ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І.МЕЧНИКОВА
(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем та технологій
(повна назва кафедри (предметної, циклової комісії)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»
(освітньо-кваліфікаційний рівень)

« Розробка системи посилення ступеню захисту інформації в комп'ютерних системах »

« Development of a protocol to Enhance the Degree of Information Security in Computer Systems »

Виконала: здобувачка заочної форми навчання
спеціальності 123 Комп'ютерна інженерія
(код, назва спеціальності)
Освітня програма ОП - Комп'ютерна інженерія
(назва)
Кішубаєва Ксенія Тімурівна
(прізвище, ім'я, по-батькові здобувача)
Керівник канд. ф-м. н., доц. Шугайло Ю.Б. _____
(науковий ступінь, вчене звання, прізвище,ініціали) (підпис)
Рецензент ст.викл. Шаріпова І.В.
(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
№__ від «___»_____ 2024 р.

Завідувач кафедри

_____ Юрій ГУНЧЕНКО
(підпис) (прізвище, ім'я)

Захищено на засіданні ЕК №_____
Протокол №__від «___»_____2024 р.
Оцінка_____/ /_____
(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

_____ Алла КОБОЗЄВА
(підпис) (прізвище, ім'я)

Одеса - 2024

АНОТАЦІЯ

Дипломна робота стосується теми “Розробка системи посилення ступеню захисту інформації в комп'ютерних системах”.

Мета роботи – вивчення проблематики секретних комунікацій і їх проведення методами правдоподібного заперечення, а також розробка програмного прототипу на базі стеганографії світлин і повного протоколу зі зберігання і приховування інформації в комп’ютерних системах, розрахована на користувача з мінімальним до середнього знаннями інформаційних технологій і інформацію рівня екстремальної чутливості. Також вивчення можливостей аналогічного методу в поєднанні з IoT, розробка повної системи для сценарію з датчиками.

Ключовим елементом роботи є розмірковування методів покращення і перспектив майбутнього, що були виявлені в результаті досліджень.

ABSTRACT

The diploma thesis addresses the topic "Development of a protocol to Enhance the Degree of Information Security in Computer Systems."

The goal of the work is to study the issues of covert communications and their implementation using plausible deniability methods, as well as to develop a software prototype based on image steganography and a complete protocol for storing and hiding information in computer systems. This system is designed for users with minimal to intermediate knowledge of information technology and aims to protect information of extreme sensitivity.

Additionally, the thesis explores the possibilities of combining this method with IoT, developing a comprehensive system for a scenario involving sensors.

A key element of the work is the consideration of improvement methods and prospects identified as a result of the research.

ЗМІСТ

ВСТУП

В часи розквіту електронних технологій і в умовах військового конфлікту як ніколи важливе усе, що пов’язане з кібербезпекою. Як на вищих інстанціях, так і серед звичайних людей, важливо розсудити про спосіб передачі і захисту екстремально чутливої інформації в умовах того, що нападник може бути більш авторитетною людиною, мати фізичний доступ до пристроїв суб’єкта та сам факт наявності чогось зашифрованого на пристрої може бути компрометуючим для суб’єкта.

В умовах сучасності ми частіше за все можемо спостерігати серед людей не-інформатичних спеціальностей дві тенденції: або вони надто легковажно ставляться до інформації, яка може виявитись для них компрометуючою, або навпаки, і намагаються завдати дій, які б їх захистили (наприклад, повністю видаляють переписки зі свого боку), але цим іноді можуть поставити себе в більш незручне положення. Хоча в другому випадку це може бути ефективним, але для того, щоб приховати екстремально чутливу інформацію, не привернувши до себе непотрібної уваги, його може бути недостатньо.

В подібних ситуаціях, коли інформація під питанням надто чутлива, щоб навіть сам факт її наявності був викритим, вступає в гру метод «правдоподібного заперечення», легальний концепт якого розглянутий в багатьох роботах з і полягає в тому, щоб відвертати наявність об’єкту під питанням до останнього, заздалегідь, звичайно, запевнивши її приховування і захист кількома рівнями, які спираються в тому числі і в великому ступені на людяний фактор.

Говорячи про людяний фактор, мається на увазі, що ми беремо за перевагу те, що «атакуючий» – теж людина і може недооцінити нас в багатьох інстанціях. В залежності від того, скільки рівнів оминання ми додамо, ми зможемо захиститись від більшої кількості потенційних атакуючих, якщо не усіх одразу. Треба також брати до уваги, що окрім того, щоб зробити свій пристрій, який має чутливу інформацію, максимально «не-підозрілим»,

потрібно також самому фізично поводитись не-підозріло. Подібний метод має застосовуватись досвідченими людьми з достатньо хорошою акторською грою і лише в ситуаціях, де інформація під питанням дійсно настільки чутлива, щоб бути схованою таким чином. Оскільки чим більше рівнів захисту, тим більше також незручностей для користувача, а також чим більш розсповсюджений метод – тим більше причин його підозрювати.

До того ж, використання цього методу має бути вдумливим також тому, що може поставити користувача в більшу небезпеку, ніж він був до того, якщо застосування ним цього методу все ж викриють. Тому що після цього суб’єкт навіть якщо викриє те, що з самого початку сховав, не зможе довести, що він не сховав щось більше.

Мета роботи полягає в тому, щоб розміркувати і сформулювати спосіб передачі або зберігання чутливої інформації від початку і до кінця, за допомоги методи правдоподібного заперечення і стеганографії в фотографіях і інш., базуючись на тому ж методі, також надавши прототип базової імплементації. Також в роботі планується розкрити потенціали для покращення і запропонувати інші способи застосування цього методу.

РОЗДІЛ ПЕРШИЙ: ТЕОРІЯ ЗА МЕТОДАМИ

1.1 Правдоподібне заперечення

Існує безліч можливих методів посилення ступеню захисту інформації в комп'ютерних системах і мережах. В залежності від структури системи, потенційних загроз і рівня чутливості інформації, необхідно розглядати різні можливі варіанти.

Метод правдоподібного заперечення – це більше, ніж набір інструкцій або конкретний алгоритм, якому потрібно слідувати, щоб захистити інформацію. Цей метод використовується не лише в інформатиці і має відношення до захисту інформації і себе в цілому, випливаючи з назви, прагнучи дати собі можливість правдоподібно заперечити своє відношення до справи під питанням, або існування цієї справи. Правдоподібне заперечення в інформатиці також представляє собою підхід в цілому, підхід, який вимагає креативності і який достатньо гнучкий, щоб до невпізнавості викривити дані, які необхідно захистити. Все залежить від кількості рівнів, які користувач бажає додати, щоб захистити свої дані. Чим більше рівнів – тим більше ймовірність, що їх не розкриють, але також більше незручностей для того, щоб пізніше повернутись до початкової інформації. [1]

Правдоподібне заперечення включає в себе сукупність дій і поведінки, спрямовану на посилення ефективності стеганографічного методу, застосованого до інформації, а також інші можливі дії, які спрямовані на те, щоб дозволити об'єкту заперечувати своє відношення до передачі або обладання інформацією під питанням. Протягом роботи ми будемо концентруватись на імплементації з стеганографією, але важливо визначити і пояснити, що для ефективності методу необхідно розуміти його історію, цілі і його "батька".

Далеко не в усіх випадках використання цього методу його можна назвати дійсно "правдоподібним". Як приклад використання методу в не-

інформатичному, а шпигунському контексті, ще з середньовіччя можна навести класичний трюк з письменністю лимонним соком, коли таємне послання на папері з не-підозрілим листом проявляється після того, як його підігрієш – і після прочитання лист одразу спалюють. В ті часи, коли цей метод був невідомим, той, хто міг перехопити листа, не здогадувався підігріти його, щоб побачити таємне послання – на це і покладалась ефективність такого способу. Так само легко можна було потім заперечувати факт обміну секретною інформацією. В наші ж часи це вже не можна назвати дійсно “правдоподібним”, оскільки набагато більше людей знають про цей трюк і викриття вкрай ймовірне. На протиставлення цьому пізніше стали використовувати невидимі чорнила, проявлення яких вимагало іншої формули, ніж просте підігрівання. Але і цей метод перестав бути актуальним, на протиставлення дешевшим і надійнішим.

