Тим не менш, тим хто вже сьогодні хоче ступити в майбутнє, варто ознайомитися за найперспективнішими "постквантовими" алгоритмами, побудованими на базі завдань теорії решіток;

– уникати обміну ключами з використанням асиметричних протоколів під час обміну конфіденційною інформацією. Вже на сьогодні існують квантові алгоритми для ефективного вирішення задач, які лежать в основі сучасної асиметричної криптографії (задачі факторизації та пошуку дискретного логарифму), що робить використання, шифрування і цифрового підпису за дорогою RSA або еліптичних кривих, протоколи Ель-Гамаля і Діффі – Хеллмана небезпечним, так як їх компрометація це тільки питання часу і нарощування потужності квантових комп’ютерів. Необхідно впроваджувати інші протоколи обміну ключами або проводити обмін ключами фізично. Наприклад, в мессенджері Signal для обміну ключами можна взаємно просканувати QR-код на телефоні співрозмовника, що є дає непогані гарантії безпеки подальшого чатингу;

– почати використовувати більш криптостійкі симетричні алгоритми. Згідно останніх досліджень і уже розроблених алгоритмів, квантові комп’ютери у скорому майбутньому дозволять ефективно підбирати паролі і ключи симетричного шифрування. На даний момент передбачається двократний ріст ефективності вирішення подібних задач порівняно з класичним комп’ютером. Це означаю, що довжину ключів в використовуваних симетричних алгоритмах варто збільшити в два рази. Що стосується найпопулярнішого симетричного алгоритму шифрування AES, доцільним виглядає збільшення розміру ключів до 256-біт.

Базуючись на вищенаведеній інформації можна впевнено говорити про перспективність квантових обчислень, які надають вагоме прискорення в вирішенні цілого класу задач. Тим не менш, варто також зауважити, що в даний момент, як проектування і створення квантових комп’ютері, так і написання програм під них є досить складними задачами.

Попри це, треба розуміти, що квантові обчислення допомагають ефективно вирішувати тільки певний спектр задач. З цього виходить, що квантові комп’ютери не є панацеєю і не є універсальним методом для подолання закону Мура про фундаментальні межі зменшення компонентів комп’ютерних систем.

Згідно з останніми дослідженнями, природний паралелізм, що надають квантові комп'ютери і нейрообчислювачі дозволяє значно збільшувати ефективність обчислень завдяки паралельній обробці інформації з величезною кількістю каналів (потоків), що приблизно дорівнює 106 і виконувати, таким чином, до 1013-1015 операцій в такт, що значно більше за максимальне число операцій в такт в інших комп’ютерних системах.

Імплементація квантових алгоритмів та побудова квантових систем - це дуже складна і трудомістка задача, через велику кількість проблем пов’язаних з ефектами квантової механіки. Серед таких проблем руйнування стану суперпозиції кубітів, під час виконання обчислень, внаслідок спостереження, неможливість забезпечення дебагу квантових додатків під час виконання, в силу вже згаданого ефекту спостереження, неможливість копіювання стану квантових регістрів, складності при

створенні та управлінні квантовими процесами всередині квантового комп'ютер. Тим не менш, навіть попри ці складності, уже існують багато реалізацій квантових комп’ютерів. Лідерами цього ринку є таки компанії Google, IBM, Intel та Microsoft. Особняком стоїть D-Wave Systems з її вузькоспеціалізованими рішеннями. Окрім цього з’явилися зручні емулятори для навчання та досліджень, такі як QCP від Google, або QDK від Microsoft. Зважаючи на це, на «паралельному фронті» активно проходить розробка та тестування пост квантових криптографічних алгоритмів та протоколів.

1.4 Постановка задачі

Виходячи з проведеного аналізу предметної галузі метою даної роботи є дослідження квантового алгоритму Шора для пошуку дискретного логарифму та його порівняння з не-квантовими аналогами.

Однією з основних задач дослідження є оцінка ефективності роботи алгоритму Шора для пошуку дискретного логарифму за умови використання сучасних інструментів для квантових обчислень. Окрім цього в рамках роботи повинні бути виконані наступні завдання:

– дослідження NP - повних задач, пов’язаних з криптографією та дискретним логарифмуванням;

– дослідження особливостей побудови квантових комп’ютерів та особливостей квантових обчислень;

– аналіз алгоритмів дискретного логарифмування для до та пост квантового періоду;

– реалізація та порівняльний аналіз реалізованих алгоритмів.

2 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ КВАНТОВИХ КОМП'ЮТЕРІВ ТА КВАНТОВИХ АЛГОРИТМІВ ДЛЯ ДИСКРЕТНОГО ЛОГАРИФМУВАННЯ

2.1 Особливості квантових комп'ютерів та обмеження, що вони накладають на квантові алгоритми

Головний компонент на якому будуються квантові комп'ютери, квантовий біт або скорочено кубіт (q-біт) - це квантова частинка, що описується ймовірнісною функцією, і з певною ймовірністю може перебувати у одному з двох станів 0 та 1. Ці два стани, або значення кубіта, може базуватися на стані атома (основному і збудженому), напрямку струму в надпровідному кільці, напрямі спина атомного ядра (вгору або вниз), двох можливих положення електрона в напівпровіднику і т. п.

Квантовий регістр має побудову досить схожу на побудову класичного регістру. Він складається з ланцюжка квантових бітів. Над цими кубітами можна проводити одно- і двох-бітові логічні операції, подібні до операцій 2I-НЕ, НЕ і т.і.

До базових станів квантового регістра, що утворений з L кубіт, відносять усі можливі комбінації нулів і одиниць довжиною L. Таким чином виходить 2^L різних комбінацій, які, по суті, уявляють собою запис десяткових чисел від 0 до 2^L -1 в двійковій формі. Однак, якби квантовий комп'ютер мав лише подібні стани, він майже нічім не відрізнявся від класичного комп'ютера. Окрім описаних базових станів існують і інші значення квантового регістра. Інші значення з'являються завдяки існуванню станів суперпозиції, які задаються комплексними амплітудами, пов'язаними умовою нормування і вектори у ймовірнісному просторі. Таким чином усі стани квантового регістра за винятком базових не мають класичних аналогів. Тож кількість його станів значно перевищує таку на класичному комп'ютері.

Основна ідея квантових обчислень, яку вперше висловили і описали Ю.І. Маніним і Р. Фейнманом, полягає у тому, що квантова система з L кубітів (квантових елементів) має 2^L лінійно незалежних станів, і це означає, що з урахуванням принципу

квантової суперпозиції, простором станів такого квантового регістра є 2^L -мірний ймовірнісний гільбертовий простір. Кожній операції над квантовим регістром відповідає поворот вектору стану регістра в гільбертовому просторі. Внаслідок таких властивостей, квантовий комп'ютера з L кубітами, що знаходяться у суперпозиції, може виконувати паралельно 2^L операцій. Загалом квантова система з L кубітів на борту, має 2^L класичних станів (00000 (L -нулів), 00001 (L -цифр), ..., 11111 (L - одиниць)), кожен з яких може бути отриманий під час вимірювання з вірогідністю від 0 до 100%.

$$|\psi\rangle = \sum_{n=0}^{2^L-1} c_n|n\rangle, \qquad (2.1)$$

де |n> - базисні квантові стани (наприклад, стан |001101>;

$|c_n|^2$ – це ймовірність отримання базисного стану |n>, як результату вимірювання

З вищеописаного виходить, що одна операція над групою кубітів впливає на вірогідності усіх можливих (базисних) значень, що і забезпечує абсолютний паралелізм обчислень.

Під час квантових обчислень, квантова програма уявляє собою послідовність квантових логічних операцій (гейтів) над введеними даними, які з математичної точки зору описуються унітарним перетворенням (матрицями певної розмірності), які впливають на ймовірнісне поле станів усього регістру, що можна спостерігати на рисунку 2.1. Як результат такої властивості, вихідний квантовий стан квантового комп'ютера стає новою суперпозицією через усього кілька тактів роботи [9].

Окрім цього, для квантового комп'ютера можна побудувати необмежену кількість перетворень, що не мають класичних аналогів.

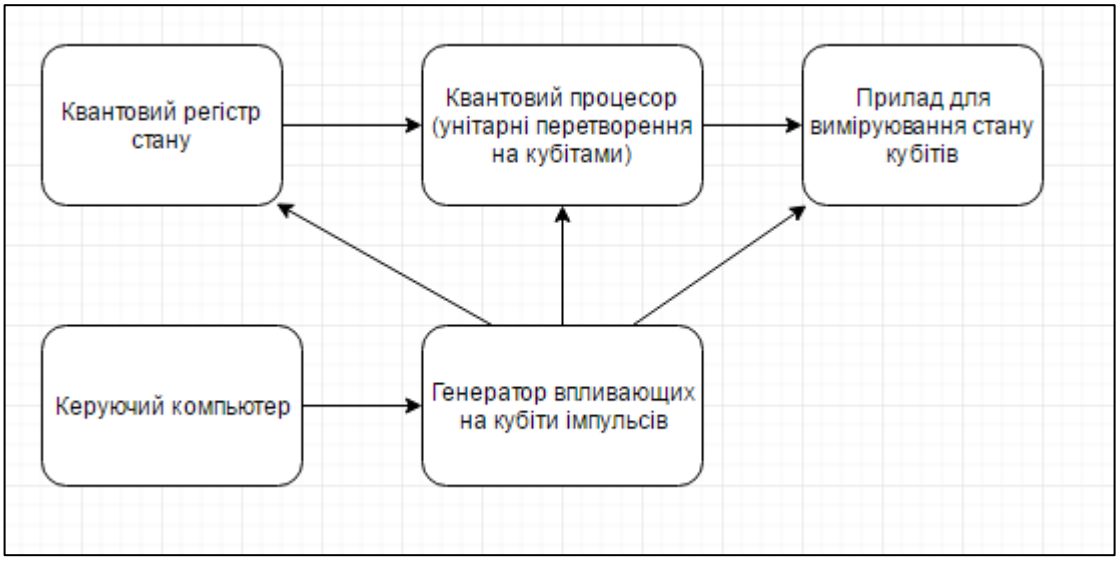

Рисунок 2.1 – Архітектура квантового комп’ютера

Обчислювальні перетворення в квантових обчисленнях прийнято називати вентилями або гейтами. Для регістрів у класичному комп’ютері існує тільки один однобітний гейт, а саме базисне перетворення not.

