МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ДЕМАШКЕВИЧ АНДРІЙ ВІКТОРОВИЧ

Допускається до захисту:
в.о.завідувача кафедри
прикладної математики та
кібербезпеки,
д. ф. з матем. А. В. Луценко
«___» _____2024 р.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРИХОВУВАННЯ ПОВІДОМЛЕННЯ У МУЛЬТИМЕДІА КОНТЕНТІ ЗА ДОПОМОГОЮ МЕТОДІВ СТЕГАНОГРАФІЇ

Спеціальність 113 Прикладна математика
ОП «Прикладна математика»

Кваліфікаційна (магістерська) робота

Науковий керівник:
Ю. С. Антонов, доцент кафедри
інформаційних технологій,
к. ф-м. наук, доцент
_____

Оцінка:___/___/_____
Голова ЕК:_____

Вінниця 2024

АНОТАЦІЯ

Демашкевич А. В. Розробка програмного забезпечення для приховування повідомлення у мультимедіа контенті за допомогою методів стеганографії. Спеціальність 113 «Прикладна математика», Освітня програма «Прикладна математика». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (магістерській) роботі досліджено розвиток та методи комп'ютерної стеганографії. Проаналізовано існуючі стеганографічні програми та описано комп'ютерно-математичну модель. Розроблено та протестовано програмне забезпечення для приховування інформації в цифрових зображеннях на базі методу заміни найменш значущого біта.

Ключові слова: стеганографія, програмне забезпечення, метод LSB, приховування повідомлень.

65 с., 11 рис., 6 табл., 57 джерел.

ABSTRACT

Demashkevych A. Software development for hiding message in multimedia content using steganography methods. Specialty 113 «Applied Mathematics», Program «Applied Mathematics». Vasyl` Stus Donetsk National University, Vinnytsia, 2024.

The qualification (master's) thesis investigates the development and methods of computer steganography. Existing steganographic programs are analyzed and a computer-mathematical model is described. The software for hiding information in digital images based on the method of replacing the least significant bit was developed and tested.

Keywords: steganography, software, LSB method, message hiding.

65 p., 11 fig., 6 tabl., 57 source.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BMP – Bitmap Picture.

GIF – Graphics Interchange Format.

JPEG – Joint Photographic Experts Group.

HTML – HyperText Markup Language.

LSB – Least Significant Bit.

PNG – Portable network graphics.

PSNR – Peak signal to noise ratio

WAV – Waveform Audio File Format.

ВСТУП

Від давніх часів до сьогодення однією з гострих проблем є захист інформації від несанкціонованого доступу. Ще у стародавні часи виникли два основні підходи до вирішення цього завдання, що існують і досі: криптографія та стеганографія. Криптографія спрямована на приховування змісту повідомлень шляхом його шифрування, стеганографія, навпаки, прагне приховати сам факт наявності повідомлення. Значним стимулом для розвитку стеганографії стало впровадження обмежень на використання криптографії в більшості країн. Наприклад, зобов'язання надавати ключі від систем шифрування, а також обов'язкова реєстрація та ліцензування криптографічних систем, незалежно від їхнього характеру. На відміну від криптографії, стеганографія не піддається зазначеним обмеженням і є ефективним способом приховування передачі даних.

Стеганографія застосовується для конфіденційної передачі повідомлень, захисту авторських або майнових прав. Вона може бути використана в ситуаціях, де потрібно приховати інформацію, не привертаючи зайвої уваги.

На тлі повномасштабного вторгнення рф в Україну все більшого розповсюдження набуває практика використання комп'ютерної стеганографії. Завантажувати стеганографічні програми та додатки рекомендують громадянам на тимчасово окупованих територіях задля збереження важливої інформації та мінімізації ризику для життя. Окрім того, стеганографічні програми допомагають приховати та передати файли з координатами місця дислокації ворога. Професійні військові використовують методи стеганографії для захисту конфіденційності, у таємних операціях та шпіонажі, дезінформації противника. Загалом, стеганографія має важливе значення, надаючи військовим можливість прихованої відправки та захисту інформації, що може бути вирішальним фактором у забезпеченні безпеки та успіху операцій.

У зв’язку з активним розповсюдженням технологій збільшується науковий інтерес до стеганографії. Значний внесок в теоретичне обґрунтування стеганографії здійснили такі вчені як В. К. Задирака [5], Н. В. Кошкіна [13], В. А. Хорошко [26], О. А. Смірнов, Є. В. Мелешко[21], M. G. Kuhn [47], R. J. Anderson [31] та інші. Питанню атак на стеганосистеми присвячені праці П. В. Римара, В. В. Крохмалюк [20]. Г. Ф. Конахович [10] в своїй роботі дав оцінку методам стаганографії для вбудовування інформації в спектральну область зображень; В. Ю. Корольов, В. В. Поліновський, В. А. Герасименко [11] розглянули принципи роботи та концепції обходу стегоаналізу; І. В. Швідченко [27] дослідила виявлення прихованого повідомлення в зображеннях. Проте у напрямку комп’ютерної стеганографії у зв’язку з невпинним розвитком технологій залишається низка неопрацьованих питань.

Актуальність теми дослідження обумовлена необхідністю розробки нових алгоритмів стеганографічних програм для безпечної передачі прихованих повідомлень, що має важливе значення в сучасному світі інформаційних технологій.

Об’єктом дослідження є комп’ютерна стеганографія.

Предмет дослідження – стеганографічний метод та засоби приховування повідомлення в мультимедіа.

Метою дослідження є вирішення питання підвищення якості стеганографічного захисту та розробка програми для приховування інформації в мультимедіа контенті методом стеганографії.

Для досягнення поставленої мети були сформовані та вирішені наступні завдання:

1. Дослідити розвиток та основні напрями стеганографії.
2. Сформувати узагальнену модель стегосистеми.
3. Проаналізувати основні методи комп’ютерної стеганографії.
4. Дослідити існуючі стеганографічні програми.

5. Розробити комп’ютерно-математичну модель програми.

6. Розробити та протестувати програмне забезпечення для приховування інформації в цифрових зображеннях.

У процесі дослідження для отримання наукових результатів та висновків використані методи аналізу та порівняння, математичного та комп’ютерного моделювання, методи стегоаналізу та інженерія програмного забезпечення. У процесі розробки програми використані стандартні можливості фреймворку . NET Core.

Наукова новизна одержаних результатів полягає у наступному:

1.Досліджений поточний стан програм для приховування інформації в файлах мультимедіа, що дозволило сформувати оцінку та вимоги до програми.

2. Розроблений алгоритм розподілу одного прихованого повідомлення у кілька контейнерів для підвищення рівня безпеки інформації.

3.Реалізовані запропоновані алгоритми в програмі, в основі якої лежить метод LSB та додаткове шифрування за допомогою криптографічних служб.

4.Розроблене програмне забезпечення для приховування повідомлень в контейнерах-цифрових зображеннях.

Практичне значення дослідження полягає у створенні програми для приховування повідомлень у файлах мультимедіа.

Магістерська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 57 найменувань, 3 додатків. Загальний обсяг роботи становить 65 сторінки.

РОЗДІЛ 1
ТЕОРЕТИЧНА ОСНОВА ТА ПРОБЛЕМАТИКА СТЕГАНОГРАФІЇ

1.1 Історія розвитку та основні напрями стеганографії

Швидкість та легкість отримання доступу до інформації значно посилюють ризик порушення безпеки даних. Питання захисту інформації є пріоритетним протягом багатьох століть. Уже в давні часи виокремилися два головні підходи до його вирішення, які актуальні й зараз: криптографія та стеганографія. Криптографія спрямована на приховування сенсу повідомлень шляхом їх шифрування. Слово «стеганографія» у перекладі з грецької буквально означає тайнопис (steganos – таємниця, секрет; graphy - запис). Стеганографія являє собою сукупність методів та засобів їхньої реалізації, які базуються на різних принципах і дозволяють приховувати сам факт існування секретної інформації в тому або іншому середовищі [26, с. 8].

Одним із перших, хто фіксував факт передачі повідомлень стеганографічними методами, вважається Геродот. Він описав використання дерев'яних дощечок, коли повідомлення наносили під шар воску. Інший метод передачі повідомлення – використання поголеної голови раба. Інформацію наносили татуюванням і після відростання волосся раб вирушав із посланням. У стародавньому Китаї практикували метод запису повідомлень на смужках шовку. У середньовіччі через посилення контролю спільно використовувались криптографічні та стеганографічні методи. У XV столітті відомі описані засоби приховування повідомлень були викладені в книзі «Steganographia». На початку XVII століття Гаспаром Скоттом було запропоновано метод приховування повідомлення в нотах, де кожній ноті відповідала певна літера. Однак, з таким методом не можна було отримати гармонійну мелодію. З XVIII століття стеганографію активно стала використовувати розвідка. Серед способів передачі повідомлень були присутні коди, що виглядали як звичайні тексти, невидимі

чорнила, повідомлення надзвичайно малих розмірів та ін. Таким чином, приховували сам факт передачі повідомлення [26, с. 10].

Під час Другої світової війни американські служби контролювали кореспонденцію на предмет використання симпатичних чорнил. Для виявлення прихованих повідомлень лаборанти використовували спеціальний пристрій із різними щітками, змоченими в різних реагентах. До того ж листи перевірялися під інфрачервоним та ультрафіолетовим світлом. Так, текст, написаний із використанням крохмалю, світився під ультрафіолетом, але залишався невидимим при стандартному освітленні, а інфрачервоне випромінювання допомагало розрізняти кольори, злиті з фоном.

Ще один метод стеганографії – це використання мікрокрапок. Зображення з даними (як, наприклад, знімок розвідданих) стискали до розміру маленької крапки і замінювали звичайну крапку в тексті. Потім одержувач міг збільшити зображення і вивчити зміст повідомлення. Спеціалісти німецької розвідки активно взялися за застосування мікрокрапок перед початком Другої світової війни, і цей метод роботи залишався ефективним аж до її завершення [26, с. 12].

До завершення XIX століття стеганографія та криптографія вважалися частинами однієї галузі науки, пов’язаної з шифруванням. Але після того як голландський офіцер Кірхгофс сформував принцип, згідно якого надійність криптографічного методу має базуватися лише на надійності ключа, криптографія стала розглядатися як окремий від стеганографії напрям. Багато методів у стеганографії базуються на методології, схожій з криптографією, яку розробив К. Шенон в теорії шифрування. Хоча його основний внесок стосується криптографії та теорії інформації, його роботи також вплинули на розвиток сучасної стеганографії.

Одним із провідних дослідників у галузі комп’ютерної безпеки та стеганографії наприкінці XX століття став Росс Андерсон [31]. Він проводив

дослідження, спрямовані на виявлення вразливостей у наявних стеганографічних системах, що допомогло поліпшити методи прихованого передавання інформації.

З 1996 року розпочалися міжнародні зустрічі, присвячені питанням приховування інформації (Information Workshop on Information Hiding). Перший форум, цілком присвячений стеганографії, пройшов у липні 2002 року. Зараз стеганографія активно та стрімко рухається вперед, опираючись на засади криптографії, обробки цифрових сигналів, теорії комунікації та інформатики [14, с. 6]. У 2008 році польські експерти з Варшавського університету технологій вперше розробили принципи та визначення для комп'ютерної стеганофонії, запропонувавши низку методів приховування даних у мережі IP-телефонії [49].

Серед вітчизняних науковців вивченням стеганографії займались Н. В. Кошкіна, В. О. Хорошко, Ю.Є. Яремчук, В.В. Карпінець. Проте теоретичні принципи стеганографії досі залишаються недостатньо вивченими.

Коли цифрові технології переплелися з усіма сферами життя, актуальність використання стеганографії лише зросла. Стеганографія дозволяє приховувати інформацію так, що зовнішній спостерігач не здогадається про її наявність. На сучасному етапі стеганографію використовують для вирішення завдань, таких як захист авторських прав, відстеження каналів витоку інформації, захист конфіденційної інформації від несанкціонованого доступу, створення секретних каналів передачі інформації, подолання систем моніторингу, камуфляж програмного забезпечення.

О. І. Стасюк виділяє наступні напрями стеганографії: класична, цифрова, лінгвістична та квантова [23, с. 1]. Класична стеганографія – це спосіб приховування даних, що здійснюється за допомогою технічних засобів. Приклади традиційної стеганографії в стародавньому світі нами були наведені вище. До сучасної стеганографії відносять фізичні та хімічні методи. Лінгвістична стеганографія заснована на приховуванні конфіденційної інформації в тексті за допомогою мовних особливостей і лінгвістичних засобів. Квантова стеганографія

заснована на поєднанні класичної та квантової інформатики, аналогічно традиційним створена з метою приховування самого факту передачі повідомлення [35, с. 12]. Цифрова стеганографія полягає у приховуванні додаткової інформації в цифрові медіа, що призводить до незначних викривлень об’єкта. Зазвичай ці об’єкти представляють собою мультимедійний контент, і внесені деформації зазвичай є непомітними для споживача, оскільки вони залишаються поза його сприйняттями [28, с. 13].

1.2 Узагальнена модель стегосистеми

Завданням стеганографії є захист інформації за допомогою вбудовування секретних повідомлень в інші дані, останні називають контейнерами. Існують два види контейнерів. Порожній контейнер – це той, в якому відсутні приховані повідомлення. Натомість заповнений контейнер або стего – це той, що містить приховані повідомлення. Ці два види контейнерів не мають відрізнятися між собою. Порожніми контейнерами можуть слугувати комп’ютерні файли, цифрові фотографії, аудіо та відео. У якості прихованого повідомлення можна використовувати текст чи чорно-білі зображення, такі як креслення або схеми. Стеганоканал – це канал для передачі стеганоконтейнера. Стеганоключ – це таємний ключ, що використовується для приховування даних. Розрізняють два типи стегоключа: секретний та відкритий. Стегосистема з секретним ключем містить один ключ, який має бути встановлений до початку передачі прихованих повідомлень або переданий через безпечний канал. У системах з відкритим ключем для вставки та вилучення інформації вживаються два різні ключі. Вони розроблені так, що на основі одного ключа неможливо обчислити інший. Таким чином, один ключ (відкритий) може бути переданий без обмежень по незахищеному комунікаційному каналу. Така система ефективно працює навіть у випадку, коли відправник і отримувач не довіряють один одному.

