МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему

«Розробка практичної моделі обміну інформації з використанням квантового розподілу ключів шифрування»

Завідувач
випускаючої кафедри — Любчак В.О
Студент гр.КБ–81 — Сагура А.Р.
Керівник роботи — Котух Є.В.

СУМИ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

Затверджую_____
Зав. кафедрою Любчак В.О.
“_____” _____2022 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи КБ-81 спеціальності «Кібербезпека» денної форми навчання Сагури Андрія Руслановича.

Тема: «Розробка практичної моделі обміну інформації з використанням квантового розподілу ключів шифрування»

Затверджена наказом по СумДУ
№_____ від _____2022 р.

Зміст пояснювальної записки: ((1) аналіз сучасних методів шифрування; 2) розгляд доцільності використання квантового каналу зв’язку; 3) опис основного алгоритму розробки додатку; 4**)** висновки до роботи)

Дата видачі завдання “_____” _____ 2022г.
Керівник випускної роботи _____ Котух Є.В.
Завдання прийняв до виконання _____ Сагура А.Р.

РЕФЕРАТ

Записка: 32 стор., 6 рис., 1 табл., 2 додатки, 15 джерел.

Об’єкт дослідження - квантовий канал передачі інформації

Мета роботи — провести аналіз існуючих технологій квантових каналів звя’язку, розглянути актуальні алгоритми шифрування та розробити практичну модель додатку з використанням кращих практик, що будуть розглянуті. Розробити практичну модель додатку для обміну повідомленнями з використанням абстрактного квантового каналу зв’язку.

Предмет дослідження - алгоритм обміну інформації за допомогою квантового каналу

В результаті роботи було - розглянуто окремі алгоритми криптографії, досліджено принципи роботи квантового каналу передачі інформації, розроблено додаток для обміну інформацією з використанням квантового каналу зв'язку.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 3
ВСТУП 4
Розділ 1 ОГЛЯД ІСНУЮЧИХ МОДЕЛЕЙ РОБОТИ 6
1.1 Симетричне шифрування 6
1.2 Асиметричне шифрування 7
1.3 Протокол квантової криптографії 10
Розділ 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ 17
2.1 Алгоритм розробки 17
2.2 Подальший план розвитку проекту 19
ВИСНОВКИ 20
СПИСОК ЛІТЕРАТУРИ 21
Додаток А 23
Додаток Б 25

ПЕРЕЛІК СКОРОЧЕНЬ

DES - Data Encryption Standard

AES - Advanced Encryption Standard

DH - Diffie-Hellman

EPR - Einstein-Podolsky-Rosen

QKD - Quantum key distribution

ВСТУП

У прогресивному світі сьогодні панують інформаційні технології тим самим задаючи вектор розвитку усім індустріям. Обмін даними через Інтернет давно став звичайною повсякденною справою як для пересічного користувача соціальних мереж та месенджерів так і для робітників приватних та державних компаній малого, середнього та великого бізнесу.

Захист інформації, що передається мережею, невпинно покращується, адже загрози крадіжки цінних знань ніколи не зникали. Шифрування інформації - невід'ємна потреба, яка виникла ще до нашої ери та вдосконалюється весь час. На сьогодення використовується класична криптографія для забезпечення безпеки кожному у мережі. Сучасна криптографія включає складні математичні розрахунки та алгоритми, ціллю операцій яких є отримання тексту що неможливо розшифрувати за прийнятний або доцільний час, який називається поліноміальним часом. Розрізняється два основних типи сучасного шифрування – симетричне [1] та асиметричне [2] шифрування, різниця яких складає оперування одним ключем і двома відповідно. Принципи роботи яких були сформовані ще в минулому тисячолітті та вдосконалені фахівцями, математиками впродовж років надійно виконує свою роботу. Алгоритми та реалізація вище названих типів шифрування є відточеними та надійними, але, наприклад, в алгоритмі симетричного шифрування існує ключ, втрата якого недопустима для збереження конфіденційності, тим самим процес передачі по мережі ключів не надає гарантій його повного захисту. Варто зауважити, що хоча зараз найдосконаліші асиметричні способи шифрування неможливо зламати за поліноміальний час все ж інновації не стоять на місці та показують світу на що будуть спроможні розрахункові машини майбутнього, наприклад квантовий комп'ютер, який неймовірно спростить дешифрування. Завдяки використанню кубітів у процесорі такого комп'ютера його швидкодія значно перевершує

звичайні комп’ютери та може виконувати алгоритм Шора [3] за лічені хвилини [4].

Видніється нова ера технологій і вже скоро квантова криптографія повноцінно посяде своє місце у світі технологій. Принцип роботи базується на законах квантової механіки, безпека гарантується законами природи та високою складністю процесів передачі ключів, шифруванням, тощо.

Однією із найскладніших умов повної конфіденційності при спілкуванні є “Проблема розподілу ключів”, і цю проблему ефективно вирішує квантова криптографія, завдяки практичній імплементації закону квантової механіки що унеможливлює клонування.

В роботі буде розглянуто існуючі моделі розподілу ключів, алгоритмів шифрування, визначення вразливих та сильних сторін кожного з підходів та розроблена практична модель системи обміну повідомленнями з використанням квантового каналу передачі інформації у вигляді додатку для Андроїд смартфону. Котрий в свою чергу оперуватиме кращими практиками та умовним практичним квантовим каналом зв’язку.

Розділ 1 ОГЛЯД ІСНУЮЧИХ МОДЕЛЕЙ РОБОТИ

1.1 Симетричне шифрування

Один із двох основних типів шифрування є симетрична криптографія. Головною ознакою якої є шифрування та розшифрування тексту одним секретним ключем, яким володіє кожна сторона. Такий вид шифрування є достатньо простим, не потребуючи великих обчислювальних потужностей та виконується швидше за аналоги асиметричного шифрування. Відомими прикладами симетричного шифрування є AES [5] та DES [6], що стали основою стандартів шифрування.

