Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Павлюк Аким Володимирович,
група РЗ-171

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Розробка месенджера для прихованої передачі повідомлень

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма Кібербезпека

Керівник:
Кушніренко Наталія Ігорівна,
к.т.н., доцент

Одеса – 2022

Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти другий (магістерський)
Спеціальність 125 Кібербезпека
Спеціалізація, освітня програма Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ

_____
д.т.н.,проф. А.А.Кобозєва
____ _____202_р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Павлюку Акиму Володимировичу

1.Тема роботи: *Розробка месенджера для прихованої передачі повідомлень*,
керівник роботи *Кушніренко Наталія Ігорівна, к.т.н., доцент*
затверджені наказом ректора від „____”_____ 20___ р. №_____ .
2.Зміст роботи*: месенджери та їх методи захисту, опис метода приховання повідомленя, програмна реалізація месенджера з методом прихованої передачі повідомлень*
3. Перелік ілюстративного матеріалу: *Слайди презентації, Результати опитування «улюбленої» соціальної платформи, Додаток «Steganography» в Google Play, Схема використання стего-відео в бот-мережі, Загальна схема методу приховання повідомлення, Приклад згенерованого словника, Приклад частини директорії зі завантаженими зображеннями, Загальна структура пакету, Вікно авторизації користувача, Загальна схема авторизації користувача, Загальний інтерфейс месенджера з відкритим чатом, Список співрозмовників,*

які додані у користувача, Помилка при спробі відправити пусте повідомлення. Загальна схема відправки пакетів в процесі транспортування повідомлення

4. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Дата видачі завдання "_____"_____2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	*Аналіз літератури за темою випускної кваліфікаційної роботи*	*02.09.2022*	*виконано*
2	*Розробка методу приховання повідомлення в зображеннях*	*19.09.2022*	*виконано*
3	*Розробка програмної реалізації месенджера з використанням запропонованого методу приховання повідомлень*	*20.10.2022*	*виконано*
4	*Підготовка тексту роботи*	*14.11.2022*	*виконано*
5	*Підготовка презентації та доповіді*	*28.11.2022*	*виконано*
7	*Попередній захист*	*02.12.2022*	*виконано*
8	*Нормоконтроль, рецензування*	*18.12.2022*	*виконано*
9	*Занесення роботи в електронний архів*	*18.12.2022*	*виконано*
10	*Допуск до захисту*	*18.12.2022*	*виконано*

Здобувач вищої освіти _____ *Павлюк А.В.*

Керівник роботи _____ *Кушніренко Н.І.*

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка месенджера для прихованої передачі повідомлень» на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 125 Кібербезпека, спеціалізація, освітня програма: Кібербезпека, містить 13 рисунків, 4 таблиці, 25 літературних джерел за переліком посилань та 3 додатки. Робота виконана на 62 сторінках загального тексту і 39 сторінках основного тексту.

Метою роботи є розробка методу приховання повідомлень, його програмна реалізація та використання в месенджері.

Об’єктом дослідження роботи є процес забезпечення безпеки особистої інформації користувача при передачі повідомлень за допомогою месенджерів. Предметом – методи приховання інформації та алгоритми транспортування повідомлень в месенджерах.

У роботі було проаналізовано існуючі популярні месенджери та їх захист, використання стеганографії в месенджерах.

У результаті виконання кваліфікаційної роботи розроблено стеганографічний метод приховання повідомлень в наборі зображень та розроблено месенджер, який використовує запропонований метод приховання.

У роботі був удосконалений метод захисту повідомлень при передачі, а також алгоритм транспортування пакетів, що дозволило підвищити безпеку особистої інформації користувача.

Результати даної роботи можуть бути використані як самостійний програмний продукт для спілкування або для імплементування методу приховання в інший програмний продукт.

ІНФОРМАЦІЙНА БЕЗПЕКА, КІБЕРБЕЗПЕКА, МЕСЕНДЖЕР, СТЕГАНОГРАФІЯ, PYTHON

ANNOTATION

Qualification work on the topic "Development of a messenger for hidden transmission of messages" for obtaining the second (master's) level of higher education in the specialty 125 Cybersecurity, specialization, educational program: Cybersecurity, contains 14 figures, 4 tables, 25 literary sources according to the list of references and 3 attachments. The work was completed on 62 pages of the general text and 39 pages of the main text.

The purpose of the work is to develop a method of hiding messages, its software implementation and use in the messenger.

The object of research is the process of ensuring the security of the user's personal information when sending messages using messengers. The subject is methods of hiding information and algorithms for transporting messages in messengers.

The work analyzed existing popular messengers and their protection, the use of steganography in messengers.

As a result of the qualification work, a steganographic method of hiding messages in a set of images was developed and a messenger that uses the proposed method of hiding was developed.

The work improved the method of protecting messages during transmission, as well as the packet transport algorithm, which made it possible to increase the security of the user's personal information.

The results of this work can be used as an independent software product for communication or for implementing the hiding method in another software product.

INFORMATION SECURITY, CYBER SECURITY, MESSENGER, STEGANOGRAPHY, PYTHON

ЗМІСТ

ВСТУП

Інформація є одним з найцінніших предметів сучасного життя. Одержання доступу до неї з появою глобальних комп'ютерних мереж стало неймовірно простим. У той же час легкість і швидкість такого доступу значно підвищили і загрозу порушення безпеки даних при відсутності засобів щодо їх захисту.

Актуальність теми полягає у тому, що в сьогоднішніх реаліях месенджери є невід'ємною частиною майже кожної людини на світі, тому безпека особистої інформації має високий пріоритет.

Найпопулярнішими у світі є такі месенджери: Telegram, Facebook Messenger, Wechat та WhatsApp [1]. А в Україні найпопулярнішою соціальною платформою для спілкування є Viber (97% українців мають завантажено на своїх мобільних пристроях), а також Telegram, Facebook Messenger та WhatsApp [2].

Метою роботи є розробка методу приховання повідомлень, його програмна реалізація та використання в месенджері.

Для досягнення визначеної мети в магістерській атестаційній роботі були сформовані для вирішення наступні задачі:

– аналіз існуючих популярних месенджерів та їх методів захисту інформації, аналіз актуальності стеганографії в месенджерах;

– розробка стеганографічного методу приховання інформації в наборі зображень;

– розробка програмної реалізації запропонованого методу приховання повідомлень;

– дослідження стійкості запропонованого методу приховання до можливих збурених дій зі сторони зловмисника;

– розробка програмної реалізації месенджера, який використовує запропонований метод приховання повідомлень.

Об'єкт дослідження – процес забезпечення безпеки особистої інформації користувача при передачі повідомлень за допомогою месенджерів.

Предмет дослідження – методи приховання інформації та алгоритми транспортування повідомлень в месенджерах [3-4].

Новизна кваліфікаційної роботи полягає в розробці методу кодування повідомлень за допомогою зображень, який вперше використано в користувацькому месенджері.

Отримані практичні результати роботи можуть бути використані як самостійний програмний продукт для спілкування або для імплементування методу кодування в інший програмний продукт.

Результати досліджень були подані у науковій статті «Розробка месенджера для прихованої передачі повідомлень», що опублікована у журналі «Інформатика та математичні методи в моделюванні» [5].

1 МЕСЕНДЖЕРИ ТА ЇХ МЕТОДИ ЗАХИСТУ ПОВІДОМЛЕНЬ

1.1 Актуальність месенджерів та захисту повідомлень

Месенджер - це спеціальний додаток або програма, яку завантажують і встановлюють на смартфон або комп'ютер. Його основна мета - це миттєвий обмін текстовими повідомленнями, фото, картинками, відео, документами з друзями, родичами, знайомими, колегами по роботі або по навчанню. Також можна здійснювати дзвінки за допомогою аудіо або відеозв'язку [6].

Слово «месенджер» походить від англійського «messenger», що означає кур'єр. Обмін повідомленнями йде миттєво, в режимі реального часу. Повідомлення відправляються співрозмовника відразу після того, як відправник закінчить введення, редагування і натисне на кнопку відправки. Але при цьому одержувач повідомлення повинен бути на зв'язку, інакше повідомлення змушене буде чекати, поки він теж запустить свій месенджер і зверне на нього увагу [7].

Програми обміну повідомленнями є одними з найпопулярніших програм по всьому світу, про це свідчать результати опитування, які можна побачити на рисунку 1.1. Мільйони повідомлень відправляються користувачами кожен день. Месенджери надають можливість спілкування з будь-якого місця, де є інтернет, та в будь-який час. З кожним роком популярність месенджерів збільшується, вони стають невід’ємним атрибутом сучасного життя, зростає кількість постійних користувачів. Але зріст попиту на програми спілкування також збільшив і кількість людей, які користуються всіма доступними методами задля викрадення особистої інформації.

Цікаво те, що месенджери є не винаходом 21 століття. Адже перші месенджери виникли ще наприкінці 1980-х у США. Це були програми Talk (1982 рік) і Zephyr (1987). На пострадянському просторі першим месенджером була програма ICQ, що з’явилась у середині 1990-х і швидко завоювала популярність. Сьогодні існує близько сотні її різноманітних аналогів [8-9].

