Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи
магістра
(ступінь вищої освіти)

на тему: Дослідження та розробка засобів демонстрації стеганографії та стегоаналізу

за освітньою програмою **Комп'ютерна інженерія**
зі спеціальності: **123 Комп'ютерна інженерія**
(шифр і назва спеціальності)

Виконав: студент **групи: КС2326**

	(підпис студента)	/ Олександр ДВЕРІС / (Ім'я ПРІЗВИЩЕ)
Керівник:	(підпис)	/ доцент, Денис ОСТАПЕЦЬ / (посада, Ім'я ПРІЗВИЩЕ)
Нормоконтролер:	(підпис)	/ доцент, Олег ЄГОРОВ / (посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу)	(підпис)	/ / (посада, Ім'я ПРІЗВИЩЕ)
(назва розділу)	(підпис)	/ / (посада, Ім'я ПРІЗВИЩЕ)
(назва розділу)	(підпис)	/ / (посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Дніпро – 2025 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
(faculty)

Department «Electronic computers»
(department)

Explanatory Note
to Master's Thesis
first (master's)
(higher education degree)

on the topic: Research and development of steganography and stegoanalysis demonstration tools
according to the educational program Computer Engineering

in the Speciality: 123 Computer Engineering
(speciality and its code)

Done by the student of the group: KC2326 / Olexander Dveris /
(name, surname)

Scientific Supervisor: / Associate Professor, Denys Ostapets/
(position, name, surname)

Normative controller : / Associate Profeessor, Oleg Yehorov/
(position, name, surname)

Supervisors

(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)

Dnipro – 2024

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: ЕОМ
Рівень вищої освіти: Другий (магістерський)
Освітня програма: Комп'ютерна інженерія
Спеціальність: 123 Комп'ютерна інженерія
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри ЕОМ
(підпис) Ігор ЧУКОВИЦЬКИЙ (Ім'я ПРІЗВИЩЕ)
Дата

З А В Д А Н Н Я

на кваліфікаційну роботу магістра
(ступінь вищої освіти)

студенту Дверісу Олександру Євгеновичу
(Прізвище, Ім'я По батькові)

1. Тема роботи: Дослідження та розробка засобів демонстрації стеганографії та стегоаналізу

Керівник роботи: Остапець Денис Олександрович, к.т.н., доцент
(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "21" 04 2024 р. № 333ст

2. Строк подання студентом роботи: 22.01.2024 р.

3. Вихідні дані до роботи:

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд та аналіз існуючих методів стеганографії	13.11.24	20%
2	Функції, режими роботи та інформаційна структура розроблюваних засобів	29.11.24	20%
3	Розробка програмного забезпечення	25.12.24	30%
4	Експериментальне дослідження ефективності застосування зрізу по молодшим бітам	14.01.25	25%
5	Реферат, вступ, висновки	17.01.25	5%
6	Подання кваліфікаційної роботи до кафедри	20.01.25	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	24.01.25	

Студент _____ Олександр ДВЕРІС
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Денис ОСТАПЕЦЬ
(підпис) (Ім'я ПРІЗВИЩЕ)

Міністерство освіти і науки України
Український державний університет науки і технологій

Відгук керівника
кваліфікаційної роботи магістра

Студент групи КС2326 Дверіс Олександр Євгенович
(шифр групи) (Прізвище, Ім'я, По батькові)

Тема випускної роботи: Дослідження та розробка засобів демонстрації стеганографії та стегоаналізу

1. Якісні відмінності кваліфікаційної роботи:
В роботі виконано огляд та порівняльну характеристику методів стеганографії та стегоаналізу, вибрано і описано відповідні методи та формат контейнеру. Розроблено алгоритми та відповідне програмне забезпечення, виконано перевірку його працездатності, наведено інструкцію з його використання. Проведено експериментальне дослідження ефективності застосування зрізу по молодшим бітам для виявлення повідомлень, прихованих за методом LSB.

Основні положення роботи доповідалися та були схвалені на XVI та XVIII Міжнародних конференціях «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2022 та 2024 роках. Опубліковані відповідні тези доповідей.

2. Зауваження:
В тексті роботи відсутній аналіз результатів експериментального дослідження.