На заміну застарілому DES прийшов удосконалений варіант стандарт криптографії - AES. Методами даного стандарту є підстановка і перестановка. В процесі шифрування використовується зсуви рядків, змішування стовпців та додавання ключів. Залежно від довжини ключа використовується різна кількість раундів - 10, 12 або 14. Останній раунд виключає процес мікшування. Перевагами AES є вагома безпека, швидкість та гнучкість. Варіативність довжини ключа являє собою додаткову перевагу.

До розгляду також можна взяти Український блочний алгоритм симетричного шифрування під назвою “Калина” [7] (ДСТУ 7624:2014). Стандарт був прийнятий у 2015 році. Співпраця провідних українських науковців і Держспецзв’язку надали країні особистий алгоритм шифрування, який відходить від радянських вже застарілих стандартів. З 1 січня 2022 року є обов’язковим до використання разом із функцією хешування “Купина” [8] (ДСТУ 7564:2014) при накладанні і перевірці електронного підпису. Алгоритм передбачає використання невеликого ключа довжиною до 512 бітів, а також включає в себе 10 етапів роботи з текстом і ключем, такі як - гамування різних видів, проста та індексована заміна, імітовставки, тощо. Криптостійкість шифрування на високому рівні, існували деякі атаки на скорочені варіанти шифрування [9] але вони виявилися не практичними.

Важливим недоліком симетричного способу шифрування вбачаю у існуванні секретного ключа який неможливо передавати у публічній мережі без криптозахищених каналів, що накладає додаткові вимоги та витрату ресурсів. Інтернеті конфіденційно по мережі, адже його перехват повністю нівелює усю секретність каналу. Також варто зазначити що обмін даними у такий спосіб значно звужує кількість користувачів, адже передача багатьом користувачам одного і того ж ключа значно підвищує ризики його крадіжки, натомість делегування різними ключами для кожного співрозмовника є не практичним.

1.2 Асиметричне шифрування

Асиметричне шифрування, також відоме як шифрування з відкритим ключем вирішує недоліки висвітлені щодо симетричного способу шифрування. Використання відкритого, або публічного ключа, який математично пов'язаний з секретним ключем надає надійність у обміні ключем мережею. Так як завдяки публічному ключу, яким шифрується повідомлення, неможливо отримати відкритий текст, його передача не є секретною і отримання його зловмисником не надасть йому доступу до чутливої інформації. Гарантія аутентифікації, яку гарантує асиметричне шифрування, є додатковим рівнем безпеки, оскільки лише користувач секретного ключа має можливість розшифрувати дані, оскільки ключі математично пов'язані.

Один із найвідоміший способів асиметричного шифрування є алгоритм RSA [10], який був відкритий у 1977 році трьома вченими - з Массачусетського технологічного інституту Ронон Рівест, Аді Шамір і Леонардо Адлеман. Ефективність даного виду шифрування полягає у методі “первинної факторизації”. Таким чином на початку шифрування обираються два простих числа, скажімо по 1024 біта кожне, і множаться для отримання великого числа. Процес визначення вихідних простих чисел з гігантського числа є непосильною задачею для сучасних комп'ютерів, тим паче для людей. Для

обчислення 768-бітного ключа має знадобитися півтори тисячі років з використанням сотні комп'ютерів, тим не менш стандартний ключ RSA є 2048-бітним, що безумовно слугує доказом високої надійності.

Підхід алгоритму RSA щодо використання ключів гнучкий, так як може використовуватись ключ довжиною 768, 1024, 2048 та 4096 біт.

Існує асиметричний алгоритм шифрування який базується на еліптичній кривій - ECC. Ніл Кобліц і Віктор Міллер запропонували ще в 1985 році використання рівняння еліптичної кривої яка представляє набір точок, які задовольняють рівняння виду - $2y = 3x + ax + b$. Число у ECC символізує точку на кривій і перемножуючись отримується нова точка фігури. Математика ECC побудована таким чином, що знаходження нової точки практично неможливо, знаючи навіть вихідну точку. Варто зазначити що використання невеликого ключа, користуючись цим алгоритмом, дає змогу продуктивно використовувати його у роботі веб сервісів, адже низьке навантаження та відносно невелика обчислювальна потужність сприяє швидкій роботі мережі.

Пропоную розглянути український асиметричний стандарт шифрування, який був затверджений у 2020 році під назвою - ДСТУ 9041:2020 [11]. Алгоритм спрямований на шифрування коротких повідомлень, що базується на скручених еліптичних кривих Едвардса. Довжина повідомлень має бути до 616 біт, задля задуманого якісного виконання шифрування. Проект алгоритму шифрування, що ліг за основу стандарту, пройшов апробацію в Україні і за її межами, а саме - в Центрально-Європейській конференції з криптографії, у червні 2020 року на форум ведучих криптологів з усього світу. Сферою застосування даного алгоритму є інкапсуляція ключів, найсучасніший математичний апарат, новий алгоритм генерації псевдовипадкових послідовностей. Алгоритм розроблений з урахуванням усіх сучасних вимог до криптостійкості, та буде під загрозою лише при існуванні квантового комп'ютера з семиста кубітами. Його перевагою є відносно невелика довжина ключа, що дозволяє пришвидчити роботу.

Варто зазначити що асиметричне шифрування також має власні недоліки, незважаючи на те що вирішує деякі проблеми симетричного шифрування. Складність процесу шифрування через наявність двох ключів а також повільніший процес адже він вимагає більше обчислювальної потужності. Шифрування невеликого об'єму даних є більш раціональним способом використання даного алгоритму, на відміну від синхронного способу, де ідеально підходить шифрування великої кількості даних.