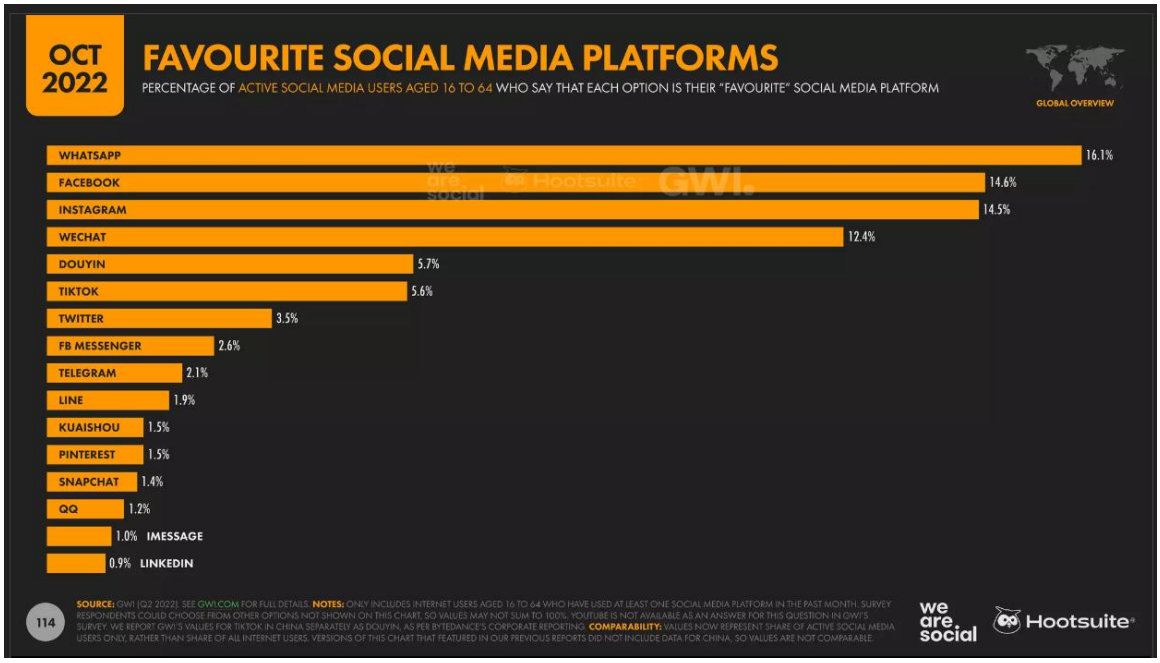

Рисунок 1.1 – Результати опитування «улюбленої» соціальної платформи

Подальший розвиток подібних додатків визначався їхньою функціональністю, кросплатформністю і мобільністю. ICQ спочатку поступився Skype, який підтримував голосові і відеодзвінки [10]. А нині їх витіснили WhatsApp, Viber, Telegram.

Додатки перестали бути виключно засобом для передачі повідомлень, а стали повноцінним джерелом розповсюдження інформації. Месенджери завдяки власному функціоналу стають «новими соціальними мережами». Відбувається це, зокрема, тому, що користувачі незадоволені захищеністю приватних повідомлень, онлайн-дискусіями, що розділяють людей на протилежні табори, і зростанням комерціалізації.

Та на відміну від соцмереж, месенджери, завдяки наскрізному шифруванню даних, набагато надійніше захищають особисту переписку та гарантують конфіденційність інформації. Окрім того, в месенджерах немає можливості коментувати записи, а отже, немає обширних дискусій, на які користувачі витрачали б свій час. У месенджерах немає єдиної стрічки новин, куди надходять

оновлення від ЗМІ і друзів, на яких підписаний користувач. Всі спільноти існують окремо на своїх майданчиках, а доступ до контенту відбувається за підпискою.

Також сповіщення про нові записи від вибраних користувачами каналів, спільнот чи груп дозволяють моментально дізнаватись про події у хронологічному порядку. Це особливо важливо, адже застосування спеціальних алгоритмів формування стрічки, наприклад, у Фейсбук, не дозволяє медіаспоживачам повністю бути у курсі подій.

З розвитком суспільство прийшло до інформаційного середовища, в якому існує маса видів комунікації, що спричинило стрибок у вдосконаленні месенджерів та популяризації таких месенджерів як Facebook Messenger, Telegram, Viber, WhatsApp, Instagram Direct та інші.

1.2 Аналіз існуючих месенджерів та захисту повідомлень

Київським міжнародним інститутом соціології було проведено опитування українців методом телефонних інтерв'ю на основі випадкової вибірки мобільних телефонних номерів. Загалом опитали 2002 респонденти, що мешкають у всіх регіонах України, крім АР Крим.

Результати досить очікувані:

- На першому місці, як і раніше, Viber, яким користується 73,6% опитаних;
- Другим за популярністю є месенджер Facebook – 42,7%;
- На третьому розташувався Telegram – 31,6%;
- Четверте зайняв WhatsApp із результатом у 25,3 відсотка;
- Останнє місце за Signal, який в Україні виявився непопулярним – всього 3,8% користувачів.

Viber оснащений сучасними засобами захисту та технологіями безпеки. Клієнтам месенджер пропонує: наскрізне шифрування, заборона відстеження скріншотів, секретні чати з функціями автоматичного видалення повідомлень, а також захист від копіювання та несанкціонованого пересилання повідомлень.

Однак величезний мінус Viber - це те, що всі резервні копії чатів зберігаються у відкритому вигляді. Це означає, що дані користувачів з серверів месенджера можуть легко викрасти цифрові зловмисники.

Telegram пропонує своїм користувачам наскрізне шифрування, яке здатне надійним щитом закрити ваше листування. Але варто пам'ятати, що ця функція доступна лише для секретних чатів та дзвінків. Звичайні повідомлення Telegram зберігає на своїх серверах, і в разі атаки особиста інформація може легко потрапити до хакерів.

WhatsApp - американський безкоштовний сервіс обміну миттєвими повідомленнями та голосовий зв'язок по IP, що належить компанії Meta. Цей додаток вважається не особливо безпечним, оскільки в пресі неодноразово з'являлися тривожні повідомлення про те, що керівництво WhatsApp надає спецслужбам доступ до особистих даних своїх користувачів. Головні мінуси сервісу - відсутність секретних чатів, зберігання інформації у відкритому вигляді та використання хмарних сховищ для резервних копій.

Коли ви надсилаєте повідомлення другові через Facebook, WhatsApp, Discord або більшість інших служб обміну повідомленнями, повідомлення передається від вас, клієнта, на центральний сервер. Потім центральний сервер направляє повідомлення другому клієнту: вашому другові. У дуже широкому сенсі клієнт запитує послуги, а сервер виконує їх. Це називається моделлю клієнт-сервер. Модель клієнт-сервер надзвичайно поширена і використовується більшістю знайомих вам онлайн-сервісів — від Netflix і Facebook до World of Warcraft. Усі ваші дані зберігаються третьою стороною, і ви повністю покладаєтеся на них. У вас немає способу перевірити, чи вони відповідально обробляють ваші дані, і ви повинні вірити, що вони й надалі дозволятимуть вам користуватися їхніми послугами.

Однорангові послуги (P2P) усувають ці проблеми [11-12]. Комп’ютери можуть обмінюватися інформацією безпосередньо й повністю обійти посередника. Замість того, щоб клієнт доставляв повідомлення на сервер для

передачі другому клієнту, клієнти просто передають дані між собою. У програмах обміну повідомленнями P2P кожен учасник фактично функціонує як клієнт і сервер одночасно.

Шифрування - це спосіб захисту даних від сторонніх очей. Усі популярні сьогодні служби обміну повідомленнями зберігають ваші повідомлення в зашифрованому вигляді, але тут є застереження — у багатьох випадках вони також можуть розкодувати повідомлення без вашого відома. Це означає, що навіть якщо ваші повідомлення можуть бути безпечними (начебто) від сторонніх зловмисників, принаймні можливо, що вони можуть бути прочитані компанією, яка зберігає їх для вас.

Існує крок у порівнянні зі звичайним шифруванням під назвою наскрізне шифрування (E2EE). Налаштування E2EE шифрують повідомлення на пристрої відправника, і повідомлення може бути розшифровано лише призначеним одержувачем(ами). Навіть ваш інтернет-провайдер (ISP) не може їх прочитати.

Поєднання наскрізного шифрування з обміном повідомленнями P2P пропонує найкраще рішення конфіденційності. Ваші повідомлення зашифровані, тобто ніхто не може прочитати їх без ключа шифрування, а копії файлів не зберігаються десь на центральних серверах.

Останній пункт важливий, якщо йдеться мова про те, щоб назавжди зберегти конфіденційність своїх розмов. Поточні схеми шифрування надійні та ефективні проти сучасних атак, але немає гарантії, що вони зможуть протистояти спробам зламати шифрування в майбутньому — особливо коли квантові комп’ютери стануть життєздатною технологією.