Схожим принципом ми користуємося, застосовуючи “правдоподібне заперечення” в інформатиці. Перш за все, ми вивчаємо очікування нашого потенційного супротивника і шукаємо способи обманути його очікування, бажано кілька разів, настільки, щоб він не зміг довести, що ми з самого початку щось приховуємо і обманули його хоч раз. Це метод, який так чи інакше застосовували в усі часи розвитку інформатичних технологій і не лише, тому важко виокремити, коли конкретно він, разом з стеганографією (див. 1.2) набув більшої популярності. Очевидно одне: для захисту інформації його використовують, коли інформація під питанням є надзвичайно чутливою, тому цільова авдиторія цього методу – це шпигуни, журналісти, посадовці і злочинці.

Коли мова йде про обмін чутливою інформацією між двома людьми, для ефективності оного необхідно заздалегідь узгодити протокол між надсилачем і отримувачем (бажано – асиметричний) який забезпечить наступне:

- Автентифікацію надсилача і адресата [2]
- Перевірку цілісності даних
- Стійкість проти цілеспрямованих атак

Коли ми говоримо про автентифікацію, в ідеальному випадку пропонується робити її на трьох рівнях з боку отримувача:

- автентифікація пристрою – класичний метод, використованих в багатьох протоколах обміну інформації, де на пристрої буде наявний ключ ідентифікації, а також “white-list” знайомих пристроїв.
- автентифікація персони – ми розглядаємо випадок, коли ми не можемо гарантувати фізичну безпеку пристрою. В такому випадку зазвичай важко говорити про інформаційну безпеку, але тим не менш, в зазначених сценаріях це необхідний етап і має критичну важливість. Автентифікація особистості – це етапи протоколу обміну, відомі лише персоні, які вона має зберігати в голові і може відмовляти своє їх знання. Це не такі очевидні речі, як пароль від комп’ютера, а більш неочевидні дії, які можна списати на випадкові. Приклад буде наведено в розділі 3, при описі пропонованого pipeline.
- автентифікація намірів – необов’язковий етап, який не буде далі розглядатись протягом роботи. Він представляє собою останній шанс для надсилача відмовити в наданні доступу до секретних даних в випадку виявлення підозрілих дій, наприклад, повторного запиту до певної інформації через довгий час. Але цей етап буде вимагати доступу до мережі після передачі стеганоконтейнера, що може не бути оптимальним рішенням. До того ж, в певній мірі захист, заміщуючий це можна забезпечити зашифровуючи всередину контейнера також умовний TTL, який буде слугувати строком придатності для секретних даних і буде “спалювати” сам себе через певний час після прочитання. Цей етап необхідний для того, щоб забезпечити безпеку даних в випадку, якщо одна з персон в обміні інформації зрадить домовленності.

Потенційну ефективність демонструє наступна схема з прикладом (рис.1), де в базі даних умовної дослідницької станції зберігається інформація під питанням. Інформація має сама по собі представляти деяку цінність, щоб

змусити супротивника повірити в те, що ми захищали саме її, коли насправді, всередині неї прихована інформація вищої цінності.

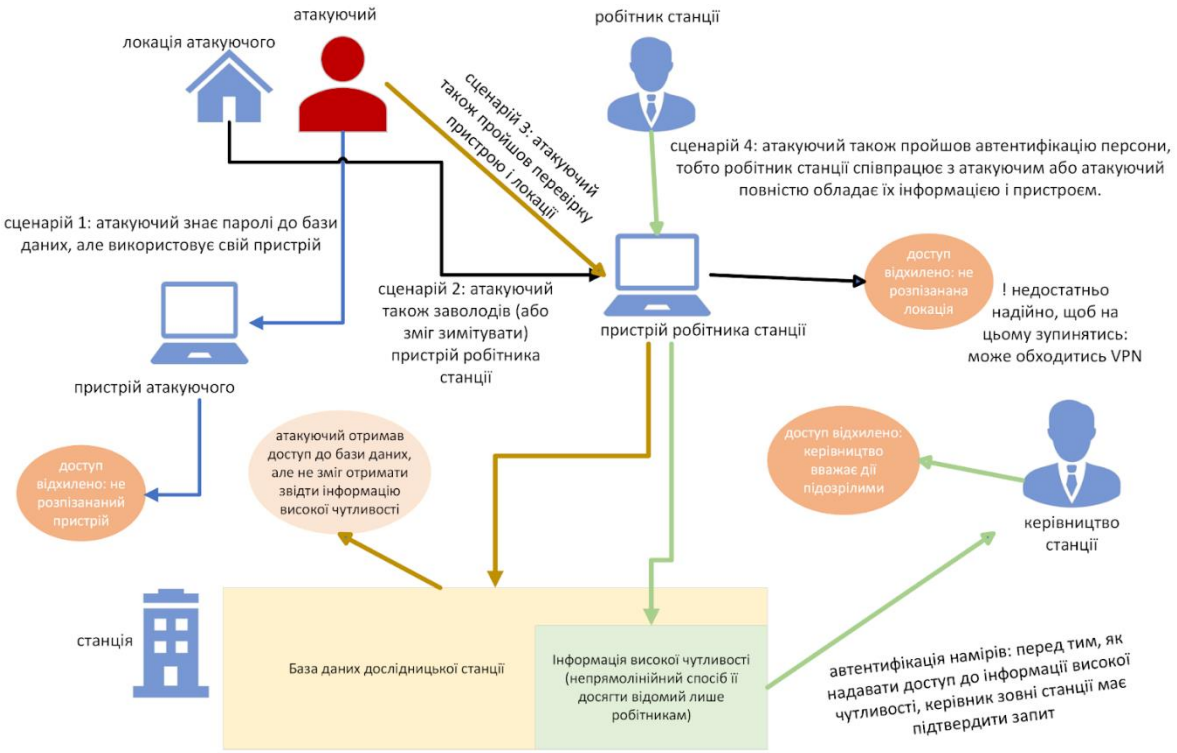

Рис. 1.1 – Захист новими шарами автентифікації в різних сценаріях

Підсумовуючи вищесказане, сутність методу полягає в тому, щоб зробити вигляд, що наші секретні дії – це випадковість, і поводити себе непідозріло, тим самим захищаючи себе і секретні дані навіть в випадку, коли атакуючий має фізичний доступ до пристрою суб'єкта або погрожує життю суб'єкта задля отримання інформації. Якщо з'являється підозра, то сенс методу знищений, незалежно від того, чи дістався атакуючий інформації.

Випадки, коли метод застосовуються, мають бути дійсно екстремальними і метод має бути використований з особливою обережністю, натренованими персонами, інакше він може поставити суб'єкта під більшу загрозу, ніж той був до того. Звичайно, можна посперечатись, що метод можна використовувати для інтересу, з не надто секретними даними, одначе при виявленні того, що цей метод використовується один раз, ви не зможете довести, що не використали його знову і відмовляти те, що зробили щось ще,

навіть якщо ви вже розкрили всю секретну інформацію, яка в вас була. Це умовна гра на довіру, і як тільки вона підірвана, ситуація стає лише гірше.

1.2 Стеганографічний метод

Стеганографія – ключовий інструмент для правдоподібного заперечення в інформатичній безпеці, коли мова йде про ситуації, де ми не можемо гарантувати приватність каналів передачі, анонімність і збереження власності фізичного пристрою з інформацією під питанням, або ж ми не можемо використати достатньо стійкий кріптографічний метод (в деяких країнах існують закони, що забороняють використання стійких кріптографічних методів без ліцензії або в принципі, або дозволяють державним органам, поліції або інш. вимагати розшифрування у суб'єкта [3]), або хочемо посилити його ефективність, приховавши сам факт його використання. Сам термін складається з двох грецьких слів – *steganós* (στεγανός), що означає «прихований», і *-graphia* (γραφή), що означає «письмовість», що повністю передає його суть.