Сукупність порожніх контейнерів, повідомлень, ключів, заповнених контейнерів та алгоритмів впровадження та вилучення називають стеганографічною системою [6].

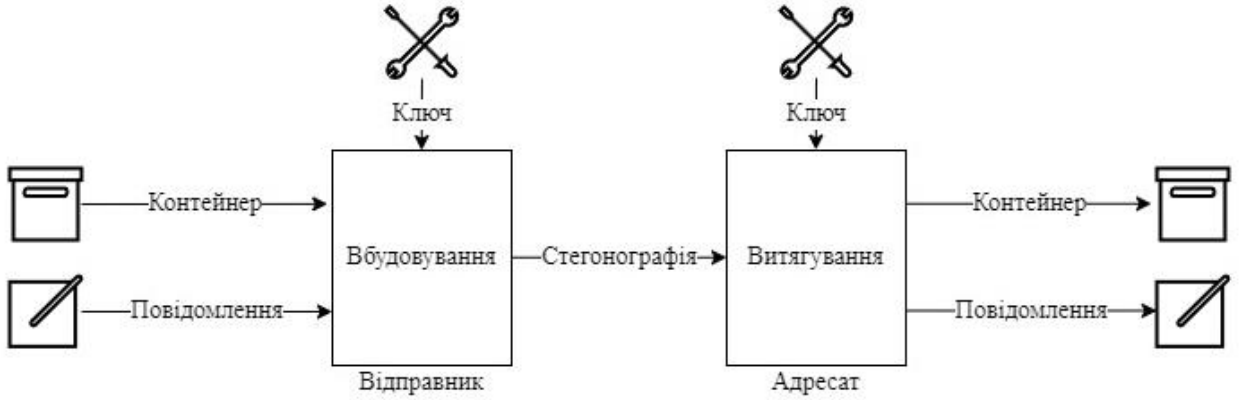


Рисунок 1.2 – Узагальнена модель стегосистеми

Джерело: [19, с. 31].

Розглянемо узагальнену модель стегосистеми (рис.1.2). Ключовими стеганографічними поняттями виступають повідомлення і контейнер. Повідомлення m∈M – це секретна інформація, наявність якої необхідно приховати, { } mn , , m2 , M = m1 K – множина всіх повідомлень. Контейнером c ∈C називається несекретна інформація, яку можна використовувати для приховання повідомлення, C = {c1 ,c2 ,...,cq } – множина всіх контейнерів, причому q >> n. Як повідомлення й контейнер можуть виступати як звичайний текст, так і файли мультимедійного формату.

Порожній контейнер (контейнер-оригінал) – це контейнер c, що не містить прихованої інформації. Заповнений контейнер (контейнер-результат) – контейнер c, що містить приховану інформацію m (cm).

Слід звернути увагу, що надійність системи залежить від обсягу вбудованих даних. Якщо збільшити обсяг даних, надійність системи знизиться (рис.1.3). Таким чином, при використанні стеганографії є обмеження на розмір вбудованих даних.

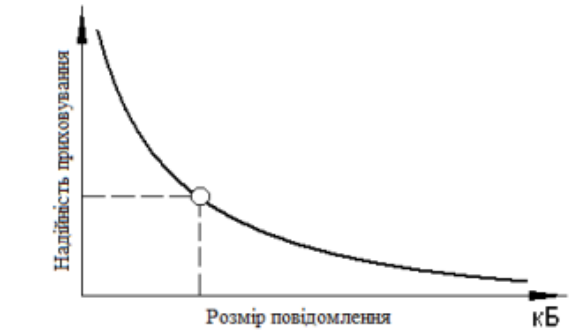


Рисунок 1.3 – Залежність надійності приховування даних від розміру повідомлення

Джерело: [26].

Для перевірки надійності стегосистем проводять стеганографічний аналіз. Стегосистеми поділяють на стійкі системи, практично стійкі і нестійкі в залежності від рівня забезпечення таємності. У теоретично стійкій стегосистемі інформація приховується лише у тих частинах контейнера, які не перевищують рівень шумів чи помилок квантування. У такій системі неможливо розробити метод для виявлення прихованих даних. У практично стійкій системі зміни у контейнері можуть бути помічені, але ефективних методів для їх виявлення у противника немає. На відміну від цього, нестабільна система приховує дані так, що вони можуть бути легко знайдені за допомогою наявних стеганолітичних інструментів. У таких ситуаціях стеганографічний аналіз допомагає виявити слабкі точки перетворення і вдосконалити його, забезпечуючи краще приховування даних [26, с. 30].

Стегосистему вважають компрометованою, якщо противник зумів підтвердити наявність прихованого повідомлення в отриманому контейнері. Якщо противник не має доказів для підтвердження наявності прихованого повідомлення в контейнері, то така стегосистема вважається надійною. Дослідники виділяють пасивні та активні види атак на стегосистеми. Зазвичай, високоякісна стегосистема пропонує трирівневу модель захисту даних, яка вирішує два ключових питання. По-перше, система маскує саме існування

захищеної інформації (це перший рівень захисту). По-друге, вона використовує методи, що перешкоджають несанкціонованому доступу до даної інформації (це другий рівень захисту). Додатково, в системі передбачено і третій рівень захисту, який полягає в криптографічному захисті (шифруванні) прихованих даних.

Для оцінки основної властивості стегосистеми – рівня конфіденційності проводяться аналітичні дослідження та практичні тести. Щоб оцінити якість стеганографічного приховування, часто використовують методи з інших дисциплін, зокрема, з криптоаналізу.

1.3 Методи комп'ютерної стеганографії

Методи комп'ютерної стеганографії розвиваються за двома основними напрямами:

1) методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

2) методи цифрової обробки сигналів, засновані на надмірності аудіо- і візуальної інформації.

До першого відносять методи, в яких застосовуються спеціальні властивості комп'ютерних форматів представлення даних. Властивості обирають таким чином, щоб забезпечити захист прихованої інформації від безпосереднього доступу, перегляду чи читання (наприклад, використання вільного кластерного простору файлів або невикористаних полів розширень). Мінусом такого підходу є обмежена прихованість та незначна кількість інформації. Використання методів другого напрямку пояснюється структурною і психофізичною надмірністю в файлах. Цифрові зображення та звуки представлені числами, що відображають інтенсивність світла чи звукових вібрацій в певні моменти часу.

Далі розглянемо детальніше методи комп'ютерної стеганографії, класифіковані за видом контейнеру [14].

1.Методи приховування даних в зображеннях.

1.1 Приховування даних у просторовій області: заміни найменш значущого біта; псевдовипадкового інтервалу; псевдовипадкової перестановки; блокового приховування; заміни палітри; квантування зображення; метод Куттера-Джордана-Боссена; метод Дармстедтера-Делейгла-Квісквотера-Макка. Основна ідея методів базується на заміні зайвих, надлишкових частин зображення бітами таємного повідомлення. Для вилучення цього повідомлення потрібно знати алгоритм, за допомогою якого прихована інформація була розміщена в контейнері. Одним із головних недоліків є нестійкість до більшості видів спотворень.

1.2 Приховування даних у частотній області: метод відносної заміни величин коефіцієнтів дискретно косинусного перетворення; метод Бенгама-Мемона Ео-Юнга; метод Хсу і Ву; метод Фрідріха. Виділення зон конфіденційності відбувається після розкладання вихідного сигналу на його складові. Залежно від потрібного рівня стійкості для конфіденційного повідомлення, обирається певний тип трансформації. Проте ємність для вбудовування може бути обмеженою.

1.3 Методи розширення спектру за допомогою прямої псевдовипадкової послідовності, за допомогою стрибкоподібного перебудовування частот, за допомогою компресії з використанням лінійної частотної модуляції. Методи базуються на розподілі прихованої інформації по всьому спектру контейнера, що забезпечує високий рівень прихованості. Однак для вилучення інформації необхідно мати доступ до оригінального немодифікованого контейнера для порівняння.

2. Методи приховування інформації в аудіо.

2.1 LSB кодування базується на алгоритмі, що замінює найменш значущий біт, щоб приховати послідовність байтів, що містять приховані дані. Проте перекодування або стиснення аудіофайлу може знищити приховану інформацію.

2.2 Паритетне кодування. Метод розбиває сигнал на окремі зразки і вставляє кожен біт секретного повідомлення в біт парності. Не вимагає значних

обчислювальних ресурсів для реалізації. У великих системах може додавати об’єму файлу.

2.3 Фазове кодування. Ґрунтується на використанні фазової інформації аудіосигналу для кодування додаткового повідомлення, прихованого у вихідному сигналі. Суть цього методу полягає в тому, що зміни в фазі аудіосигналу можуть бути використані для кодування додаткової інформації, не спричиняючи значущих артефактів або втрати якості звуку.

2.4 Передача секретної інформації виконується шляхом розподілу спектра у частотному діапазоні аудіофайлу за допомогою коду, що не взаємодіє з фактичним сигналом. Інформація розподіляється по широкому спектру, вона стає менш помітною, але загальна ємність для вбудовування даних може бути обмеженою.

2.5 Приховування відлуння. Інформація вбудовується шляхом додавання відлуння до оригінального сигналу. Є малозамітним для людського вуха. Якщо файл піддається модифікаціям, відлуння може бути змінено або втрачено.

3. Методи приховування даних у відео.

3.1 Метод вбудовування на рівні коефіцієнтів. Біти приховуваного повідомлення вбудовуються в коефіцієнти дискретно-косинусного перетворення. Приховувана інформація зберігається при фільтруванні, зашумленні (адитивним шумом) і дискретизації. Повідомлення може бути виявлено за допомогою статистичних методів аналізу, які досліджують аномалії або зміни в розподілі коефіцієнтів.

3.2 Метод вбудовування інформації на рівні бітової площини здійснюється шляхом взаємодії з окремими бітами (або бітовими площинами) відео даних. При повторному накладенні послідовності біт якість відео погіршиться, а інформація буде видалена.

3.3 Метод вбудовування інформації за рахунок енергетичної різниці між коефіцієнтами. Інформація вбудовується шляхом видалення декількох коефіцієнтів дискретно-косинусного перетворення. Може забезпечити високу стійкість проти простих зовнішніх атак, таких як шум або стиснення.

Таким чином, основним напрямком комп'ютерної стеганографії є використання властивостей надмірності контейнера-оригіналу, але водночас Важливо враховувати, що при приховуванні інформації може відбутися спотворення певних статистичних характеристик контейнера або порушення його структури.

1.4 Огляд сучасних рішень приховування інформації в мультимедіа

Великий вибір програм стегоаналізу представлений на сторінках закордонних веб-ресурсів, в той час як вітчизняних програмних продуктів майже немає у вільному доступі. Проаналізуємо програми для приховування даних у контейнерах мультимедіа: зображення, аудіо, відео.

Програмне забезпечення StegSpy v2.1 [51] здатне виявити факт існування зашифрованої інформації програмами Hiderman, JPHide and Seek, Masker, JPegX, Invisible Secrets. До недоліків програми відносять можливість проводити одночасно аналіз тільки одного файлу, виявлення прихованих даних тільки однією з відомих застосунку програм, використання тільки сигнатурного методу.

Програмне забезпечення з відкритим кодом Stegdetect 0.6 [54] здатне працювати з цифоровими зображеннями форматів JPEG, BMP, PNG. Програма використовує статистичні методи аналізу, щоб виявити зміни в зображеннях, які могли бути викликані впровадженням прихованої інформації. В результаті виводиться звіт аналізу, в якому демонструється ймовірність наявності стеганографії в зображенні. Зазвичай використовується через командний рядок, що дає змогу користувачам налаштовувати параметри аналізу.

XstegSecret beta v0.1 націлений на компіляцію, втілення та спрощення використання методів стеганоаналізу для цифрових медіа-файлів, таких як зображення, аудіо та відео. Серед переваг програми – вбудована база, що розпізнає понад 40 стеганографічних програм. Вона дозволяє провести сканування файлової системи чи конкретних папок на досліджуваних носіях, щоб

виявити файли-артефакти, пов’язані з певними стеганографічними додатками. Серед недоліків дослідники акцентують увагу на недостатньо зручному інтерфейсі, також останній раз програма оновлювалась у 2007 році [27].

У дослідженні стеганографічного програмного забезпечення [57] було відібрано п’ять програм випадковим методом для вивчення їх особливостей та можливостей. Для порівняння були обрані Invisible Secrets 4, Hermetic Stego 8.04, Puffer 4.04, Xiao Steganography 2.6.1 та S-Tools.

Invisible Secrets 4 дає змогу приховувати і захищати дані у файлах зображень, звукових і відеофайлах, а також інших типах файлів, щоб забезпечити їхню секретність і цілісність. Програма підтримує п’ять форматів файлів: JPEG, PNG, BMP, HTML і WAV; надає можливість шифрування файлів з використанням різних алгоритмів, включно з Advanced Encryption Standard, Blowfish та іншими. Це забезпечує додатковий рівень безпеки.

Puffer версії 4.04 було випущено як доопрацьовану версію Puffer 4.03 у 2009 році. Це шифрувальне і стеганографічне програмне забезпечення загального призначення, яке використовується для захисту даних від несанкціонованого перегляду для безпечного обміну, працює з контейнерами форматів PNG та BMP.

Xiao Steganography 2.6.1 розроблено компанією Nakasoft у Венесуелі 2005 року. До сильних сторін можна віднести простоту у використанні та інтуїтивно зрозумілий інтерфейс.

Hermetic Stego 8.04, є частиною пакету програм Hermetic, яка була випущена в 2009 році. Hermetic Stegois може приховати будь-який тип файлу в зображеннях BMP за допомогою ключа шифрування, так що присутність прихованого файлу неможливо виявити, якщо при приховуванні даних був використаний ключ.