Стеганографія — темний родич криптографії, використання кодів. У той час як криптографія забезпечує конфіденційність, стеганографія призначена для забезпечення секретності. Конфіденційність – це те, що вам потрібно, коли ви використовуєте свою кредитну картку в Інтернеті – ви не хочете, щоб ваш номер став відкритим. Для цього ви використовуєте криптографію та надсилаєте закодовану купу тарабарщини, яку може розшифрувати лише веб-сайт. Хоча ваш

код може бути незламним, будь-який хакер може подивитися та побачити, що ви надіслали повідомлення. Для справжньої таємниці ви не хочете, щоб хтось знав, що ви надсилаєте повідомлення.

Геродот, цікавий грецький історик, повідомляє про геніальний метод застосування стеганографії [13]. Гістей, правитель Мілета, хотів надіслати послання своєму другові Арістагору, закликаючи до повстання проти персів. Гістей поголив голову свого найбільш довіреного раба, а потім витатуював повідомлення на шкірі раба. Коли волосся відросло, раба відправили до Арістагора з надійно схованим повідомленням.

Пізніше в історіях Геродота спартанці отримали повідомлення про те, що Ксеркс готується вторгнутися в Грецію. Їхній інформатор, Демерат, був греком у вигнанні в Персії. Побоюючись розкриття, Демерат написав своє повідомлення на дерев'яній підкладці воскової таблички. Потім він сховав повідомлення під свіжим шаром воску. Очевидно порожня табличка легко пройшла повз увагу вартових на дорозі.

Більш тонкий метод, майже такий самий старий, полягає у використанні невидимого чорнила. Описані ще в першому столітті нашої ери, невидимі чорнила зазвичай використовувалися для серйозних комунікацій аж до Другої світової війни. Найпростішими є органічні сполуки, такі як лимонний сік, молоко або сеча, які темніють, якщо їх тримати над полум'ям.

У 1641 році єпископ Джон Вілкінс запропонував цибульний сік, галун, солі аміаку, а для написання світяться в темряві — «дистильований сік світлячків» [14]. Сучасні невидимі чорнила флуоресціюють під ультрафіолетом і використовуються як засоби захисту від підробок. Наприклад, "VOID" друкується на чеках та інших офіційних документах чорнилом, яке з'являється під сильним ультрафіолетовим світлом, яке використовується для фотокопій.

1.3 Аналіз використання стеганографії в месенджерах

Можливість приховувати повідомлення за допомогою стеганографії має соціальні та етичні наслідки, подібні до криптографії. Деякі уряди вжили заходів щодо заборони використання криптографії або певних типів криптографії. Ці заборони часто застосовуються, щоб дозволити органам влади контролювати комунікації; Вважається, що безпека людей потребує обмежень конфіденційності комунікацій. В даний час стеганографічні методи можна використовувати, коли шифрування заборонено.

Стеганаліз, виявлення стеганографічних повідомлень в електронних носіях, часто може виявити наявність стеганографічних повідомлень. Таким чином, зовнішні сторони (наприклад, уряди) можуть докладати зусиль, щоб виявити та, можливо, відстежити як відправника, так і одержувача стеганографічних повідомлень.

В інтернеті можна знайти також месенджери, в яких для захисту, а точніше приховання повідомлень, використовується стеганографія. В деяких таких проєктах опублікований вихідний код або доданий додаток на такі платформи, як Google play [15] (див. рис.1.2).

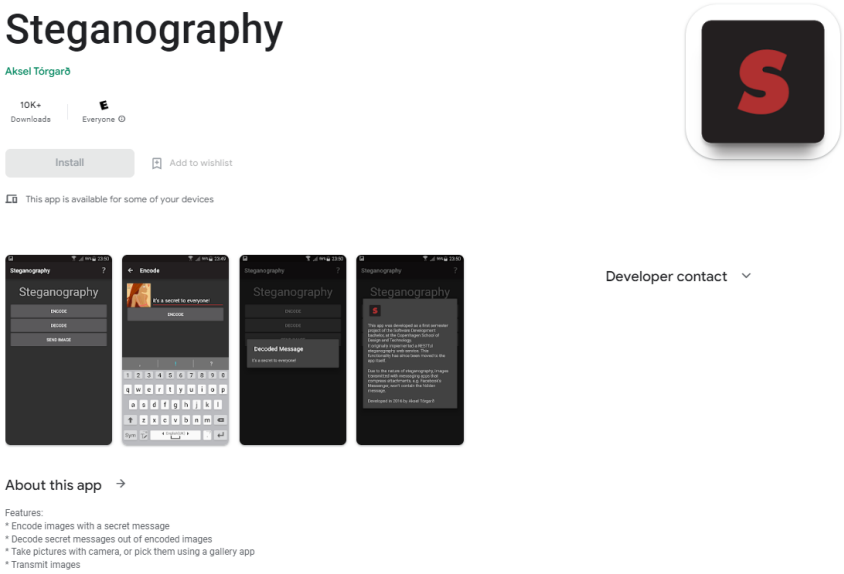

Рисунок 1.2 – Додаток «Steganography» в Google Play

Спочатку програма «Steganography» була розроблена як екзаменаційний проект у Копенгагенській школі дизайну та технологій. Вона використовувала веб-сервіс для функції стеганографії. Пізніше, програму було перенесено на телефон та опубліковано в Google Play.

Нещодавно з'явилися бот-мережі на основі стеганографії (стего-ботнети), які дозволяють системам виявлення ботнетів виглядати звичним трафіком [16-18]. У стего-ботнетах кожне повідомлення вбудовано в мультимедійний файл, наприклад файл зображення, за допомогою методів стеганографії та розміщується на веб-сайтах служби соціальних мереж (таких як Facebook) або в онлайн-месенджерах (таких як WeChat або KakaoTalk).

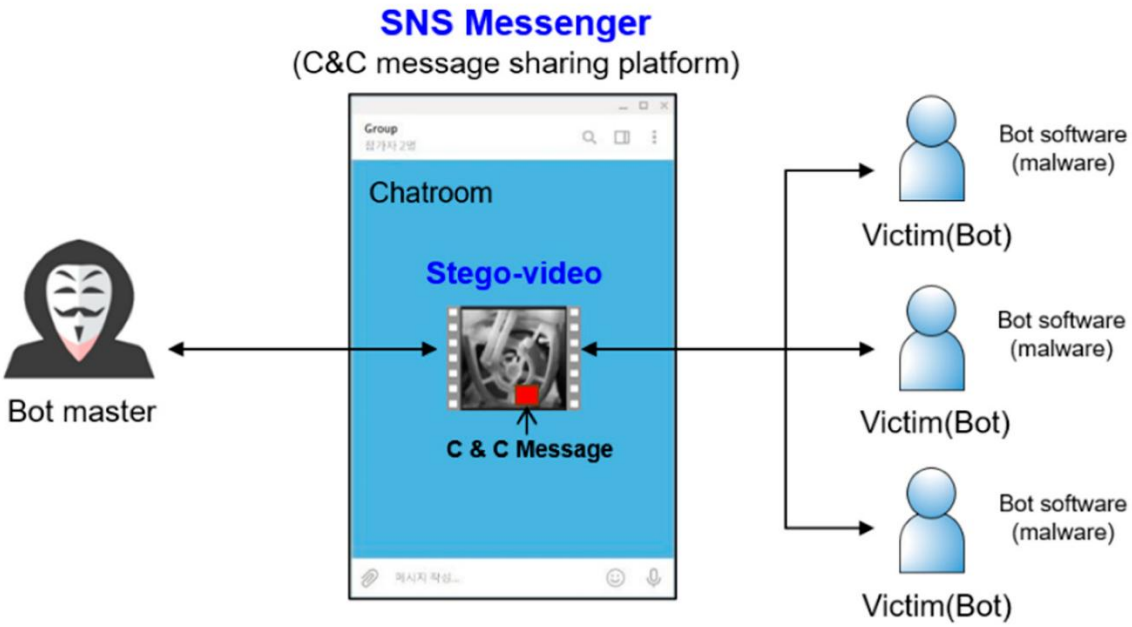

Рисунок 1.3 – Схема використання стего-відео в бот-мережі

Традиційні системи виявлення ботнетів без методів виявлення стеганографії не можуть їх виявити. Тим часом, згідно з опитуванням, помічено, що існуючі дослідження ботнету стеганографії обмежені використанням лише методів стеганографії зображень, хоча метод відеостеганографії має деякі очевидні переваги перед методом стеганографії зображень. З огляду на цю мотивацію, у статті досліджено ботнет на основі відеостеганографії на платформах соціальних мереж.