В той же час для квантового регістру існує набагато більше однокубітних перетворень. Серед таких перетворень, для яких існує проста фізична інтерпретація, можна виділити наступні перетворення: S-фазовий гейт, π / 8-гейт, H-гейт Адамара (2.2).

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$
$$R = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \qquad (2.2)$$

де H - гейт Адамара;

S - фазовий гейт;

R - π / 8-гейт.

Для визначення стану n-розрядного регістру необхідно тензорно помножити n кубітів регістру.Так, наприклад, дано два кубіта у станах| Q_1> і | Q_2>, тоді для визначення стану 2-розрядного регістра необхідно розрахувати їх тензорний добуток (2.3).

$$|Q_{12}>=|Q_1>\otimes|Q_2>, \quad (2.3)$$

де |Q_1> та|Q_2> - певні стани регістра;

|Q_{12}> - загальний стан регістра.

Якщо кубіти знаходяться у зплутаному стані то описання стану регістра за допомогою тензорного добутку стає неможливим. Саме завдяки таким зплутаним станам квантові обчислення придатні для вирішення складних задач і саме завдяки ним забезпечується квантовий паралелізм.

Загалом же, стан регістра з двох кубітів описується формулою (2.4).

$$|Q_{12}>=a|00>+b|01>+c|10>+d|11>, \quad (2.4)$$

де a,b,c та d – деякі коефіцієнти ймовірності при можливих станах;

|Q_{12}> - загальний стан регістра.

Ввівши умови нормування і ототожнення представлені формулою (2.5).

$$|ij> \equiv |i> \otimes |j>, \quad (2.5)$$

де |i> та |j> - певні стани регістра;

|ij> - загальний стан регістра.

Маємо, що умовою невиконання є нерівність (2.6).

$$AD - BC \neq 0, \quad (2.6)$$

де A,B,C та D – деякі квантові стани системи.

Умова невиконання еквівалентна тому, що стан регістру | Q_{12}> заплутаний Будь-який регістр у незаплутаному стані можна перетворити у регістр зі заплутаним станом використовуючи універсальний квантовий гейт, такий, що після його виконання сума ймовірностей всіх станів зберігається [10].

З опису квантових обчислювальним систем випливає те, що квантові алгоритмі маються деякі цікаві особливості. Базовою такою особливістю є те, що усі операції над кубітами оперують ймовірностями і усі обчислення мають ймовірнісний характер.

Зазвичай квантові алгоритми поділяють на три наступних класи:

– алгоритми для моделювання квантових систем та систем мікросівту;
– алгоритми пошуку на квантовому комп’ютері;
– алгоритми, які засновані на використанні квантового перетворення Фур'є.

Для кожної з перелічених задач квантовий комп’ютер дозволяє домогтися великого прискорення, яке досягається завдяки використанню, в обчисленнях, квантового паралелізму. Квантовий паралелізм робить можливим обчислення функції f (x) для великої кількості значень x одночасно [11].

2.2 Головні проблеми квантових обчислень

Квантова теорія, незважаючи на квантовий паралелізм, що полегшує роботу укладачів програм та алгоритмів, трохи обмежує їхні сфери діяльності. Вона створює певні проблеми, основні з яких ми розглянемо нижче.

2.2.1 Проблема клонування стану регістра на квантовому комп’ютері

Копіювання або передача інформації в регістр або між двома точками пам’яті є однією з базових операцій класичного комп’ютера. Але у квантовій теорії існує "Теорема про заборону клонування" , яку сформулювали Вуттерс, Зурек та Дієкс [12] у 1982 році. Вона стверджує, що створення ідеальної копії довільного невідомого квантового стану є неможливим, та є фундаментальною у площині знань щодо квантової теорії інформації, квантових обчислень та суміжних.

Явище квантової телепортації допомагає обійти зазначену вище проблему. Цей термін вперше був згаданий у статті в журналі "Physical Review Letters" за 1993 рік. Саме там є чіткий опис квантового явища, що запропонували назвати "телепортацією" (англ. Teleporting), та його відмінності від досить популярного поняття "телепортації" серед авторів наукової фантастики. Квантова телепортація у науковому сенсі не може використовуватися як канал передачі енергії або речовини на будь-які відстані[13]. Замість того йде передача квантового стану на відстань за допомогою заплутаної пари та класичного каналу зв’язку. При такій телепортації квантовий стан руйнується під час проведення вимірювання у точці відправлення, та потім відтворюється у точці прийому.

Окрім передачі квантової інформації за допомогою квантового каналу, за класичним каналом обов’язково передається інформація, без якої повідомлення не може бути прочитано. Кореляції Ейнштейна - Подільського - Розена, що характерні для квантово-заплутаних часток використовуються для передачі “квантової” інформації. Класичну інформацію можливо передати через звичайні канали зв’язку.

Зараз ми розглянемо досить простий приклад. Візьмемо квантову систему, що має 2 можливих стана $\psi 1$ та $\psi 2$ Для простоти розглянемо квантову систему з двома можливими станами $\psi 1$ і $\psi 2$ (проекції спіна фотона або електрона на вісь). Не зважаючи на те, що системи такого зразка мають назву “кубіти”, за допомогою

способу, який описаний нижче можна передати стан будь-якій системі з кінцевою кількістю станів.

Перейдемо до розгляду опису квантової телепортації. Наприклад, відправник має частку A, вона знаходиться у довільному квантовому стані $\psi A = \alpha\psi 1 + \beta\psi 2$, та відправнику треба передати цей квантовий стан одержувачу. Мета відправника зробити так, щоб у розпорядженні одержувача опинилася частка B у ідентичному стані. Так, необхідно з максимальною точністю зробити трансфер значень двох комплексних чисел α і β. Головний критерій — не швидкість передачі інформації, а точність. Щоб досягти цієї мети, послідовно виконуються такі кроки:

– одержувач та відправник мають попередню домовленість про створення часток C та B, що є квантово заплутаними. Частка C потрапить до відправника, а частка B до одержувача. Обидві частки є квантово заплутаними, тож ступені свободи усієї пари часток описуються єдиним чотиривимірним вектором стану ψBC;

– загалом, квантова система частинок A і C має чотири стани, але її стан неможливо описати вектором, тому що повністю визначеним, або чистим станом володіє лише система з трьох частинок A, B, C. В той час, як відправник проводить вимірювання над системою з двох частинок A і C він отримує одне з 4 можливих значень вимірюваної величини. Така як в ході виміру система з трьох частинок A, B, C колапсує в певний новий стан і стани частинок A і C стають повністю відомі, - заплутаність часток руйнується і частка B переходить до деякого певного квантового стану;

– саме перехід частки B до певного стану можна назвати процесом передачі "квантової частини" інформації, або ж квантовою телепортацією Тим не менш, на цьому етапі дізнатися передану інформацію неможливо. Уся доступна одержувачеві інформація, це інформація про зв’язок між станами частинок B та A, проте природа цього зв’язку, одержувачу, невідома;

– для того щоб одержувач отримав змістовну інформацію збережену у B, відправник повинен повідомити йому результат свого вимірювання використовуючи

класичні канали зв’язку. При цьому два біта, відповідні зачепленому стану AC втрачаються. Таким чином, слідуючи законам квантової механіки, приходимо до висновку, що знаючі стани частинок A і C, отримані шляхом вимірювання проведеного над парою цих частинок, і маючи частинку B заплутану з часткою C, одержувач матиме змогу зробити певні перетворення над станом частинки B і дізнатися початковий стан A.

Обмеженням такого виду передачі інформації є те, що одержувач зможе остаточно отримати інформацію тільки після того, як матиме доступ до даних, отриманими по обох каналах (класичному та квантовому), тобто одержувач не зможе дізнатися інформацію передану квантовим шляхом до того як отримає необхідні дані за класичним каналом.

Це явище отримало назву квантової телепортації (а не просто передачі даних) внаслідок того, що початковий стан частинки A руйнується після проведення над ним усіх необхідних дій. Внаслідок чого, можна сказати, що стан частинки A був не скопійований, а телепортований (перенесений) з одного місця в інше [14].

2.2.2 Проблема декогеренції

Теорія квантових обчислень розрахована на те, що квантові суперпозиції, якими описується стан L-кубітного регістру, в процесі обчислень, завжди залишаються когерентними. Проте, фізичні реалізації квантових комп’ютерів піддаються таким явищам, як: неточності в значеннях параметрів керуючих імпульсів, вплив неконтрольованого оточення на регістри, а також хаотична взаємодія між кубітами в регістрах. Ці явища призводять до декогеренції квантового стану регістрів. Це означає, що когерентний, тобто узгоджений стан системи) перетворюється в змішаний, хаотичний стан, для опису якого, використовують матрицю щільності.

Перехід до опису системи матрицею щільності означає, що інформації про фази базисних станів зникає, і позбавляє елементи квантової системи змоги інтерферувати і заплутуватися. Простими словами, декогеренцію можна описати, як втрату системою її квантових властивостей та перехід роботи системи з площини квантової у площину класичної фізики.