S-Tools є інструментом для стеганографії, який був створений Енді Брауном у 1996 році. Ця програма дає змогу використовувати зображення або аудіофайли для приховування інших зображень або аудіофайлів і навпаки.

Існують деякі обмеження щодо формату файлів: аудіофайл має бути типу WAV, а файл зображення – BMP або GIF, однак це обмеження стосується лише зображення-контейнеру, яке приховуватиме секретне зображення. Програма прагне зберегти високу якість і непомітність змін у зображеннях, щоб вони мали натуральний вигляд. Має простий і зрозумілий користувацький інтерфейс, що робить його відносно легким у використанні.

Показник якості зображення після вбудовування зазвичай відображає успішність роботи стеганографічної системи. Інструментом для вимірювання якості зображення після вбудовування в експерименті став показник PSNR (табл. 1.1). Пікове відношення шуму – технічний термін для відношення між максимально можливою потужністю сигналу та потужністю спотворювального шуму, який впливає на точність його представлення.

Таблиця 1.1 – Тестування критерієм оцінки якості стискання PSNR

Зображення	**Invisible Secrets 4**	**HermeticStego 8.04**	**Puffer 4.04**	**Xiao Stegano-graphy2.6.1**	**S-Tools**
1	2	3	4	5	6
Зображення 1	51.14	50.81	43.21	51.7689	56.39
Зображення 2	51.14	50.07	49.15	57.7948	62.43
Зображення 3	51.14	50.81	43.15	51.7704	56.40
Зображення 4	51.14	50.85	43.06	51.7911	56.38
Зображення 5	51.14	50.81	43.17	51.7508	56.41
Зображення 6	51.14	50.07	49.15	57.7956	62.43

Джерело: [57].

Результати експерименту показали, що всі п’ять програм працюють відносно на одному рівні. Проте ці методи мають певні недоліки, серед яких обмежена ємність сховища інформації, а алгоритми, як правило, не є стійкими до геометричних атак та атак стиснення.

У роботі Д. О. Тарасова, А. С. Мельник та М. М. Голобородько [24] були сформовані критерії порівняння засобів стеганографії та виконано практичний аналіз 15 програмних продуктів для стеганографії. Більшість проаналізованих програм для стеганографічного захисту використовують в якості контейнерів 24-

розрядні зображення BMP та 16-розрядні аудіозаписи WAV. Така популярність обумовлена розповсюдженістю цих форматів і простотою у взаємодії з ними. Програма Puff визнана найзручнішою з числа інших завдяки таким перевагам: підтримка найширшого спектру форматів даних; можливість використання складеного контейнера для приховування великих обсягів інформації; здатність обробляти як файли, так і папки одночасно; можливість архівування та шифрування схованих даних; висока продуктивність, особливо при роботі з великими обсягами інформації.

В останні роки передача відео є типовою подією. До прикладу, на сервісі YouTube розміщено сотні мільйонів відеозаписів, причому однаковий відеоматеріал може бути представлений в різних форматах. Також численні відеофайли знаходяться в P2P-мережах. Що стосується використання відеофайлів як контейнерів, то стеганографія все ще сильно залежить від академічних досліджень. Практичні інструменти, опубліковані для публічного використання, майже відсутні. Також згідно з дослідженнями [40], не існує програм у вільному доступі, що спеціалізується на виявленні прихованих даних у відеофайлах. Поточні дослідження проводяться в основному університетами. Деякі з цих робіт пропонують демонстраційні інструменти або програмні скрипти для підтвердження концепції виявлення прихованого вмісту, але повноцінний інструментарій, який може бути використаний широкою аудиторією, відсутній.

Є. Ю. Катаєва та А. Г. Ребріков [9] проаналізували використання прихованої передачі даних у відеофайлах. Вчені порівняли існуючі алгоритми відеостеганографії та дослідили, як людське око реагує на відтінки кольорів. Людський організм ставить пріоритет на розпізнавання декількох типів кольорів (конкретно червоного і зеленого) за рахунок неможливості сприйняття більшої кількості кольорів. Отже, відтінки синього кольору є найменш помітними для

людського ока, тому вбудовування інформації саме в синій кольоровий канал є найбільш оптимальним рішенням.

Таким чином, стегоаналіз цифрових медіафайлів, особливо аудіо та відеофайлів, все ще перебуває на стадії досліджень і розробок. Іноді можливість знищити приховані дані може бути єдиним доступним вибором, коли неможливо довести існування прихованих даних та/або неможливо їх вилучити.

Окрім готових програм зручними інструментами для приховування повідомлень є веб-застосунки та доступні рішення на сервісах для сумісної розробки і хостингу проектів. Розглянемо детальніше один із інструментів [53]. Інтерфейс подібних сервісів досить простий та зрозумілий для користувачів. Щоб закодувати повідомлення в зображення, необхідно вибрати файл, який буде слугувати контейнером, ввести текст і через кнопку запустити процес кодування. Не дивлячись на легкість у використанні, даний сервіс має ряд обмежень та проблем. По-перше, чим більше тексту треба приховати, тим більшим має бути зображення. По-друге, після кодування в рази збільшується розмір зображення. При декодуванні зображення у тому ж сервісі, де воно було закодоване, інструмент не видає повне повідомлення.

На основі вищенаведених фактів можемо стверджувати, що даний інструмент не є надійним та не може гарантувати захист передачі повного повідомлення.

Висновки до розділу 1

У першому розділі роботи розглянуто теоретичну основу та проблематику стеганографії. Вона з давніх часів була одним із двох головних підходів вирішення питання захисту інформації. На відміну від криптографії, яка шифрує саме повідомлення, стеганографія приховує сам факт передачі повідомлення. Окреслені основні напрями стеганографії в залежності від галузей використання.

Наведено визначення та узагальнена модель стегосистеми, математично описано процес стеганографічного перетворення. Тип контейнера, в якому приховується інформація, є важливим для стеганографії. Його особливості використовують для створення алгоритмів, виявлення яких вимагає значного витрати ресурсів. Сформовані основні вимоги до стегосистеми.

Проведено порівняння методів стеганографії, в яких в якості контейнерів використовуються зображення, аудіо та відеофайли, розглянуто переваги та недоліки.

Здійснено огляд існуючих у відкритому доступі програм для приховування даних у контейнерах, таких як зображення, аудіофайли та відеофайли. Найбільш популярними контейнерами є файли зображень, найменше прикладів програмного забезпечення з контейнерами відеофайлів.

РОЗДІЛ 2
РОЗРОБКА АЛГОРИТМУ ПРОГРАМИ

2.1 Постановка завдання

На сьогоднішній день у доступі є велика кількість програмних продуктів для стеганографії, які пропонують безкоштовно чи за умовами, що майже не вимагають оплати. Вони працюють на принципі вбудовування таємних повідомлень у контейнери, такі як зображення, аудіо та відеофайли. Використовуючи подібне програмне забезпечення, можна передавати конфіденційні дані, маскуючи їх під звичайною інформацією, яка доступна всім.

У результаті аналізу доступних в мережі Інтернет стеганографічних програм було зроблено висновок, що найпоширенішим методом приховування є метод найменшого значущого біта. При застосуванні даного методу можливо приховати значну кількість інформації у файлах, забезпечити підвищений рівень безпеки. Однак були виявлені наступні недоліки:

– для приховування повідомлення використовується один контейнер;

– великі повідомлення можуть вимагати зміни великої кількості бітів, що може призвести до видимих аномалій у контейнері;

– сучасні методи стеганоаналізу можуть виявляти зміни в найменш значущих бітах, особливо якщо змінено багато бітів у контейнері;

– багато програм, що використовують метод LSB, не використовують додаткового шифрування для прихованого повідомлення. Це означає, що якщо приховане повідомлення буде виявлено, його можна легко прочитати.

Таким чином необхідно розробити програму для приховування повідомлень на основі одного із стеганографічних методів з можливістю шифрувати більше інформації в кілька контейнерів для підвищення рівня безпеки.

Програма має відповідати наступним функціональним вимогам:

1.Використовувати як контейнер цифрові зображення форматів PNG, BMP.

2.Приховувати в контейнерах текстову інформацію.

3. Забезпечувати візуальну ідентичність стего та оригіналу.

4.Проводити валідацію приховування повідомлення в контейнері.

5.Передбачати можливість обирати кілька контейнерів для приховування повідомлення.

6.Проводити стегоаналіз контейнерів з прихованою інформацією.

7.Відображати результати приховування.

8.Відображати результати стегоаналізу.

Перша вимога полягає у можливості використання в якості контейнерів файли форматів PNG, BMP, оскільки такі контейнери з прихованим повідомленням людському оку важко відрізнити від оригіналу. До того ж, формат BMP є некомпресованим, PNG використовує втратно-вільний метод компресії, що не змінює оригінальні пікселі. Вибір саме цифрових зображень в якості контейнерів обумовлений низкою причин, а саме:

– доступністю інструментів для розробки;

– наявністю широкої бази стеганографічних програм, можливістю вивчити та виявити їх недоліки;

– існування практично важливого виклику щодо захисту фотографій та картин від незаконного копіювання та поширення;

– розмір контейнера відомий наперед, і відсутні обмеження, пов'язані з вимогами реального часу;

– присутністю шумової структури в більшості зображеннях, які як раз підходять для непомітного вбудовування інформації;

– низькою можливістю людиною побачити незначні зміни в зображенні, яскравості, контрастності, вмісту в ньому шуму, спотворень поблизу контурів;

– вдосконаленими методами цифрової обробки зображень. Згідно з

дослідженням, найбільш популярним типом даних, які використовуються в стеганографії є цифрові зображення (рис.2.1).

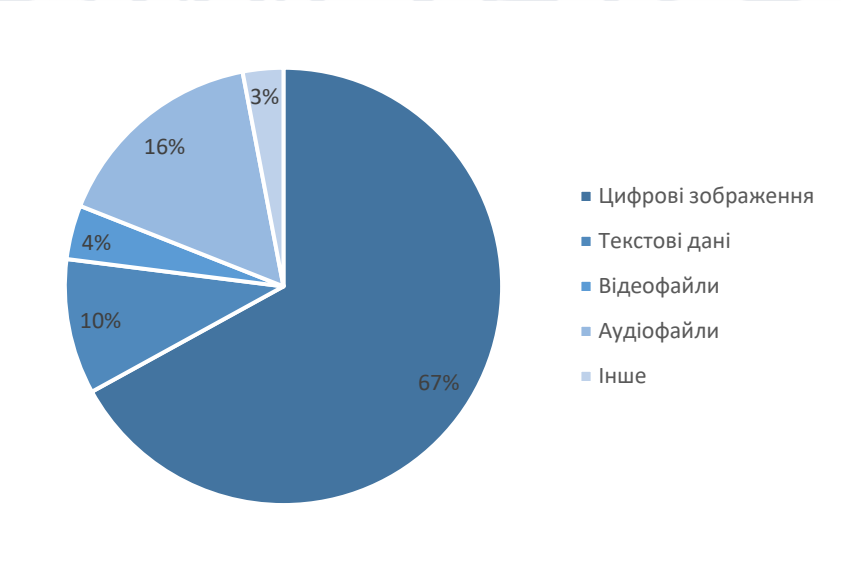


Рисунок 2.1 – Розподіл стеганографічних програм за видом файлів-контейнерів

Джерело: [4].

Друга та третя вимога є логічними для стегосистеми, виконання функції передачі повідомлення, скриваючи сам факт цієї дії. Четверта вимога необхідна в тих випадках, коли програма буде не здатна обробити та приховати велику кількість інформації. Для запобігання втрати частини повідомлення при кодуванні, програма має дати сигнал наперед про неможливість приховати повідомлення цілим. Виконання п'ятої вимоги закриває дві потреби – передача більшої кількості інформації та підвищення безпеки. Одне повідомлення має бути зашифровано в кількох контейнерах. Навіть у випадку, коли один із контейнерів може бути скомпрометований, секретне повідомлення цілим не вилучать. Шоста вимога передбачає декодування повідомлення з кількох контейнерів. Виконання заключних вимог забезпечить комфортну та швидку взаємодію користувача з програмою.

До нефункціональних вимог можемо віднести наступні:

– програма має працювати на операційній системі Windows;

– з метою отримання кращого результату приховування, використовувати зображення, де буде розподіл 8 пікселів на 3 символи;

– простий у використанні для користувача інтерфейс програми.

2.2 Дослідження методу LSB

Найпоширенішим серед методів заміни в просторовій множині є метод заміни найменш значущого біта. Молодший значущий біт зображення несе в собі найменше інформації. У більшості ситуацій люди не помічають незначних змін у бітах. Згідно з дослідженнями [47], для сприйняття не настільки важливо, чи будуть пікселі в масиві представлені одним і тим самим відтінком із доступної палітри, чи чергуванням двох кольорів із відповідними характеристиками, які разом відтворюють ідентичний колір. Такі невеликі відхилення можна розглядати як шум, що дозволяє вбудовувати інформацію, замінюючи найменш важливі біти пікселів на біти прихованого тексту. У зображеннях в градаціях сірого, де кожен піксель представлений одним байтом, кількість інтегрованих даних може досягти 1/8 від загального розміру файлу. Цей метод набув популярності завдяки своїй простоті та можливості вміщення відносно великої кількості інформації в невеликі файли. Ефективність прихованого каналу зв'язку від 12,5 до 30% [14, с. 58]. Розглянемо цей метод на прикладі 24-бітного растрового RGB-зображення. Кожна точка в такому зображенні кодується 3-ма байтами, кожен байт визначає інтенсивність червоного (Red), зеленого (Green) і синього (Blue) кольору. Сукупність інтенсивностей кольору в кожному з 3-х каналів визначає відтінок пікселя (рис. 2.2).

Рисунок 2.2 – Приклад отримання відтінку в RGB зображенні

Джерело:[3].