Кемпбелл-Мур розробила плагін Secretbook, щоб приховати текстові повідомлення довжиною до 140 символів у зображеннях JPEG у Facebook за допомогою браузера Google Chrome [19-22]. Стаття Бекхузена чудово розкриває проблему стеганографії Facebook і пояснює рішення Кемпбелла-Мура. Коли хтось завантажує зображення на Facebook, воно автоматично стискається. Якщо на зображенні є стеганографія, Facebook спотворює його. Алгоритм Secretbook автоматично стискає зображення JPEG, як це зробив би Facebook, а потім додає приховані дані стеганографії. Алгоритм також додає надлишковість, тому будь-які спотворення, що залишилися, можна виправити шляхом реконструкції з копій.

В цій роботі запропоновано розробити власний месенджер для спілкування, який буде використовувати стеганографічний метод приховання повідомлення, який також запропоновано в цій роботі та перевірена його стійкість до можливих збурених дій зловмисника.

2 ОПИС МЕТОДА ПРИХОВАННЯ ПОВІДОМЛЕННЯ

2.1 Загальний опис методу

В роботі запропоновано новий метод приховання повідомлення за допомогою зображень. Для користуванням методу потрібен приватний ключ і набір випадкових змістовних зображень.

В самому месенджеру приватний ключ отримується за допомогою обміну публічними ключами алгоритмом Діффі-Геллмана. Обмін ключами між користувачами детально описано в розділі 3.2.

За допомогою приватного ключа формується спеціальний словник відношення «піксель-символ», який в подальшому буде використаний для перетворення повідомлення на набір послідовних зображень. Формування словника відношення «піксель-символ» детально описано в розділі 2.2.

Набір випадкових змістовних зображень автоматично завантажується з інтернету за випадковим пошуковим запитом. Кількість зображень буде становити більш, ніж 200 (в залежності від часу, але не більше 5 хвилин). Як і за допомогою чого виконується завантаження зображень з інтернету описано в розділі 2.3.

Після того, як було підготовлено словник відношення «піксель-символ» та набір зображень, метод приховання повідомлення готовий до кодування тексту.

По черзі, починаючи з першого символу, метод починає перетворювати кожен символ на зображення, яке підходить для його кодування. Сам метод нічого не вбудовує в зображення, якщо відповідне до символу зображення було завантажене. В іншому випадку метод змінює лише значення одного пікселя.

Через те, що алгоритм використовує для приховання повідомлення лише значення одного пікселя, цей метод є стійким до виявлення наявності контейнера в зображенні стеганоаналізом.

На рисунку 2.1 схематично відображено загальну схему роботи запропонованого методу приховання повідомлення.

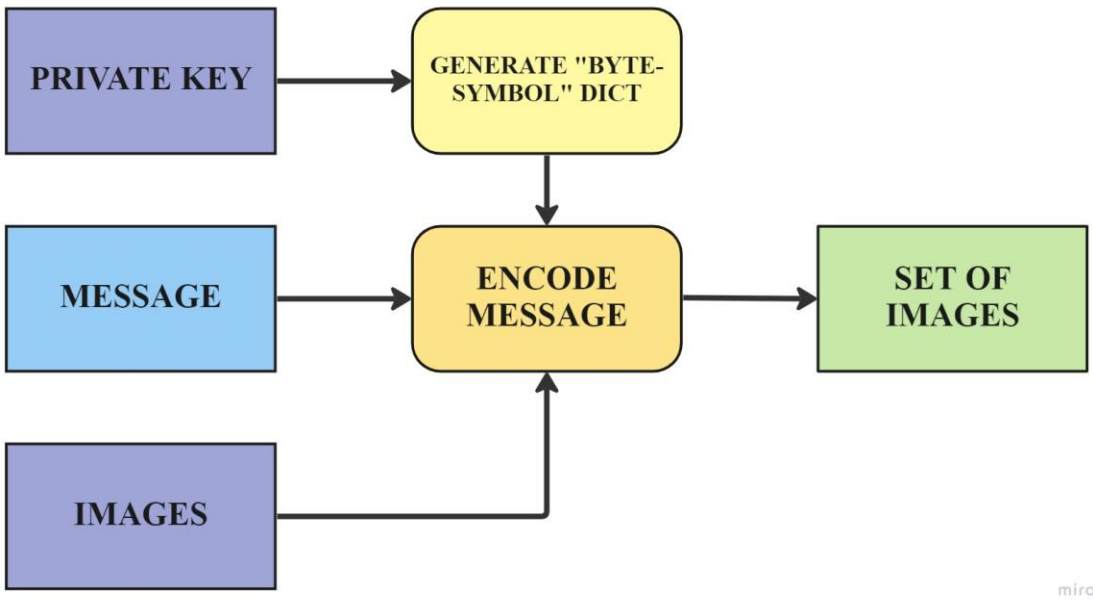

Рисунок 2.1 – Загальна схема методу приховання повідомлення

В розділі 2.5 перевірена стійкість методу приховання до збурених дій, які може використати зловмисник по відношенню до зображень, які передаються.

2.2 Формування словника відношення «піксель-символ» за допомогою ключа

Для формування словника відношення «піксель-символ» потрібен ключ. Опис обміну ключами між співрозмовниками детально описано в розділі 3.2.

Алфавіт месенджера, який було використано в цій роботі, складається з великих літер англійської абетки («A-Z», всього 26 літер) та символів «.» (точка), «,» (кома), «!» (знак оклику), «?» (знак питання), « » (пробіл). Всього - 31 символ.

Для кожного символу з алфавіту месенджера було відведено визначену кількість значень пікселів, а саме:

«A – Z» - 8 значень;

«!» - 9 значень;

«?» - 9 значень;

«,» - 9 значень;

«.» - 10 значень;

« » - 11 значень.

Всього – 256 значень, що дорівнює кількості значень, яких може набувати яскравість пікселя.

Всі символи згруповані в кортежі: кожен контейнер містить символ у відповідній йому кількості (наприклад, «AAAAAAAA», «BBBBBBBB», …). Такий підхід використано з метою, щоб у подальшому збільшити захист методу від збурених дій (стеганографічних атак), які можуть бути використані по відношенню до зображень, які використовуються як контейнери для передачі повідомлення.

Для того, щоб сформувати випадковий словник відношення «піксель-символ», де тепер за символ можна рахувати одразу відрізок символів, використовується бібліотека *random* в Python. За допомогою методу *sample* в модулі *random* формуємо випадкову послідовність вище зазначених кортежів символів, після чого задаємо відповідність значенням пікселів від 0 до 255 відповідно до сформованої випадкової послідовності.

Щоб відправник і отримувач повідомлення мали той самий словник відношення «піксель-символ» використовується ключ як *seed* для модуля *random*. Такий *seed* дозволяє використати властивості псевдовипадкового

формування послідовностей, тому сформований словник буде однаковий як у відправника, так і в отримувача [23-24].

На виході функції формування словника відношення «піксель-символ» отримуємо словник в Python такого виду, як на рисунку 2.2.

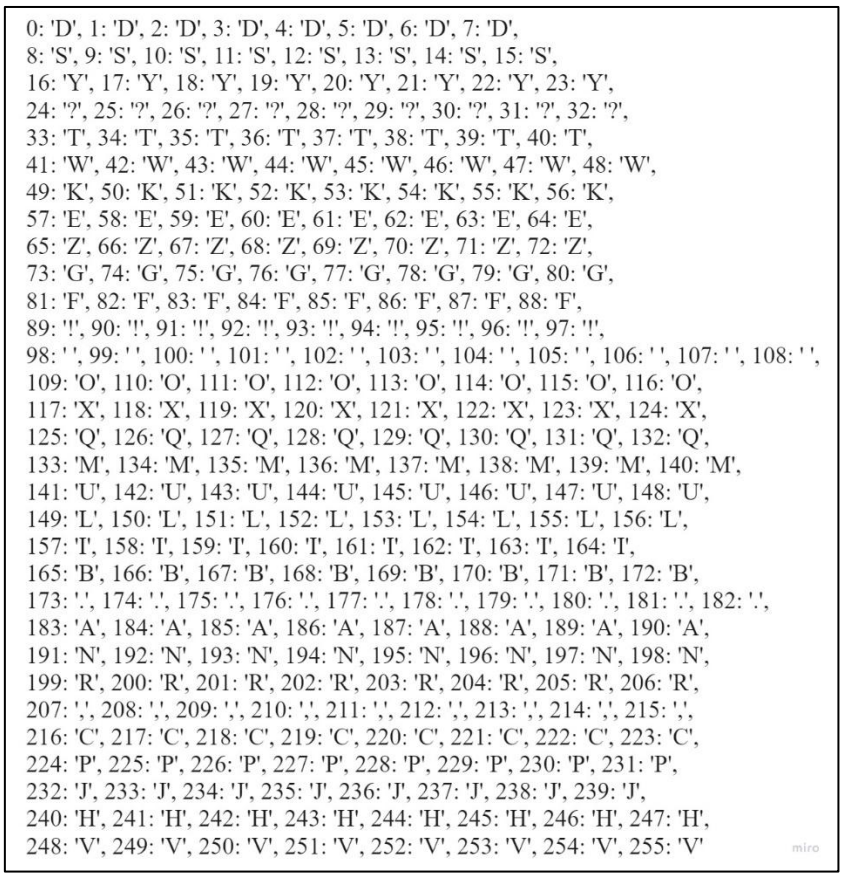

Рисунок 2.2 – Приклад згенерованого словника

2.3 Завантаження набору зображень, які будуть передаватись

Завантаження зображень виконується за допомогою бібліотек *urllib*, *json*, *imghdr*, *random_word*. Процес виконується поки не набереться хоча б по одному зображенню на двісті будь-яких значень пікселів або поки не вийде тайм аут в 5 хвилин. Завантаження зображень відбувається за допомогою Google API [25].