В цілому же стеганографія має декілька застосувань: електронні відбитки, вотермарки, виявлення модифікацій файлу, обмін секретною інформацією, тощо. Оскільки робота центрується навколо захисту інформації за допомоги правдоподібного заперечення, то опис, наданий далі, посилаючись на стеганографічний метод, має на увазі саме його застосування в контексті передання секретної інформації. Деякі джерела часто використовують поняття стеганографії лише в контексті секретних комунікацій і виділяють вотермарки в окрему категорію, хоч технічно їх принцип і схожий, [4] але вотермарки в будь-якому випадку не є ціллю дослідження.

Принцип стеганографічного методу полягає в тому, щоб заповнити стеганоконтейнер інформацією, не піддаючи початковий зміст контейнеру видимій людському оку зміні. Саме тому для найбільш класичного випадка як

контейнери використовують медіа аналогового характеру, e.g. світлини, аудіо, відео. В цьому випадку незначні зміни контейнеру списуються на похибку . В сучасності, однак, також існують методи, які дозволяють програмно передати секретну інформацію, не генеруючи змін контейнеру, такі як, наприклад: генерування валідного тексту, де слова будуть індикаторами, або шифрування інформації за допомогою шахової дошки [5]. В роботі надалі буде використовуватись класичний метод, тому мова нижче буде йти про нього.

Принцип роботи класичного методу доволі уніфікований, і включає в себе дві сторони, між якими стається введення й виведення інформації, самі дані, які будуть приховані, та стеганоконтейнер, який буде в ході передання між цима двома інстанціями експозований потенційному атакуючому (рис. 1.2). Варто зазначити, що “надсилач” і “отримувач” – це умовні позначення і можуть фізично бути однією і тією самою людиною.

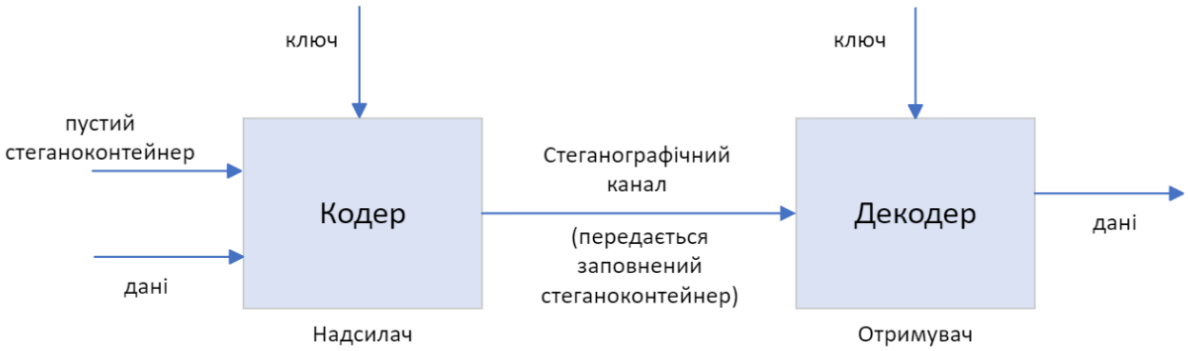

Рис. 1.2 – Модель стеганосистеми

Стеганографічним каналом виступає будь-який обраний метод передачі інформації, від соціальної мережі до фізичного конверту з листом. Проведучи паралелі зі старими стеганографічними методами, таким як невидимі чорнила, пустим контейнером буде виступати звичайний, не-підозрілий лист. “Кодером” буде написання певними невидимими чорнилами справжніх даних, а ключем декодування буде формула чи протокол проявки. Стеганографічним каналом

в такому сценарії, відповідно, буде метод передання листа, пошта чи гонець, а декодування – це проявка.

Але навіть в такому, симпліфікованому сценарії, легко бачити, що до обрання стеганографічного каналу необхідно підходити з уважністю. Оскільки ті, хто контролюють канал, можуть, якщо не виявити початкове повідомлення, то принаймні пошкодити контейнер так, що воно не зможе бути декодоване в результаті.

Оскільки класичний метод продукує зміни початкового контейнеру, то критично важливим в протоколі передачі є те, щоб початковий контейнер у вигляді до додання туди інформації не потрапив в руки атакуючого.

В роботі ми розглядатимемо класичний варіант розвитку подій багатьом відомого методу з відносно очевидними пропозиціями адаптації алгоритму. Хоча основна мета цієї роботи полягає в тому, щоб розміркувати можливості використання методу, необхідно зазначити, що найчастіше на практиці він використовується з зловмисними намірами. Тому подальші розмірковування мають також слугувати застереженням.

Коротко також розглянемо, як виявити сліди застосування стеганографії?

Найлегше це, звичайно, зробити, якщо на руках мається початковий контейнер, щоб порівняти його з заповненим. Ще легше, якщо, погано розраховуваши пропорції контейнеру до прихованої інформації, користувач деформував контейнер настільки, що зміни стають видимими людському оку. Але обидва ці випадки на практиці виникають вкрай рідко.

Для того, щоб виявити сліди застосування стеганографії, можна спробувати виявити в ньому повторюючиїся патерни за допомогою статистичного аналізу, методи якого включають в себе, але не обмежені хістоаналізом (де актуально), трансформативним, аналізом JPEG-компресії і так далі. За іншим методом, шукаються патерни чи характерні маркери відомих стеганографічних систем. [6] Також пошук полегшується, якщо ми маємо уявлення про те, що саме необхідно шукати – як, наприклад, з вотермарками.

Також, в залежності від використаного каналу, існують більш доступні для людини індикатори, які можуть містити сліди застосування стеганографії. В випадку, наприклад, коли каналом передачі є картинка на вебсайті, її несправедливо величезний розмір може вже вказувати на наявність всередині прихованої інформації, оскільки зазвичай медіа на веб сторінках намагаються оптимізувати (хоча це і може, звичайно, бути й всього лише індикатором поганих практик розробки).

Загалом, існують застосунки, такі як Stego Suite, які здатні виконувати стеганографічний аналіз для виявлення слідів застосування стеганографії, але з часом з'являється більше нових методов і алгоритмів, і обидві сторони – приховування і виявлення – постійно розвиваються, тому неможливо цілковито покладатись на застосунки і існуючі програми, хоч їх приклад і, звичайно, стане в нагоді, а в випадку виявлення – додаткова перевірка ніколи не буває зайвою, щоб почувати себе спокійніше.

Якщо, наприклад, певний співробітник компанії поводиться себе підозріло, а компанія має на це ресурси, то не грішно перевірити всі файли на його пристрої на наявність слідів стеганографії відомими методами аналізу. Але не-виявлення слідів ще не буде стовідсотковою гарантією, що ця людина не користувалась методом.

Підсумовуючи, стосовно термінології варто зазначити, що стеганографія не обов'язково використовується для правдоподібного заперечення, а правдоподібне заперечення не обов'язково використовує стеганографію. Але в контексті уявної ситуації, для якої буде використовуватись розроблений в ході роботи програмний прототип, ці два поняття нероздільні. (див. 3 розділ)

Також, цікаво розглянути один з історичних способів стеганографії "маска". Відомим прикладом його використання є лист від Генрі Клінтона до Джона Бургойне 1777 року [7]. Його принцип складається в тому, що лист з самого початку складається таким чином, що виглядає як звичайний, не викликаючий підозру лист. Насправді же, якщо накласти на нього лист-

аплікацію певної форми, яка буде показувати лише певні слова (або літери), то буде виявлене справжнє, секретне повідомлення. Цей метод раніше потребував багато зусиль задля справжньої ефективності.

РОЗДІЛ ДРУГИЙ: СТЕГАНОГРАФІЯ КАРТИНКИ

Як ми визначили в першому розділі (див. 1.2) світлини є одним з ідеальних варіантів стеганоконтейнера в контексті інформаційних технологій. Одна з однозначних переваг є тим, що вони з самого початку зберігають багато інформації і мають тенденцію багато важити, особливо коли мова йде про фотографії високої якості.