Розглянемо алгоритм обміну ключами Діффі-Хеллмана [12] він є одним із перших практичних реалізацій обміну відкритими ключами у сфері криптографії. Алгоритм обміну ключами Діффі-Хеллмана - це один із способів, яким можна генерувати спільний ключ та обмінюватися чутливою інформацією між двома сторонами таким чином, щоб бути переконаними, що ніхто не зможе виявити комунікацію. Про алгоритм слід пам'ятати один важливий факт: передається не інформація при обміні, а створюється ключ, який можна буде використовувати для обміну інформацією. Оскільки цей метод дозволяє створити ключ шифрування з іншою стороною, можна почати шифрування поточних та прийнятих повідомлень. Встановлено, що навіть якщо хтось записує дані передачі, вони не можуть бути розшифровані.

Алгоритм заснований на криптографії еліптичних кривих, яка є методом криптографії з відкритим ключем, заснований на алгебраїчній структурі еліптичних кривих над кінцевими полями. Даний алгоритм також використовує функцію trapdoor [13], як і багато інших способів криптографії з відкритим ключем. Проста ідея розуміння алгоритму DH полягає в наступному

- Перша сторона обирає два простих числа g і p(мат редактор) та повідомляє їх другий.

- Потім друга сторона вибирає секретний номер (назвемо його a), а потім обчислює g mod p і відправляє результат назад першій стороні, назвемо результат A. Секретний номер не відправляється.

- Потім перша сторона робить те саме, вона вибирає секретне число b і обчислює результат B, аналогічний

- Потім цей результат відправляється другій стороні.

- Друга сторона бере отримане число B та обчислює $B^a \ mod \ p$

- Перша партія бере отримане число A та обчислює $A^b \ mod \ p$

Обидві сторони отримають ту саму відповідь незалежно від порядку зведення в ступінь. Число, прийняте на кроках 5 і 6, буде прийнято як загальний секретний ключ. Тепер цей ключ можна використовувати для будь-якого шифрування даних, що використовується, наприклад в AES.

1.3 Протокол квантової криптографії

Варто розглянути сучасні можливості квантового комп'ютера та перспективи розвитку задля розуміння важливості квантових каналів обміну ключами. Назараз існують прототипи квантових компютерів які мають малу кількість кубітів та відносно невелику стабільність і точність, але в змозі виконувати мінімально можливі алгоритми, наприклад алгоритм Шора. Розвиток даної технології дозволить збільшити обчислювальні можливості та виконувати квантову факторизацію з високою швидкістю настільки, що алгоритми асиметричного шифрування втратять актуальність. Оскільки в такому випадку можно буде дискредитувати шифр знаючи публічний ключ, який за технологією має бути і вільному доступі. Квантові канали передачі інформації можуть вирішити проблему легкого злому шифрування, так як перехопити інформацію неможливо тим самим нанести шкоду чи поставити під питання конфіденційність.

В рамках моєї роботи пропоную розглянути протоколи квантової криптографії, як найбільш перспективні для майбутньої реалізації та використання криптографічного захисту інформації. Перший протокол квантової криптографії (BB84) [14] був запропонований і опублікований в 1984 Беннетом і Брассардом. Пізніше ідея була розвинена Екертом у 1991 році. В основі методу квантової криптографії лежить спостереження квантових

станів фотонів. Відправник визначає ці стани, а одержувач їх реєструє. Тут використовується квантовий принцип невизначеності, коли дві квантові величини не можуть бути виміряні одночасно з необхідною точністю. Так поляризація фотонів може бути ортогональною, діагональною або циркулярною. Вимірювання одного виду поляризації рандомізує іншу складову. Таким чином, якщо відправник та одержувач не домовилися між собою, який вид поляризації брати за основу, одержувач може зруйнувати надісланий відправником сигнал, не отримавши жодної корисної інформації.

Відправник кодує дані, що надсилаються, задаючи певні квантові стани, одержувач реєструє ці стани. Потім одержувач та відправник спільно обговорюють результати спостережень. Зрештою з якоюсь високою достовірністю можна бути впевненим, що передана і прийнята кодові послідовності тотожні. Обговорення результатів стосується помилок, внесених шумами або зловмисником, і ні в якому разі не розкриває вміст переданого повідомлення. Може обговорюватися парність повідомлення, але не окремі біти. Під час передачі даних контролюється поляризація фотонів. Поляризація може бути ортогональною (горизонтальною або вертикальною), циркулярною (лівою або правою) та діагональною (45 або 135 градусів).

Як джерело світла може використовуватися світло випромінюючий діод або лазер. Світло фільтрується, поляризується та формується у вигляді коротких імпульсів малої інтенсивності. Поляризація кожного імпульсу моделюється відправником довільним чином відповідно до одного з чотирьох перелічених станів.

Одержувач вимірює поляризацію фотонів, використовуючи довільну послідовність базових станів (ортогональна або циркулярна). Одержувач відкрито повідомляє відправнику, яку послідовність базових станів використовував. Відправник відкрито повідомляє отримувача про те, які базові стани використані коректно. Усі виміри, виконані при неправильних базових станах, відкидаються. Вимірювання інтерпретуються згідно з

двійковою схемою: ліво-циркулярна або горизонтальна поляризація - 0, право-циркулярна або вертикальна - 1. Реалізація протоколу ускладнюється присутністю шуму, який може викликати помилки. Помилки, що вносяться, можуть бути виявлені і усунені за допомогою підрахунку парності, при цьому один біт з кожного блоку відкидається. Беннет у 1991 році запропонував наступний протокол.