По-перше, за допомогою модуля *random_word* генерується випадкове існуюче слово англійською мовою. Це слово або набір слів використовується

для реквесту як пошуковий запит. Таким чином кожне завантажене зображення буде мати різну тематику.

Далі, формується реквест для googleapis.com з використанням наступних параметрів:

- API_KEY;
- SEARCH_ENGINE_ID;
- згенероване випадкове англійське слово;
- тип пошуку – в нашому випадку це image.

Запит вертає набір параметрів для перших 20-ти зображень, які знайдені за цим словом. Випадковим чином обирається одне із знайдених зображень та починається завантаження зображення на комп’ютер. Якщо зображення неможливо завантажити, то процес починається знову. Якщо зображення неможливо зберігти, то збережені дані по цьому зображенню видаляються і процес починається знову. Якщо зображення дуже маленьке (менше 100x100) або формат збереження зображення не «jpg», «jpeg», «png», «bmp», то зображення видаляється і завантажується нове.

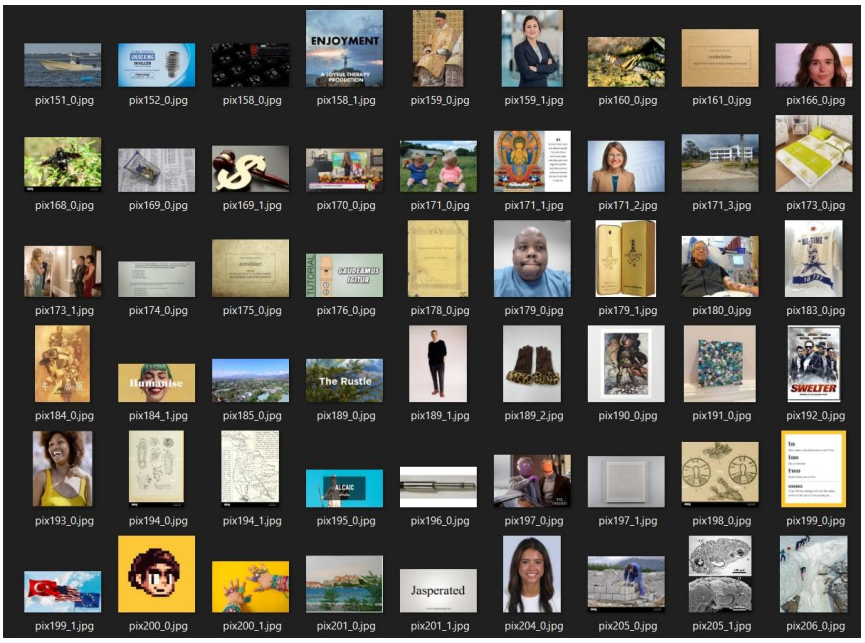

Рисунок 2.3 – Приклад частини директорії зі завантаженими зображеннями

В нових реквестах до пошукового запиту додаються ключові слова «jpg», «photo» або інше англійське згенероване слово, щоб повторний реквест не привів до безкінечної рекурсії непідходящих зображень.

Таким чином, на виході ми отримуємо 200+ випадкових зображень з інтернету, які будуть використані в подальшому для кодування повідомлень, які відправляють.

2.4 Перетворення повідомлення на набір зображень

Формування набору зображень з повідомлення відбувається за допомогою двох вхідних параметрів: ключа та набору випадкових зображень.

Ключ отримано в ході обміну ключами (розділ 3.2). Набір зображень відбувається на початку сесії користувача.

Перед кодуванням повідомлення, воно перевіряється на наявність символів не з алфавіту – такі символи видаляються. Потім, всі інші символи міняються на заголовні, якщо вони були маленькі. Повідомлення готове до наступного етапу.

Починаючи з першого символу повідомлення починається кодування:

1. Для обраного символу підбирається відповідність значень яскравості пікселя за допомогою згенерованого словника відношення «піксель-символ». В результаті отримуємо перелік значень яскравості пікселя.
2. Серед завантажених зображень відбувається пошук таких, в яких обраний піксель яскравості є в переліку значень, який сформовано в пункті 1. Серед зображень, які підходять для кодування обраного символу, випадковим чином обирається будь-яке. Якщо не знайдено жодного зображення, яке підходить для кодування символу, то обирається випадкове серед усього переліку завантажених зображень та автоматично міняє обраний піксель на значення, яке підходить для кодування символу. В результаті отримуємо зображення.

3. Обране зображення додається до переліку зображень для відправки. Пункти 1-2 виконуються поки не закодується повідомлення повністю.

Таким чином, на виході отримуємо перелік зображень, в яких закодовано повідомлення і які готові до відправки іншому користувачу.

2.5 Перевірка стійкості методу приховання повідомлень до збурених дій

Зображення, які передаються, можуть бути перехоплені та піддатися збуреної дії зі сторони зловмисника, після чого знову передані до отримувача. Такими діями можуть бути:

1. накладення непомітного шуму;
2. розмиття;
3. повторне стиснення.

На практиці було перевірено стійкість методу до накладення шуму Гауса з різним коефіцієнтом на 100-та випадкових зображеннях. Результати перевірки можна побачити в таблиці 1.

Таблиця 2.1 – Результати перевірки стійкості методу до накладання шуму Гауса

Коефіцієнт шуму Гауса	Кількість закодованих символів	Кількість успішно декодованих символів
0.08	100	96
0.1	100	95
0.15	100	90
0.2	100	86
0.3	100	73
0.5	100	59

1	100	32

За результатами перевірки стійкості метода до шуму Гауса можна сказати, що метод є стійким по відношенню до непомітного для користувача шуму. При коефіцієнті до 0.15 зберігається більше, ніж 90% повідомлення.

Для наступного тестування стійкості було використано шум Лапласа.

Таблиця 2.2 – Результати перевірки стійкості метода до накладання шуму Лапласа

Коефіцієнт шуму Лапласа	Кількість закодованих символів	Кількість успішно декодованих символів
0.03	100	96
0.05	100	95
0.08	100	93
0.1	100	93
0.15	100	93
0.2	100	86
0.3	100	85
0.5	100	77
1	100	50

За результатами перевірки стійкості метода до шуму Лапласа можна сказати, що метод є стійким по відношенню до непомітного для користувача шуму. При коефіцієнті до 0.15 зберігається більше, ніж 93% повідомлення.

Наступною збуреної дією було розмиття для перевірки стійкості метода приховання повідомлень. В ході перевірки було використано конкретний алгоритм розмиття зображення, а саме – розмиття Гауса. В таблиці 2.3 можна побачити результати перевірки стійкості методу приховання з різними коефіцієнтами розмиття.

Таблиця 2.3 – Результати перевірки стійкості метода до накладання розмиття Гауса

Коефіцієнт розмиття Гауса	Кількість закодованих символів	Кількість успішно декодованих символів
0.05	100	89
0.2	100	86
0.5	100	86
1	100	84
2	100	77
3	100	73
5	100	70

За результатами перевірки стійкості метода до розмиття Лапласа можна сказати, що метод є стійким по відношенню до непомітного для користувача розмиття. При коефіцієнті до 1 зберігається більше, ніж 84% повідомлення.

Наступною збуреною дією було стиснення за алгоритмом JPEG. Результати перевірки можна побачити в таблиці 2.4.

Таблиця 2.4 – Результати перевірки стійкості метода до повторного стиснення зображення

Коефіцієнт якості зображення	Кількість закодованих символів	Кількість успішно декодованих символів
95	100	98
90	100	96
80	100	91
70	100	84
60	100	81
50	100	73

За результатами перевірки стійкості метода до повторного стиснення JPEG можна сказати, що метод є стійким по відношенню до непомітного для користувача стиснення. При коефіцієнті якості до 70 зберігається більше, ніж 84% повідомлення.

Отже, стійкість запропонованого методу приховання повідомлення було перевірено різними збуреними діями з різними коефіцієнтами та визначено, що метод є стійким по відношенню таких дій, які намагаються атакувати повідомлення так, щоб цього не помітив отримувач.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕСЕНДЖЕРА З МЕТОДОМ ПРИХОВАНОЇ ПЕРЕДАЧІ ПОВІДОМЛЕНЬ

3.1 Сервер та авторизація користувача

Для функціонування месенджера було створено окремий сервер. Сервер має три запущених потоки для приєднання:

1. Потік для передачі повідомлень користувачам;
2. Потік для отримання повідомлень від користувачів;
3. Потік для авторизації користувачів.