3. Висновок щодо дотримання академічної доброчесності
Академічну доброчесність дотримано.

Комплексна оцінка кваліфікаційної роботи:
Дипломна робота заслуговує позитивної оцінки.

Керівник: доцент каф. ЕОМ Денис ОСТАПЕЦЬ
Дата: 23.01.2025р.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра:

74 с., 37 рис., 3 табл., 1 додаток, 14 джерел.

Об'єкт розробки – програмні засоби демонстрації стеганографії та стегоаналізу.

Мета роботи – дослідження та розробка програмних засобів для реалізації та демонстрації роботи стеганографії та стегоаналізу.

Приведено опис та порівняльну характеристику методів стеганографії та стегоаналізу. Обґрунтовано вибір графічних контейнерів та методів LSB та LSB slicing. Описано функціонування засобів та структури даних. Розроблені блок-схеми узагальнених алгоритмів роботи комплексу в режимах приховування, отримання повідомлень та зрізу по молодшим бітам. Написане програмне забезпечення комплексу, виконано перевірку його працездатності, наведено інструкцію з його використання комплексу. Проведено експериментальне дослідження ефективності застосування зрізу по молодшим бітам для виявлення повідомлень, прихованих за методом LSB.

Розроблені програмні засоби можуть використовуватися на практиці для приховування повідомлень у графічні контейнери BMP або їх стеганографічного аналізу, а також в цілях навчання.

Ключові слова: СТЕГАНОГРАФІЯ, СТЕГОАНАЛІЗ, ГРАФІЧНИЙ КОНТЕЙНЕР, BMP-24, МЕТОД НАЙМЕНШ ЗНАЧУЩИХ БІТ, LSB, ЗРІЗ ПО МОЛОДШИМ БІТАМ, LSB SLICING, C#.

ЗМІСТ

ВСТУП

З розвитком мультимедійних технологій стеганографія перейшла на новий рівень. Однак, на сьогодні лише окремі її напрямки знаходять широке застосування.

Стеганографія є перспективною сферою для збереження та передачі інформації з обмеженим доступом, оскільки приховування самого факту існування інформації значно підвищує рівень її захисту.

Застосування методів стегоаналізу дозволяє ефективно виявляти стеганографічні техніки та розкривати приховані дані, що робить цю галузь важливою для забезпечення інформаційної безпеки. Ця робота присвячена стеганографічному приховуванню даних у контейнерах та виявленню прихованої інформації за допомогою стегоаналізу, що підкреслює актуальність обраної теми.

Мета роботи – дослідження та розробка програмних засобів для реалізації та демонстрації роботи стеганографії та стегоаналізу.

Основні положення даної роботи доповідались та були схвалені на XVI та XVIII Міжнародних конференціях «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2022 та 2024 роках [1, 2].

Робота складається із вступу, чотирьох розділів, висновків та додатку.

1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ СТЕГАНОГРАФІЇ

1.1 Загальні відомості

Двома важливими методами захисту інформації є криптографія та стеганографія. Обидва давно відомі та широко використовуються.

У криптографії ключ шифрування використовується для перетворення повідомлення. Ніхто не може отримати доступ до повідомлення без використання такого ключа. Однак передача зашифрованої інформації може легко викликати підозру, тому зашифрована інформація підлягатиме підвищеній перевірці. Для усунення цього недоліку криптографічних методів були запропоновані методи стеганографії. Стеганографія – це техніка вбудовування секретної інформації в «невинні» закриті об’єкти (зображення, аудіо, відео) [3].

Таким чином, стеганографія приховує наявність вбудованих секретних даних настільки, що ніхто не може навіть виявити факт їх існування. Обидва методи (криптографія та сётеганографія) можуть використовуватися разом для підвищення конфіденційності передачі даних. У цьому контексті стеганографія (приховування факту передачі секретних даних) і криптографія (захист інформації, що міститься) дуже відрізняються одна від одної, і необхідно розрізняти легітимну та нелегітимну потребу користувачів виявити факт. Оскільки в стеганографії неможливо відновити інформацію без відомого процесу вилучення, передбачається, що необхідні виявлення самого стеганографічного процесу відомі. Достатня кількість секретних даних, які неможливо виявити, визначає ефективність стегосистем.