Відомим випадком імплементації є використання методу терористичними організаціями для секретних комунікацій, фіксоване, наприклад, так рано як в 2002 і 2010 роках [8]. Заповнені стеганоконтейнери розміщувались в інтернеті на вебсайтах по типу блогів, наче звичайні баннери або ілюстрації, насправді маючи в собі критичну інформацію атак і операцій.

Повертаючись до термінології, стеганографічним каналом в цьому випадку буде виступати веб сторінка, а контролювати її буде серверний провайдер. Хоч і існує багато способів виявлення, існує так само багато способів застосування. Це також об’єктивно нереалістично намагатись моніторувати усі веб сторінки стеганоаналізом. Деякі провайдери, щоб захистити себе від можливих недоброчесних клієнтів, можуть спробувати, наприклад, застосувати додання шуму до усіх завантажуваних медіа – тим самим знищуючи можливе повідомлення всередині. Натомість, і застосовуючий може поквапитись і застосувати методи детекції і відновлення повідомлення, які використовуються в протоколах комунікацій, і можуть бути влаштовані напряму в світлину – про це пізніше у розділі.

Просунута стеганографія, виконана над картинкою, за принципом схожа на ту саму маску-аплікацію, яку історично накладали на листи, де маскою слугує індикатор того, які пікселі або кольори були затронуті кодувальником, або селектор додатково з порядком читання, в випадку, якщо порядок “слів” не співпадає з природнім способом читання мовою листа. Але для повного розуміння, розглянемо, як працює стеганографія картинок по порядку.

Як відомо, світлина зазвичай складається з пікселів, а пікселі – з трьох кольорів. Навіть якщо це векторна картинка, її можна конвертувати в такий формат. В деяких форматах також зберігається прозорість (альфа-канал) – це також може дати потенційний простір для запису інформації.

2.1. Алгоритм LSB (Least Significant Bit)

LSB - один з найпростіших і найпоширеніших методів цифрової стеганографії. Він полягає в заміні найменш значущих бітів пікселів зображення бітами прихованих даних.

Кожен колір є числом від 0 до 255, і в реаліях світу середньостатистична людина не факт, що побачить різницю між 235 і 255. А якщо різниця між кольорами складає всього одиницю значення, то не побачить жодна.

Послідовність застосування алгоритму LSB:

1. Вибір носія: Обираємо цифрове зображення як носій (наприклад, BMP або PNG). На цьому етапі варто зауважити, що зображення повинно бути достатньо унікальним (особисті фото наприклад) щоб знизити вірогідність наявності у атакуючого копії оригінального (незміненого) зображення.
2. Підготовка повідомлення: Перетворюємо приховане повідомлення на двійковий формат. Це можна здійснити, наприклад, переведенням Unicode коду кожного символу у двійкову систему числення. Але на цьому етапі можна задіяти також будь-яке інше (бажано бієктивне) відображення множини символів на множину двійкових послідовностей певної довжини (7-8 біт)
3. Заміна бітів: Замінюємо найменш значущі біти пікселів зображення на біти прихованого повідомлення. На цьому етапі слід приділити увагу тому, які саме біти будуть змінені. Це може бути, наприклад якась конкретна послідовність бітів (ключ), яку знають складач та отримувач повідомлення. Але за наявності багатьох повідомлень з застосуванням

одного й того ж ключа статистично можуть бути виявлені аномалії та скомпрометовано сам ключ. Тому зазвичай у більшості застосувань стеганоключем виступає алгоритм обрання пікселю, біту або їх комбінація.

4. Збереження стего-носія: Зберігаємо модифіковане зображення як стего-зображення. При цьому оригінальне зображення варто знищити для того щоб знизити можливість порівняння з ним

Такий метод не генерує значних змін контейнеру і, окрім того, зовсім не змінює його початковий розмір. Однак з-за того, що зміни дуже невеликі, він стає більш чутливим до атак і наймаліша компресія світлини може повністю знищити повідомлення всередині. Тому цей метод може використовуватись лише в тому випадку, коли наш стеганографічний канал передачі, хоч і не стовідсотково захищений від атакуючих, принаймні стовідсотково гарантує незмінність даних.

Варто зазначити, що в захисті заповненого стеганоконтейнеру грає роль не лише обраний ключ, а й обрані дані.

Приклади ідеальних даних, які можна передати: id, пароль, ключ, координати, номер телефону. В випадку з таким типом передаваємих даних навіть у випадку їх розшифрування, це не дасть багато інформації атакуючому і це можна доволі легко списати на набір рандомних символів, особливо якщо шлях їх знаходження був достатньо складним, щоб його функцію важко було категоризувати.

Складаючи саме повідомлення, варто уникати легко-вгадуємих слів в встраюванні, таких як ФІО, назви населених пунктів, або, ще гірше, розповзюджені “індикативні” слова, як “password, coordinates,”. В випадку з приховуванням, наприклад, номеру телефону, також можна забезпечити кращий захист, відмовившись від коду країни, оскільки це також легко-вгадуєма частина, яка може дати супротивнику підказку про використаний нами алгоритм вибору біта.

Підсумовуючи, з вищесказаного можемо уявити собі, в якому сценарії буде застосовуватись подібний метод. Наше повідомлення має бути скоріше коротке, ніж довге, ми маємо контролювати або бути впевненими в тому, що стеганографічний канал не застосовує методів протидії стеганографії, а ключовим елементом задля ефективності буде виступати функція обрання біту. [9]

2.2 Застосування генетичного алгоритму в алгоритмі LSB

Генетичні алгоритми (ГА) можуть бути ефективно використані в стеганографії для оптимізації процесу приховування інформації в носії. Тобто, за допомогою генетичного алгоритму для певного зображення обирається така послідовність бітів, яка забезпечить найкращу “прихованість” змінення в них даних від статистичного аналізу та інших методів виявлення стеганографії.

Цей процес складається з етапів, описаних нижче.

2.2.1. Початкова популяція

Створюється початкова популяція можливих розташувань бітів повідомлення в носії. Кожен індивід в популяції буде представляти можливий спосіб приховування повідомлення і мати розмір що дорівнює довжині повідомлення. Зазвичай ця популяція генерується випадковим чином (random choice).

2.2.2. Функція пристосованості

Визначається функція пристосованості, яка оцінює якість кожного індивіда. Це може бути поєднання критеріїв, таких як:

- Непомітність (наскільки зміни в носії малопомітні для ока або вуха).

- Стійкість (наскільки стійка інформація до атак, наприклад, стиснення або фільтрації).
- Ємність (скільки інформації може бути приховано).

Наприклад такою функцією може служити лінійна комбінація $\alpha * I + \beta * x^2$, де α, β - деякі ваги, що відповідають важливості кожного з 2 критеріїв, $\alpha + \beta = 1$.

$$I = \sum_i |stego_i - cover_i|$$

- являє собою суму абсолютних різниць в бітах зміненого ($stego_i$) та покривного ($cover_i$) зображень та відображає фактор непомітності змін в зображенні (Invisibility).

$$x^2 = \sum_{i=0}^{255} hcover(i)(hstego(i) - hcover(i))^2$$

- являє собою характеристику Хі квадрат, що є показником відмінності гістограм покривного ($hcover(i)$) та стеганографічного ($hstego(i)$) зображень відповідно, де i — значення інтенсивності пікселів (від 0 до 255 для 8-бітного зображення).

Цей приклад функції буде оцінювати пристосованість кожного індивіда до атак що статистично аналізують зміни в зображеннях на основі обчислення значущості змін та аналізу гістограм рівнів інтенсивності пікселів.

2.2.3. Генетичні оператори

Використовуються стандартні генетичні оператори для еволюції популяції:

Селекція: вибір найкращих індивідів для відтворення.

Вся популяція сортується за значеннями функції пристосування і відбирається певний процент найкращих індивідів

Схрещування: для доповнення до первинного розміру популяції генеруються нові індивіди шляхом комбінування частин двох батьківських індивідів (обраних навмання з кращих індивідів з попередньої популяції).