Вплив молодших розрядів на остаточне зображення менший у порівнянні з іншими розрядами. З цього можна зробити висновок, що заміна одного або двох менш значущих бітів на інші буде майже непомітною для відтінку пікселя, і глядач, ймовірно, не виявить змін. Припустимо, нам потрібно приховати в пікселі зображення (з параметрами R=56, G=200, B=53) шість біт: 100111. Для цього розіб'ємо їх на три пари і замістимо ними по два молодших біти в кожній колірній компоненті. Замість пікселя з параметрами R=56 (00111000), G=200 (11001000), B=53 (00110101), отримаємо R=58 (00111010), G=201 (11001001), B=55 (00110111). У результаті маємо новий відтінок, дуже схожий на вихідний. Як бачимо, ці кольори важко розрізнити навіть на великій площі (рис.2.3).

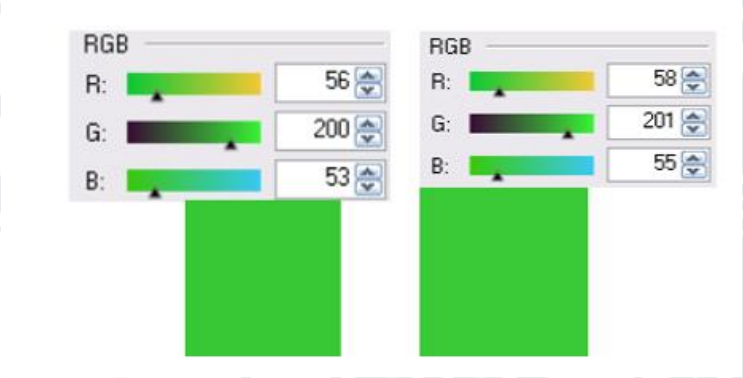


Рисунок 2.3 – Порівняння двох відтінків

Джерело:[3].

Використовуючи стеганографічний метод LSB, в середньому потрібно змінити лише половину бітів зображення-носія. Однак, контейнери можуть піддаватися різним видам атак. Виявлення факту передачі прихованої інформації займається окремий розділ стеганографії – стегоаналіз. У контексті комп'ютерної безпеки та інформаційної безпеки, стегоаналіз використовується для виявлення наявності та вилучення стеганографічно вбудованої інформації з мультимедійних файлів, текстових даних чи інших форматів, де інформація може бути таємно вбудована. Типи атак, що існують, можна умовно розділити на ті, що тільки компрометують наявність прихованого повідомленні та ті, що порушують його цілісність (табл.2.1).

Таблиця 2.1 – Типи атак на заповнений контейнер методом LSB

Стан прихованого повідомлення	**Тип атаки**	**Метод реалізації**
1	2	3
Компрометація факту наявності прихованого повідомлення	Атака на основі відомого порожнього контейнера	Порівняння порожнього з заповненим контейнерів та виділення шуму.
	Атака на основі відомого заповненого контейнера	Виявлення наявності повідомлення через часте вбудовування одним і тим самим способом та повторюваних ознак.
	Атака на основі відомої математичної моделі контейнера або його частини	Визначення відмінностей через підозрілий шум у відомій моделі. Біти всередині можуть бути скоррельовані один з одним.
	Суб’єктивна атака проти незахищених стеганограм	Порушник уважно розглядає зображення, намагаючись визначити «на око», чи є в зображенні приховане повідомлення.
Вплив на заповнений контейнер з метою руйнування стеганограми	Пошкодження файлу в процесі його передачі	Помилки під час передавання файлу каналом зв'язку. Перетворення файлу в інший формат (наприклад, зображення з BMP в JPEG). Стиснення зображення з втратами.
	Проведення маніпуляцій з файлом	Зміна розрядності колірної шкали з 8 до 7 біт. Застосування методів, що очищають від шумів. Звичайна обрізка файлів.

Джерело: складено автором на основі [20].

В залежності від вихідних даних, які використовуються, методи стегоаналізу можна умовно поділити на дві групи:

1) методи, що призначені для атак на конкретні контейнери, заздалегідь відомі;
2) методи, що призначені для атак на будь-які алгоритми стеганографії [4, с. 132].

При використанні останніх немає потреби знати конкретний стеганографічний метод. Найбільш розповсюджені з цієї категорії, як правило, базуються на алгоритмах, які потребують попереднього навчання за допомогою

наборів порожніх та із вбудованою інформацією контейнерів. Ці методи, в свою чергу, можна класифікувати як візуальні та статистичні.

Візуальні техніки ґрунтуються на можливості людського зору вивчати та розпізнавати різницю в зображеннях. Коли контейнер максимально заповнений, візуальні атаки стають найефективнішими. Проте, чим менше інформації в контейнері, тим важче виявити приховані дані. Зазвичай аналізують не саме зображення, а його бітові рівні. На поверхні, порівнюючи порожній та наповнений контейнер, візуальна різниця може бути непомітною. Однак при детальному вивченні бітового рівня, що включає найменш значущі біти, іноді можна виявити сліди вбудованої інформації. Для візуального дослідження бітових рівнів важливій спосіб вбудови прихованого повідомлення та тип контейнера. Коли прихована інформація додається до кожного пікселя послідовно, існує висока ймовірність виявлення цього. Проте якщо пікселі, в які додавалася інформація, були визначені за допомогою алгоритму псевдовипадкового вибору, завдання виявлення стає значно складнішим, зокрема, залежно від початкового «шуму» в контейнері. На рис.2.4 показані контейнери і відповідні їм бітові рівні з найменш значимими бітами в різних контекстах: а – зображення, яке слугує контейнером; б – бітовий зріз порожнього контейнеру; в – бітовий зріз частково заповненого контейнеру, стегоповідомлення записане послідовно в кожний піксель; г – бітовий зріз частково заповненого контейнеру, стегоповідомлення записане у пікселі, які обрані генератором псевдовипадкових чисел. Як можна побачити, найменш значущі біти вже мають деякий «шум», що ускладнює візуальне розпізнавання.

Популярними є і методи статистичного стегоаналізу. В основі цих методів лежить дослідження статистичних характеристик зображень, аудіофайлів та інших даних, щоб виявити аномалії, що можуть вказувати на наявність стеганографічного контенту.

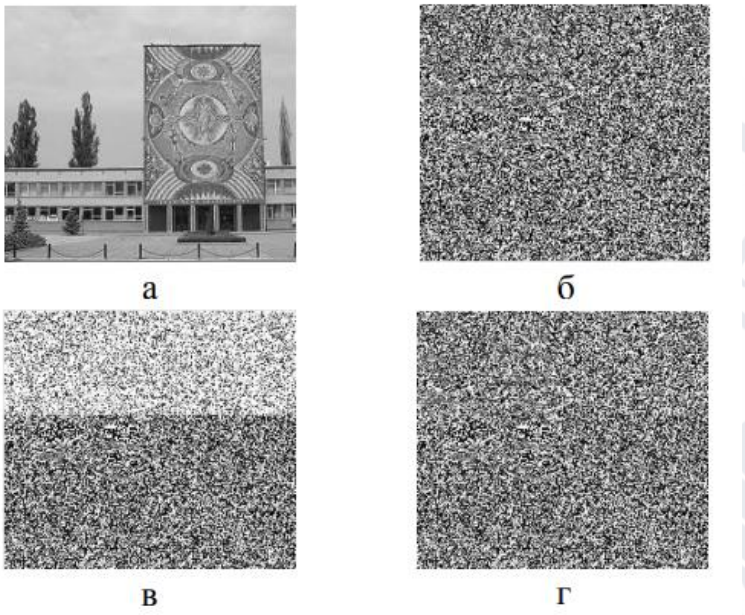


Рисунок 2.4 – Візуальний стегоаналіз контейнеру з великою первісною зашумленістю

Джерело: [21].

Одним із статистичних методів стегоаналізу є RS-аналіз. Зображення поділяється на сегменти, що складаються з n пікселів, позначених як G(x1, x2,...,xn), де n – парне число. Зазвичай вибирають 2 сусідніх пікселя, розташованих горизонтально. Для таких сегментів пікселів обчислюється регулярність або «гладкість», позначена як f(G). Як варіант, цей показник може базуватися на дисперсії величин пікселів у сегменті або просто на різниці між значеннями сусідніх пікселів. Значенням пікселя тут є ціле число в діапазоні від 0 до 255.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{n-1} |x_{i+1} - x_i| \quad (2.1)$$

Функція F(x) називається фліпінгом [21] і має властивість F(F(x)) = x. Визначаються дві функції фліпінгу – F_1, відповідає інверсії молодшого біта пікселя, і F_2, що представляє собою інверсію з переносом у старший біт (додавання одиниці):

$$F_1: 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255, \quad (2.2)$$

$$F_2: 255\leftrightarrow 0,\ 1\leftrightarrow 2,\ 3\leftrightarrow 4,\ \dots,\ 253\leftrightarrow 254,\ 255\leftrightarrow 0. \quad (2.3)$$

При застосуванні фліпінгу до групи отримують перетворену групу пікселів. Далі, всі групи пікселів розділяються на класи в такий спосіб:

(1) Регулярні групи: $G\in R \Leftrightarrow f(F(G)) > f(G)$,

(2) Сингулярні групи: $G\in S \Leftrightarrow f(F(G)) < f(G)$,

(3) Невикористовувані групи: $G\in U \Leftrightarrow f(F(G)) > f(G)$.

Метод засновується на припущені, що незаповнений контейнер, має такий вигляд:

$$RM \cong R-M \text{ та } SM \cong S-M. \quad (2.4)$$

Виходячи з припущення, якщо ми застосуємо F_{-1} до зображення, його розподіл буде відповідати розподілу, отриманому за допомогою F_1, де значення пікселів зміщені на один. У випадку нормального зображення (без прихованого повідомлення) різниця між групами не має бути значущою. Проте якщо існує велика розбіжність між цими значеннями, це може вказувати на використання LSB-стеганографії в найменших бітах зображення [21, с. 96].

Наступним є метод аналізу розподілу пар значень на основі критерію хі- квадрат. Проводиться дослідження гістограми, яка базується на компонентах зображення, та оцінюється розподіл сполучних значень цієї гістограми. У випадку BMP-файлів сполучні значення створюються на основі величин пікселів. Щодо JPEG, ці значення базуються на квантованих коефіцієнтах дискретного косинусного перетворення, які мають відмінності в найменшому біті. Молодші біти у зображеннях не мають випадкового характеру. Частоти двох суміжних елементів контейнера зазвичай суттєво відхиляються від середнього значення цих елементів. У «чистому» зображенні випадки, коли частоти елементів з показниками 2N та 2N + 1 майже однакові, зустрічаються нечасто. Але при вбудовуванні даних ці частоти наближаються одна до одної або навіть стають ідентичними. Суть атаки на основі хі-квадрат полягає у виявленні цих схожих

частот та визначенні імовірності втручання, враховуючи близькість частот парних та непарних елементів досліджуваного контейнера. Цей метод характеризується послідовним вивченням всього зображення та акумуляцією частот елементів. Статистичні атаки, засновані на критерії хі-квадрат, зазвичай не можуть виявити стегоканал, якщо контейнер заповнений лише на 50% або менше, особливо коли приховані дані розподілені рівномірно по контейнеру.

Таким чином, метод LSB є вразливим до багатьох типів атак і рекомендується до використання при чистому каналі передачі даних без шумів. Для розробки стеганографічних програм на основі даного методу необхідна його модифікація та застосування додаткових засобів захисту. Проте перевага даного методу є в тому, що він підходить для стеганографічних програм з різними типами контейнерів, таких як цифрові зображення, аудіофайли та відеофайли.

2.3 Математична модель стеганографічного перетворення

Математична модель. Математично процес стеганографічного перетворення можна описати наступними залежностями:

$$E : C\times M \rightarrow S \quad (2.5)$$

$$D : S \rightarrow M \quad (2.6)$$

$S = \{(c_1, m_1), (c_2, m_2), \dots, (c_q, m_q)\} = \{s_1, s_2, \dots, s_q\}$ – множина заповнених контейнерів. Залежність (2.5) описує процес приховування інформації, залежність (2.6) – витягування прихованої інформації. Однією із обов'язкових умов при цьому є відсутність «перетину», тобто якщо $m_a \neq m_b$ (причому $m_a, m_b \in M$, а $(c_a, m_a), (c_b, m_b) \in S$, то $E(c_a, c_m) \cap E(c_b, m_b) = \emptyset$.

Стегосистему можна представити як сукупність ∑(C,M,S,E,D) – контейнерів, повідомлень та перетворень, що їх зв'язують. Завжди контейнери с обираються таким чином, щоб заповнений контейнер майже не відрізнявся від порожнього контейнера. Стегосистему можна вважати надійною, коли *sim[c, E (c, m)*=1 (де sim – функція подібності). Контейнер може обиратися двома

способами: довільно (сурогатний метод) та підбором найбільш придатного у конкретному випадку контейнера, який зміниться найменше при перетворенні. В останньому випадку контейнер обирається виходячи із умови:

$$c = \max \text{sim}[x,E(x,m)] \quad (2.7)$$

В будь-якому випадку пряме та зворотне перетворення (E та D) мають відповідати одне одному та підлягати умові, що незначне викривлення контейнера (на величину δ) не має призводити до викривлення прихованої інформації:

$$E(c,m) \approx E(c + \delta,m) \text{ або} \quad (2.8)$$

$$D[E(c,m)] \approx D[E(c + \delta,m)] = m \quad [28, \text{с. } 14]. \quad (2.9)$$

Алгоритм розподілу повідомлення по контейнерах в розробленій нами програмі базується на пропорційному підході. У разі використання в одному наборі кількох зображень різними за розміром, в менший контейнер програма розподілить меншу кількість інформації, відповідно, в більшу – більший об'єм інформації. Формула (2.10) описує знаходження коефіцієнту пропорційності, формула (2.11) визначає довжину підрядка для контейнера.

$$k = \frac{\sum_{i=1}^{n} l_i}{l_t} \quad (2.10)$$

$$l_{t_i} = \frac{l_i}{k} \quad (2.11),$$

де n – кількість контейнерів, l_i – максимально допустима довжина повідомлення i–му контейнері, l_t– довжина вихідного повідомлення, l_{t_i} – довжина частини повідомлення в i–му контейнері, k – коефіцієнт.