1.2 Термінологія сучасної стеганографії

Стеганографія – це метод передачі чи зберігання інформації з урахуванням того, щоб залишити в таємниці сам факт передачі чи зберігання [3]. Термін був введений абатом Йоганном Тритемієм у 1499 році у трактаті "Стеганографія", прихованому під формою магічної книги у бенедиктинському монастирі Святого Мартіна у Шпонгеймі [3, 4].

На відміну від криптографії, яка приховує дуже таємні повідомлення, стеганографія стежить за тим, щоб навіть сам факт існування повідомлення залишався непоміченим. Зазвичай повідомлення виглядає як інше, наприклад, зображення, список покупок або лист. Часто стеганографія використовується разом з криптографією для додаткового захисту.

Головна перевага стеганографії, що вона дозволяє уникнути звернення уваги до передачі інформації. У той час як криптографія приховує зміст повідомлення, стеганографія маскує сам факт наявності прихованих повідомлень, уникаючи викриття у тих країнах, де заборонена криптографія. Таким чином, криптографія захищає зміст, а стеганографія – сам факт наявності прихованих повідомлень від сприйняття.

На рис. 1.1 показано основні компоненти стеганографічної системи та процес вбудовування секретної інформації в накладені об’єкти [3, 4, 5].

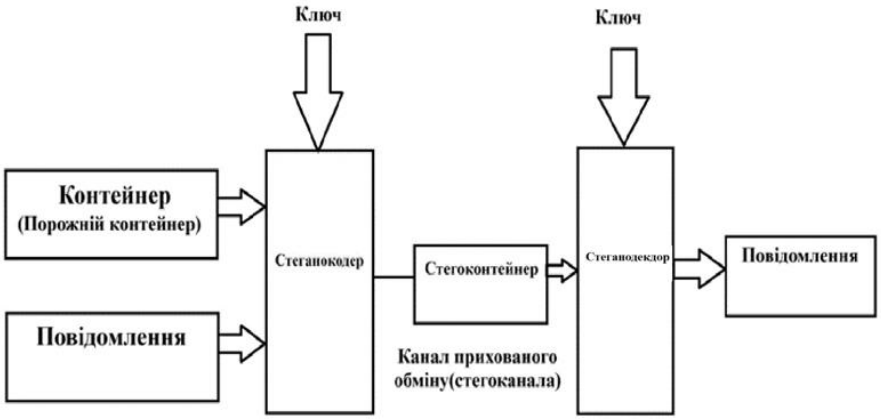


Рисунок 1.1 – Схема стеганографічної системи

На рисунку 1.1 показана базова схема будь-якої стеганографічної системи. Такою секретною інформацією може бути текст, зображення, відео, аудіо. Як правило, вбудовані повідомлення попередньо шифруються за допомогою спеціально вибраних шифрів і ключів.

Склад стеганосистеми [3, 4, 6]:

– Носій (контейнер) – це будь-яка інформація, яка використовується для приховання секретних повідомлень.

– Порожній контейнер – це контейнер, який не містить вбудованого повідомлення.

– Заповнений контейнер (стеганоконтейнер) – це контейнер, який містить інтегровану інформацію.

– Вбудоване (приховане) повідомлення – це повідомлення, яке інтегрується в контейнер.

– Стеганографічний канал – це засіб передавання прихованого повідомлення.

– Стегоключ – це секретний ключ, що використовується для приховування інформації. У стегосистемі може бути один або кілька стегоключів, залежно від рівня захисту (наприклад, вбудовування попередньо зашифрованого повідомлення).

– Стеганокодер – це пристрій, що використовується для інтеграції повідомлень у контейнер із застосуванням стеганографії.

– Стегонодекордер – пристрій, що відновлює приховане повідомлення.

1.3 Класифікація стеганографії

1.3.1 Методи стеганографії

Наприкінці 1990-х років виділилося кілька напрямків стеганографії (див. рис. 1.2) [3].