Передача інформації між сервером та користувачем відбувається шляхом надсилання послідовних пакетів з байтами. Пакети мають структуру як показано на рисунку 3.1.

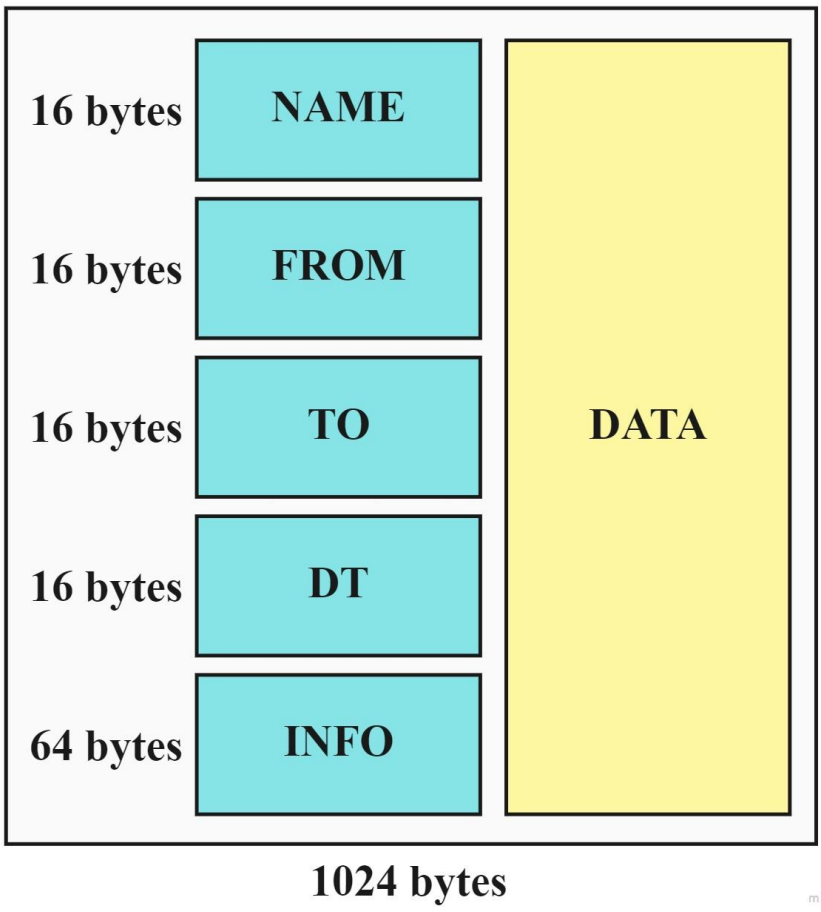

Рисунок 3.1 – Загальна структура пакету

Поля «NAME», «FROM», «TO», «DT» в заголовку мають розмір 16 байтів, поле «INFO» - 64 байти. Все інше місце до 1024 байтів займає поле «DATA», яке за замовчуванням при розпаковці пакета не декодується.

Для безпечного обміну паролем між користувачем та сервером пароль передається у вигляді хешу. Хешування паролю відбувається за допомогою хеш-функції SHA-256. На сервері пароль також зберігається у вигляді хешу.

Щоб увійти в систему користувачу потрібно ввести логін та пароль у поля логін системи, які можна побачити на рисунку 3.2, після чого відправляється пакет з цими даними на сервер.

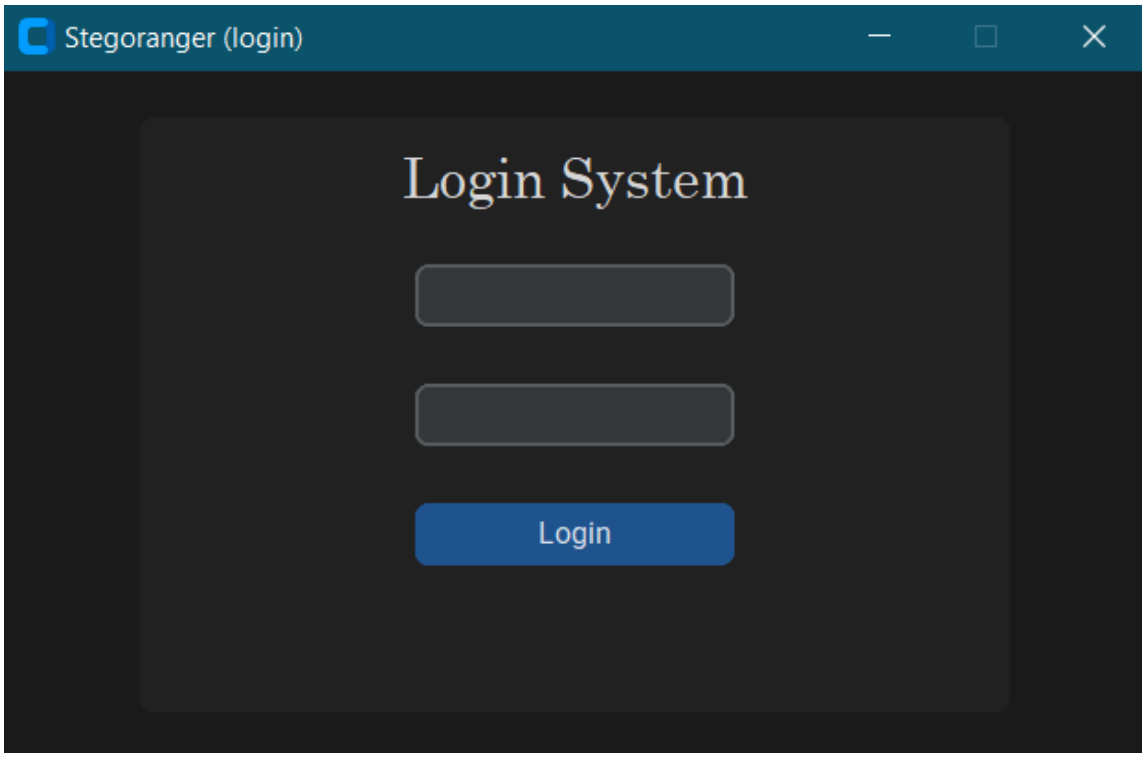

Рисунок 3.2 – Вікно авторизації користувача

Сервер спочатку перевіряє чи зареєстрований користувач з даним юзернеймом в мережі. Якщо такий користувач не зареєстрований, то сервер відправляє пакет-відповідь з заголовком «AUTH_FAIL», де у додатковій інформації позначає, що даний користувач не зареєстрований. Якщо ж користувач зареєстрований, то сервер перевіряє чи відповідає введений пароль паролю, який знаходиться в базі акаунтів на сервері. Якщо пароль не відповідає, то сервер відправляє пакет-відповідь з заголовком

«AUTH_FAIL», де у додатковій інформації позначає, що введений пароль не відповідає паролю на сервері. Якщо ж пароль підходить, то сервер відправляє пакет-відповідь «AUTH_OK» і юзер входить до системи. На сервері одразу реєструється його чинна IP адреса та призначається статус online.

На рисунку 3.3 відображена загальна схема авторизації користувача.