У заповненому контейнері 15 пікселів першого рядка містять системну інформацію, а саме: наявність прихованого повідомлення, порядковий номер частини повідомлення, загальна кількість частин вихідного повідомлення, довжина прихованого повідомлення в контейнері. Максимально допустимий

об’єм повідомлення для одного контейнеру розраховується по кількості його пікселів.

Набори даних. Вхідними даними для стеганографічної програми обрані:

– зображення форматів BMP і PNG в якості контейнерів, які приховають інформацію:

– текстові повідомлення – інформація, яку необхідно приховати у зображеннях.

Формати зображень BMP і PNG мають низку особливостей, для використання в програмах стеганографії на основі методу LSB:

– BMP – це формат без втрат, у якому дані зображення зберігаються у вигляді послідовності пікселів. Його структура прямолінійна, що полегшує роботу з ним;

– відсутність стиснення в BMP означає, що після впровадження інформації немає ризику її втрати через будь-яке перетворення або стиснення;

– в BMP можна легко перетворювати пікселі та біти, особливо до молодші біти, що ідеально підходить для LSB;

– незважаючи на стиснення, PNG зберігає всі піксельні дані, що дає змогу впроваджувати інформацію в молодші біти.

Суттєвий вплив на надійність стегосистеми та можливість виявлення факту передавання прихованого повідомлення має вибір контейнера. Тому для гарантії результату слід керуватися наступними критеріями вибору контейнерів:

– відмова від загальновідомих зображень як контейнера, як, наприклад, картини «Джоконда»;

– відмова від використання в якості контейнера зображень, конвертованих з JPEG-формату у формат BMP;

– отримання зображення за допомогою фотоапарата або сканера, а не за допомогою графічних редакторів;

– великий розмір контейнера;

– зашумленість;

– відсутність плавних переходів і монотонних областей;

– «строкатість»;

– значна кількість перепадів яскравості;

– наявність великої кількості пікселів, відтінки кольорів яких погано розрізняються оком людини (зелений, жовтий).

Наведені вище параметри в достатній мірі відображають усі характеристики, які потрібні для створення контейнера методом LSB, стійкого до візуального стеганоаналізу.

Криптографічні алгоритми. Для забезпечення більшої стійкості до атак та надійності приховування інформації задіяні криптографічні алгоритми. Криптографічний захист являє собою шифрування – перетворення інформації для приховування її від сторонніх осіб. Першочерговою метою шифрування вважається засекречування від несанкціонованого доступу даних, що досягається за допомогою застосування спеціального ключа. За допомогою шифрування реалізуються аспекти безпеки даних, такі як конфіденційність, цілісність, доступність.

Алгоритми шифрування поділяють на дві основні групи: симетричні та асиметричні. Основна відмінність між ними полягає у використанні ключів. У разі симетричного шифрування застосовується один ключ, тоді як асиметричне шифрування містить два різні, але взаємопов'язані ключі. Ця технологічна особливість незважаючи на простоту, формує значні функціональні відмінності між обома формами шифрування та визначає методи їхнього застосування.

Асиметричне шифрування знаходить своє застосування в ситуаціях, коли необхідно зашифрувати і розшифрувати повідомлення або пакети даних для безлічі користувачів. Це особливо актуально у випадках, де швидкість і обчислювальна потужність не є пріоритетом. Прикладом такого використання може бути зашифрована електронна пошта, де відкритий ключ застосовується для

шифрування повідомлень, а приватний – для їхнього розшифрування. Проте на поточному етапі розвитку асиметричних алгоритмів існують певні обмеження. Наприклад, тривалість процесу шифрування значно збільшується, виходячи за часові рамки, встановлені для симетричного шифру. Крім того, необхідність у створенні довших ключів виникає для забезпечення стійкості шифру.

Що стосується симетричного шифрування, як було підкреслено раніше, воно базується на одному ключі, який використовується двома або більше користувачами. Цей ключ виконує функцію як для шифрування, так і для дешифрування відкритого тексту, що містить повідомлення або частину закодованих даних. Процес шифрування включає в себе обробку відкритого тексту (введення) алгоритмом шифрування, відомим як шифр. Шифр, своєю чергою, породжує зашифрований текст (висновок).Схема симетричного шифрування зображена рис.2.5.

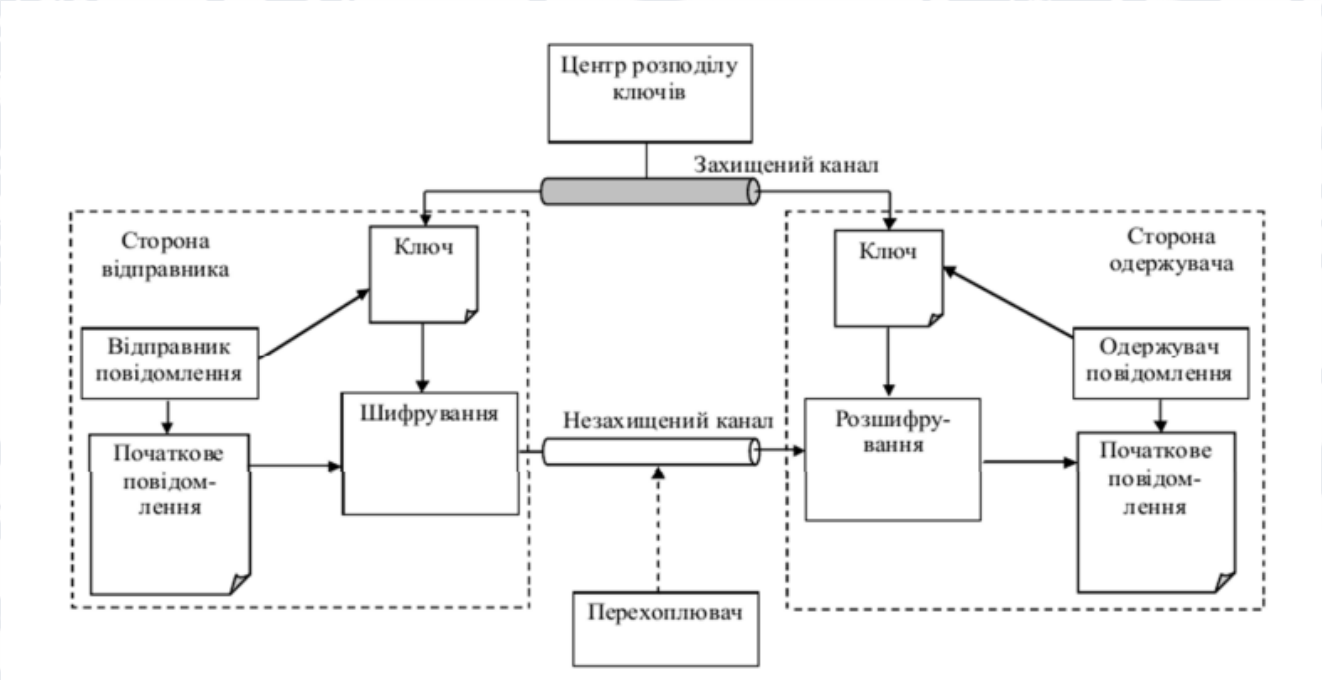


Рисунок 2.5 – Схема симетричного шифрування

Джерело: [6].

Симетрична схема шифрування є досить надійною, де єдиний спосіб отримати доступ до зашифрованої інформації – використати відповідний ключ

для розшифрування. Процес розшифрування в основному перетворює зашифрований текст назад у відкритий. Забезпечення безпеки симетричного шифрування полягає у складності випадкового вгадування ключа. Ключі з довжиною 256 біт зазвичай розглядаються як високо-захищені й теоретично стійкі до атак методом «грубої сили» за використання квантового комп'ютера.На сьогодні дві найпоширеніші симетричні схеми шифрування ґрунтуються на блоковому і потоковому шифрі. Блокові шифри групують дані в блоки заздалегідь встановленого розміру, водночас кожен блок шифрується відповідним ключем і алгоритмом шифрування (наприклад, 128-бітний текст перетворюється на 128-бітний зашифрований текст). У потоковому шифрі дані відкритого тексту шифруються не блоками, а інкрементом в 1 біт (1-бітний текст перетворюється на 1-бітний зашифрований текст).

Симетричні алгоритми мають ряд переваг, серед яких високий рівень безпеки, швидкість шифрування та дешифрування. Відносна простота симетричних систем також є логістичною перевагою, оскільки вони вимагають меншої обчислювальної потужності, ніж асиметричні. Крім того, безпека, що забезпечується симетричним шифруванням, може бути поліпшена простим збільшенням довжини ключа. Для кожного окремого біта, що додається до довжини симетричного ключа, складність злому шифрування за допомогою атаки методом перебору, експоненціально збільшується.

У розробленому програмному забезпеченні повідомлення шифрується за допомогою стандартної бібліотеки для криптографічних операцій System.Security.Cryptography. Реалізується функцією формування ключа на основі пароля (PBKDF2) за допомогою генератора псевдовипадкових чисел HMACSHA1. Використовується клас RFC2898DeriveBytes. RFC 2898 містить методи для створення ключа і вектора ініціалізації з пароля і солі. Є можливість використовувати PBKDF2, функцію успадкування ключів на основі пароля, щоб отримати ключі за допомогою псевдовипадкової функції, яка дозволяє

створювати ключі практично необмеженої довжини. Клас RFC2898DeriveBytes можна використовувати для створення похідного ключа з базового ключа та інших параметрів. У функції успадкування ключа на основі пароля базовий ключ є паролем, а інші параметри – значенням солі та кількістю ітерацій. Використання RFC2898 дає ряд переваг в безпеці, захисті від атак та універсалізації. RFC2898 надає можливість налаштування кількості ітерацій хешування, що дає змогу збільшувати складність процесу хешування залежно від необхідного рівня безпеки. Оскільки RFC 2898 є стандартом, його реалізації та застосування забезпечують сумісність між різними системами та платформами. Шляхом збільшення числа ітерацій PBKDF2 може запобігати атакам за словником, збільшуючи час, необхідний для перевірки кожного можливого пароля. PBKDF2 може також використовуватися для генерації ключів шифрування з паролів для захисту даних, що робить його корисним інструментом для забезпечення безпеки інформації. Таким чином, перелічені RFC 2898 та методу PBKDF2 є ефективними в забезпеченні безпеки зберігання паролів та інших конфіденційних даних.

Висновки до розділу 2

У другому розділі поставлено завдання на розробку стеганографічної програми. Досліджений метод LSB та види його стегоаналізу, встановлені основні недоліки методу.

Метод LSB, вразливий до різноманітних видів атак, рекомендується для використання лише в умовах чистого каналу передачі даних, позбавленого будь-якого впливу шуму. Таким чином, для створення стеганографічних програм на основі цього методу, необхідно ввести модифікації та використовувати додаткові засоби захисту. Описано математичну модель та задіяні для додаткового захисту криптографічні алгоритми.

Наборами даних для стеганографічної програми є цифрові зображення форматів BMP і PNG, що можуть слугувати контейнерами, та текстові повідомлення – інформація, яку програма буде приховувати в контейнерах. Перелічені параметри для вибору оптимальних контейнерів.

РОЗДІЛ 3
ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1.Архітектура програми

Стеганографічна програма містить дві основні частини, кожна з них поділена на декілька функціональних підсекцій. Основним елементом є головне вікно програми, яке водночас є частиною системного компонування. Така конфігурація надає змогу легко та інтуїтивно взаємодіяти з програмою, оскільки модулі організовані у такій послідовності, яку зазвичай використовують для вирішення подібних завдань.

Програма розроблена шляхом організації архітектури, яка містить інтерфейс користувача та обчислювальний прошарок (рис.3.1). Для реалізації моделі програми було розроблено алгоритм, що відображає послідовність дій, необхідних для виконання. Детальна блок-схема роботи програми наведена в додатку А. Для приховування повідомлень в зображеннях виконуються наступні кроки: вибір та завантаження пустих контейнерів, введення повідомлення, введення ключа шифрування, вибір папки для завантаження модифікованих контейнерів, розділення його для вбудовування в різні контейнери, запис частин повідомлення в контейнери та збереження заповнених контейнерів.

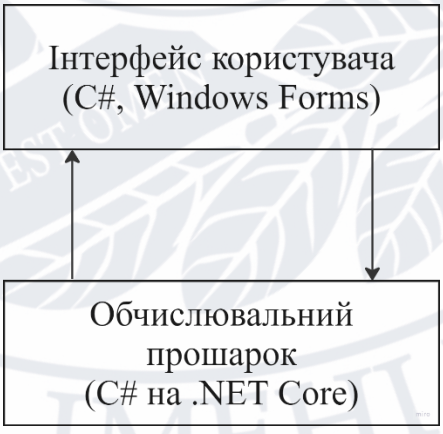


Рисунок 3.1 – Архітектура програми

Джерело: складено автором.

Другою важливою функцією програми є вилучення з контейнерів повідомлення, яке попередньо в ній було приховано. Для отримання результату користувачу необхідно завантажити всі контейнери, які містять частини зашифрованого повідомлення, адже навіть без одного, програма не дасть повного повідомлення, ввести ключ. Відповідно, без наявності ключа або з некоректним ключем, вилучення вихідного повідомлення буде неможливим. Виконання описаного етапу проходить в наступній послідовності: вибір заповнених контейнерів, введення ключа шифрування, отримання даних з контейнеру, валідація даних. Останнім етапом є отримання користувачем прихованого повідомлення. Якщо при перевірці контейнерів програма не виявила ні одного заповненого з наявних, користувач отримує повідомлення про відсутність прихованої інформації. Операція продовжується, якщо на попередньому етапі відбулась успішна валідація даних – програма провела перевірку на визначення хоча б одного заповненого контейнеру із всіх завантажених. Перед виведенням прихованого повідомлення програма проводить дешифрування тексту.