Протокол Беннета

- Відправник та одержувач домовляються про довільну перестановку бітів у рядках, щоб зробити положення помилок випадковими.
- Рядки поділяються на блоки розміру k (k вибирається так, щоб ймовірність помилки в блоці була мала).
- Для кожного блоку відправник та одержувач обчислюють та відкрито сповіщають один одного про отримані результати. Останній біт кожного блоку видаляється.
- Для кожного блоку, де парність виявилася різною, одержувач та відправник роблять ітераційний пошук та виправлення невірних бітів.
- Щоб виключити кратні помилки, які можуть не помічені, операції пунктів 1-4 повторюються для більшого значення k.

Для того, щоб визначити, чи залишилися не виявлені помилки, одержувач і відправник повторюють псевдовипадкові перевірки:

Одержувач і відправник відкрито оголошують про випадкове перемішування позицій половини біт у їх рядках.

Одержувач та відправник відкрито порівнюють парності. Якщо рядки відрізняються, парності повинні не співпадати з ймовірністю 1/2.

Якщо має місце відмінність, одержувач і відправник, використовує двійковий пошук та видалення невірних бітів.

Якщо відмінностей немає, після m ітерацій одержувач та відправник одержують ідентичні рядки з ймовірністю помилки 2-m.

Схема реалізації односпрямованого каналу з квантовим шифруванням показана на мал. 1. Передавальна сторона знаходиться ліворуч, а приймаюча – праворуч. Осередки Покеля служать для імпульсної варіації поляризації потоку квантів передавачем і аналізу імпульсів поляризації приймачем. Передавач може формувати один із чотирьох станів поляризації (0, 45, 90 і 135 градусів). Власне передані дані надходять у вигляді керуючих сигналів на ці комірки. Як канал передачі може використовуватися оптичне волокно. Як первинне джерело світла можна використовувати і лазер.

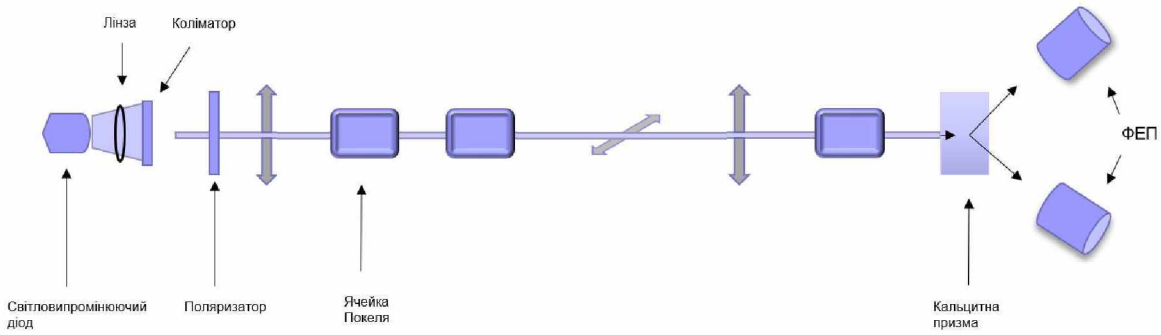


Рисунок 1.1 - Реалізація односпрямованого квантового каналу

На стороні, що приймає, після осередку Покеля ставиться кальцитова призма, яка розщеплює пучок на два фотодетектори (ФЕП), що вимірюють дві ортогональні складові поляризації. При формуванні імпульсів квантів, що передаються, доводиться вирішувати проблему їх інтенсивності. Якщо квантів у імпульсі 1000, є ймовірність того, що 100 квантів на шляху буде відведено зловмисником на свій приймач. Аналізуючи пізніше відкриті переговори між стороною, що передає і приймає, він може отримати потрібну йому інформацію. В ідеалі кількість квантів в імпульсі має бути близько одного. Тут будь-яка спроба відведення частини квантів зловмисником призведе до суттєвого зростання кількості помилок у сторони, що приймає. У цьому випадку прийняті дані повинні бути відкинуті та спроба передачі повторена. Але, роблячи канал більш стійким до перехоплення, ми в цьому випадку

стикаємося з проблемою "фонового" шуму (видача сигналу без фотонів на вході) приймача (адже ми змушені підвищувати його чутливість). Для того щоб забезпечити надійне транспортування даних логічному нулю та одиниці можуть відповідати певні послідовності станів, що припускаються корекції одинарних і навіть кратних помилок.

Подальшого покращення надійності криптосистеми можна досягти, використовуючи ефект EPR [15]. Ефект EPR виникає, коли сферично симетричний атом випромінює два фотони в протилежних напрямках у бік двох спостерігачів. Фотони випромінюються з невизначеною поляризацією, але з симетрії їх поляризації завжди протилежні. Важливою особливістю цього ефекту є те, що поляризація фотонів стає відомою лише після виміру. На основі EPR Екерт запропонував крипто-схему, яка гарантує безпеку пересилання та зберігання ключа. Відправник генерує кілька EPR фотонних пар. Один фотон з кожної пари він залишає собі, другий посилає своєму партнеру. При цьому якщо ефективність реєстрації близька до одиниці, при отриманні відправником значення поляризації 1, його партнер зареєструє значення 0 і навпаки. Ясно, що таким чином партнери щоразу, коли потрібно, можуть отримати ідентичні псевдовипадкові кодові послідовності. Практично реалізація даної схеми проблематична через низьку ефективність реєстрації та вимірювання поляризації одиночного фотона.

Неефективність реєстрації є платою за таємність. Слід враховувати, що з однофотонному режимі виникають суто квантові ефекти. При горизонтальній поляризації (H) та використанні вертикального поляризатора (V) результат очевидний - фотон не буде зареєстровано. При 45^0 поляризації фотона та вертикальному поляризаторі (V) ймовірність реєстрації 50%. Саме ця обставина і використовується у квантовій криптографії. Результати аналізу під час передачі двійкових розрядів представлені таблиці 1. Тут передбачається, що з передавача логічному нулю відповідає поляризація V, а

одиниці - +45^0, для приймаючої сторони логічному нулю відповідає поляризація -45^0, а одиниці - H.