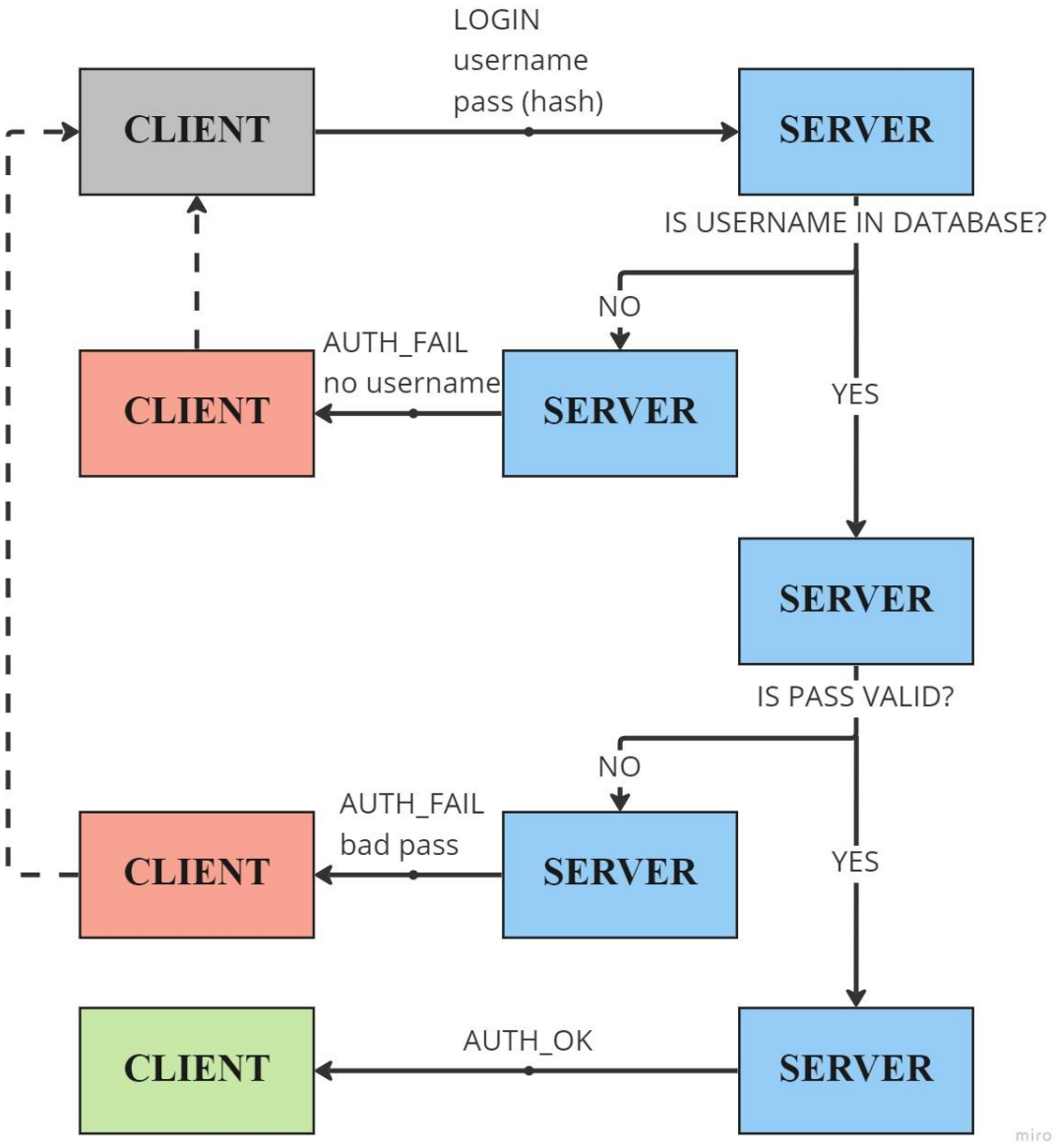

Рисунок 3.3 – Загальна схема авторизації користувача

Після успішної авторизації користувач переходить до вікна спілкування з іншими юзерами, яке відображено на рисунку 3.4.

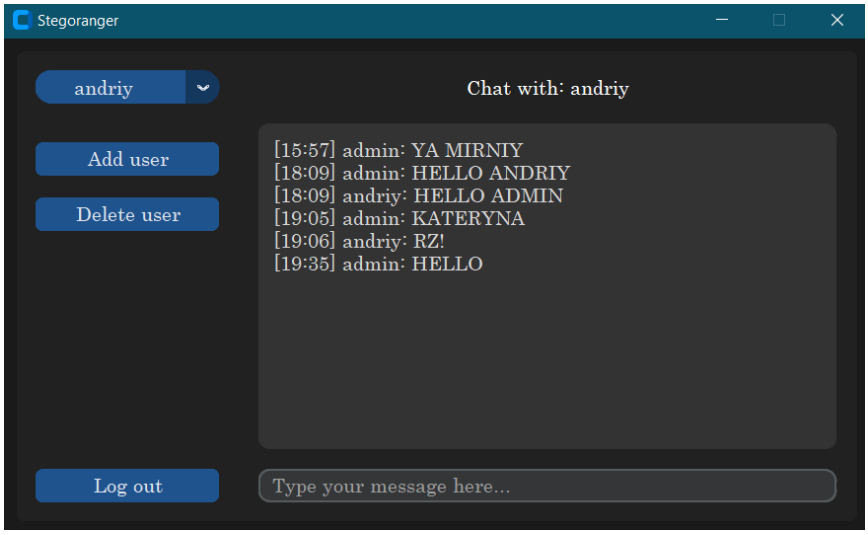

Рисунок 3.4 – Загальний інтерфейс месенджера з відкритим чатом

Також, користувач підключається до сервера на два різні порти: для відправки повідомлень і для отримання. Підключення до сокету авторизації більше не потрібне, тому воно припиняється.

Для того, щоб вийти з акаунта, можна скористуватись кнопкою «Log out», яке припинить всі підключення користувача до сервера та поверне до вікна авторизації.

3.2 Обмін ключами між користувачами

Кодування повідомлення можливе за наявністю однакового ключа як у відправника, так і у отримувача. Тому, перед відправленням чи отриманням повідомлення потрібно отримати одразу ключ або відкритий ключ, який допоможе згенерувати закритий.

Мета цього месенджера орієнтована на прихованні передачі повідомлень через незахищений канал зв'язку. Але для передачі повідомлення спочатку потрібен ключ, який повинен передаватись через незахищений канал зв'язку. Тому одним з найкращих вирішень проблеми є обмін ключами за допомогою протокола Діффі-Геллмана.

На початку сесії користувача генеруються значення g, p, a. Далі, за допомогою формули 3.1 рахується відкритий ключ x користувача.

$$x = g^a mod\ p, \tag{3.1}$$

де g і p – прості числа, які відповідають умовам протоколу Діффі-Геллмана, і є відкритими ключами;

a – випадкове ціле позитивне число, яке є закритим ключом цього користувача.

Після створення відкритих ключів, вони передаються на сервер, щоб інші користувачі могли згенерувати ключ для спілкування з цим користувачем.

Після виходу користувача відкриті ключі зберігаються на сервері поки користувач офлайн, до наступної його сесії.

3.3 Відправлення повідомлення користувачем

Для передачі повідомлення користувач має спочатку обрати співрозмовника зі списку, який випадає (див. рис.3.5). Якщо у користувача вже є отримані або відправленні повідомлення з цим співрозмовником, то вони завантажуються з локального файлу, який зберігає всі бесіди для користувача.

Якщо користувач зробить спробу відправити повідомлення до того, як вибере співрозмовника, то програма попередить користувача червоним текстом, що він має обрати співрозмовника.

Рисунок 3.5 – Список співрозмовників, які додані у користувача

Користувач не може відправити пусте повідомлення, тому для того, щоб щось відправити, він має ввести в строку хоча б один символ. При спробі відправити пусте повідомлення відобразиться червоне попередження, як на рисунку 3.6.

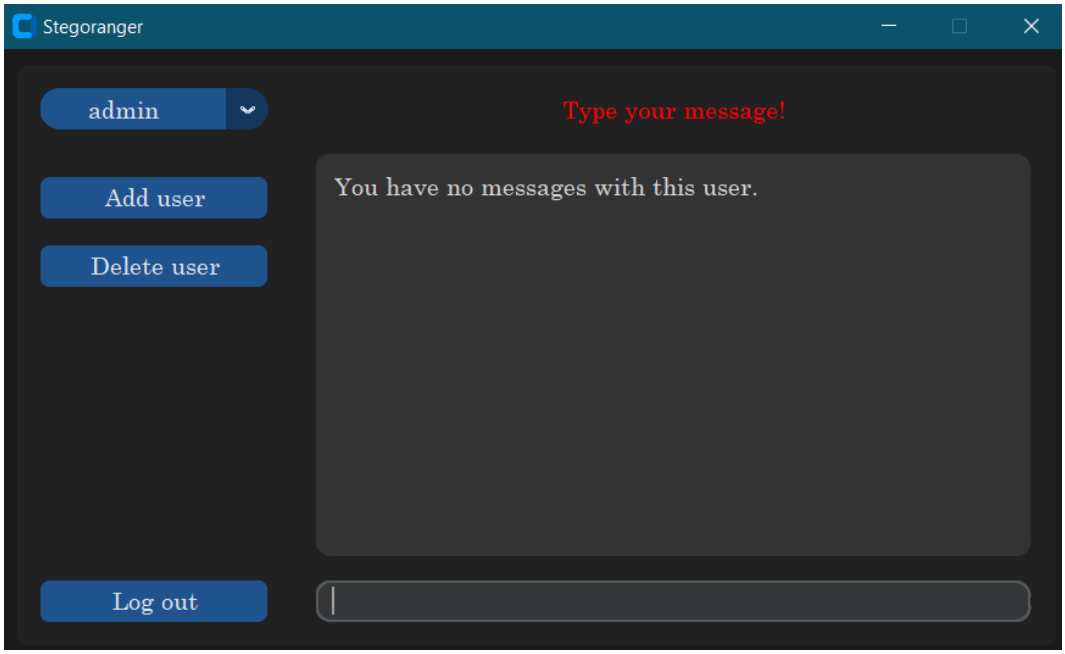

Рисунок 3.6 – Помилка при спробі відправити пусте повідомлення

Після введення тексту юзер натискає клавішу Enter і програма починає процес передачі повідомлення.

В першу чергу, програма відправляє на сервер пакет із заголовком «SEND» для отримання відкритих ключів співрозмовника. Користувач отримує від сервера пакет-відповідь з заголовком «SEND_OK» та публічними ключами g, p, x співрозмовника в самому пакеті.