Враховуючи вищеописане час за який починається декогеренції стану квантового комп’ютера, вважається важливою властивістю. Декогеренція квантових станів є однією з головних перешкод на шляху побудови великого практичного квантового комп’ютера. Особливо критичним є те що час за який квантова система втрачає когерентність має обернено пропорційну залежність до кількості операцій виконуваних на цим станом. Тобто для боротьби з декогеренцією треба, або скорочувати час виконання операцій, або ж збільшувати час за який відбувається декогеренція кубіта (тобто збільшувати резистентність кубітів до декогеренції). Тому вчені зараз активно шукають способи утримання квантового комп'ютера в когерентному стані протягом тривалого часу, щоб отримати можливість обчислення будь-якої задачі з великим (але поліноміальним) алгоритмом обчислень. На сьогодні головним методом для цього є метод квантової корекції помилок.

Цей метод полягає в періодичному очищенні стану квантового комп'ютера від малих помилок, які накопичуються в векторі стану через процеси декогеренції. Завдяки такому процесу очищення стан регістру квантового комп’ютера не деградує до того ріквня, коли будь-які обчислення преестають бути доцільними і ведуть лише до хибних реузльтатів. Окрім цього метода, зараз ведеться дослідження активних методі боротьби з декогеренції. Тим не менш квантовий метод корекції помилок залишається головною надією дослідників на можливість побудови повномасштабного квантового комп'ютера, що працює в когерентному стані скільки завгодно тривалий час [15].

2.2.3 Ймовірнісний характер квантових обчислень

У квантових комп’ютерах стан системи описується суперпозицією, що означає наявність багатьох дискретних станів, пов’язаних за певними ймовірностями, одночасно. Це призводить до того, що правильний результат виконання алгоритму теж можна отримати лише з деякою ймовірністю, нехай і відмінною від 0. І це означає, що правильний результат можна отримати далеко не при кожному запуску квантового алгоритму. Зважаючи на це, можна виділити два способи вирішення цього питання і прискорення отримання правильного результату. Потрібно чи створювати такі квантові алгоритми, які будуть приводити до правильних результатів з досить високою ймовірністю і таким чином зменшать кількість запуску алгоритму, необхідну для отримання правильного результату, чи змиритися з необхідністю багаторазового запуску алгоритму, і після кожного запуску перевіряти результат на правильність. Другий спосіб підходить тільки для задач, вирішення яких можна перевірити за поліноміальний час. До таких задач, наприклад, відносяться задачі класу складності NP.

2.2.4 Проблема вимірювання поточного стану системи

До проблеми вимірювання поточного стану квантової системи призводить парадокс Ейнштейна - Подільського - Розена. Парадокс полягає в тому, що координатна частинки і її імпульс не можуть бути виміряні в один і той же час відповідно до співвідношення невизначеності Гейзенберга. Якщо припустити, що подібний парадокс має місце тому, що при вимір однієї з величин приводить до принципово непереборних обурень у стані частинки і ці обурення радикально

змінюють значення іншої величини, то прийти до гіпотетичного способу, який дав би змогу обійти співвідношення невизначеності і власне парадокс. Спосіб полягає в наступному. Нехай дві однакові частинки А і В були утворені в результаті розпаду третьої частинки С., Тоді їх сумарний імпульс p_A+p_B має дорівнювати вихідному імпульсу частинки С p_c за законом збереження імпульсу, іншими словами, імпульси частинок А і В мають бути пов'язані. Це надає нам змогу дізнатися імпульс однієї з частинок, нехай це буде А, і, використовуючи закон збереження імпульсу, розрахувати імпульс частинки В ($p_B=p_C-p_{A)}$, не призводячи при цьому ні до яких обурень в її русі. Якщо тепер виміряти координату частинки В, то ми отримуємо значення двох одночасно невимірних величин для цієї частинки, а це неможливо за законами квантової механіки.

З цього виходить, що, можливо, співвідношення невизначеності не є абсолютним, а закони квантової механіки не повні, не відображають усієї картини реального світу і повинні бути уточнені і доповнені у майбутньому.

Якщо ж припустити, що закони квантової механіки в даному випадку не порушуються, це призводить до твердження, що вимір імпульсу однієї з таких частинок еквівалентний виміру імпульсу другої з них.

Проте це твердження веде до протиріччя з принципом причинності, так як походить на явище миттєвого впливу однієї з частинок ні іншу.

При побудові квантових комп’ютерів вищенаведені гіпотези і парадокси призводять до того, що перехоплення стану квантових регістрів, під час виконання програм, стає неможливим без втручання в роботу програм і пошкодження стану квантових регістрів, що може призвести до руйнування стану суперпозиції та квантової заплутаності і припинення роботи програм. Через це, процес відлагодження (або ж дебагу), звичний на класичних комп’ютерах, на квантових комп’ютерах стає неможливим.

2.3 Фізична реалізація квантових комп’ютерів

До фізичних реалізацій квантових комп’ютерів, наразі висувають наступні вимоги :

– стійкість до декогеренції, яка виражається в тому що програма складена з квантових гейтів (вентилів) виконується швидше за час наступу критичної декогеренції. На даному етапі для виконання цієї вимоги час за який наступає декогеренція повинен в 10^4 або більше разів перевершувати час виконання основних квантових гейтів (час такту). При цьому допустима помилка обчислень при виконанні окремого гейту має бути менше або рівною 10^{-4};

– архітектура квантового комп’ютера має бути розрахована для масштабування, тобто збільшення числа кубітів. Окрім цього архітектура повинна дозволяти виділення і фіксацію у просторі великого числа ($10^2/10^3$) 2-х рівневих частинок - кубітів, які можуть бути піддані певному впливу для організації їх квантової еволюції у відповідності з виконуваним квантовим алгоритмом;

– підтримка повного набору гейтів за Тюрінгом. Враховуючі, що усі унітарні квантові операції можуть бути зведені до сукупності одно- і дво-кубітних операцій, при розробці квантової системи необхідно гарантувати наявність певних нелінійних взаємодій між керованими кубітами, які б забезпечили виконання дво-кубітних операцій. Точність контролювання імпульсами, що керуються операціями, повинна бути не менше ніж 10^4;

– можливість ініціалізації кубітів у певний базовий стан. Тут мається на увазі можливість встановлення L-кубітного вхідного регістра в вихідний базовий стан| O_1, O_2, O_3 ,…., O_L > (ініціалізація);

– можливість зчитування стану. Підтримка операції вимірювання стану на виході квантової системи з високою точністю.

Цими днями відомо кілька варіантів реалізації фізичних систем, які можна вважати базою для квантових комп’ютерів:

– системи з використання квантових точок, що мають два електронних орбітальних та спінових стана;

– системи, що базуються на використанні надпровідникових структур з переходами Джозефсона;

– системи на базі нейтральних атомів або іонів з двома низьколежачими коливальними або надтонкими рівнями, які утримуються в силових пастках, що створені в вакуумі за допомогою електромагнітних полів, з використання лазерного охолодження до температур, що близькі до абсолютного нуля;

– системні створені на основі певних станів квантового електромагнітного поля в електродинамічних резонаторах і фотонних кристалах;

– системи на основі окремих електронних та ядерних спінів в магнітному полі.

3 АНАЛІЗ АЛГОРИТМІВ ДИСКРЕТНОГО ЛОГАРИФМУВАННЯ ДЛЯ ДО ТА ПОСТ КВАНТОГО ПЕРІОДІВ

3.1 Класичні алгоритми пошуку дискретного логарифму

У криптографії атака є методом вирішення проблеми. Більш конкретно, мета атаки - знайти швидкий спосіб вирішення проблеми, на якій базується алгоритм шифрування. Відомі способи атаки на проблему дискретного логарифму еліптичних кривих, які працюють для всіх кривих є повільними, тим самим роблячи шифрування на основі цієї проблеми практичним.

Проте відомо кілька ефективних методів вирішення проблеми дискретного логарифму на еліптичних кривих для певних типів еліптичних кривих. Це означає, що під час вибору еліптичної кривої для шифрування необхідно переконатися, що обрана крива не входить до одного з класів для яких проблема дискретного логарифму може бути вирішена достатньо швидко.

Розглянемо декілька алгоритмів для злому дискретного логарифму у загальному випадку та на еліптичних кривих:

– алгоритм Гельфонда - Шенкса [16].

Ідея алгоритму полягає у виборі оптимального співвідношення часу і пам'яті, а саме в удосконаленому пошуку показника ступеня. Нехай задана циклічна група G порядку n, генератор групи α і деякий елемент групи β. Завдання зводиться до знаходження цілого числа x, для якого виконується $a^x = \beta$. Алгоритм Гельфонда - Шенкса заснований на поданні x у вигляді x = i · m - j, де $m = \lceil \sqrt{n} \rceil + 1$,і переборі 1 ≤ i ≤ m і 0 ≤ j <m. Обмеження на i і j випливає з того, що порядок групи не перевищує m, а значить зазначені діапазони достатні для отримання всіх можливих x з напівінтервала [0; m). Таке уявлення рівносильно рівності $a^{im} = \beta a^j$ (1). Алгоритм попередньо обчислює a^{im} для різних значень i і зберігає їх в структурі даних, що дозволяє ефективний пошук, а потім перебирає всілякі значення j і перевіряє,

якщо βa^j відповідає якомусь значенню i. Таким чином знаходяться індекси i і j, які задовольняють співвідношенню (1) і дозволяють обчислити значення x = i · m – j.

– алгоритм Менезеса-Окамото-Ванстона.