Таблиця 1.1 - Результати аналізу передачі двійкових розрядів

Біт, що передається	1	0	1	0
Поляризація передачі	+45	V	+45	V
Поляризація прийому	-45	-45	H	H
Біти коду при отриманні	0	0	1	1
Результати прийому	-	-	+	-

Зрозуміло, що в першій та четвертій колонці поляризації передачі та прийому ортогональні і результат детектування буде відсутній. У колонках 2 і 3 коди двійкових розрядів збігаються і поляризації не ортогональні. Тому з ймовірністю 50% може бути позитивний результат у будь-якому з цих випадків (і навіть в обох). У таблиці передбачається, що успішне детектування фотона відбувається випадку колонки 3. Саме цей біт стає першим бітом загального секретного ключа передавача і приймача.

Фізичні принципи які закладено у квантову комунікацію дозволяє забезпечити криптографічні функції на рівні фізики процесу, така властивість на логічному рівні дозволяє вирішити проблеми аутентифікації на принципово іншому рівні.

Розглянемо на практичному прикладі властивості та принципи квантової комунікації, що забезпечують безпеку передачі даних. Уявимо, є дві людини, Аліса та Боб, які хочуть відправити один одному секрет, який ніхто інший не зможе перехопити. За допомогою QKD Аліса відправляє Бобу серію поляризованих фотонів оптоволоконним кабелем.

Якщо підслуховувач на ім'я Єва спробує підслухати розмову, їй доведеться прочитати кожен фотон, щоб прочитати секрет. Потім вона має

передати цей фотон Бобу. Читаючи фотон, Єва змінює квантовий стан фотона, що робить помилки в квантовому ключі. Це попереджає Алісу та Боба про те, що хтось підслуховує та ключ був скомпрометований, тому вони відмовляються від ключа. Аліса повинна надіслати Бобові новий ключ, який не був скомпрометований, і тоді Боб може використовувати цей ключ для читання секрету.

Розглянуті підходи широко використовуються в комерційних моделях передачі інформації за допомогою квантових каналів. В роботі буде запропоновано практичну реалізацію передачі інформації з використанні квантових каналів з метою вивчення особливостей реалізації та можливостей на практичному досвіді.

Алгоритм включає в себе використання симетричного шифрування на основі ключа, який генерується базуючись на отриманій, за допомогою квантового каналу, числовій послідовності. Оскільки в даний спосіб забезпечується повна конфіденційність в листуванні на відміну від асиметричного алгоритму, який передбачає використання публічного ключа, який буде головною вразливістю даного підходу в майбутньому.

Розділ 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ

2.1 Алгоритм розробки

Враховуючи переваги квантового каналу та недоліки асиметричного прийнято рішення використовувати виключно симетричне шифрування, яке має також наступні переваги - швидкість роботи алгоритму та великий об'єм даних, що можна зашифрувати не втрачаючи ресурсів та часу. Додаток реалізовуватиме спілкування типу broadcast (кожен - усім). Таким чином матимемо копію додатку на кожному пристрої, які отримують числову послідовність та формують ключ, обмінюючись зашифрованими повідомленнями, які зберігатимуться на хмарному сервері, не боячись витоку даних з нього.

Розроблення додатку планується в середовищі Android Studio мовою програмування Java. З основними виконуючими класами додатку можна ознайомитись в додатку Б. Сервер на базі Google додатку FireBase, яка забезпечує роботу у реальному часі. Числову послідовність додаток братиме з сайту про кількість населення землі, цю передачу чисел вважатимемо за абстрактну модель квантового каналу передачі даних. Знімки екрану розробленого додатку представлені в додатку А.

В загальному плані принципова схема роботи додатку представлена на рисунку 2.1, також більш детально можна ознайомитися за допомогою Діаграми взаємодії 2.2.