Програма генерує приватний ключ b та на основі публічних ключів співрозмовника вираховує публічний ключ y, який буде надіслано разом із повідомленням співрозмовнику. Після чого рахує ключ k для подальшого листування.

Використовується ключ *k* для створення словника відношення «піксель-символ». Детальний опис процесу створення словника описано в розділі 2.1. За допомогою цього словника програма відбирає зображення для відправки, в яких закодовано повідомлення. Коли повідомлення закодовано, всі зображення по черзі починаються відправлятись на сервер пакетами по 1024 байт. Сервер зберігає всі ці пакети в свої базі. В додатковій інформації в кожному пакеті фіксуються два параметри: IMG_END, MSG_ENG.

IMG_END – параметр в додатковій інформації пакета, якій відповідає за те, чи є цей пакет останнім пакетом зображення, яке зараз передається.

MSG_END – параметр в додатковій інформації пакета, якій інформує про те, що цей пакет є останнім пакетом в процесі передачі цього повідомлення.

Після успішної передачі останнього пакета сервер відправляє пакет-відповідь з заголовком «MESSAGE_OK» і програма фіксує повідомлення як відправлене та записує його в локальний файл бесіди.

Загальну схему відправки пакетів можна побачити на рисунку 3.7.

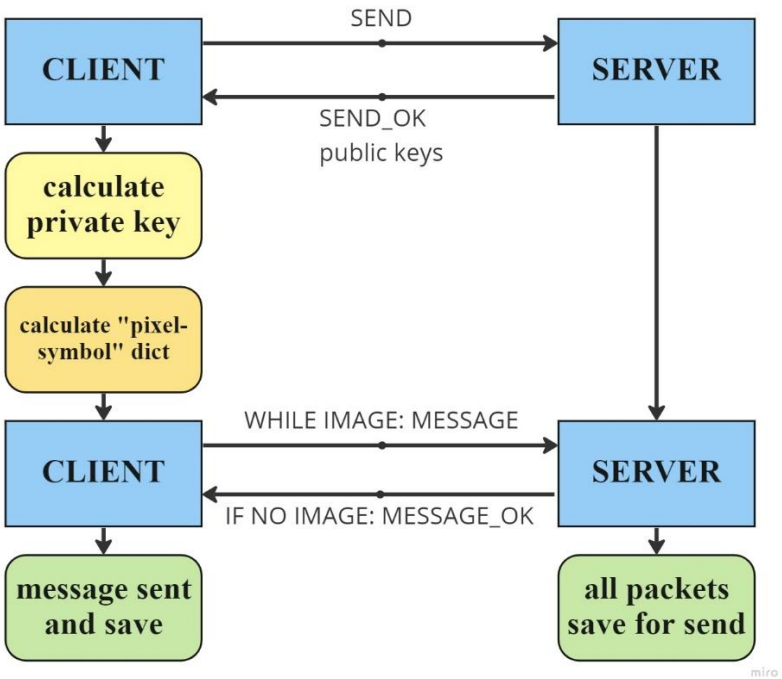

Рисунок 3.7 – Загальна схема відправки пакетів в процесі транспортування повідомлення

Таким чином відправляється повідомлення від користувача до його співрозмовника.

3.4 Отримання повідомлення користувачем

Якщо співрозмовник, якому було надіслано повідомлення, тільки зайшов у месенджер або вже знаходиться онлайн, то сервер починає відправку всіх пакетів, які йому надіслав відправник повідомлення.

Першим пакетом від сервера до співрозмовника є привітальним, де зберігаються дані про публічні ключі. Користувач перевіряє чи збігається публічний ключ, який вказаний в пакеті, з ключом, який збережено у цього користувача. Отримання повідомлень та перевірка ключа відбувається до зміни публічного ключа при старті сесії.

Якщо ключі співпадають, то починається отримання всіх інших пакетів. З отриманих пакетів формуються зображення та зберігаються локально в директорії, яка призначена для отриманих зображень.

Починаючи з першого зображення месенджер починає декодувати повідомлення, використовуючи при цьому сформований словник відношення «піксель-символ». Словник було сформовано за принципом, який використовується у відправника.

Коли повідомлення декодовано, то воно зберігається в локальному файлі бесід. Якщо користувач відкриє чат з юзером, який відправив йому це повідомлення, то воно відобразиться на екрані. Після цього всі отримані зображення видаляються з директорії.

ВИСНОВКИ

У результаті виконання магістерської атестаційної роботи: була. доведена актуальність обраної теми, обґрунтована мета розробки методу приховання інформації та поставлені задачі для досягнення виконання мети.

У ході роботи були виконані наступні задачі:

1. Проведено аналіз існуючих популярних месенджерів та їх методів захисту інформації. Розглянуто слабкі сторони месенджерів, а також деякі історії витоку особистої інформації. Проаналізовано використання стеганографії в сучасних месенджерах.
2. Було розроблено стеганографічний метод приховання інформації без модифікації самих зображень.
3. Розроблено програмну реалізації запропонованого методу приховання повідомлень та проведено тестування стійкості методу до можливих збурених дій зі сторони зловмисника.
4. Розроблено програму реалізації месенджера, який використовує запропонований метод приховання повідомлень, з користувацьким інтерфейсом.

За результатами магістерської наукової роботи можна зробити висновок, що поставлена мета – розробка методу приховання інформації та програмної реалізації месенджера, який використовує запропонований метод, була успішно виконана.

ПЕРЕЛІК ПОСИЛАНЬ

1. WhatsApp, WeChat and Facebook Messenger: global usage of messaging apps and statistics. URL: https://www.messengerpeople.com/global-messenger-usage-statistics/

2. Які мобільні додатки є найбільш популярними? URL: https://www.kiis.com.ua/?lang=ukr&cat=reports&id=1072&page=1

3. Кінзерявий О.М. Стеганографічні методи приховування даних у векторні зображення, стійкі до активних атак на основі афінних перетворень: дис. ... канд. техн.. наук. Спеціальність 05.13.21 Системи захисту інформації. Київ , 2015. 324 с.

4. Sun Y., Lu Y., Chen J., Zhang W., Yan X. Meaningful secret image sharing scheme with high visual quality based on natural steganography. *Mathematics*. 2020, 1452 c.

5. Павлюк А.В., Кушніренко Н.І., Троянський О.В. Розробка месенджера для прихованої передачі повідомлень. *Інформатика та математичні методи в моделюванні.* 2022. URL: http://immm.op.edu.ua

6. Audio and video calls. URL: https://www.facebook.com/help/messenger-app/1673374996287506

7. Messenger. URL: https://en.wikipedia.org/wiki/Messenger_(software)

8. Zephyr: Hiding Metadata in a Messaging System. URL: https://doi.org/10.48550/arXiv.1910.13337

9. Zephyr the Roaming Dinosaur. URL: https://www.gazette-drouot.com/en/article/zephyr-the-roaming-dinosaur/38628

10. Як месенджери змінили спілкування. URL: https://www.imena.ua/blog/the-messenger-changed-humanity/

11. What is Peer to Peer (P2P) Messaging? URL: https://shazzle.com/articles/what-is-peer-to-peer-p2p-messaging/#:~:text=Peer%20to%20peer%20technology%20and,and%20any%20other%20private%20information.

12. Why You Should Use Peer-to-Peer Messaging Apps. URL: https://www.howtogeek.com/790612/why-you-should-use-peer-to-peer-messaging-apps/

13. Herodotus. Greek historian. URL: https://www.britannica.com/biography/Herodotus-Greek-historian

14. John Wilkins. URL: https://mathshistory.st-andrews.ac.uk/Biographies/Wilkins/

15. Stenography. Mobile app. URL: https://play.google.com/store/apps/details?id=com.akseltorgard.steganography&hl=en&gl=US&pli=1

16. A Novel Video Steganography-Based Botnet Communication Model in Telegram SNS Messenger Bot. URL: https://doi.org/10.3390/sym13010084

17. Enhanced PeerHunter: Detecting Peer-to-Peer Botnets Through Network-Flow Level Community Behavior Analysis. URL: https://doi.org/10.1109/TIFS.2018.2881657

18. A Survey on Botnet Detection Techniques. URL: https://doi.org/10.1109/ic-ETITE47903.2020.Id-70

19. Steganography: how to send a secret message. URL: http://strangehorizons.com/non-fiction/articles/steganography-how-to-send-a-secret-message/

20. Using Facebook for Image Steganography. URL: https://www.researchgate.net/publication/277959373_Using_Facebook_for_Image_Steganography

21. Secretbook. URL: https://www.secretbook.com.br/

22. Castiglione A., Cattaneo G., De Santis A. A forensic analysis of images on online social networks. *Third International Conference on Intelligent Networking and Collaborative Systems*, 2011, P. 679-684.

23. Thoughts on pseudorandom number generators. URL: https://doi.org/10.1016/0377-0427(90)90346-2

24. A Comparative Study of Some Pseudorandom Number Generators. URL: http://dx.doi.org/10.1016/0010-4655(95)00015-8

25. Google APIs Explorer. URL: https://developers.google.com/apis-explorer?hl=en