Ідея алгоритму полягає в переході від проблеми дискретного логарифму на еліптичній кривій $E(F_q)$ до проблеми дискретного логарифму в $F_{q^m}^x$ для деяких m. Після цього переходу, проблема може бути вирішена досить швидко, використовуючи атака обчислення індексу, якщо m - не велике. Невеликі m завжди можуть бути отримані для певних типів кривих. Почнемо з визначення групування Вейля для кривих $E(F_q)$.

– ρо-алгоритм Полларда для дискретного логарифму [17]

Алгоритм Полларда є імовірнісним алгоритмом, що дозволяє знаходити дискретний логарифм виду $Q = xP$, де $0 < x < n, n -$ порядок групи, і працює зі складністю, що залежить лише від величини модуля. Примітний також тим, що існує варіант реалізації такого алгоритму, який має більш складну ітераційну функцію, яка призводить до пришвидшення алгоритму;

– алгоритм Поліґа-Геллмана [18]

Спеціалізований алгоритм, який отримує перевагу від факторизації порядку n мультиплікативної групи порядку G , де n – гладке число, нехай $n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$, буде розкладом на прості множники. Якщо $x = \log_\alpha \beta$, тоді підхід до вирішення задачі полягає в знаходженні $x_i = x \; mod \; p_i^{e_i}$, для $1 \le i \le r$ і подальшому розрахунку x. Кожен з x_i визначається обчисленням $l_0, \; l_1, \dots, l_{e_i - 1}$ для його $p_i -$ арного представлення $x_i = l_0 + l_1 p_i + \dots + l_{e_i - 1} p_{e_i - 1}$, де $0 \le l_j \le p_i - 1$.

Передбачувана велика обчислювальна складність завдання пошуку дискретного логарифму лежить в основі крипостійкості деяких алгоритмів шифрування з відкритим ключем, таких як ECDH та ECDSA.

3.2 Алгоритм пошуку дискретного логарифму Шора та його варіації.

Дискретний алгоритм Шора [19][20] – квантовий алгоритм пошуку дискретного логарифму, квантова реалізація якого, дозволяє знайти дискретний логарифм у групі з порядком n бітів за час $O(n^3)$, використовуючи O(6n) логічних кубітів. Алгоритм був розроблений Пітером Шором у 1994 році.

Обидва алгоритма Шора (для факторизації та дискретного логарифму) пізніше почали розглядатися, як спеціальні випадки більш загального фреймворку, який називається проблема схованої абеліанівської підгрупи. В той час, як в алгоритмі факторизації розглядається підгрупа групи цілих чисел Z, у випадку дискретного логарифму розглядається підгрупа Z^2. Зокрема, просторова підрешітка решітки Z^2, така, що її елементи можуть бути записані як лінійна комбінація двох (лінійно не залежних) векторів в Z^2. Таким чином, алгоритм для дискретного логарифму можна розглядати, як двовимірну версію алгоритму факторизації.

Для алгоритму дискретного логарифма над еліптичними кривими, перш за все робиться припущення, що період базової точки α еліптичної кривої – це просте число і воно відоме до початку роботи алгоритму. Якщо період α не відомий, то його можна знайти за допомогою алгоритму пошуку періоду, що використовується в алгоритму факторизації. У таблиці 1.2 наведено приклад періодичної функції на еліптичній кривій для базової точки α.

Таблиця 3.1 – Значення періодичної функції для різних n (n α) на кривій $y^2 = x^3 + 2x + 2 \pmod{17}$ з початковою точкою $\alpha = (5,1)$

n		1	2	3	4	5	6	7	8
n α mod p	n(5,1) mod 17	(5,1)	(6,3)	(10,6)	(3.1)	(9,16)	(16,13)	(0,6)	(13,7)

Кінець таблиці 3.1

n		9	10	11	12	13	14	15	16
n α mod p	n(5,1) mod 17	(7,6)	(7,11)	(13,10)	(0,11)	(16,4)	(9,1)	(3,16)	(10,11)
n		17	18	19	20	21			
n α mod p	n(5,1) mod 17	(6,14)	(5,16)	(0,0)	(5,1)	(6,3)			

Алгоритм пошуку дискретного логарифма полягає в наступному: нехай задано $\alpha^q = e$, де q – просте число і $\beta = \alpha^d$, де d – не відоме і знаходиться у проміжку від 0 до q – 1. Розглянемо функцію $f(x, y) = \alpha^x \beta^y$ для цілих чисел x та y. Ця функція має два незалежні періоди на площині Z^2, що видно з функції 3.1.

$$f(x + q, y) = f(x, y) \text{ та } f(x + d, y - 1) = f(x, y) \qquad (3.1)$$

Таким чином усі x, y разом з $f(x, y) = e$ формують просторову підрешітку від Z^2.

Двовимірне перетворення Фур'є приводить до подвійної решітки з якої можна визначити d. Треба зауважити, що $f(x, y)$ можна роздивлятися, як функцію визначену над Z_q^2, як $f(x, y) = f(x \bmod q, y \bmod q)$.

Далі, спочатку уявимо, що нам відомий спосіб виконати квантове швидке перетворення Фур'є з періодом q ($QFFT_q$), тому, що у цьому випадку алгоритм міг би виглядати краще.

Тоді алгоритм починається з наступного стану двох квантових регістрів, з яких, за допомогою «квантового паралелізму» вираховується $\alpha^x \beta^y$ як видно з формули (3.2).

$$\frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1}|x,y> \ \rightarrow \frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1}|x,y,\ \alpha^x\beta^y> \qquad (3.2)$$

Знову уявимо, що ми зараз виміряємо значення останнього регістра. У такому разі ми отримаємо випадковий елемент α^{x_0} з групи згенерованої α, при цьому значення x_0 знаходиться у проміжку від 0 до q – 1. Тоді необхідно знайти два перші регістри у суперпозиції всіх x, за формулою (3.3).

$$\alpha^x\beta^y = \alpha^x(\alpha^d)^y = \alpha^{x_0} \qquad (3.3)$$

Так як період α це q, цей вираз еквівалентний формулі (3.4).

$$x + dy \equiv x_0\ (mod\ q), \text{або } x = (x_0 - dy)\ mod\ q \qquad (3.4)$$

Тож для y існує тільки одне рішення і стан двох перших регістрів наступний відображено у формулі (3.5).

$$\frac{1}{\sqrt{q}}\sum_{y=0}^{q-1}|x_0 - dy, y> \qquad (3.5)$$

Тепер варто виконати перетворення Фур'є над кожним з двох регістрів використовуючи наше (гіпотетичне) квантове перетворення Фур'є періоду q, яке діє на базові стани згідно з формулою (3.6).

$$|z> \rightarrow \frac{1}{\sqrt{q}}\sum_{\dot{z}=0}^{q-1}|\omega_q^{z\dot{z}}, \dot{z}>, \qquad (3.6)$$

де $\omega_q = e^{2\pi i/q}$

Після перетворення формули (3.6) ми отримаємо формулу (3.7).

$$\frac{1}{\sqrt{q}}\frac{1}{q}\sum_{\dot{y}=0}^{q-1}\sum_{y=0}^{q-1}\omega_q^{(x_0-dy)\dot{x}}\omega_q^{y\dot{y}}|\dot{x},\dot{y}> \tag{3.7}$$

Тепер можна розрахувати суму по у. Вона дорівнює $q\omega_q^{x_0\dot{x_0}}$, якщо $q\omega_q^{x_0\dot{x_0}}$ і зводиться до 0 у іншому випадку. Таким чином, після розрахунку, ми отримаємо формулу (3.8).

$$\frac{1}{\sqrt{q}}\sum_{\dot{x}=0}^{q-1}\omega_q^{x_0\dot{x}}|\dot{x},\dot{y}=d\dot{x}\ mod\ q> \tag{3.8}$$

Тепер можна побачити, що вірогідність виміру базового стану не залежить від x_0, і тому не має значення яке x_0 ми виміряли вище. Після вимірювання ми отримуємо пару $\dot{x},\dot{y}$, з якої ми можемо розрахувати $d=\ \dot{y}(\dot{x})^{-1}mod\ q$, якщо $\dot{x}\ \neq 0$.

На практиці нам необхідно замінити кожне $QFFT_q$ на швидке квантове перетворення Фур'є з періодом $2^n(QFFT_{2^n})$, тому що його можна фінітно реалізувати. Для розглянутого вище $QFFT_q$, ми завжди можемо отримати пару $\dot{x},\dot{y}$ з $\dot{y}\ \equiv\ d\dot{x}\ mod\ q$ при фінальному вимірі. Однак для $QFFT_{2^n}$ ми отримаємо велику вірогідність виміряти пару $\dot{x},\dot{y}$, якщо умова у формулі (3.9) виконується.

$$\left(\frac{\dot{x}q}{2^n},\frac{\dot{y}q}{2^n}\right)\approx(k,dk), \tag{3.9}$$

де k – деяке ціле число

Для $2^n\approx q$ ми маємо хорошу (константну) вірогідність отримати правильні значення в Z_q^2 після округлення.

Окрім алгоритму Шора над простим полем F_p^* існують різні модифікації алгоритму для пошуку дискретного логарифму в особливих випадках, або ж над іншими скінченними групами. Як приклад такого алгоритму можна привести алгоритм для розрахунку дискретного логарифму d у групі F_p^* для випадку де $d \lll q$ [21]. Цей алгоритм складається з двох кроків:

– власне квантовий алгоритм, який приймає на вхід генератор групи g і елемент $x = [d]g$ і повертає, як результат пару (k, j) і $[e]g$, що ігнорується;

– класичний алгоритм, що приймає на вхід пару (k, j) і повертає, як результат шуканий дискретний логарифм d, якщо пара «хороша», тобто відповідає певним вимогам.