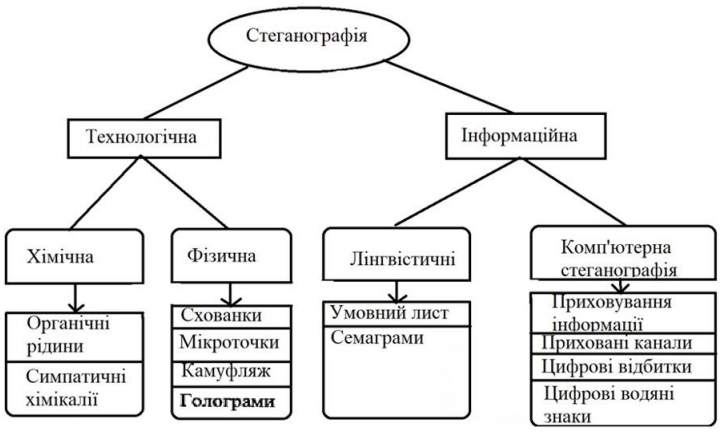


Рисунок 1.2 – Класифікація методів стеганографічного захисту

1.3.2 Хімічна стеганографія

Хімічні методи стеганографії застосовують невидимі чорнила, які можуть бути представлені у двох варіантах: органічні рідини та симпатичні хімікалії.

Органічні рідини – до таких речовин відносяться молоко, фруктові соки та оцет. Щоб виявити приховане повідомлення, написане речовинами на папері, слід злегка нагріти місце його нанесення.

Симпатичні хімікалії – хімічни розчини, що залишаються безбарвними після висихання. При обробці іншими хімічними речовинами, вони дозволяють виявити прихований текст.

1.3.3 Фізична стеганографія

Фізичні методи стеганографії використовують тайники, мікроточки, камуфляжі та голограми. Це включає запис даних на різні носії інформації, які неможливо виявити звичайними методами. Є стандартні носії інформації, яким приділяється більше уваги, такі як: комп'ютерні диски, аудіо/відеообладнання.

Приклад:

– Мікроточки – це надзвичайно маленькі точки, що застосовуються для приховування інформації. Вони настільки малі, що їх неможливо побачити без спеціальних інструментів, що робить їх ідеальними для стеганографії. Мікроточки можуть бути інтегровані в зображення, текст або інші носії інформації, і для їх виявлення потрібні спеціальні методи;

– Тайники – місця зберігання фізичних носіїв інформації;

– Камуфляж це метод, який застосовують для приховання інформації у межах іншого носія даних. Він дозволяє передавати дані так, що їх існування та зміст залишаються невидимими для сторонніх осіб;

– Голограми – тривимірні зображення, що містять інформацію.

1.3.4 Лінгвістична стеганографія

Лінгвістичні методи стеганографії використовують умовний лист і семаграми (див. рис. 1.3) [7].

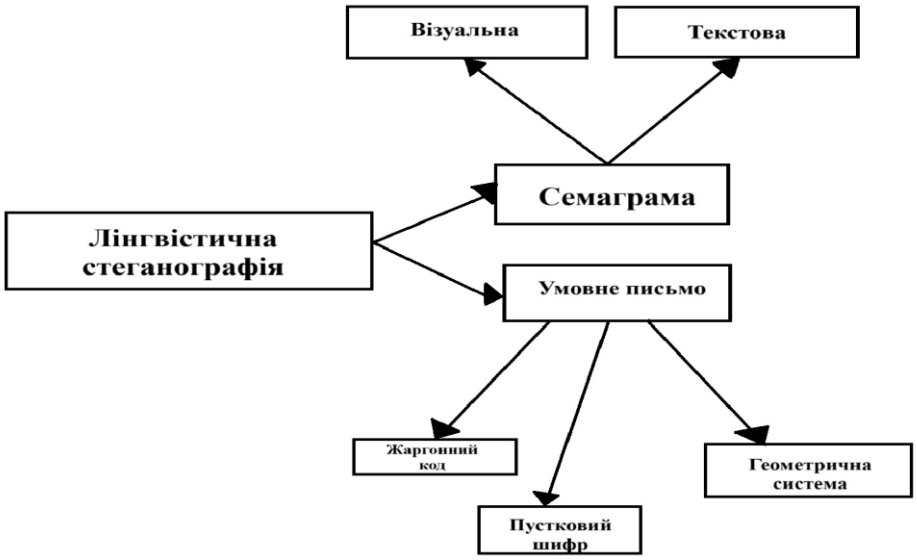


Рисунок 1.3 – Лінгвістичні методи стеганографії

Існує три види умовного листа [3, 7]:

– Жаргонний код – спосіб спілкування, який використовує спеціальні слова та вирази, зрозумілі лише певним групам людей.