У процесі проектування програми використана модель багатошарової архітектури з урахуванням особливостей впровадження та масштабування у майбутньому. Функціональна область розділена на шари, що дає можливість працювати з кожним окремо. Програма містить UI module, Crypto module та Stego module (табл.3.1), адже поділену на модулі програму легше розуміти та супроводжувати. Кожен з трьох модулів реалізує конкретний функціонал. UI відповідає за взаємодію програмного забезпечення з користувачем та містить класи Main, EventsService, MessageBoxHelper, ValidationHelper. Основним його завданням є надання інтерфейсу та спрощення процесу роботи користувача з програмою. Crypto модуль відіграє ключову роль у забезпеченні безпеки та конфіденційності даних, відповідає за шифрування та дешифрування повідомлень, містить клас EncryptionHelper. Stego відповідає за функціонал запису повідомлень в контейнер та вилучення прихованих повідомлень з

контейнеру. До нього відносять класи ImageLsbService, StringExtension. Опис методів кожного класу наведений в Додатку Б.

З метою підвищення якості та надійності програмного забезпечення в процесі його розробки використані архітектурні патерни. Клас EventsService реалізований за допомогою патерну Command. Даний поведінковий патерн перетворює запити в об’єкти, що дає змогу передавати їх в якості аргументів під час виклику методів, додавати запити до черги, вести логування і забезпечувати можливість скасування операцій. У роботі програми патерн Command дозволяє прибрати пряму залежність між об’єктами, які викликають операції, та об’єктами, що їх виконують; дозволяє реалізовувати відміну та повторення дій; надає можливість впровадження відкладеного запуску операцій та формувати складну команду з набору простих команд.

Для класу, що містить методи запису та читання повідомлень контейнера використаний патерн Strategy [55]. Даний поведінковий патерн визначає сімейство схожих алгоритмів і розміщує кожен із них у власний клас, після чого алгоритми можна взаємо замінювати безпосередньо під час виконання програми. Не дивлячись на те, що використання патерну Strategy ускладнює програму за рахунок наявності додаткових класів, перевага його використання полягає у спрощеній заміні алгоритмів та заміні наслідування делегуванням.

Інтерфейс користувача створений мовою програмування C# із застосуванням інтерфейсу програмування додатків Windows Forms. Основними елементами керуванням інтерфейсу є наступні:

– form – основний компонент інтерфейсу програми, де розміщені усі додаткові складові;

– button – кнопка, елемент інтерфейсу, при натисканні якого починають виконуватись задані функції в програмі;

– label – елемент інтерфейсу, який демонструє інформацію, але не дозволяє її змінювати на формі;

– textbox – елемент інтерфейсу, який відображає, модифікує та демонструє інформацію відповідно до функціонування програми;

– richtextbox – елемент інтерфейсу, який фіксує, коригує та показує дані в залежності від роботи програми.

Таблиця 3.1 – Опис спроектованих модулів програми

Модуль	**Клас**	**Опис**
1	2	3
UI module	Main	Клас, який відповідає за відображення форми. Містить методи обробки дій на формі.
	EventsService	Клас, що містить функціонал для поділу повідомлення на підрядки, для взаємодії з іншими модулями програми.
	ValidationHelper	Клас, що відповідає за валідацію даних.
	MessageBoxHelper	Клас, що містить загальні методи для взаємодії над формою.
Crypto module	EncryptionHelper	Клас, що відповідає за шифрування/дешифрування повідомлень.
Stego module	ImageLsbService	Клас, який містить методи для запису/читання повідомлень контейнера.
	StringExtension	Клас, який містить методи для конвертації/деконвертації символів у/з бінарного коду.

Джерело: складено автором.

Обчислювальна частина створена мовою програмування C# за допомогою фреймворку .NET Core. Детальне обґрунтування вибору інструментів буде надано нижче.

3.2 Засоби розробки програмного забезпечення

Програма функціонує в середовищах ОС сімейства Windows. Для виконання завдання була використана мова програмування C# та платформа .NET Core.

.NET Core – це платформа з відкритим вихідним кодом для створення різних типів додатків, як десктопних, так і веб-версій. Інсталюється на комп’ютерах з операційною системою Microsoft Windows. Була обрана для використання в процесі розробки програми через наступні переваги:

– можливість взаємодіяти з іншими програмами, оскільки в перспективі може виникнути необхідність інтеграції з іншими застосунками;

– наявність спільного середовища виконання, усі програми на .NET Core працюють у конкретній віртуальній машині. Це гарантує, що всі програми будуть мати однорідну поведінку у відношенні до управління пам’яттю, безпеки та реагування на помилки;

– .NET Core розроблено неприв’язаною до платформи, отже вона є крос- платформеною;

– оскільки .NET Core містить так звану спільну систему типів CommonTypeSystem, бібліотеки класів, написані для Common Language Runtime однією мовою, можуть використовуватись в проєктах, що розроблені іншою мовою.

Вибір мови програмування C# обумовлений початковою орієнтацією на безпеку коду, просте використання, розширеною підтримкою подійно-орієнтованого програмування, уніфікованою системою типізації, найбільшою і тісною інтеграцією з .NET Core. Через велике розмаїття синтаксичних конструкцій і можливості працювати з платформою .NET Core, C# дає змогу швидше, ніж будь-яка інша мова, розробляти програмні рішення. Простота є ключовим фактором у розробці програмного продукту, оскільки вона дає змогу ефективно створювати функціональні та високопродуктивні додатки в короткі терміни. Цьому сприяють унікальні конструкції мови та специфічний синтаксис, які максимально органічно втілюють поставлені цілі.

Для реалізації програми було обрано інструмент VisualStudio 2022. Дане середовище розробки забезпечує мінімізацію синтаксичних помилок завдяки перевірці на стадії компіляції. Також інструмент надає зручні функції для роботи з WindowsForms, дозволяючи створити інтуїтивний користувацький інтерфейс з використанням базових елементів управління. Windows Forms забезпечує можливість візуального проектування, що дає змогу переглядати та вносити зміни

до інтерфейсу прямо в середовищі розробки. Ця функціональність спрощує весь процес створення, даючи змогу швидко тестувати і вносити необхідні корективи в дизайн користувацького інтерфейсу.

3.3 Результати дослідження

Для коректного функціонування розробленого програмного засобу потрібна наявність певної технічної та програмної конфігурації пристроїв. Програма розроблена для використання на персональному комп'ютері. Далі перелічено необхідні параметри пристроїв: достатній обсяг оперативної пам'яті, не менше 4 ГБ, тактова частота процесора від 2 ГГЦ. Інші елементи комп'ютера можуть мати різні характеристики, оскільки вони не мають значного впливу на функціонування програми. Також на пристрої має бути встановлена ОС Windows, від 10 версії та .NET Core не менше 7.0.14.

Для початку використання програмного засобу необхідно запустити виконуваний файл. Після запуску програми користувачу стає доступне головне вікно (рис. 3.2). Ліворуч розташований модуль для процесу приховування повідомлень, праворуч – для вилучення прихованих повідомлень.

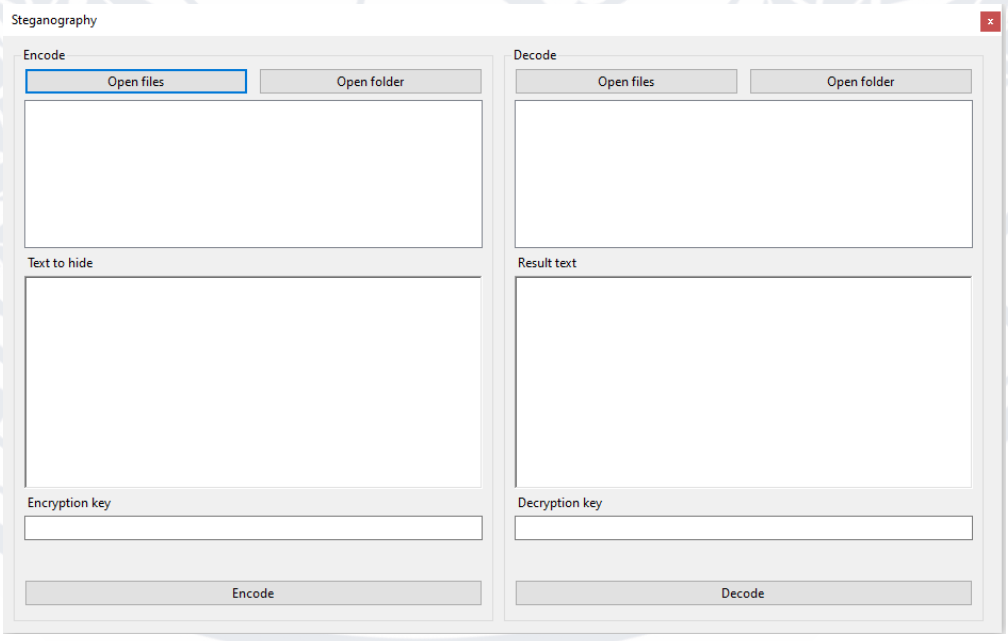


Рисунок 3.2 – Головний екран програми

Головний екран містить наступні елементи управління:

– кнопка «Open files» дозволяє обрати кілька або одне зображення;

– кнопка «Open folder» дозволяє обрати конкретну папку, де знаходяться всі зображення (пусті або заповнені контейнери);

– listbox, розташований під названими вище елементами, представляє собою елемент управління для відображення списку обраних для роботи контейнерів;

– кнопка «Encode» запускає процес приховування тексту в контейнери;

– кнопка «Decode» запускає процес вилучення прихованого повідомлення;

– textbox «Text to hide» та «Result text» – поля для введення текстового повідомлення та виведення вилученого повідомлення відповідно;\

– «Encryption key» та «Decryption key» поля для введення ключів шифрування.

Розглянемо результати тестування програми. У розробленому програмному застосунку приховаємо в одне зображення PNG (1900×1200 пікселів) фрагмент тексту на 100 000 символів. Для початку використаємо візуальний метод аналізу. Він є найпростішим методом стегоаналізу, базується на здатності зорової системи людини аналізувати зорові образи та виявляти суттєві відмінності в зображеннях, які можна зіставити. Як бачимо на рис.3.3, візуально порожній контейнер не відрізняється від заповненого.

Рисунок 3.3 – Порівняння пустого та заповненого контейнерів

Варто зауважити, що такий візуальний метод стегоаналізу, так само як і будь-які інші візуальні методи, виявляється малоефективним у протистоянні приховування інформації в зображеннях, які спеціально націлені на максимальну візуальну непомітність.

Для виявлення факту існування прихованого каналу передачі інформації ефективними є дві атаки, а саме хі-квадрат та RS-атака. Ці підходи засновані на статистичних методах і застосовуються для виявлення прихованих повідомлень у піксельних областях. Особливо ефективні вони в контексті стандартного послідовного методу вбудовування, що використовує найменш значущий біт. За допомогою різних методів відбувається пошук відхилення в розглянутому контейнері від оригінального. До переваг цієї групи методів належить необмежена сфера застосування, що важливо, якщо врахувати, що не завжди відомий метод стеганографії. Для виконання цього завдання було використано онлайн-ресурс під назвою Lsbtools. Щоб отримати можливість тестувати зображення атаки на контейнери з вбудованим повідомленням, слід перейти за посиланням [49], завантажити контейнер та вибрати тип атаки.

Проведемо хі-квадрат тест. Під час хі-квадрат атаки зображення розбивається на блоки, що не перетинаються, і для кожного блоку обчислюється значення хі-квадрат, яке відображає ступінь відхилення фактичного розподілу значень пікселів від очікуваного. Якщо це відхилення значне, то це може свідчити про наявність прихованої інформації. Результати хі-квадрат атаки в даному випадку хибні, програма не виявила прихованого тексту, це свідчить про високу стійкість методу приховування (рис.3.4).

Тепер проведемо RS-атаку. Одна із проблем, яку ми виявили при дослідженні існуючих стеганографічних програм, це зменшення стійкості до атак при приховуванні повідомлення великого об’єму. Рішенням в нашому програмному застосунку стала функція приховування об’ємного повідомлення в кількох контейнерах. При виконанні операції програма розподіляє частини

повідомлення пропорційно розмірам контейнера. Протестуємо рішення, взявши 10 наборів зображень з однаковими властивостями, кількість контейнерів в кожному наборі більше на 1, ніж в попередньому.

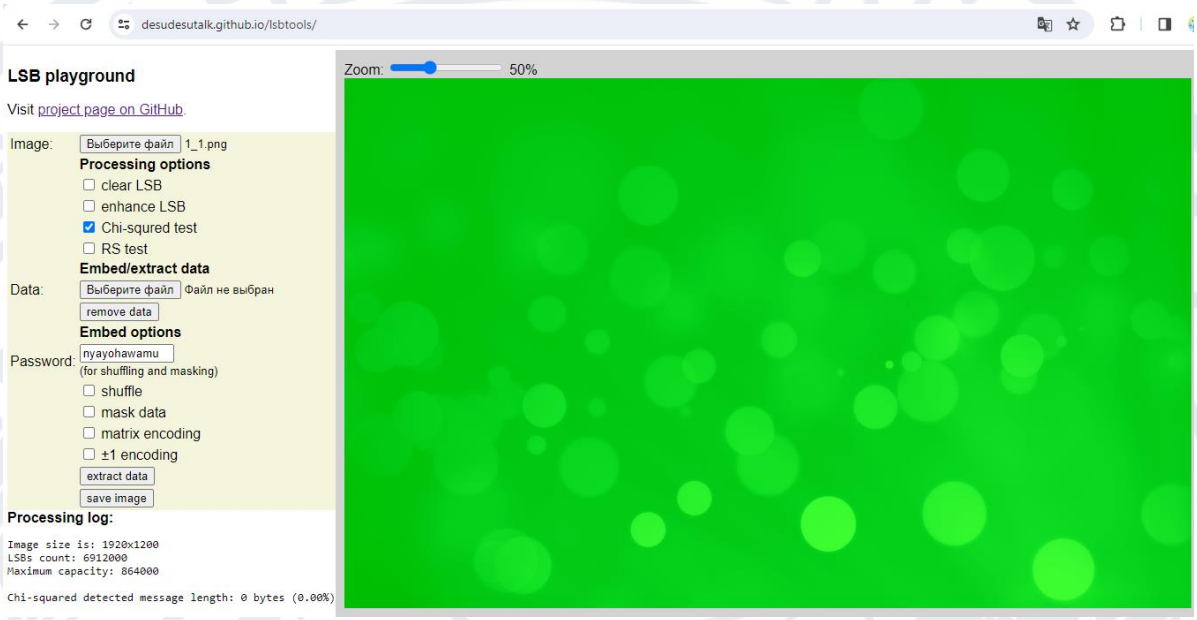


Рисунок 3.4 – Результати хі-квадрат атаки на заповнений контейнер

У таблиці 3.2 наведені результати RS-атаки 10 наборів контейнерів, в яких приховано повідомлення на 100 000 символів.