Мутація: робляться випадкові зміни в індивідах для введення нових варіантів. (випадковим чином обираються деяка невелика кількість індивидів і бітів в них та змінюються на протилежне значення)

2.2.4. Еволюційний процес

Повторюється процес селекції, схрещування та мутації для певної кількості поколінь, поки не буде досягнуто бажаного рівня якості або не мине певна кількість поколінь.

2.2.5. Остаточне рішення

Обирається найкращий індивід з фінальної популяції як остаточне рішення для приховування повідомлення в носії. [10]

Варто зазначити, що після застосування генетичного алгоритму застосований ключ (набір бітів в зображенні, який був змінений) буде для кожного випадку генерації стеганографічного зображення свій і постане проблема його передання отримувачеві кожного разу. Це суттєво скорочує сферу застосування алгоритму.

2.3. Застосування алгоритму кластеризації в алгоритмі LSB

Цей метод використовує алгоритм кластеризації для вибору пікселів зображення, в які буде додаватися повідомлення, після чого додається випадковий шум в інші пікселі для підвищення безпеки.

Кроки алгоритму:

Вибір носія: Вибирається зображення для приховування даних.

Кластеризація: Використовується алгоритм k-середніх для кластеризації пікселів зображення. Як це відбувається:

- Обирається кількість k кластерів, на які буде розбите зображення
- Обирається метрика, за якою буде відбуватися кластеризація. Для чорно-білих світлин за відстань між двома пікселями може бути взяте просто абсолютне значення різниці в їхній інтенсивності. Для кольорових це може бути сума таких абсолютних різниць за всіма каналами, або евклідова відстань у 4-вимірному просторі
- Довільним образом пікселі розбиваються на k кластерів. В кожному з кластерів обчислюється “центр”, тобто середня за значеннями інтенсивності каналів серед пікселів даного кластеру.
- За допомогою обраної метрики обчислюється відстань від кожного пікселю до всіх k “центрів” і за найменшим значенням відбувається нове розподілення за кластерами
- В новому розподіленні на кластери відбувається переобчислення “центрів” і процес перерозподілу на кластери за найменшою відстанню до нових “центрів” відбувається знов
- Процес повторюється поки наступне розподілення на кластери не стане дорівнювати попередньому.

Це не завжди може буде досягнуто, тому слід цей процес припинити після певної кількості ітерацій, або при виконанні певних умов (наприклад, черговий перерозподіл на кластери міняє менше певного % маркувань пікселів). Сходимість, а також єдиність отриманого розподілу можуть суттєво залежати від першого вибору “центрів”.

Вбудовування повідомлення: Повідомлення вбудовується в певний кластер, або декілька кластерів. До незмінених пікселів додається “шум”. Ключем будуть служити ті кластери, в яких вбудоване повідомлення. Оскільки генерування кластерів може відбуватися по-різному для одного і того ж носія

та числа k, передавати доведеться увесь ключ, тобто розташування всіх змінених пікселів. [11]

РОЗДІЛ ТРЕТІЙ: ІМПЛЕМЕНТАЦІЇ І ІДЕЇ ПОКРАЩЕННЯ

3.1 Прототип, розроблений для роботи

Для ознайомлення з імплементацією алгоритму LSB і можливості його протестувати, було розроблено програмний прототип аплікації кодування повідомлення в картинку на операційну систему Windows, згідно з методом, описаному в другому розділі.

Уявімо наступний сценарій: нам необхідно зберігати інформацію критичної важливості (наприклад, власні паролі) – або обмінюватись нею в рамках контрольованого нами, але не цілковито захищеного стеганографічного каналу (наприклад, фізичне передання USB-флеш-накопичувача з “галереєю” заповнених картинок з рук в руки). В будь-який момент в нас можуть запросити доступ до канала передачі або до нашого пристрою, з якого ми проводили кодування чи декодування. Наша мета – замаскувати комунікацію на всіх етапах передачі і залишати факт проведення комунікацій невикритим і не запідозрюваним. Фінальний протокол має включати в себе послідовність дій, модифікацію коду наданої аплікації і патерни поведінки.

Розглянемо структуру аплікації. Вона розділена на два основні модулі, кодування і графічного інтерфейсу. Графічний інтерфейс в даному випадку є мінімалістичним і має лише виконувати функцію для ознайомлення. Запустивши аплікацію командою “python gui.py”, де python - системна змінна, що веде до шляху коду версії пітона на пристрої використовуючого. Ми бачимо мінімалістичне і інтуїтивне головне меню, єдині два вікна якого дозволяють нам закодувати чи декодувати світлину. (рис. 3.1)

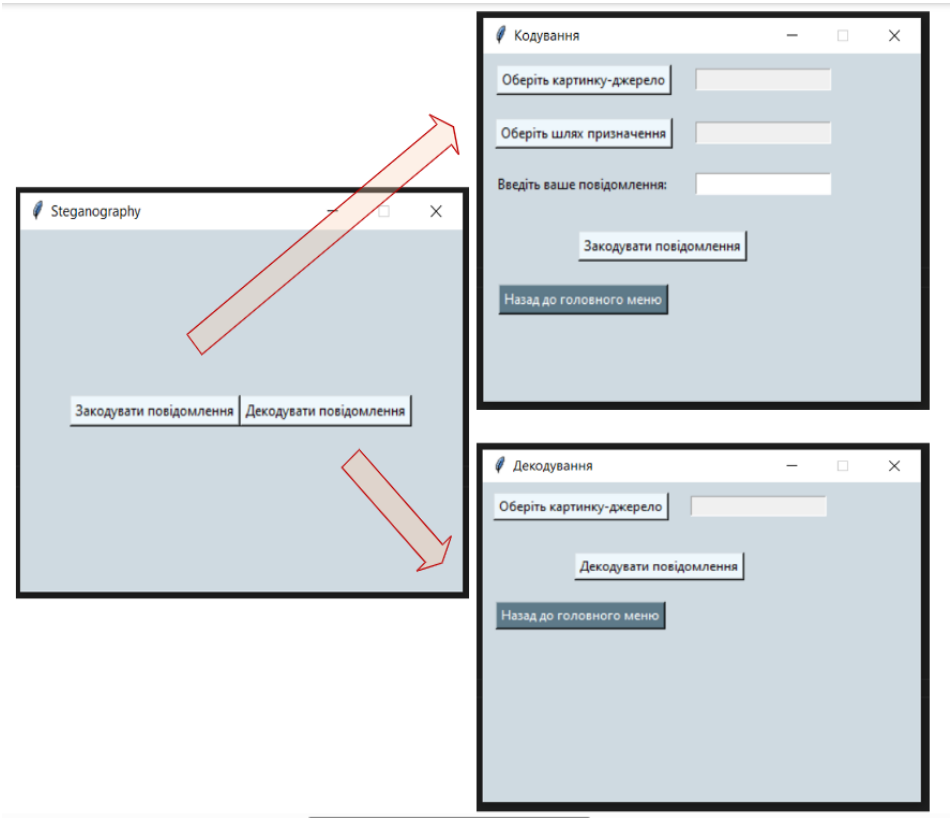

Рисунок 3.1 – інтерфейс прототипу 1

Перед використанням застосунку, користувач має обрати, як зазначено раніше, унікальні стеганоконтейнери, які будуть використані для передання (зберігання) інформації. Хоча ми будемо працювати таким чином, щоб зміни здавались випадковими і в крайньому випадку звинувачувати в них неточність аналогового сигналу, або погрішність, створену компресією файлу, найкращим варіантом буде, щоб файл-контейнер секретної інформації після шифрування став єдиним екземпляром цього файлу. Як файл-контейнер ідеально використовувати звичайне, непідозріле фото, зроблене власними руками – наприклад, свіжа фотографія домашнього улюбленця, їжі або селфі. Після шифрування в це фото повідомлення, оригінал необхідно видалити.

В програмі обираємо пустий стеганоконтейнер як джерело, і задаємо повний шлях файлу призначення.

Далі, в програмі же задаємо повідомлення, яке збираємося приховувати – для приклада, просте текстове повідомлення (про інші можливості див.