– Пустковий шифр – це метод кодування інформації, який використовує певні символи або літери для заміни істинних символів у повідомленні. Повідомлення здається безглуздим, але якщо певне коло осіб знає метод розшифрування, вони можуть зрозуміти приховане значення.

– Геометрична система – це метод приховування інформації за допомогою геометричних фігур або форм. Такі системи застосовують для створення схованих повідомлень, які можна розгадати, використовуючи спеціальні методи

інтерпретації цих форм.

Друга категорія лінгвістичної стеганографії включає семаграми, що є прихованими повідомленнями, які використовують будь-які символи, крім букв і цифр.

1.3.5 Комп'ютерна стеганографія

Комп'ютерна стеганографія – це спосіб приховування інформації всередині цифрових файлів. Для цього використовуються різні технології та алгоритми, що дозволяють впроваджувати приховані повідомлення у такі файли, як зображення, аудіо, відео та текстові документи [3].

Приховування інформації – це методи стеганографії, що дозволяють впроваджувати приховані дані в цифрові файли, такі як зображення, аудіо, відео або текстові документи, без змін їх зовнішнього вигляду чи якості.

Приклад:

– Зображення (метод Least Significant Bit – LSB) – найбільш простий та популярний метод, який використовує найменш значущі біти пікселів для приховування даних у зображеннях;

– Аудіо – інформація може бути прихована у звукових файлах завдяки маніпуляціям з амплітудами звукових хвиль;

– Відео – інформація приховується у відеозаписах шляхом зміни окремих кадрів або пікселів.

Приховані канали (Hidden Channels) – це способи передачі інформації через цифрові мережі, які залишаються непоміченими для стандартних методів моніторингу.

Цифрові відбитки – це метод біометричної ідентифікації, що використовує унікальні риси пальців для підтвердження особи. Цей процес включає сканування пальців, створення цифрового відбитка та його збереження у базі даних. При подальшому вході новий скан пальців порівнюється з збереженим відбитком для автентифікації особи.

Цифрові водяні знаки (Digital Watermarks) – це спеціальні метадані, що інтегруються у цифрові файли, такі як зображення, відео та аудіо, з метою захисту авторських прав та ідентифікації власників. Вони також використовуються для виявлення та запобігання піратству.

Зараз у світі активно прогресує цифрова стеганографія з використанням комп'ютера. Це охоплює створення нових алгоритмів та інструментів, що дають змогу ефективно й надійно приховувати інформацію [7].

Сучасні стеганографічні методи можуть застосовувати машинне навчання для вдосконалення технік приховування, роблячи їх менш помітними та стійкішими до виявлення.

Порівняльна характеристика методів стеганографії наведена в табл. 1.1.

1.4 Області застосування стеганографії

Можливі сфери застосування стеганографії (див. рис. 1.4) [5, 6, 8].

Захист від незаконного копіювання:

– Електронна комерція, де стеганографія може застосовуватися для захисту цифрових товарів;

– Контроль тиражування DVD та інших носіїв мультимедійної інформації;

– Розповсюдження мультимедійної інформації, зокрема відео по запиту.

Автентифікація:

– Системи відеоспостереження, де стеганографія може бути використана для забезпечення конфіденційності та цілісності відео-записів;

– Електронна комерція, де стеганографія служить для автентифікації електронних транзакцій;

– Голосова пошта та електронне конфіденційне діловодство, де стеганографія допомагає в підтвердженні автентичності та безпеки обміну інформацією.