Таблиця 3.2 – Результати RS-атаки

№ набору	Кількість контейнерів у наборі	Сума знайдених байтів в наборі
1	1	62515
2	2	61998
3	3	60828
4	4	60627
5	5	60059
6	6	60025
7	7	58432
8	8	58145
9	9	58687
10	10	57756

З отриманих даних видно, що в наборі №10, де повідомлення приховано у 10 контейнерах в результаті RS-атаки знайдено значно менше символів, ніж у №1,

де все повідомлення приховано в одному зображенні. Вихідні дані по кожному протестованому контейнеру наведені в Додатку В.

Можемо зробити висновок, що модифікований алгоритм працює та збільшує можливість приховування великого об’єму повідомлення у не один, а цілу групу контейнерів. Отже, навіть після вилучення з контейнеру частини даних, у зловмисника не буде достатньо інформації для розкриття цілісного повідомлення.

Висновки до розділу 3

У третьому розділі роботи описані результати роботи над розробкою програмного забезпечення, наведені основні функції, продемонстрований інтерфейс та проведено тестування. Викладена інформація про засоби розробки та вимоги до програмного забезпечення.

Проведено демонстрацію функціоналу програми. Під час проведення тестів збоїв у програмного застосунку не відбулось. Обране рішення модифікувати алгоритм приховування повідомлення не в один, а в декілька контейнерів забезпечує надійнішу роботу програми.

Результати випробувань підтвердили коректність функціонування системи. Загалом, розроблений програмний застосунок є стабільним і відповідає встановленим вимогам.

ВИСНОВКИ

У магістерській роботі було розроблено програмне забезпечення для приховування повідомлень у мультимедіа контенті за допомогою методу стеганографії. Для цього був здійснений огляд існуючих стеганографічних програм, виявлені їх проблеми та сформовані підходи до вирішення.

Була розглянута теоретична основа та проблематика стеганографії. Проведений аналіз методів стеганографії, які використовуються у випадках приховування інформації у мультимедіа контенті. Наведена типова схема стегосистеми.

Наступним етапом стали розробка комп'ютерно-математичної моделі програми. Описано алгоритм роботи застосунку, архітектура та математична модель. До опису розробленої архітектури додана детальна блок-схема роботи програми. На цьому етапі також досліджений метод LSB, який було обрано як основу для приховування даних в контенті. На основі проведених досліджень було прийнято рішення про використання додаткових способів захисту вбудованої інформації в контейнери за допомогою криптографічних служб, а також модифікованого алгоритму, коли програма буде розподіляти повідомлення в одразу декілька завантажений контейнерів.

Під час перевірки роботи програмного застосунку стеганографії для цифрових контейнерів у формі зображення не виявлено помилок чи недоліків. Наведені результати RS-атаки та атаки хі-квадрат на стеганоконтейнери, в які було вкраплено інформацію.

У результаті роботи було досягнуто мети розробки програмного забезпечення для приховування інформації в файлах мультимедіа.

Отримано наступні наукові та практичні результати:

1. Розроблений алгоритм розподілу одного прихованого повідомлення у кілька контейнерів.

2. Реалізовані запропоновані алгоритми в програмі, в основі якої лежить метод LSB та додаткове шифрування за допомогою криптографічних служб.

Практичне значення отриманих результатів полягає у створенні програмного забезпечення для приховування повідомлень у файлах мультимедіа. Поставлені нами завдання були виконані.

Результати даної роботи можуть бути корисними для організації захисту передачі прихованих повідомлень, а функціонал розробленого нами програмного забезпечення може бути розширений до використання у якості контейнерів аудіо та відеофайлів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Головін М., Головіна Н. Навчальний приклад маскування інформації в акустичному сигналі. *Наукові записки Бердянського державного педагогічного університету.* 2021. № 2. С. 203–210.

2. Денисюк В. О. Стеганографічний алгоритм захисту даних з використанням файлів зображень. *Ефективна економіка*. 2017. № 5. URL: http://www.economy.nayka.com.ua/?op=1&z=5584 (дата звернення: 02.10.2023).

3. Дорожан А.В., Астраханцев А.А., Вовк О.О. Исследование характеристик методов скрытия на основе НЗБ на фоне аддитивного шума. *Вісник національного технічного університету «ХПІ»*. 2012. № 18. С. 37–40.

4. Дослідження статистичних методів стегоаналізу цифрових зображень / В. І. Чистов та ін. *Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України*. 2023. № 1. С. 130–135.

5. Задирака В. К., Кудін О. М., Швідченко І. В. Стеганографія в хмарних інформаційно-комунікаційних системах. *Компьютерная математика*. 2014. № 1 (19).

6. Захист інформації в телекомунікаційних системах : навч. посіб. / Г. Ф. Конахович та ін. Київ : НАУ, 2009. 380 с.

7. Іванов В.Г. Захист інформації засобами комп'ютерної стеганографії. *Безпекове інноваційне суспільство: взаємодія у сфері правової освіти та правового виховання* : матеріали міжнар. інтернет-конф, м. Харків, 25 трав. 2016 р. Харків, 2016. С. 53–56.

8. Інформаційна безпека в комп'ютерних мережах : навч. посіб. / Смірнов О. А. та ін. Кропивницький : Вид. Лисенко В. Ф., 2020. 295 с.

9. Катаєва Є.Ю., Ребріков А.Г. Актуальність використання прихованої передачі даних у відеофайлах. *Управління розвитком складних систем*. 2021. № 46. С. 48–54.

10. Конахович Г. Ф., Прогонов Д. О., Пузиренко О. Ю. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних. Київ : «Alex Print Centre», 2018. 558 с.

11. Корольов В. Ю., Поліновський В. В., Герасименко В. А. Визначення можливостей RS-стегоаналізу для дослідження статистичних властивостей зображень. *Вісник Хмельницького національного університету*. 2010. № 4. С. 102–110.

12. Корольов В. Ю., Ходзінський О. М. JPEG стеганографія на базі теоретико-чисельних перетворень. *Вісник Хмельницького національного університету*. 2014. № 1. С. 61–69.

13. Кошкіна Н.В. Спектральні методи комп’ютерної стеганографії та методи стеганоаналізу з навчанням і класифікацією : автореф. дис. …д. техн. наук: 05.13.21. Київ, 2016. 41 с.

14. Кузнєцов О.О., Євсеєв С.П., Король О.Г. Стеганографія : навч. посіб. Харків : ХНЕУ, 2011. 232 с.

15. Кущ С.М., Луценко В.М., Прогонов Д.О. Виявлення прихованих повідомлень як складова комплексних систем захисту інформації. *Наук.-техн. журнал «Захист інформації»*. 2012. № 3(56). С. 65–71.

16. Кущ С.М., Прогонов Д.О. Алгоритм формування стеганограм на основі LSB-методу та його використання для оцінювання ефективності методів активного стегоаналізу. *Вісник Національного університету «Львівська політехніка»: «Автоматика, вимірювання та керування»*. 2013. № 774. С. 21–27.

17. Мельник С. В., Кондакова С. В. Світові тенденції розвитку цифрової стеганографії в контексті завдань забезпечення інформаційної безпеки

держави. *Актуальні проблеми управління інформаційною безпекою держави* : зб. матер. наук.-практ. конф., м. Київ. Київ, 2019. С. 134–138.

18. Навроцький Д.О. Методи комп'ютерної стеганографії. *Вісник Національного технічного університету України «КПІ»*. 2007. № 35. С. 105–108.

19. Різуненко А. Теорія та практика цифрової обробки зображень : монографія. Полтава : РВВ ПУСКУ, 2009. 195 с.

20. Римар П.В., Крохмалюк В.В. Атаки на стеганосистеми. Криптографічні атаки. *Матеріали наукової конференції професорсько-викладацького складу, наукових працівників і здобувачів наукового ступеня за підсумками науководослідної роботи за період 2019–2020 рр.*, м. Вінниця, квіт. – трав. 2021 р. Вінниця, 2021. С. 344–346.

21. Смірнов О.А., Мелешко Є.В. Дослідження методів стегоаналізу цифрових зображень. *Наука і техніка Повітряних Сил Збройних Сил України*. 2012. № 2(8). 92-99.

22. Сучасні методи квантової стеганографії / Г. Ф. Конахович та ін. *Науково-технічний журнал «Захист інформації»*. 2011. № 2. С. 5–9.

23. Стасюк О.І., Гнатюк С.О., Літош М.С. Сучасні стеганографічні методи захисту інформації. *Науково-технічний журнал «захист інформації»*. 2001. № 1. С. 1–7.

24. Тарасов Д.О., Мельник А.С., Голобородько М.М. Класифікація та аналіз безкоштовних програмних засобів стеганографії. *Інформаційні системи та мережі. Вісник НУ «Львівська політехніка»*. 2010. № 673. С. 365–374.

25. Терейковський І. А., Гнатюк С. О. Захист інформації в комп'ютерних системах : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2022. 135 с.

26. Хорошко В.О., Яремчук Ю.Є, Карпінець В.В. Комп'ютерна стеганографія: навчальний посібник. Вінниця : ВНТУ, 2017. 155 с.

27. Швідченко І. Аналіз програмного забезпечення зі стеганоаналізу. *Штучний інтелект*. 2012. № 3. С. 487–495.

28.Юдін О. К., Зюбіна Р. В., Фролов О. В. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів. *Радіоелектроніка та інформатика*. 2015. № 3. С. 13–21.

29.Юдін О. К., Корченко О. Г., Конахович Г. Ф. Захист інформації в мережах передачі даних. Київ : ТОВ «НВП» ІНТЕРСЕРВІС», 2009. 716 с.

30.Anderson R. J., Petitcolas F. A. P. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*. 1998. Vol. 16, no. 4. P. 474–481.

31.Anderson R. Stretching the limits of steganography : Lecture Notes in Computer Science. 1174th ed. Berlin : Springer, 1996. 364 p.

32.A Novel Approach of Image Steganography for Secure Communication Based on LSB Substitution Technique / S. Rahman et al. *Computers, Materials & Continua*. 2020. Vol. 64, no. 1. P. 31–61.

33.An overview of steganography techniques applied to the protection of biometric data / M. Douglas et al. *Multimed Tools App*. 2018. No. 77. P. 17333–17373

34.Ansari A. S., Mohammadi M. S., Parvez M. T. A Comparative Study of Recent Steganography Techniques for Multiple Image Formats. *I. J. Computer Network and Information Security*. 2019. No. 1. P. 11–25.

35. Curty M., Santos D. J. Quantum steganography. *2nd bielefeld workshop on quantum information and complexity, bielefeld*, Bielefeld, 12–14 October 2000. 2000. P. 12–14.

36. Digital image steganography: Survey and analysis of current methods / A. Cheddad et al. *Signal Processing*. 2010. Vol. 90, no. 3. P. 727–752.

37. Digital image steganography survey and investigation (goal, assessment, method, development, and dataset) / De Rosal Ignatius Moses Setiadi et al. *Signal Processing*. 2023. Vol. 206. URL: https://doi.org/10.1016/j.sigpro.2022.108908 (date of access: 02.10.2023).

38. Fridrich J. Steganography in Digital Media: Principles, Algorithms, and Applications. New York : Cambridge University Press, 2009. 437 p.

39. Haider TH. Salim ALRikabi, Hazim H. T. Enhanced Data Security of Communication System Using Combined Encryption and Steganography. *iJIM*. 2021. Vol. 15, no. 16. P. 144–157.

40. Hassan N. A., Hijazi R. Data hiding forensics. ScienceDirect. URL: https://www.sciencedirect.com/topics/computer-science/steganography-program (date of access: 09.10.2023).

41. High Capacity Image Steganography System based on Multi-layer Security and LSB Exchanging Method / R. S. Hameed et al. *International Journal of Advanced Computer Science and Applications*. 2022. Vol. 13, no. 8. P. 108–115.

42. Image Steganography: A Review of the Recent Advances / N. Subramanian et al. *IEEE Access*. 2021. Vol.9.URL: https://ieeexplore.ieee.org/abstract/document/9335027 (date of access: 02.10.2023).

43. Implementation of Hybrid Cryptography in Steganography for Augmented Security / V. Kalaichelvi et al. *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing*, Villupuram, 21–22 April 2023. Villupuram, 2023. P. 1–5. URL: https://ieeexplore.ieee.org/document/10151554 (date of access: 02.10.2023).

44. Improvement of Image Steganography Scheme Based on LSB Value with Two Control Random Parameters and Multi-level Encryption / M. H. Mahdi et al. *ICSET*. 2019. P. 1–14.

45. Keller J., Langsdorf S. Error Codes in and for Network Steganography. *Architecture of Computing Systems* : International Conference on Architecture of Computing Systems, Athens, 13–15 June 2023. 2023. P. 81–93.

46. Kirti, Sarowa P. Current trends in steganography. *International Journal of Computer Sciences and Engineering*. 2018. Vol. 6, no. 7. P. 879–882.

47. Kuhn M. G., Anderson R. J. Soft tempest: Hidden data transmission using electromagnetic emanations. *International Workshop on Information Hiding*. Berlin, 1998. P. 124–142.

48. Large-Capacity Image Steganography Based on Invertible Neural Networks / L. Shao-Ping et al. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. P. 10816–10825.