Розділ 3.2). Як ми визначили в першому розділі, чим коротші дані, тим легше і ефективніше їх можна приховати, а за другим визначили, які саме найкращі для приховування. Також увага до того, що повідомлення, яке вводиться, має бути вже зашифрованим обраним користувачем криптографічним методом – сам застосунок не імплементує криптографії. Кодування символів відбувається за UTF-8, що підтримує кирилицю.

Алгоритм кодування використовує бібліотеку для python PIL (Pillow), щоб працювати з пікселями завантаженого повідомлення. Також в біти, які не затронуті кодуванням повідомлення, додається шум – для цього використовується бібліотека random.

В індикованому місці коду задана функція, яка використовується для кодування повідомлення в картинку і слугує селектором бітів повідомлення. В прототипі, для прикладу, це steps(x) = 2x, stepb(x) = 1, що в коді виглядає так:

```
def writable_pixel(i: int) -> bool:
    return (i // 3) % 2 == 0
```

Функція повертає булеве значення, чи поточний піксель стосується повідомлення. Вираз після ‘return’ виражає собою функцію, в нашому випадку, обрання набору з трьох пікселів через один пропуск.

Користувач вільний і навіть має змінити цю функцію на складнішу, яка буде відома лише користувачу і отримувачу (якщо отримувач відрізняється від користувача). В ідеальному випадку варто змінювати функції від повідомлення до повідомлення, щоб атакуючому було важко виявити кореляцію за умови, що він має декілька наших контейнерів і підозрює наявність в них повідомлень. При побудові цієї функції також можно використовувати алгоритми, описані в п 2.2 та 2.3 або інші алгоритми вибору бітів.

В випадку спроби декодування контейнеру, в якому немає повідомлення, алгоритм не видає помилок, а видає той набір випадкових символів, який спробував

Якщо користувач відрізняється від отримувача, то першим етапом протоколу можна вважати обмін ключами іншим методом (або стеганографічним, за допомоги старого відомого ключа).

Робота алгоритма прототипу за замовчуванням досконало представлена на рисунку 3.2 на прикладі ASCII.

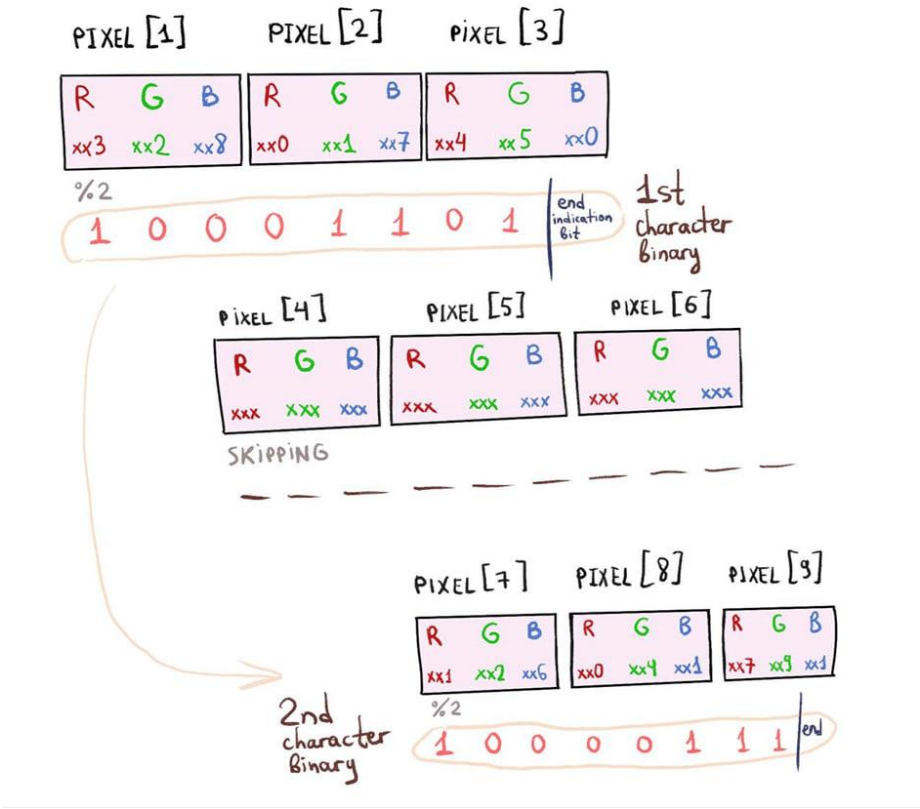

Рис. 3.2 – ілюстрація алгоритму дефолтної функції, вбудованої в прототип 1

Але залишається проблема того, що якщо програма для кодування буде в нас викрита, то це одразу підвищить підозру і може спровокувати атакуючого більш обережно і досконало шукати в нас інформацію, що може призвести до викриття комунікації. Як було зазначено в розглядаємому сценарії, атакуючий може в будь-який момент отримати доступ до всіх наших пристроїв. Повний протокол має не лише працювати з стеганографічним каналом, а й з не гарантовно захищеними кінцями передачі.

Розглянемо можливі легко імплементуємі модифікації протоколу.

По-перше, щоб дистанціювати компрометуючу програму від нашого пристрою, переносимо її на USB-флеш-накопичувач, за її природою вона портабельна. Однак цього, звичайно, все ще недостатньо.

По-друге, важливим кроку протоколу буде розділення модулів графічного інтерфейсу і кодування. Графічний інтерфейс для атакуючого не представляє ніякої суттєвої цінності, і за легких модифікацій може бути схожим на будь-яку програму криптографічного захисту.

Повертаємось до того, що програма призначена бути використаною на Windows, оскільки більшість не-підозрілих користувачів мають саме її як операційну систему. Це грає нам на руку.

Далі, все ще як частина протоколу, пропонується перенести зміст файла з алогритмом, тобто всі його функції, в будь-який файл бібліотек справжнього пітона на пристрої користувача – в ідеальному випадку також перейменовуючи функції кодування. Необхідно змінити включення файлів графічного інтерфейсу на включення бібліотеки, в яку був скопійований код. Ідеальним також буде кожен раз при використанні стирати лінію коду з підключенням, щоб не давати атакуючому індикацій того, де спробувати шукати алгоритм, а в випадках питань відмовляти, що це “недописана програма в рамках тренувань”.

Таким чином, маємо графічний інтерфейс на USB-флеш-накопичувачі, використання якого залежить від пристрою, який має в собі додатковий невеликий ключ знання того, яку бібліотеку необхідно підімкнути, щоб все запрацювало, і викриття якого не дасть атакуючому ніякої інформації. А наш алгоритм прихований десь глибоко в коді пітону, і навіть за його викриття, для декодування функція обрання біту в ньому має бути модифікована таким чином, який знає лише користувач (і отримувач, якщо вони різняться) – і немає жодних причин вірити, що ці функції взагалі були ним використані, якщо не порівнювати зі справжнім кодом тої ж версії пітона – тому можна також

поквапитись про те, щоб встановити стару або менш відому версію, аніж, наприклад, Python 3.9.

Звичайно, в такому випадку все ще залишається шанс, що комунікація може бути викритою, тому що це не стовідсотково стійка система. Але ймовірність того, що атакуючий, що не має до нас жодних підозр, одночасно вкраде і застосує з нашим пристроєм наш USB-флеш-накопичувач, вирішить просканувати на зміни доволі типові бібліотеки, що мають багато комп'ютерів, і після цього, виявивши функції для стеганографії, вирішить спробувати застосувати їх на світлинах і якимось чином вгадати ключ, а потім розгаданий набір виглядаючий випадковими символів вирішить спробувати розшифрувати відомими криптографічними методами, – вкрай мала.

Отже, навіть такий мінімізований протокол може послугувати великим кроком в захисті критичних даних.

3.2 Ідеї для покращення і перспективи

Запропонований застосунок, звичайно, не є повноцінною реалізацією потенціала методу, або навіть його частиною в класичному розумінні. Існує безліч можливостей модифікувати його таким чином, щоб він ставав ефективніше. Розглянемо кілька цікавих ідей, що вар'юються від простих в імплементації до складних.