Таблиця 1.1 - Порівняння методів стеганографії

Методи стеганографії / Характеристика	Наочність	Стеганостійкість	Використання контейнера	Складність реалізації
Хімічна стеганографія	Висока, оскільки використовує хімічні реакції	Висока, оскільки інформація вбудована на молекулярному рівні	Використовує хімічні речовини як контейнери	Висока, вимагає точного контролю хімічних реакцій
Фізична стеганографія	Висока, оскільки використовує фізичні зміни	Висока, оскільки інформація вбудована на фізичному рівні	Використовує фізичні об'єкти як контейнери	Середня, вимагає меншого контролю фізичних змін
Лінгвістична стеганографія	Низька, оскільки інформація приховується у тексті	Залежить від складності тексту та контексту	Використовує текст як контейнер	Відносно низька, вимагає доброго знання мови та стилістики
Комп'ютерна стеганографія	Висока, оскільки використовує цифрові носії, такі як зображення, аудіо або відео	Висока, оскільки використовує цифрові алгоритми для приховування інформації	Використовує цифрові файли як контейнери	Висока, вимагає технічних знань та використання спеціалізованого програмного забезпечення

Прихована анотація документів:

– Медична сфера, зокрема прихована анотація на медичних знімках;

– Картографія, де стеганографія використовується для прихованої анотації географічних карт;

– Мультимедійні бази даних, де інформація може бути прихована в мультимедійних файлах.

Прихований зв'язок:

– Військові та розвідувальні цілі, де стеганографія допомагає приховати інформацію від неприязних сил;

– Ситуації, коли використання криптографії не є можливим чи доцільним.

Рисунок 1.4 – Потенційні множини застосування стеганографії

1.5 Вибір стеганографічного методу

У роботі прийнято рішення використовувати графічний контейнер.

Для графічних контейнерів можуть використовуватися методи, наведені нижче [3, 8].

– **Метод блочного приховання:**

Оригінальне зображення розглядається як набір неперетинаючих блоків довільної форми, і для кожного з цих блоків створюється біт парності.

– **Метод найменш значущих біт:**

Метод найменших значущих бітів (НЗБ) - є методом обробки зображень та стиснення, що включає в себе видалення або заміну менш важливих бітів у кожному байті або пікселі зображення. Ця техніка може бути використана для оптимізації розміру як зображень, так і інших видів даних.

– **Метод псевдовипадкової перестановки:**

Метод псевдовипадкової перестановки є технікою шифрування або обробки даних, де псевдовипадкові числа використовуються для формування перестановок або ключів перестановок.

– **Метод дискретно косинусного перетворення:**

Метод дискретного косинусного перетворення (DCT) представляє собою техніку обробки сигналів та стискання даних, широко використовувану в області обробки зображень та аудіо. DCT конвертує послідовність дискретних даних, таких як пікселі у зображенні чи відгуки сигналу, у послідовність коефіцієнтів, які відображають внесок різних частот у ці дані.

З перерахованих методів мною обрано метод НЗБ з наступних причин:

– Простота реалізації – легко реалізується навіть для початківців;

– Висока ємність даних – дозволяє впроваджувати значну кількість прихованих даних;

– Непомітність – зміни в зображенні важко виявити без спеціального аналізу.

1.6 Висновки за розділом

Розглянуто основні поняття стеганографії та класифікацію методів стеганографії. Проведено порівняльний аналіз методів стеганографії. Для подальшої реалізації в роботі прийнято рішення у якості контейнера використовувати графічні файли формату BMP-24. В результаті проведеного аналізу вибрано метод найменших значущих бітів (LSB, Least Significant Bit).

2 ФУНКЦІЇ, РЕЖИМИ РОБОТИ ТА ІНФОРМАЦІЙНА СТРУКТУРА РОЗРОБЛЮВАНИХ ЗАСОБІВ

2.1 Структура контейнера

Формат BMP (від англ. bitmap) є популярним у операційній системі (ОС) Windows для обміну різними зображеннями між програмами. Цей формат добре відомий і підтримується майже усіма додатками, що працюють в середовищі Windows. Файли BMP займають значну кількість пам'яті, маленькі зображення з роздільною здатністю 640 × 480 пікселів можуть займати кілька мегабайтів простору. BMP-файл має просту структуру [9].

Структура кожної частини файлу, що містить 256-колірне зображення, показана в таблиці 1.2 [6].