49. Lsbtools. URL: https://desudesutalk.github.io/lsbtools/ (дата звернення: 06.11.2023).

50. Mazurczyk W., Lubacz J., Szczypiorski K. Hiding Data in VoIP. *Proceedings of the 26th army science conference (ASC 2008)*, Orlando, 1–4 December 2008. Orlando, 2008.

51. Raggo M. T. StegSpy v2.1. URL: http://www.spy-hunter.com/stegspy (date of access: 01.10.2023).

52. Singh A. K., Singh J., Singh H. Steganography in Images Using LSB Technique. *International Journal of Latest Trends in Engineering and Technology*. 2015. Vol. 5, no. 1. P. 426–430.

53. Steganography online. URL: https://stylesuxx.github.io/steganography/ (date of access: 02.10.2023).

54. Stegdetect 0.6. Niels Provos. URL: https://www.provos.org/p/outguess-and-stegdetect-downloads/ (date of access: 01.10.2023).

55. Strategy. *Behavioral Patterns*. URL: https://refactoring.guru/design-patterns/strategy (date of access: 06.11.2023).

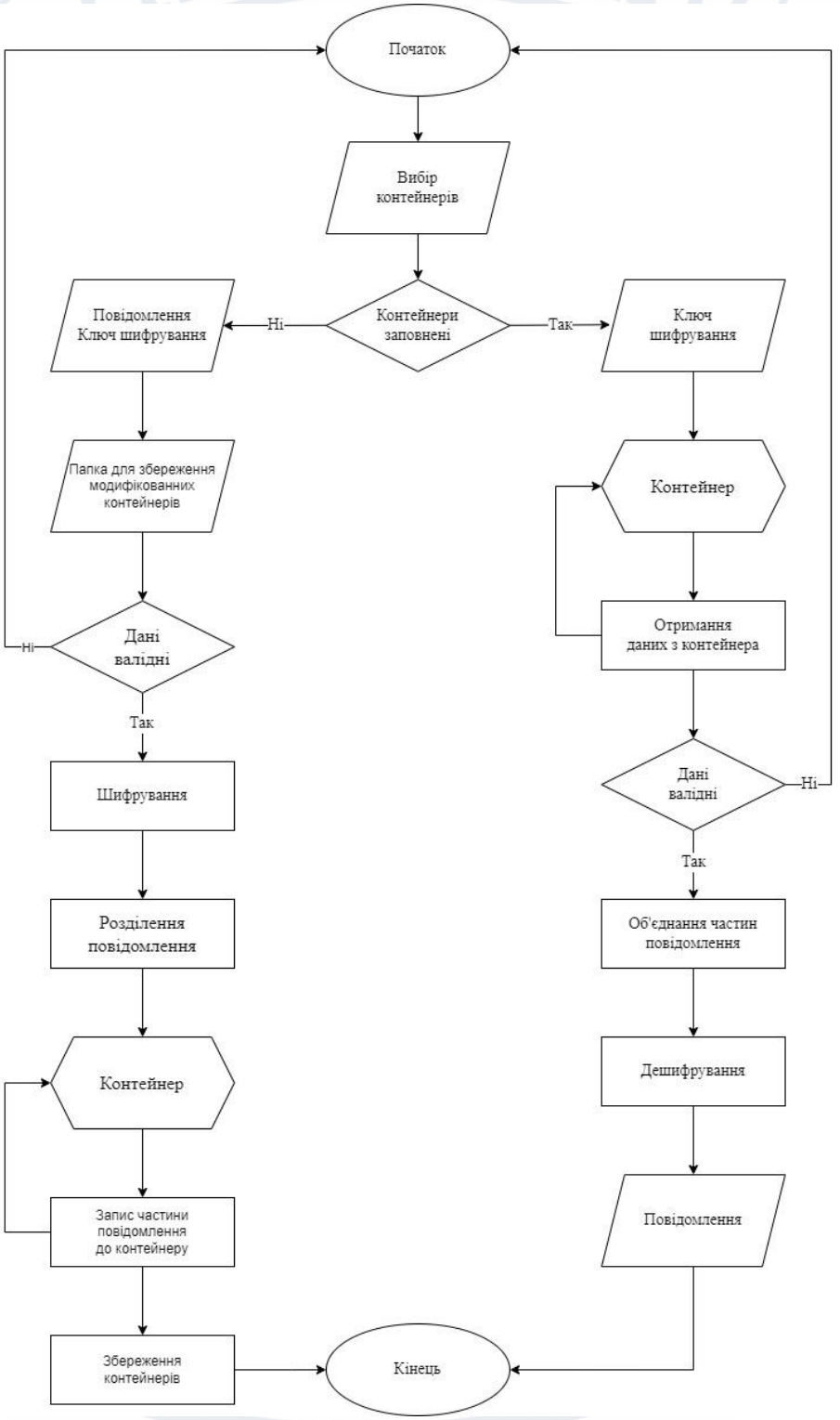
56. Yahya A. Steganography Techniques for Digital Images. Springer Cham, 2019. 122 p.

57. Zeki A. M., Ibrahim A. A., Manaf A. A. Steganographic software: analysis and implementation. *International journal of computers and communications*. 2012. Vol. 6, no. 1. P. 35–42.

ДОДАТКИ

ДОДАТОК А

Блок-схема роботи програмного забезпечення



ДОДАТОК Б

Опис використаних методів в програмному забезпеченні

Клас	Метод	Параметри	Опис
1	2	3	4
Main	Main		Конструктор, що створює форму
	buttonOpenFileForEncode_Click	object sender, EventArgs e	Метод викликає метод з EventsService
	listBoxFilesForEncode_KeyUp	object sender, KeyEventArgs e	Видалення одного з обраних контейнерів для запису повідомлення
	listBoxFilesForDecode_KeyUp	object sender, KeyEventArgs e	Видалення одного з обраних контейнерів для читання повідомлення
	buttonEncode_Click	object sender, EventArgs e	Метод викликає метод з EventsService
	buttonOpenFilesForDecode_Click	object sender, EventArgs e	Метод викликає метод з EventsService
	buttonDecode_Click	object sender, EventArgs e	Метод викликає метод з EventsService
	buttonOpenFolderForEncrypt_Click	object sender, EventArgs e	Метод викликає метод з EventsService
	buttonOpenFolderForDecode_Click	object sender, EventArgs e	Метод викликає метод з EventsService
MessageBoxHelper	ShowWarning	string message	Метод, що показує попередження на формі
	ShowSuccess	string message	Метод показує повідомлення про успішність виконання
EncryptionHelper	EncryptAesString	string value, string encryptionKey	Метод, що шифрує повідомлення з використанням введеного ключа
	DecryptAesString	string value, string encryptionKey	Метод, що дешифрує повідомлення з використанням введеного ключа
	GetKeyAndIv	string encryptionKey	Метод повертає функцію формування ключа на основі пароля (PBKDF2) за допомогою генератора псевдовипадкових чисел HMACSHA1

Продовження таблиці

1	2	3	4
StringExtensions	ToIntFromBinary	this char[] value	Перетворює цілочисельне число в двійковий код
	ToStringSymbol	this int number	Перетворює рядок у двійковий масив зазначеної довжини
	ToBinaryArray	this string value, int byteCount = 1	Перетворює рядок у двійковий масив зазначеної довжини
	ToBinaryArray	this int value, int byteCount = 1	Перетворює цілочисельне число в двійковий масив зазначеної довжини
	ToBinaryArray	this byte value, int byteCount = 1	Перетворює значення byte у двійковий масив зазначеної довжини
	NormalizeArray	char[] array, int byteCount	Нормалізує вихідний двійковий масив. Довжина масиву має бути кратна 8
EventService	GetFileExtension	string filePath	Метод, що повертає розширення файлу
	IsAllowedFile	string filePath	Метод, що перевіряє доступність файлу виступати в якості контейнера
	IsWrongParts	IEnumerable<SubstringDataModel> parts	Метод перевіряє можливість складання повного повідомлення з виділених частин
	GetContainers	ListBox listBox	Метод, що забезпечує відкриття файлів-контейнерів
	GetContainersFolder	ListBox listBox	Метод дає змогу вибрати папку, що містить контейнери
	ItemsKeyUpEvent	ListBox listBox, KeyEventArgs e	Метод дає змогу видалити один з обраних контейнерів
	HideMessage	RichTextBox richTextBoxForEncode, ListBox listBoxFilesForEncode, TextBox textBoxEncryptKey	Метод валідує вхідні дані, шифрує повідомлення з використання введеного ключа і записує повідомлення в контейнери

Продовження таблиці

1	2	3	4
EventService	GetMessage	ListBox listBoxFilesForDecode, TextBox textBoxDecryptKey, RichTextBox richTextBoxDecode	Метод валідує вхідні дані для вилучення повідомлення з обраних контейнерів, витягує повідомлення з контейнерів і заповнює поле на формі
ValidateHelper	IsInvalideToHide	RichTextBox richTextBoxEncode, ListBox listBoxFilesForEncode, TextBox textBoxEncryptKey	Метод валідації вхідних даних для приховування повідомлення
	IsInvalideToGet	ListBox listBoxFilesForDecode, TextBox textBoxDecryptKey	Метод валідації вхідних даних для вилучення повідомлення
ImageLbsSerivce	GetAllowedSymbolsCount	string filePath	Метод повертає кількість допустимих для запису в контейнер символів
	DivideMessage	string message, string key, string[] files	Метод ділить вихідне повідомлення на частини
	HideMessage	string filePath, string message, int number, int count, string destinationFolder	Метод, що записує службову інформацію в перший рядок контейнера і повідомлення в контейнер
	HideSymbol	Bitmap image, char[] symbols, params (int x, int y)[] pixels	Метод записує символ у вигляді доїчного масиву в зазначені пікселі
	HideMessage	string[] files, string message, string destinationFolder	Метод, що записує повідомлення в обрані контейнери
	HideMessage	Bitmap image, string message	Метод записує повідомлення в контейнер, починаючи з другого рядка
	SaveFile	Bitmap image, string originalFilePath, string destinationFolder	Зберігає заповнений контейнер у вказану папку
	GetMessage	string filePath	Метод дістає повідомлення з контейнера

Продовження таблиці

1	2	3	4
	GetSymbol	Bitmap image, params (int x, int y)[] pixels	Метод отримує символ у вигляді двійкового масиву із зазначених пікселів
	GetMessage	Bitmap image, int substringSymbolsCount	Метод отримує повідомлення вказаної довжини з контейнера
	GetFromPixel	Bitmap image, int x, int y	Метод отримує частину двійкового коду символу із зазначеного пікселя
	SetToPixel	Bitmap image, int x, int y, char rValue, char gValue, char bValue	Метод записує частину двійкового коду символу в зазначений піксель

ДОДАТОК В

Результати RS-атаки на набори заповнених контейнерів

Назва файлу	**№ набору**	**№ з/п в наборі**	**Кіл-ть символів в контейнері**	**Заповненість контейнеру, %**	**Сума знайдених байтів**	**Виявлено, %**
1	2	3	4	5	6	7
1_1.png	1	1	99999	8,68	62515	7,24
1_2.png	2	1	49991	4,34	30997	3,59
2_2.png		2	50008	4,34	31001	3,59
1_3.png	3	1	33334	2,89	20247	2,34
2_3.png		2	33334	2,89	20272	2,35
3_3.png		3	33331	2,89	20309	2,35
1_4.png	4	1	25004	2,17	15126	1,75
2_4.png		2	25004	2,17	15140	1,75
3_4.png		3	25004	2,17	15179	1,76
4_4.png		4	24987	2,17	15182	1,76
1_5.png	5	1	19999	1,74	11986	1,39
2_5.png		2	19999	1,74	12012	1,39
3_5.png		3	19999	1,74	12042	1,39
4_5.png		4	19999	1,74	12040	1,39
5_5.png		5	20003	1,74	11979	1,39
1_6.png	6	1	16667	1,45	9975	1,15
2_6.png		2	16667	1,45	10003	1,16
3_6.png		3	16667	1,45	9993	1,16
4_6.png		4	16667	1,45	10002	1,16
5_6.png		5	16667	1,45	10045	1,16
6_6.png		6	16664	1,45	10007	1,16
1_7.png	7	1	14284	1,24	8334	0,96
2_7.png		2	14284	1,24	8364	0,97

Продовження таблиці

1	2	3	4	5	6	7
3_7.png	7	3	14284	1,24	8374	0,97
4_7.png		4	14284	1,24	8326	0,96
5_7.png		5	14284	1,24	8312	0,96
6_7.png		6	14284	1,24	8331	0,96
7_7.png		7	14295	1,24	8391	0,97
1_8.png	8	1	12500	1,09	7240	0,84
2_8.png		2	12500	1,09	7278	0,84
3_8.png		3	12500	1,09	7247	0,84
4_8.png		4	12500	1,09	7275	0,84
5_8.png		5	12500	1,09	7266	0,84
6_8.png		6	12500	1,09	7296	0,84
7_8.png		7	12500	1,09	7255	0,84
8_8.png		8	12499	1,08	7288	0,84
1_9.png	9	1	11111	0,96	6487	0,75
2_9.png		2	11111	0,96	6528	0,76
3_9.png		3	11111	0,96	6516	0,75
4_9.png		4	11111	0,96	6546	0,76
5_9.png		5	11111	0,96	6491	0,75
6_9.png		6	11111	0,96	6509	0,75
7_9.png		7	11111	0,96	6570	0,76
8_9.png		8	11111	0,96	6549	0,76
9_9.png		9	11111	0,96	6491	0,75
1_10.png	10	1	9999	0,87	5760	0,67
2_10.png		2	9999	0,87	5770	0,67
3_10.png		3	9999	0,87	5768	0,67

Продовження таблиці

1	2	3	4	5	6	7
4_10.png	10	4	9999	0,87	5799	0,67
5_10.png		5	9999	0,87	5783	0,67
6_10.png		6	9999	0,87	5768	0,67
7_10.png		7	9999	0,87	5766	0,67
8_10.png		8	9999	0,87	5778	0,67
9_10.png		9	9999	0,87	5788	0,67
10_10.png		10	10008	0,87	5776	0,67