Першою ідеєю, простішою, є модифікація генерування шуму і перетворення ключів (інформації про функцію). Як відомо, генератор випадкових чисел, влаштований в пітон, використовує псевдовипадковий алгоритм. Тому, оскільки ми заповнюємо контейнер шумом в усіх бітах, окрім інформаційних, теоретично атакуючий міг би виявити кореляцію від зворотнього, виявивши справжні біти з повідомленням. Також використання конкретної, навіть складної функції, все одно представляє собою патерн, які програми статистичного аналізу теоретично можуть помітити. Отож, ідеєю буде замість функції використовувати ключ довжини повідомлення, який буде

генеруватись з більш стійко випадкових чисел, вигляду “17239834129”, де кожне число (чи два) буде індикувати крок між бітами або символами. Цей ключ має бути переданий іншим (або зазначеним в прикладі методом), але своїм форматом являє собою ідеальний набір даних, які можна сховати.

Наступною ідеєю буде, звичайно, перенесення алгоритму на компільовану мову. Якщо атакуючий ще може подивитись початковий код бібліотек пітону і підвищити рівень підозри після виявлення стеганографічного алгоритму, то якщо функції нашого алгоритму буду лежати в .dll серед бібліотек, наприклад, Qt, або, ще краще, якоїсь гри на нашому пристрої, то він з меншою ймовірністю буде щось підозрювати.

З боку графічного інтерфейсу також можна зробити модифікації, які суттєво підвищать ефективність. Якщо пояснювати присутність на флеш-накопичувачі “недописаної програми” може накликати непотрібні підозри, то до, наприклад, “Libre Office Portable” не має виникнути жодних питань. Можна модифікувати відомі open-source застосунки (в ідеальному випадку, якщо вони самі написані на компільованих мовах) таким чином, щоб вони при запуску і при поверхневому аналізі структури, виглядали і працювали, як звичайні. Але, лише користувач знає, що, наприклад, якщо натиснути певну комбінацію клавішу, обравши жирний шрифт, то відкриється графічний інтерфейс застосунку кодування, який був надійно схований всередині.

Останньою ідеєю складних модифікацій буде перенести програму за тим же принципом на мобільний пристрій. Оскільки найкращим контейнером є фото, що існує в одному екземплярі, то ідеальним не привертаючим уваги графічним інтерфейсом, що запускає кодувальник, буде сам застосунок “камери”. Уявімо наступну систему: спершу, в нотатках на телефоні звичайними словами записане повідомлення. В іншому заданому файлі нотаток, записана, здавалося б, звичайна інформація, але насправді, довжина кожного слова в ній – це ключ того, який крок використовується в обранні біту. Потім, користувач робить фотографію за допомоги модифікованого застосунку камери, що автоматично підвантажує повідомлення з нотаток,

інкрустуючи в зроблену фотографію, і за можливості також стираючи початкове повідомлення з нотаток. Таким чином зроблена фотографія автоматично стає єдиним екземпляром і готова для проведення секретних комунікацій в найкоротші строки.

В випадку застосування останньої ідеї, можна також поквапитись про моментальне вивантаження контейнеру з повідомленням в cloud, що може дати змогу провести комунікацію в небезпечних для життя ситуаціях, при цьому роблячи дії, які мають виглядати невинно, такі як письмо у власних нотатках і фотографування себе.

Загалом, стеганографія має неймовірні перспективи. Сучасні досягнення в розвитку моделей штучного інтелекту по сутності можуть дозволити нам автоматизувати шифрування за допомоги бесшумного методу стеганографії (тобто, методу, що не залишає слідів на контейнері) .

Але й в більш вузькому просторі стеганографії картинок мається великий потенціал потенційних досліджень. Вище були розглянуті варіанти для покращення захисту методом змінення протоколу. Але мається також великий потенціал для зберігання величніших обсягів інформації.

Доволі легко уявити собі модифікацію алгоритму таким чином, щоб він послідовно записував дані у серію фотографій – в такому випадку, ми можемо розраховувати на набагато більший простір для інформації. Так, невинний жорский диск з сімейними фотографіями може стати повноцінною секретною базою даних. Яку саме послідовність необхідно обрати для дешифрування може стати додатковим ключем захисту.

З цікавого також, можна розглянути варіант нелінійного запису даних – до генерації кроків необхідно буде підійти з особливою обережністю, обираючи функцію, яка точно не призведе до обрання одного і того самого біту двічі. Зате, в такому випадку ми отримуємо новий рівень захисту за рахунок того, що можемо розділити передання світлин в кілька різних етапів, а без повного набору неможливо буде відновити початкове повідомлення. Як

варіант також, знову ж таки, обирати строго зростаючу або убиваючу функцію, але застосовувати результати почергово до усіх світлин набору (рис. 3.3).

Рисунок 3.3 – Приховування повідомлення в кількох світлинах

Повертаючись до тем розвитку штучного інтелекту, окрім того, з просуванням моделей, їх можна буде використовувати для пошуку патернів і згодом подібні алгоритми і конкретні функції може бути легше виявляти. Але суть з самого початку полягала в тому, щоб додати так багато кроків, щоб навіть якщо в решті решт повідомлення змогли відновити, шлях до нього здавався таким випадковим і складним, що здавалося, що це просто випадковість і можна було продовжувати відмовляти своє до нього відношення. Оскільки в теорії, таким чином з остаточних бітів світлин можна скласти абсолютно будь-яке повідомлення.

Заплутаність шляху до інформації, хоч і ускладнює нам життя, але додає ефективності, згідно з параметрами методу правдоподібного заперечення.

3.3 Стеганографія та IoT

В процесі роботи і розмірковувань також виникла ідея використання LSB в потоці даних з IoT. Постійна і росповсюджена проблема збору інформації з датчиків для аналізу часових рядів, потенційно прогнозування і

детекції аномалій, дозволяє проводити обміни даних в великих кількостях, не викликаючи підозр.

Було розроблено складові сценарію, які передбачені за звичайного функціоналу системи, адаптовані запропонованими позиціями, що дозволять обмінюватись секретними повідомленнями.

Сценарій представляє собою розповсюджену проблему збору інформації про температуру пристроя, яку вивантажують в базу даних на cloud, для подальшого аналізу, візуалізації і налаштування слідкуваннь.

В якості брокера було використано Mosquitto, як розповсюджений вибір для таких цілей. В якості бази даних і провайдера було використано InfluxDB – також як популярний вибір для даних часових рядів. Для достовірності було інтегровано Grafana з попередженнями про перегріви, як в звичайній системі контролю.

Було також розроблено скріпт для імітації даних з датчиків для різних сценаріїв – звичайної роботи, аномального стрибку температури, і перегріву. В випадку з аномальним стрибком, попередження про перегрів не мають на це реагувати, оскільки попередження налаштовані таким чином, що реагують на середнє значення за певний останній проміжок часу.

Всі посередники і проміжкові скрипти були розроблені мовою програмування python, також для побудови реквестів і тестування було використано SQL, а для налаштування Grafana також Flux.

Робота системи в звичайному режимі представляє собою наступне: девайс, за допомоги скрипта, що крутиться на ньому, считує інформацію з датчиків і публікує її на брокер. Опісля, скрипт-підписник брокеру публікує її в базу даних на cloud, до якої законфігурована окремо Grafana виводить графік-звіт роботи, і з якої будуються попередження і розсилаються на пошти відповідальних. Налаштований звіт Grafana виглядає, як показано на рисунку 3.4.