Цей алгоритм потребує $2[\log_2 q]$ регістрів для індексу і обчислення двох квантових перетворень Фур'є розміру $2^{[\log_2 q]}$. Теоретична нижня границя вірогідності отримання шуканого дискретного логарифму з першого запуску дорівнює 2^{-10}.

Зокрема, ми розглянемо особливий випадок, де $q \geq 2^{2\ell} - (2^{\ell} - 1)d$ і де d - на інтервалі $0 < d < 2^{\ell}$ для якогось цілого числа ℓ.

– нехай $|\Psi\rangle = \frac{1}{\sqrt{2^{3\ell}}} \sum_{a=0}^{2^{2\ell}-1} \sum_{b=0}^{2^{\ell}-1} |a\rangle|b\rangle|0\rangle$ де перший і другий регістри мають довжину мають довжину ℓ та 2ℓ відповідно;

– розрахуємо $[a]g \odot [-b]x$ і збережемо результат у третій регістр і отримаємо формулу (3.10).

$$|\Psi\rangle = \frac{1}{\sqrt{2^{3\ell}}} \sum_{a=0}^{2^{2\ell}-1} \sum_{b=0}^{2^{\ell}-1} |a, b, [a]g \odot [-b]x\rangle = \\ = \frac{1}{\sqrt{2^{3\ell}}} \sum_{a=0}^{2^{2\ell}-1} \sum_{b=0}^{2^{\ell}-1} |a, b, [a - bd]g\rangle \qquad (3.10)$$

– розрахуємо QFT розміру $2^{2\ell}$ на першому регістрі і QFT розміру 2^{ℓ} на другому регістрі і одержимо стан вираженний формулою (3.11);

$$|\Psi\rangle = \frac{1}{\sqrt{2^{3\ell}}}\sum_{a=0}^{2^{2\ell}-1}\sum_{b=0}^{2^{\ell}-1}|\mathrm{a}, b, [a-bd]g\rangle \xrightarrow{QFT}$$

$$\xrightarrow{QFT} \frac{1}{\sqrt{2^{6\ell}}}\sum_{a=0}^{2^{2\ell}-1}\sum_{b=0}^{2^{\ell}-1}\sum_{j=0}^{2^{2\ell}-1}\sum_{k=0}^{2^{\ell}-1} e^{2\pi i(aj+2^{\ell}bk)/2^{2\ell}}|\mathrm{j}, k, [a-bd]g\rangle \qquad (3.11)$$

– проведемо вимірювання для того щоб отримати пару (j, k) і $[e]g$.

Пара (j, k) використовується далі в послідовному алгоритмі для знаходження дискретного логарифму.

Таким чином, за допомогою вищенаведеної модифікаціїї алгоритму Шора можна дость швидко знайти дискретний логарифм для у випадку використання особливих груп для яких виконується наступна тотожність $q \geq 2^{2\ell} - \left(2^{\ell}-1\right)d$.

Окрім описаних модифікації існують і інші модифікаіїї алгоритму Шора для певних абелієвиї груп та еліптичних кривих, але їх опис виходить за рамки даної роботи.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ДЛЯ ОБЧИСЛЕННЯ ДИСКРЕТНОГО ЛОГАРИФМУ

4.1 Реалізація не квантових алгоритмів пошуку дискретного логарифму

Для реалізації класичних алгоритмів для пошуку дискретного логарифму було обрано мову програмування Java. Вибір мови було обумовлено попереднім досвідом використання і тим що Java зарекомендувала себе, як мова програмування що підходить для розробки, як невеликих алгоритмів, так і величезних програмних систем. Для тестування продуктивності було обрано інструмент JMH [22], який немає гідних альтернатив в Java екосистемі і забезпечує правильні умови для вимірювання продуктивності коду, як то:

– виконання циклів розігріву перед початком основного тестування, завдяки яким JIT-компілятор встигає приступити до роботи і тим самим забезпечується реалістичність тестування;

– створення окремого інстансу JVM для забезпечення ізоляції коду під час тестування;

– врахування статистичної похибки у отриманих результатах та багато іншого.

В наступних розділах описані реалізовані класичні алгоритми для пошуку дискретного логарифму.

4.1.1 Алгоритм Гельфонда – Шенкса

На вхід алгоритму подаються: G – циклічна група порядку n, генератор α і деякий елемент β

Алгоритм складається з наступних кроків:

а) Знаходимо m $\leftarrow \sqrt{n} + 1$

б) Обчислюємо $\gamma \leftarrow \alpha^m$.

в) Для будь якого i де $0 < i < m$:

1) Записати в таблицю (i, γ).

2) Вирахувати $\gamma \leftarrow \gamma \cdot \alpha^m$

г) Для будь-якого j.де $0 \leq j \leq m$:

1) Вирахувати α

2) Перевірити, чи знаходиться $\beta\alpha^j$ в таблиці

3) Повернути $im - j$ якщо так, або продовжити виконання циклу.

Теоретична складність цього алгоритму в найгіршому випадку дорівнює $O(\sqrt{n})$.

4.1.2 ρо-алгоритм Полларда для дискретного логарифма

Цей алгоритм виконує випадковий пошук дискретних логарифмів, як і алгоритм Гельфонда - Шенкса, але він потребує меншого об'єму пам'яті для збереження даних, тому кращій для практичних цілей.

В основі даного алгоритму лежить та ж ідея, що й в основі ро-алгоритму для факторизації – спочатку будується псевдовипадкова послідовність, в якій знаходяться співпадаючи елементи за допомогою алгоритму Флойда, а потім на базі отриманої величини вираховується шуканий дискретний логарифм.

Нехай треба розрахувати $\log_g a$ в кінцевій групі G порядку n

Група G розбивається на три непересічних підмножини приблизно рівної потужності: $G = S_1 U S_2 U S_3$, так щоб $1 \notin S_2$. Причому розбиття повинно бути побудоване таким чином, щоб перевірка, до якої підмножини належить даний елемент x, була простою.

Наприклад, якщо $G = Z_p$, де p – просте число, то можна задати розбиття $S_1 = \left\{1, \dots, \left[\frac{p}{3}\right]\right\}, S_2 = \left\{\left[\frac{p}{3}\right], \dots, \left[\frac{2p}{3}\right]\right\}, S_3 = \{\left[\frac{2p}{3}\right], \dots, p-1\}$, або розбиття може бути таким: якщо $x\ mod\ 3 = 1$, то $x \in S_1$, якщо $x\ mod\ 3 = 2$, то $x \in S_2$, якщо $x\ mod\ 3 = 0$, то $x \in S_3$.

Далі на G задається послідовність $x_0, x_1, x_2, \dots$, де $x_0 = 1$, а x_{i+1} вираховується по x_i за допомогою функції f для $i \geq 0$:

$$x_{i+1} = f(x_i) = \begin{cases} ax_i, & \text{якщо } x_i \in S_1 \\ x_i^2, & \text{якщо } x_i \in S_2 \\ gx_i, & \text{якщо } x_i \in S_3 \end{cases}$$

Обчислення проводяться у групі G, тобто якщо $G = Z_m$, то розрахунки треба робити по модулю m.

Така послідовність групових елементів може бути представлена двома послідовностями: $\{u_0, u_1, u_2, \dots\}$ та $\{v_0, v_1, v_2, \dots\}$ такими, що $x_i = g^{u_i}a^{v_i}$,1 при $u_0 = v_0 = 0$, і

$$u_{i+1} = \begin{cases} u_i, & \text{якщо } x_i \in S_1 \\ 2u_i, & \text{якщо } x_i \in S_2 \\ u_i + 1, & \text{якщо } x_i \in S_3 \end{cases}$$

$$v_{i+1} = \begin{cases} v_i + 1, \text{якщо } x_i \in S_1 \\ 2v_i, \text{якщо } x_i \in S_2 \\ v_i, \text{якщо } x_i \in S_3 \end{cases}$$

Обчислення в послідовностях u и v виконуються по модулю n.

В силу того, що група G – кінцева, за допомогою алгоритму Флойда можна знайти такі x_iи x_{2i}, что $x_i = x_{2i}$. Тогда $g^{u_i}a^{v_i} = g^{u_{2i}}a^{v_{2i}} \Rightarrow a^{(v_{2i}-v_i)} = g^{(u_{2i}-u_i)}$. Логарифмувавши за основою g обидві частини даного рівняння, отримуємо:

$$(v_i - v_{2i}) \log_g a \equiv (u_{2i} - u_i)(mod\ n)$$

Вирішивши це порівняння, отримуємо шуканий логарифм. (Варто зауважити, що якщо $G = Z_m^*$, то $n = \varphi(m)$.

Асимптотична складність даного метода дорівнює $O(\sqrt{n})$, где n – порядок групи G.

4.1.3 Алгоритм Поліґа-Геллмана

На вхід алгоритму подаються генератор α циклічної групи G порядку n і елемент β циклічної групи.

Алгоритм складається з наступних кроків:

а) Знайти розкладення на прості множники для n $n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$, де $e_i \geq 1$

б) Для i від 1 до r робимо наступне: (Обчислити $x_i = l_0 + l_1 p_i + \dots + l_{e_i-1} p_i^{e_i-1}$ де $x_i = x\ mod\ p_i^{e_i}$):

1) Покласти $\gamma \leftarrow 1$ і $l_{-1} \leftarrow 1$
2) Обчислити $\alpha \leftarrow \alpha n / p_i$
3) Обчислити l_j. Для j від 0 до $e_i - 1$ робимо наступне:
   - Обчислити $\gamma \leftarrow \gamma \alpha^{l_{j-1} p_i^{j-1}}$ і $\bar{\beta} \leftarrow (\beta \gamma^{-1})^{n/p_i^{j+1}}$
   - Обчислити $l_j \leftarrow \log_\alpha \beta$
4) Встановити $x_i \leftarrow l_0 + l_1 p_i + \dots + l_{e_i-1} p_i^{e_i-1}$

в) Використати, наприклад, алгоритм Гаусса для обчислення цілого x, $0 \leq x \leq n - 1$.такий що $x \equiv x_i\ (mod\ p_i^{e_i})$ для $1 \leq i \leq r$

г) Повернути (x)

Асимптотична складність цього алгоритму в найгіршому випадку дорівнює $O(\sqrt{n}))$, де n – порядок групи G.