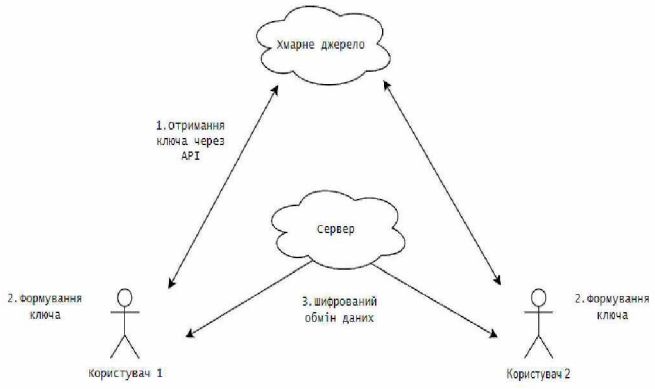
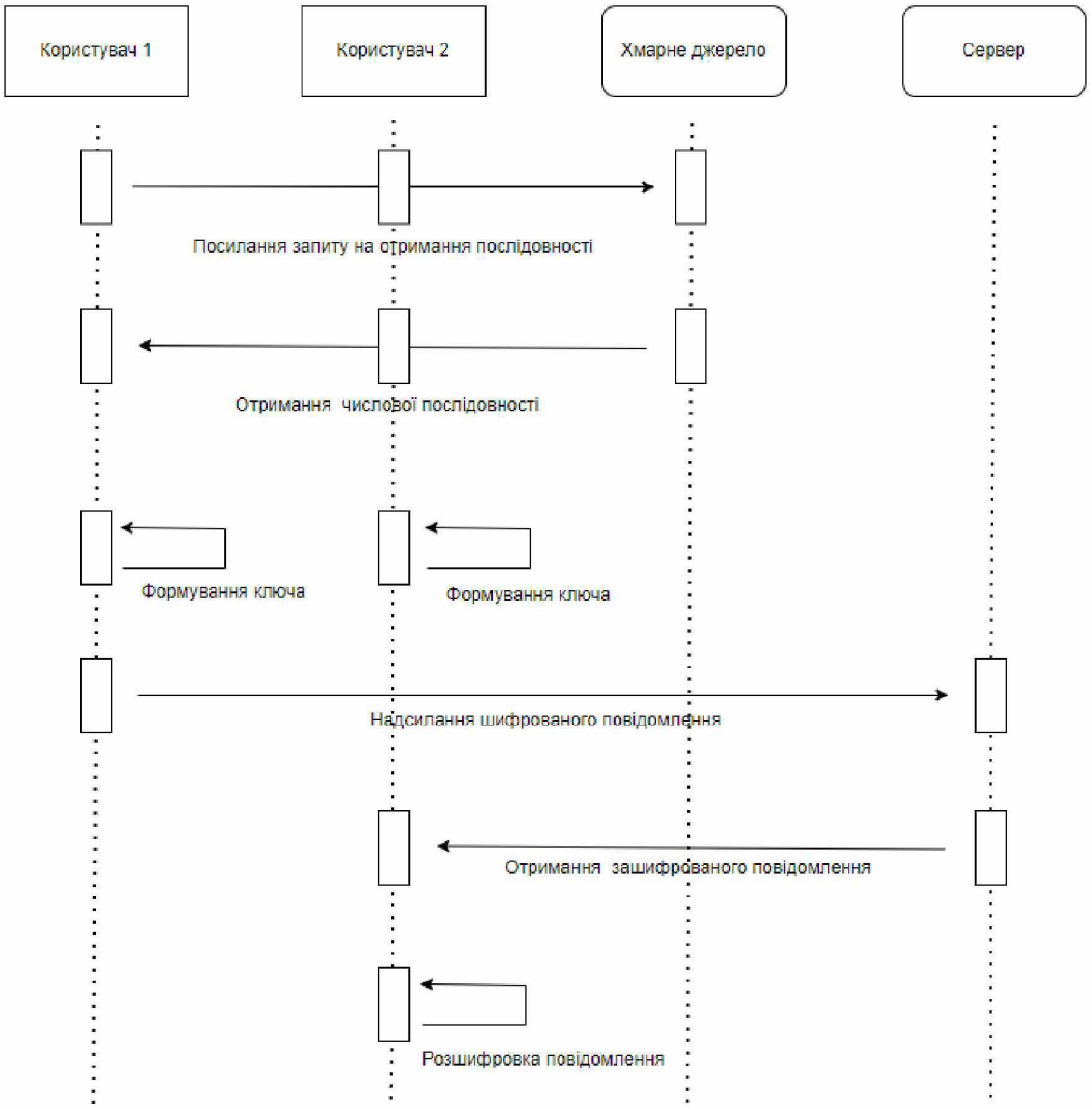


Рисунок 2.1 - Високорівнева архітектура роботи додатку



Діаграма взаємодії 2.2 - Детальна схема роботи додатку

Роз'яснення діаграма взаємодії

1) В першу чергу користувач надсилає запрос на отримання числової послідовності.
2) Кожен користувач отримує надсилання числової послідовності, необхідної для ключа шифрування.
3) Третій крок - формування ключа на основі отриманих даних.
4) Остання стадія - обмін повідомленнями, використовуючи отриманий ключ для шифрування, використовуючи хмарний сервер.

2.2 Подальший план розвитку проекту

Вбачаю доцільну потребу подальшого розвитку даної роботи, оскільки в додатку ще не реалізовано аккаунт систему, та можливість листування з конкретними особами. Також варто зазначити, що існує чимало розповсюджених практик додатків для обміну інформацією з яких варто взяти приклад та вдосконалити власний додаток. Не менш важливою є криптографічна частина, яка також може бути удосконалена впровадженням покращених шифрувальних алгоритмів.

Базуючись на роботі студентки групи КБ-81 Панченко Юлії, в майбутньому доцільно додати ті функціональні та нефункціональні вимоги, що описані в напрацюваннях. Технології не стоять на місці та розвитку немає кінця, тому завжди є місце для сміливих рішень та впровадження кращих практик.

ВИСНОВКИ

У бакалаврській роботі розглянуто окремі алгоритми криптографії, досліджено принципи роботи квантового каналу передачі інформації, виокремлені слабкі та сильні сторони методів шифрування та вибрані кращі практики, обґрунтовано використання квантового каналу передачі інформації, як засобу досягнення безумовного криптографічного засобу.

Для розробки моделі додатку з використанням кращих технологій було детально розібрано методології обміну ключами, та принципи роботи квантового каналу.

Було розроблено практичну модель системи обміну повідомленнями, з використанням симетричного шифрування та реалізовано абстрактну схему квантового каналу зв'язку.

Описаний подальший план розвитку проекту, задля вдосконалення функціоналу та приміненні все кращих практик кіберзахищеності.

СПИСОК ЛІТЕРАТУРИ

1. Принципи симетричного шифрування [Електронний ресурс] : – режим доступу https://hostpro.ua/wiki/ua/security/encryption-types-algorithms

2. Принципи асиметричного шифруваня [Електронний ресурс]: – режим доступу https://wiki.tntu.edu.ua/

3. Алгоритм Шора [Електронний ресурс]: – режим доступу https://uk.wikipedia.org/wiki/

4. Інформація про квантовий комп'ютер [Електронний ресурс] : – режим доступу https://www.tadviser.ru/index.php/

5. Принципи алгоритму AES [Електронний ресурс] : – режим доступу https://br.atsit.in/ru/?p=7715

6. Принципи алгоритму DES [Електронний ресурс] : – режим доступу https://kaf401.rloc.ru/Criptfiles/DES.htm

7. Інформація про алгоритм “Калина” [Електронний ресурс] : – режим доступу https://www.slideshare.net/oliynykov/kalyna

8. Інформація про алгоритм “Купина” [Електронний ресурс] : – режим доступу https://www.slideshare.net/oliynykov/kalyna

9. Огляд шифру “Калина” з розглядом атак [Електронний ресурс] – режим доступу: https://uk.upwiki.one/wiki/Kalyna_(cipher)

10. Інформація щодо алгоритму RSA [Електронний ресурс] : – режим доступу https://habr.com/ru/post/534014/

11. Відомості про ДСТУ [Електронний ресурс] : – режим доступу http://online.budstandart.com/ru/catalog/doc-page.html?id_doc=90523

12. Принцип роботи алгоритму Діффі-Хеллмана [Електронний ресурс] : – режим доступу https://tproger.ru/translations/diffie-hellman-key-exchange-xplained/#:.