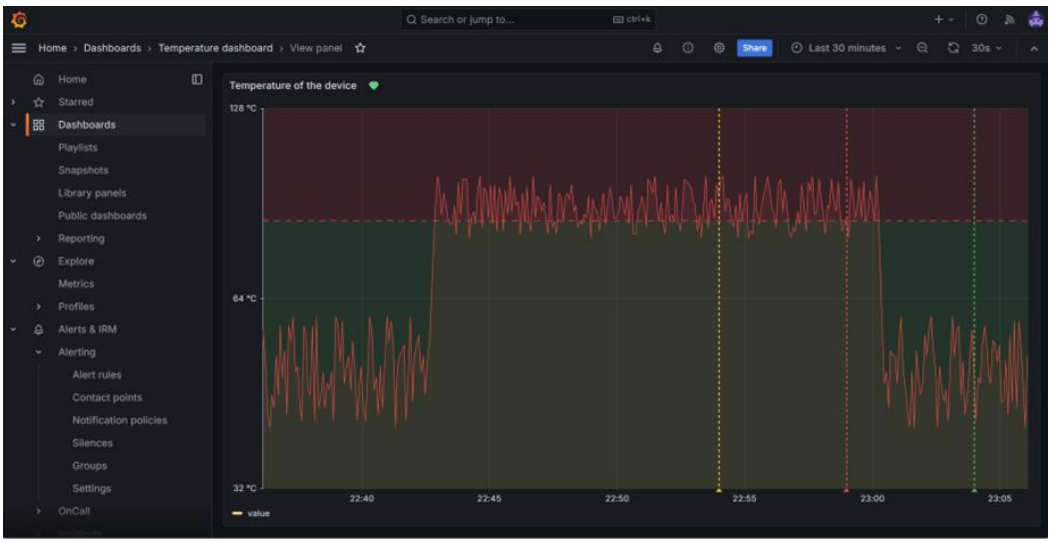

Рисунок 3.4 – Візуалізація даних в Grafana

Інкрустувати інформацію всередину повідомлення пропонується на етапі публікації даних з девайсу, або на етапі передачі їх в базу даних (рис.3.5) – там, де задіяні кастомізовані скрипти. Опісля, будь-хто, хто має доступ до бази даних, може забрати їх звідти селект реквестом і застосувати функцію декодування, де ключем буде таймштамп початку – і можлива функція кроку.

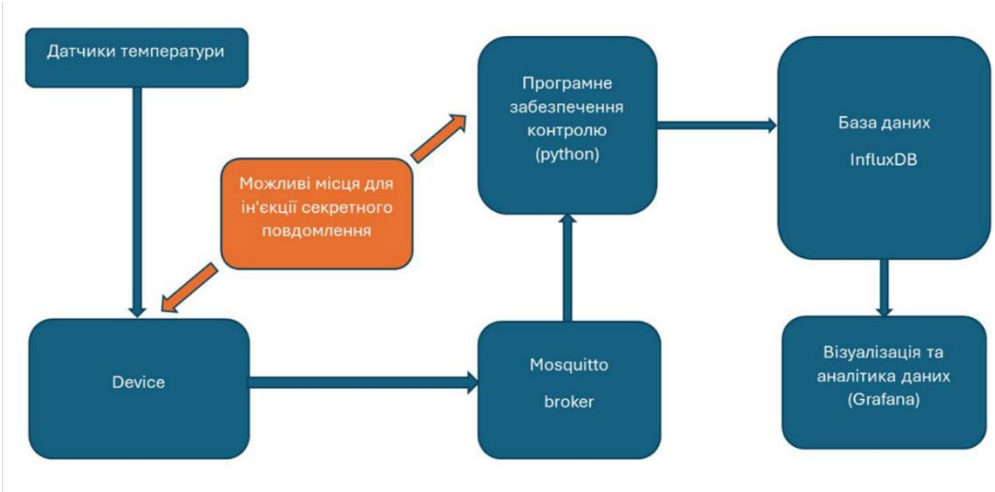

Рисунок 3.5 – схема системи прототипу 2

Влаштовування повідомлення полягає в тому, що на одному з запропонованих етапів ми змінимо парність певного біту значення – в прикладі це в одиницях, але в залежності від точності датчиків конкретної системи, це можуть бути будь-які малі цифри після коми.

Зручність використання стеганографії з IoT буде полягати в тому, що згодом повідомлення саме буде знищене, також за природою даних можна не очікувати пошкоджень повідомлення, і в решті решт, його буде складніше помітити, оскільки мається на увазі великий поток даних, а точність часу-ключа важко вгадати. Це вид передаваємої на постійній основі інформації, яка не викликає особливих підозр і важко піддається аналізу.

За підсумком, прототип 2 імплементує всі принципи правдоподібного заперечення і може дозволити повноцінну і самодеструктуючуся коммунікацію.

ВИСНОВКИ

Представлена робота спрямована на вивчення стеганографічного методу, креативних способів його застосування і моделей систем, які зможуть дозволити правдоподібне заперечення.

Можна помітити, що протягом роботи мова часто заходила про протоколи, автентифікацію, поведінкові фактори і послідовність дій. Це підкреслює, що стеганографія сама по собі не є достатньою для захисту інформації, але в поєднанні з криптографічними методами, виконаними над інформацією, а також в комбінації з якісно продуманим протоколом, що імплементує філософію методу правдоподібного заперечення (включно з автентифікацією), це може бути неймовірно сильним методом захисту.

За результатами, був розроблений ознайомлювальний програмний прототип, який дозволяє заховати чутливу інформацію у фотографіях, або для подальшого персонального використання (наприклад, зберігання важливих паролів) або для обміну чутливою інформацією. Для цього, за допомоги алгоритма модифікації кольорів LSB, в нього шифрується прихована інформація. Потрібну інформацію зберігає не кожен піксель, а лише обрані за функцією, яка влаштована в програму і має бути задана користувачем. Решта же пікселів теж модифікується, щоб створити «шум». Також було розроблено систему, яка застосовує аналогічний принцип в поєднанні з IoT системами для збору температурних даних, дозволяючи розглядання потокових даних часових рядів як потенційного каналу секретних комунікацій.

В роботі також був проведений докладний аналіз історичного розвитку правдоподібного заперечення в інформатиці, розбір алгоритмів посилення стійкості, і були запропоновані потенційні модифікації системи задля покращення методу. Були також розглянуті перспективи майбутнього і затронуті методи протидії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhang, Q., Jia, S., Chang, B. *et al.* Ensuring data confidentiality via plausibly deniable encryption and secure deletion – a survey. *Cybersecur* 1, 1 (2018). Access mode: https://doi.org/10.1186/s42400-018-0005-8
2. Microsoft Security, What is an authentication? Access mode: https://www.microsoft.com/en-us/security/business/security-101/what-is-authentication (дата звернення: 18.03.24)
3. Global Partners Digital, World map of encryption laws and policies / London, 2022. – Access mode: https://www.gp-digital.org/world-map-of-encryption/#
4. Sahu, Aditya Kumar and Sahu, Monalisa. "Digital image steganography and steganalysis: A journey of the past three decades" *Open Computer Science*, vol. 10, no. 1, 2020, pp. 296-342. Access mode: https://doi.org/10.1515/comp-2020-0136
5. Abdelrahman Desoky, Noiseless steganography: the key to covert communications / CRC Press , An Auerbach Book , 2012
6. Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, Brendan Halloran, Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research, Neurocomputing, Volume 335, 2019, Pages 299-326, ISSN 0925-2312, Access mode: https://doi.org/10.1016/j.neucom.2018.06.075
7. Henry Clinton Mask Letter to John Burgoyne, August 18, 1777. Henry Clinton Papers / William L. Clements Library, University of Michigan. Access mode: https://clements.umich.edu/exhibit/spy-letters-of-the-american-revolution/gallery-of-letters/clinton-burgoyne-mask-letter/
8. krypt3ia, “Al-Qaida goes “Old School” With Tradecraft and Steganography”, 13.03.2010, Access mode: http://krypt3ia.wordpress.com/2010/03/13/al-qaida-goes-old-school-with-tradecraft-and-steganography/
9. Bailey, Karen and Kevin Curran. “An evaluation of image based steganography methods.” *Multimedia Tools and Applications* 30 (2006): 55-88.

10. Wang, Yongjie & Stojiljković, Nina & Jehle, Johannes. (2016). 2010-Wang et al-J Virol Methods.

11. M. Sownya, G. Manikandan, Clustering based steganographic approach for secure data transfer / Contemporary Engineering Sciences, Vol. 8, 2015, no. 12, 525-531