4.1.4 Структура програмного коду

Вищеописані алгоритми були реалізовані з використанням мови програмування Java. Алгоритм прямого перебору було розпаралелено, так як в однопоточному варіанті його продуктивність є дуже низькою і не варта розгляду.

Алгоритми були реалізовані керуючись принципами і найкращими практиками ООП. Окрім безпосередньо алгоритмів, були реалізовані класи які містять код для генерації випадкових простих чисел та генерації примітивних коренів, а також класи для рішення систем конгруенцій, які використовуються у алгоритмі Пухлінга-Хелмана.

Розглянемо діаграму класів зображену на рисунку 4.1.

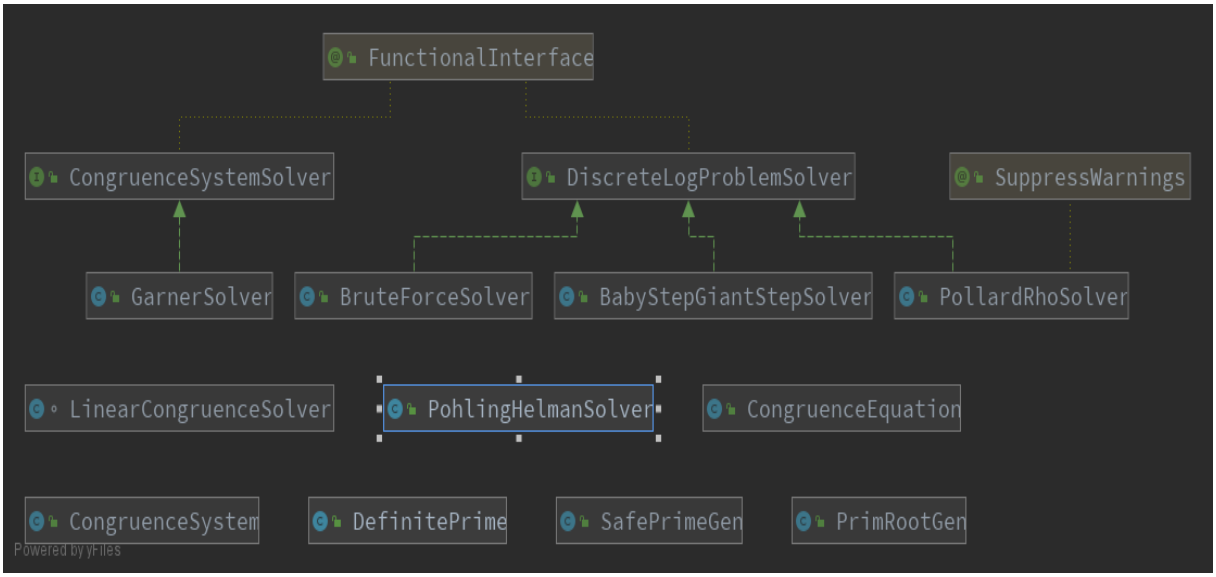

Рисунок 4.1 – Діаграма класів реалізації класичних алгоритмів для пошуку дискретного логарифму

З рисунку видно, що основна реалізована логіка була схована за двома інтерфейсами, а саме інтерфейсом DiscreteLogProblemSover, який реалізують усі алгоритми пошуку дискретного логарифму (окрім алгоритму Пухлінга-Хелмана, так як він має дещо відмінній інтерфейс і, у свою чергу, використовує алгоритм Полдарда), і CongruenceSystemSolver, реалізація якого направлена на вирішення системи конгруенцій. Окрім класів, які містять код безпосередньо алгоритмів, було додано кілька службових класів які інкапсулюють деякі примітивні операції над простими числами, а також алгоритми генерації простих чисел та примітивних коренів, які використовуються, як в алгоритмах, так і для генерації даних для тестування.

Ці класи – це DefinetePrime (містить операції над простими числами), SafePrimeGen (використовується для генерації простих чисел) і PrimRootGen (використовується для генерації примітивних коренів).

4.1.5 Результати тестування продуктивності алгоритмів за допомогою JMH

JMH дозволяє досить точно вимірювати продуктивність функцій за декількома критеріями, такими, як пропускна здатність, середній час виконання, обраний час виконання (коли один результат з великої кількості запусків обирається за еталон) і т.і. Завдяки цим якостям інструмент JMH де-факто визнаний чи не єдиним інструментм для більш-менш точного виміру продуктивності Java-додатків, який дозволяє розробляти досить гнучкі і правильні, з точки зору теорії вимірювання продуктивності, тести.

На рисунку 4.2 наведено результати тесту продуктивності для реалізованих класичних алгоритмів пошуку дискретного логарифму.

Рисунок 4.2 – Результати тестування продуктивності не квантових алгоритмів пошуку дискретного логарифма

Для кожного за алгоритмів наведено середній час виконання і окремо, випадково взятий зразковий час (sample time). Тобто результати можна вважати досить показовими. Результати тестів приведені згори вниз для: алгоритму простого перебору, алгоритму Гельфонда-Шенкса, алгоритму Пухлінга-Хелмана і алгоритму Полларда-Ро.

Власне порівняння швидкодії алгоритмів зображене на рисунку 4.3. З рисунку видно результати вимірювання продуктивності, з яких виходить, що найповільнішим алгоритмом є BruteForce, тобто простий перебір варіантів (в нашому випадку він був розпаралелений на 8 потоків, для 1 потоку результати ще гірші). Він у середньому виконувався за 28 мілісекунд, наступний за продуктивністю алгоритм Гельфонда – Шенкса, або BabyStepGiantStep, він, у середньому, виконувався за 5 мілісекунд. 1 та 2 місце ділять алгоритм Поліґа-Геллмана та ρ-алгоритм Полларда, з 0,9 та 1,4 мілісекундами відповідно.

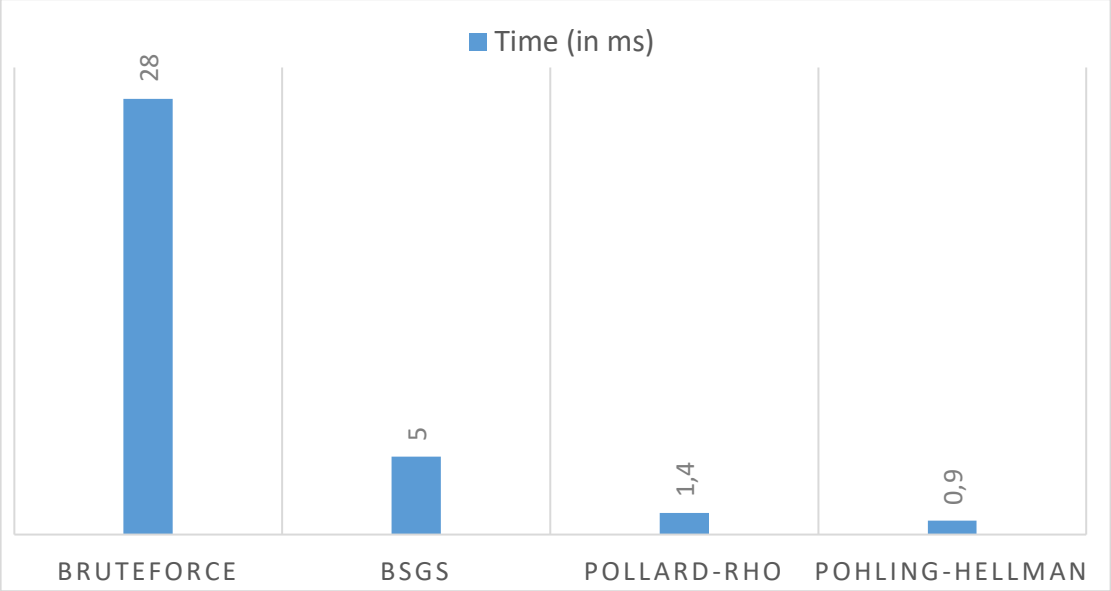

Рисунок 4.3 – Результати тестування продуктивності не квантових алгоритмів пошуку дискретного логарифма

Тести виконувалися на наступному обладнанні з ОС Windows: Intel Core i7-6700k 4.5GHz, DDR4 32Gb ОЗП.

4.2 Реалізація квантового алгоритму Шора пошуку дискретного логарифму

У цьому розділі наведено відомості про обрання відповідного інструментарію для розробки алгоритму Шора. Опис інструментарію та різних його характеристик, таких як якість документаціїї, накільки інструментарій інтегрований в інтегровані середовища розробки, та наскільки багаті стандартні бібліотеки представлені у розгладаному інструментарії. Деталі реалізації та результати тестування квантового алгоритму.

4.2.1 Вибір інструментів для реалізації

На сьогоднішній день існують десятки, як бібліотек, так і окремих інструментів (або мов) для написання квантових програм і їх запуску на емуляторі квантового комп’ютера [23]. При виборі інструменту для реалізації квантового алгоритму Шора для пошуку дискретного логарифма було визначено ряд критеріїв, яким повинен відповідати обраний інструмент. На основі цих критеріїв було вибрано і порівняно декілька підходящих інструментів. Інструменти повинні відповідати наступним критеріям:

- підтримка написання коду статично типізованою мовою програмування;
- наявність ефективного квантового емулятора;
- багата стандартна бібліотека, яка полегшувала та пришвидшувала процес розробки;
- зручна інтеграція в інтегровані середовища розробки;
- наявність добре сформованих навчальних матеріалів та документації.