13. Коротке пояснення значення функції [Електронний ресурс]: – режим доступу https://dic.academic.ru/dic.nsf/ruwiki/1584874

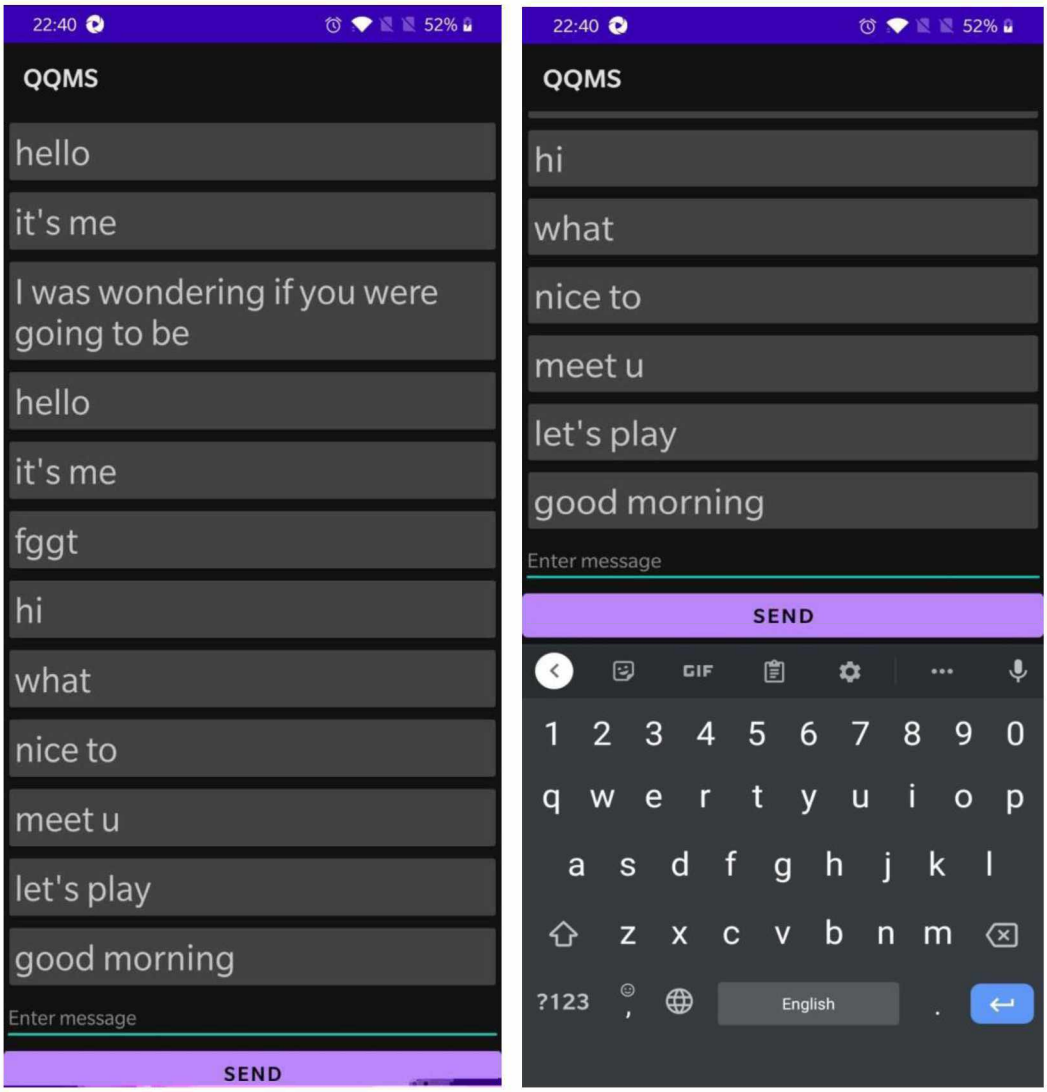
14. Інформація про протокол BB84 [Електронний ресурс] : – режим доступу https://ru.wikipedia.org/wiki/BB84

15. Інформація про ефект EPR [Електронний ресурс] : – режим доступу https://habr.com/ru/post/316252/

16. Код проекту [Електронний ресурс] : – режим доступу https://github.com/AndrewSan/Diplom_QQMS