Серед наявних інструментів було обрано три, які повністю відповідають вищенаведеним критеріям:

- QCGPU;
- QuEST;
- Microsoft QDK.

Для порівняння цих інструментів скористуємося методом варіантних мереж. Оцінимо за п'ятибальною шкалою наступні характеристики програмних продуктів:

а) швидкість і простота розробки (5).

б) підтримка інструменту в інтегрованих середовищах розробки (3).

в) багатство стандартної бібліотеки (5).

г) якість документації (4).

д) легкість інсталяції розробленого програмного забезпечення (3).

е) підтримка мультиплатформеності (2).

В круглих дужках вказана важливість кожної з характеристик.

Для рішення завдання вибору найкращого інструменту для розробки скористуємося методом варіантних мереж. Результат розрахунку придатності того чи іншого інструменту наведено у таблиці 4.1.

Таблиця 4.1 – Рішення завдання вибору програмного забезпечення

Середовище розробки	Характеристика						Сума
	а (5)	б (3)	в (5)	г (4)	д (3)	е (2)	
QCGPU	4	3	3	3	4	4	76
QuEST	5	3	3	3	4	4	75
Microsoft QDK	5	5	5	5	4	4	105

Як видно з таблиці, найкращім інструментом за сукупністю результатів по усіх характеристик виявися набір для розробки квантових програм від компанії Microsoft – QDK.

QDK включає в себе емулятор квантового комп’ютера який можна запускати в CLI середовищі, мову програмування Q# та багату стандартну бібліотеку.

Мова програмування Q# була нещодавно розроблена компанією Microsoft з метою розповсюдження знань про квантові алгоритми та надання можливості реалізації квантових алгоритмів за допомогою оптимізованих квантових примітивів, які входять в стандартну бібліотеку мови Q#. На зображено структуру QDK.

Q# підтримує парадигму квантового програмування і може бути легко інтегрована з іншими технологіями та мовами Microsoft (C#, F#, ASP.NET і т.д.).

Стандартна бібліотека QDK включає усі необхідні для реалізації функції та примітиви. Так, наприклад, простір імен Microsoft.Quantum.Convert, як можна здогадатися з назви, включає в себе набір функцій для конвертації між різними типами

Q#. Простір імен Microsoft.Quantum.Intrinsic складається з методів що відтворюють роботу базових гейтів, а Microsoft.Quantum. Canon містить методи для перетворення гейтів (наприклад метод CControlled, що перетворює звичайну операцію на контрольовану). Структуру стандартної бібліотеки QDK можна побачити на рисунку 4.4

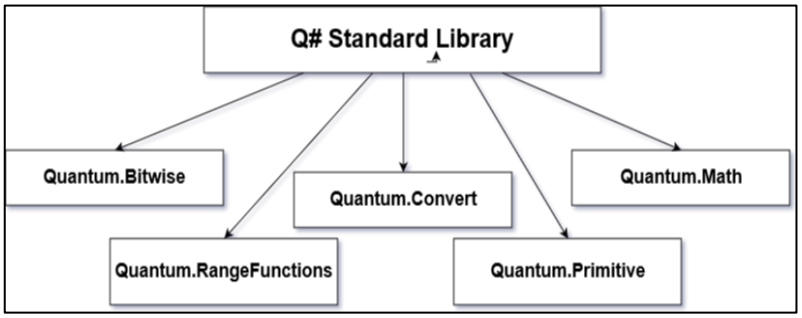

Рисунок 4.4 – Структура QDK

QDK має інтеграцію з такими середовищами розробки, як Visual Studio, Visual Studio Code і навіть IntelliJ IDEA.

4.2.2 Реалізація алгоритму і її результати

Для практичної реалізації за основу було взято алгоритм описаний Нільсеном М.А. та Чуангом І.Л [24]

Структуру проекту реалізації квантового алгоритму Шора можна побачити на рисунку 4.5

Реалізація алгоритму складається з частини коду написаній на C# (ShorRunner.cs) та частини коду написаній на Q# (DiscreteLogShor.qs).

ShorRunner.cs містить у собі код для взаємодії з користувачем (зчитування користувацького вводу) та для запуску квантового алгоритму на симуляторі.

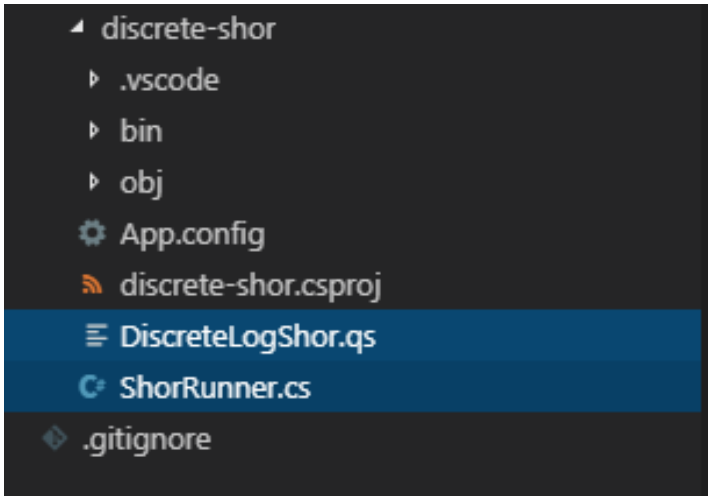

Рисунок 4.5 – Структура проекту

DiscreteLogShor.qs містить власне квантовий алгоритм Шора пошуку дискретного логарифму, розбитий на декілька функцій. Далі наведено код основної функції алгоритму:

```
// Discrete logarithm Algorithm
    // Input: integers a, b, N
    // Goal: compute the discrete logarithm log_a(b), returns -1 on failure cases
    operation DiscreteLogShor (a: Int, b: Int, N: Int) : Int {

        mutable appr_slmodr = 0;
        mutable appr_l = 0;
        let order = N - 1; // Euler function
        Message($"Order is ={order}");
        let t = BitSizeI(N) * 2 + 1;
        Message($"Register size is ={t}");
        using ((x1, x2, y) = (Qubit[t], Qubit[t], Qubit[t])) {
            ApplyToEach(H, x1);
            ApplyToEach(H, x2);
            DLOracle(a, b, N, x1, x2, y);
            InverseQFT(x1);
            InverseQFT(x2);
            set appr_slmodr =
MeasureInteger(BigEndianAsLittleEndian(BigEndian(x1)));
            set appr_l = MeasureInteger(BigEndianAsLittleEndian(BigEndian(x2)));
            ResetAll(x1);
            ResetAll(x2);
            ResetAll(y);
        }
        let (slmodr, _) = (ContinuedFractionConvergentI(Fraction(appr_slmodr, t),
N))!;
```

```
        let (l, _) = (ContinuedFractionConvergentI(Fraction(appr_l, t), N))!;
        if (GreatestCommonDivisorI(l, order) != 1) {
            Message($"Algorithm failed due to measure l={l}, no coprime to
r={order}");
            return -1;
        }
        let l_inv = InverseModI(l, order);
        let s = slmodr * l_inv % order;
        return s;
    }
```

На вхід алгоритму подається 3 числа: основа логарифму (генератор циклічної групи), число що треба логарифмувати і модуль циклічної групи (N). Далі виділяються 3 квантові регістри розміру: кількість бітів $BitSize(N) * 2 + 1$. Далі до перших двох регістрів застосовується гейт Адамара.

Після цього виконується DiscreteLogOracle(DLOracle), в рамках якого проходять основні математичні дії над квантовими регістрами, а саме зведення у ступінь по модулю кожного з вірогідних значень регістрів за формулою (4.1)

$$\frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1}|x,y> \ \rightarrow \frac{1}{q}\sum_{x=0}^{q-1}\sum_{y=0}^{q-1}|x,y,\ \alpha^x\beta^y> \qquad (4.1)$$

У коді ця операція виглядає наступним чином:

```
    // Oracle U|x1⟩|x2⟩|y⟩ → |x1⟩|x2⟩|y ⊕ f(x1,x2)⟩ where f(x1,x2)=b^x1*a^x2 mod N
    // Input: integers a, b, and N, registers x1, x2, and qs.
    operation OracleFunc (a : Int, b : Int, N : Int, x1 : Qubit[], x2 : Qubit[],
qs : Qubit[]) : Unit {
        body(...) {
            X(qs[Length(qs) - 1]);
            PowerOfa(b, N, x1, qs);
            PowerOfa(a, N, x2, qs);
        }

        adjoint auto;
        controlled auto;
        adjoint controlled auto;
    }
```

PowerOfa у даному випадку відповідає за операцію, яка описана формулою (4.2).

$$|x\rangle|y\rangle \rightarrow |x\rangle|a^x * y \bmod N\rangle \quad (4.2)$$

Після завершення роботи оракула, над першим та другим регістром виконується інвертоване перетворення Фур'є і з регістрів дістаються числові значення. Далі виконується пошук збіжного безперервного дробу (ContinuedFractionConvergent) по N, як знаменнику на основі, виміряних з квантових регістрів, приблизних значень.

В результаті зберігаємо необхідні значення чисівників знайдених дробів, як slmodr та l. Далі l та N – 1 на взаємну простоту. Якщо вони не взаємно просто, цей запуск алгоритму провалено і треба робити новий запуск. Якщо ж умова взаємної простоти виконана, то знаходимо логарифм, як slmodr помножений на (зворотній модуль l по N -1 по модулю N -1).

Реалізований алгоритм завершив роботу тільки для найпростішої циклічної групи за модулем 3. При цьому час його виконання наблизився дорівнював 2 годинам 57 хвилинам. Таке довгострокове виконання пояснюється використанням емулятора замість реального квантового комп'ютера, так, як емулювання квантових ефектів потребує значного часу і оперативної пам'яті. Навіть для циклічної групи за модулем 5, споживання оперативної пам'яті виходило за межі 100 гігабайт (так як треба було емулювати роботу 28 кубітів не враховуючи проміжні кубіти, що використовувалися у бібліотечних методах) , після чого робота програми завершувалася з помилкою в емуляторі.

Наразі можна впевнено констатувати, що в найближче десятиріччя квантові комп'ютери загалом і зокрема алгоритм Шора неможливо буде використати для злому криптографічних систем, робота яких базується на використанні складності знаходження дискретного логарифму. Цей висновок випливає з факту, що навіть у теорії алгоритм Шора пошуку дискретного логарифму потребує 6n кубітів. Де n – число байт в модулі циклічної групи.

Враховуючи що сучасний рекомендований розмір модуля для протоколу Діффі-Хеллмана дорівнює 2048 бітам [25], для його злому буде потрібен 12288-кубітний

квантовий комп’ютер. В той час, як найбільший сучасний квантовий комп’ютера складається з 72 кубіт (не враховуючи квантові комп’ютери від D-Wave Systems, які не являються такими для повного спектра задач) [26]. Тобто квантові комп’ютери все ще на здатні конкурувати зі звичайними та суперкомп’ютерами у вирішенні NP-задач. Тим не менш, працездатність алгоритма Шора практично доведена і область досідження і розробки квантових комп’ютерів розвивається досить швидко. Зважаючи на це, вже варто починати впровадження пост-квантових криптографічних алгоритмів на підприємствах, і, що більш важливо, необхідно почати перехід до пост-квантової криптографії для клієнтів та серверів, чия комунікація відбувається через відкриті інтернет канали. Варто також зауважити, що деякі хеш-функції таком можуть бути зламані за допомогою квантових комп’ютерів, тому перехешування паролів користувачів – це також важливий корок у пост-квантову еру.

ВИСНОВКИ

У ході виконання атестаційної роботи було досліджено проблему вирішення NP-задач, а саме проблему пошуку дискретного логарифму за допомогою парадигми квантових обчислень. Було виконано порівняння існуючих, не квантових, методів вирішення проблеми пошуку дискретного логарифму. Для цього було розглянуто особливості квантових алгоритмів, проблеми квантових обчислень, таких як когерентизація, клонування, перегляд поточного стану регістрів, імовірнісний характер обчислень. Окрім цього було досліджено актуальний стан розвитку квантових комп'ютерів так квантових обчислень, був відмічений посилений розвиток пост квантових рішень та розглянуто успіхи у реалізації квантових комп'ютерів таких організацій, як D-Wave Systems, Google, Intel, IBM та Microsoft.

Був проведений детальний аналіз декількох варіантів квантового алгоритму Шора для пошуку дискретного логарифму, а також досліджено не квантові алгоритми для пошуку дискретного логарифму. В ході дослідження було виявлено, що алгоритм Шора пошуку дискретного логарифму все ще не може конкурувати навіть з найпростішими класичними алгоритмами для пошуку дискретного логарифму, такими, як простий перебір, та алгоритм Гельфонда – Шенкса, а тим паче алгоритмом Поліґа-Геллмана, який у ході дослідження показав найкращу продуктивність.

Такий результат пояснюється, в першу чергу, недостатньою кількістю кубітів у сучасних квантових комп’ютерах, що являється наслідком складності боротьби з небажаними квантовими ефектами, які виникають під час роботи квантового комп’ютера, та зі складністю утримання атомів у необхідному стані протягом достатнього часу.

Тим не менш, у ході роботи було практично доведено можливість застосування алгоритму Шора пошуку дискретного логарифму з використанням емулятора квантового комп’ютера від Microsoft. Таким чином розробка та адаптування криптографічним алгоритмів пост-квантової ери (стійких до квантових алгоритмів

злому) є актуальною та затребуваною темою, яку вже сьогодні треба массово впроваджувати задля забезпечення охорони персональних даних користувачів та, що важливіше, для забезпечення безпеки банківської системи, яка в наш цас дуже залежить від алгоритмів і протоколів асиметричної криптографії, які можуть бути зламані за допомого алгоритму Шора на квантовому комп'ютері. Також не можна виключати вірогідність розробки нових квантовиї алгоритмів, які будть загрожувати усе більшій кількості криптографічниї алгоримтів, не тільки асиметричним, але й симетричним та хеш-функцій. Щоб бути готовим до цього, варто вже зараз розглядати адаптаційю та використання найновішиї пост-квантових протоколів та алгоритмів, які квантові комп'ютерів наразі просто не в змозі златмати.

Область квантових обчислень тільки зароджуєтся, і може принести безліч сюрпризів, як криптоаналітикам так і науковцям різниї напрямів.

ПЕРЕЛІК ДЖЕРЕЛ

1. List of companies involved in quantum computing or communication [Электронный ресурс] / Wikipedia – Режим доступу: https://en.wikipedia.org/wiki/List_of_companies_involved_in_quantum_computing_or_communication - 20.03.2019 г. - Загл. з екрану.

2. National Security Agency, Cryptography today, August 2015, archived on 23 November 2015, tinyurl.com/SuiteB..

3. Тесленко Н.О. Криптоаналіз методів несиметричного шифрування [Текст]: диплом/ Н.О. Тесленко.-Харьків.:ХНУРЕ, 2016. – 71 с.

4. Функция односторонняя [Электронный ресурс] / Математическая криптография - Режим доступу: http://cryptography.ru/ref/функция_односторонняя/ - 20.04.2019 г. - Загл. з екрану.

5. М.Г. Адигеев Введение в теорию сложности [Текст] / М.Г. Адигеев.-Ростов-на-Дону:РГУ, 2004.-35с.

6. Томас Х. Алгоритмы. Построение и анализ [Текст]/ Х. Томас, И. Чарльз.-М.:Вильямс, 2016.-1328с.

7. VS Denchev, S Boixo, SV Isakov, N Ding, R Babbush. What is the computational value of finite-range tunneling? - Physical Review X, 2016 - APS

8. Q# [Электронный ресурс] / The Q# Programming Language - Режим доступу: https://docs.microsoft.com/en-us/quantum/quantum-qr-intro/ - 13.03.2019 г. - Загл. з екрану.

9. В.Н. Ручкин Естественный параллелизм квантовых компьютеров и нейровычислителей/ В.Н. Ручкин, В.А. Романчук, В.А. Фулин.- Рязань.:Рязанский государственный университет им. С.А. Есенина, 2013 – 9 с.

10. С.А. Дуплий. Квантовая информация, кубиты и квантовые алгоритмы/С. Дуплий, В. Калашников, Е. Маслов. – Харьків.:Харьківський національний університет ім. В. Н. Каразіна, 2005.- 6 с.

11. Дж.Прескилл Квантовая информация и квантовые вычиселения / Д. Прескилл.-М.:Ижевск, 2008. - 464 с.

12. Wootters W.A Single Quantum Cannot Be Cloned [Text] / Wootters W., Zurek W.//Nature.-1982.-№5886.-802-803pp.

13. Bennett C. Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels [Text] /Bennett C., Brassard G., Crépeau C. et al.//Physical review letters.-1993.-№13.-1895-1899pp.

14. А.Г. Грозин Квантовый компьютер для чайников/ А.Г. Грозин.-Новосибирск.: Препринт, 2004.-24 с.

15. П.А. Правильщиков Квантовый параллелизм и решение уравнений в задачах управления на базе новой модели вычислений / П.А. Правильщиков.-М.:Институт проблем управления им. В.А. Трапезникова, 2014.- 17 с.

16. D. Shanks. The infrastructure of a real quadratic field and its applications. Proceedings of the Number Theory Conference. — University of Colorado, Boulder,, 1972. — C. pp. 217-224.

17. Pollard, J. M. Monte Carlo methods for index computation (mod p). / Pollard, J. M. / Mathematics of Computation - 1978 – Vol. 32, no. 143. – P. 918-924.

18. S. C. Pohlig and M. E. Hellman. An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance. IEEE Transactions on Information Theory. - 1978. - Vol. 1, no. 24. - P. 106-110.

19. Shor P. Algorithms for quantum computation: discrete logarithms and factoring [Text] /Shor P./ Foundations of Computer Science.—1994.—№10.-124–134pp.

20. Proos J., Zalka C. Shor's discrete logarithm quantum algorithm for elliptic curves [Text] /Shor P./ QIC 3 (No. 4) (2003) pp.317-344

21. Martin Ekerå, “Modifying Shor’s algorithm to compute short discrete logarithms”, in IACR Cryptology ePrint Archive, 2016

22. JMH [Электронный ресурс] / OpenJDK - Режим доступу: https://openjdk.java.net/projects/code-tools/jmh/ - 15.04.2019 г. - 1. Загл. з екрану.

23. List of QC simulators [Электронный ресурс] / Quantiki - Режим доступу: https://www.quantiki.org/wiki/list-qc-simulators - 15.05.2019 г. - 1. Загл. з екрану.

24. Nielsen, M. A. Quantum Computation and Quantum Information. / Nielsen, M. A., Chuang, I. L. - 2010 - pp.238-240

25. RFC 8270 [Электронный ресурс] / IETF - Режим доступу: https://tools.ietf.org/html/rfc8270 - 20.04.2019 г. - 1. Загл. з екрану.

26. A Preview of Bristlecone, Google’s New Quantum Processor [Электронный ресурс] / Google AI Blog - Режим доступу: https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html - 25.04.2019 г. - 1. Загл. з екрану.