Додаток А

Інтерфейс додатку



Рисунки А.1,А.2 - Інтерфейс діалогу користувача

Знімок екрану сайту хмарного серверу

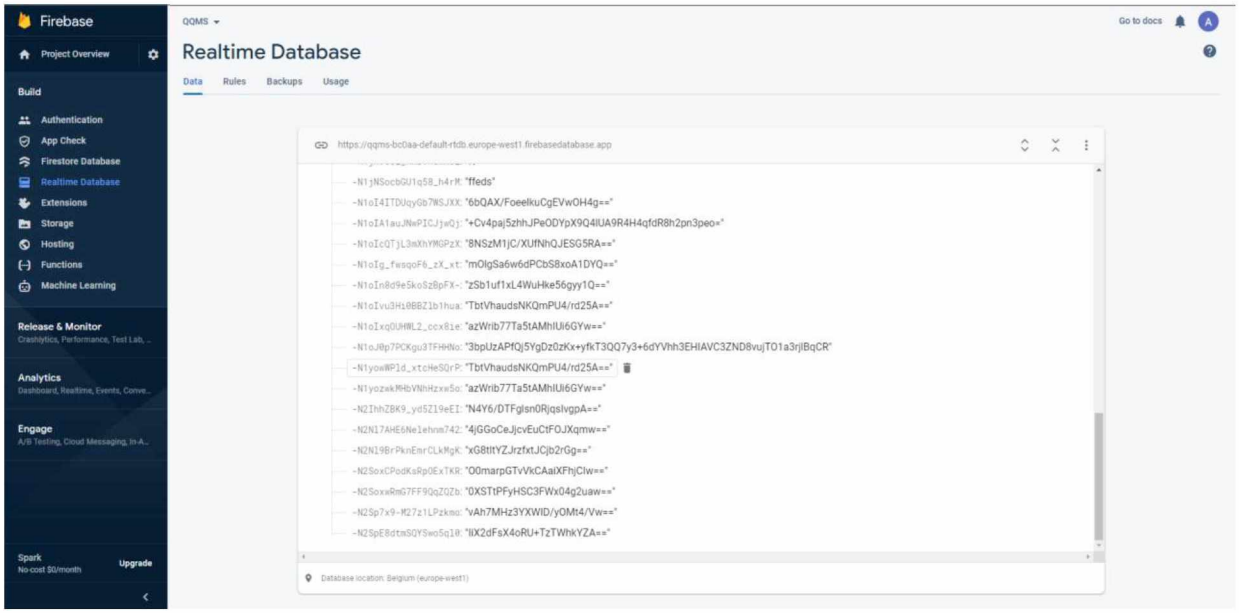


Рисунок А.3 - Журнал реєстрації зашифрованих повідомлень на сервері

Додаток Б

Лістинг коду

Повну версію проекту можна знайти на GitHub за посиланням [16]

```
package com.steam.qqms;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
import java.io.IOException;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {


private static int MAX_MESSAGE_LENGTH = 40;
```

```
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://qqms-bc0aa-
default-rtdb.europe-west1.firebasedatabase.app/");
        DatabaseReference myRef = database.getReference("messages");

        private Document doc;
        private Thread secThred;
        private Runnable runnable;
        EditText mEditTextMessage;
        Button mSendMessage;
        ArrayList<String> messages = new ArrayList<>();
        RecyclerView mMessagesRecycler;

        String key = "0101001000010110";
      //  String key = getNumber();
        String bufer = "";
        @Override
        protected void onCreate(Bundle savedInstanceState) {
          init();
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);
          mSendMessage = findViewById(R.id.send_message_b);
          mEditTextMessage = findViewById(R.id.message_input);
          mMessagesRecycler = findViewById(R.id.message_recycler);
          DataAdapter dataAdapter = new DataAdapter(this, messages);
          mMessagesRecycler.setLayoutManager(new LinearLayoutManager(this));
          mMessagesRecycler.setAdapter(dataAdapter);
          mSendMessage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
              String mag = mEditTextMessage.getText().toString();
              if (mag.equals("")){
                Toast.makeText(getApplicationContext(), "Enter a message",
Toast.LENGTH_SHORT).show();
                return;
```

```
                }

                if (mag.length() > MAX_MESSAGE_LENGTH){
                    Toast.makeText(getApplicationContext(), "Too long message",
Toast.LENGTH_SHORT).show();
                    return;
                }
                mag = Crypto.encrypt(mag, key);
                myRef.push().setValue(mag);
                mEditTextMessage.setText("");
            }
        });

        myRef.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable
String previousChildName) {
                String mag = snapshot.getValue(String.class);
                mag = Crypto.decrypt(mag, key);
                messages.add(mag);
                dataAdapter.notifyDataSetChanged();
                mMessagesRecycler.smoothScrollToPosition(messages.size());
            }


            @Override
            public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable
String previousChildName) {}

            @Override
            public void onChildRemoved(@NonNull DataSnapshot snapshot) {}

            @Override
```

```
            public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable
String previousChildName) {}

            @Override
            public void onCancelled(@NonNull DatabaseError error) {}
        });
    }

    private void init(){
        runnable = new Runnable() {
            @Override
            public void run() {
                getNumber();
            }
        };
        secThred = new Thread(runnable);
        secThred.start();
    }

    private String getNumber(){
        try {
            doc = Jsoup.connect("https://gtmarket.ru/ratings/world-population").get();
            Elements elements = doc.getElementsByTag("tbody");
            Element first = elements.get(4);
            Elements number = first.children();
            bufer = number.get(0).text();
            Log.d("MyLog", "Number - " + number.get(0).text());
            System.out.println("new - " + bufer);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return bufer;
    }
}
```

Клас, що відповідає за шифрування

```
package com.steam.qqms;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class Crypto {
  private static SecretKeySpec secretKey;
  private static byte[] key;
  public static void setKey(final String myKey) {
    MessageDigest sha = null;
    try {
      key = myKey.getBytes("UTF-8");
      sha = MessageDigest.getInstance("SHA-1");
      key = sha.digest(key);
      key = Arrays.copyOf(key, 16);
      secretKey = new SecretKeySpec(key, "AES");
    } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {
      e.printStackTrace();
    }
  }

  public static String encrypt(final String strToEncrypt, final String secret) {
    try {
      setKey(secret);
      Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```

```
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            return Base64.getEncoder()
                    .encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
        } catch (Exception e) {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static String decrypt(final String strToDecrypt, final String secret) {
        try {
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            return new String(cipher.doFinal(Base64.getDecoder()
                    .decode(strToDecrypt)));
        } catch (Exception e) {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }
}
```