Заголовок файлу починається з "BM" [6], після чого вказується загальна довжина файлу в байтах. Далі йдуть 4 забронюванні байти для можливих майбутніх розширень формату, а в кінці заголовка знаходиться зсув від початку файлу до даних зображення. Для 256 кольорів цей зсув становить 1078 байтів, що означає, що потрібно пропустити кількість байтів, щоб отримати інформацію.

Інформаційний розділ починається з довжини (може перетворюватися для файлу з 256 кольорами складає 40 байт) і містить дані про розміри зображення, його роздільну здатність, особливості кольорового представленнята інші параметри [6, 9].

Кількість площин, які можуть використовуватися у файлах з невеликою глибиною кольору. Файли з кількістю 256 кольорів і в кілька разів більше – це становить 1, по цьому поле вважається застарілим, але зберігається для забезпечення сумісності [9].

Глибина кольору - це ключова характеристика відображення кольору у файлі, що визначається кількістю біт на пік сель [6]. У такому разі значення дорівнює 8 біт.

Стиснення в BMP-файлах зазвичай не застосовується, однак для цього передбачено спеціальна зона в заголовку. Як правило воно дорівнює 0, що

означає відсутність стиснення зображення.

Розмір зображення – це обсяг пам'яті в байтах, необхідний для зберігання зображення, без урахування палітри [6].

Таблиця 1.2 - Структура BMP-файла

Ім’я	Довжина	Зсув	Опис
Заголовок файла (BitMapFileHeader)			
Type	2	0	Сигнатура "BM"
Size	4	2	Розмір файла
Reserved 1	2	6	Зарезервовано
Reserved 2	2	8	Зарезервовано
OffsetBits	4	10	Зсув зображення від початку файла
Інформаційний заголовок (BitMapInfoHeader)			
Size	4	14	Довжина заголовка
Width	4	18	Ширина зображення, точка
Height	4	22	Висота зображення, точка
Planes	2	26	Кількість площин
BitCount	2	28	Глибина кольору, бітів на точку
Compression	4	30	Тип компресії (0 – незжате зображення)
SizeImage	4	34	Розмір зображення, байт
XpelsPerMeter	4	38	Горизонтальна роздільна здатність, точка на метр
YpelsPerMeter	4	42	Вертикальна роздільна здатність, точка на метр
ColorsUsed	4	46	Кількість використовуваних кольорів (0 – максимально можливе для даної глибини кольору)
ColorsImportant	4	50	Кількість основних кольорів
Таблиця кольорів (палітра) (ColorTable)			
ColorTable	1024	54	256 елементів по 4 байти
Дані зображення (BitMap Array)			
Image	Size	1078	Записане рядками зліва направо і знизу вгору

Горизонтальна та вертикальна роздільна здатність визначається кількістю точок на метр. Ці параметри важливі для точного відтворення масштабу відсканованих зображень. У графічних редакторах створені зображення зазвичай мають нульові значення в цих полях [9].

Обсяг таблиці палітри можна зменшити завдяки меншій кількості кольорів у зображенні, ніж дозволяє обрана глибина. На практиці файли рідко

зустрічаються. На практиці такі файли зустрічаються рідко. У випадку, коли число кольорів досягає максимально можливого значення для цієї глибини кольору, скажімо, 256 кольорів при 8 бітах, поля встановлюються на нуль [6].

Кількість основних кольорів розпочинається з початкових позицій палітри, і рекомендується виводити її без змін. Це стає критичним, якщо підтримувана дисплеєм максимальна кількість кольорів менша, ніж у палітрі BMP-файлу. Під час створення формату, мабуть, припускали, що найчастіше вживані кольори розташуються на початку таблиці. На практиці це вимога не завжди дотримується, і кольори не сортуються за частотою їхнього використання в зображенні. Це має значення, оскільки палітри, навіть якщо вони складаються з однакових кольорів, можуть розташовувати їх в іншій послідовності, що робить це більш складним, правильне відображення таких зображень одночасно [3, 6].

Після заголовка з інформацією йде таблиця кольорів, набір з 256 елементів, кожен з яких складається з 4 байтів, кожен з яких відповідає певному кольору в палітрі. Три байти описують компоненти синього, зеленого та червоного кольору, а залишковий байт (старший байт) кожного поля дорівнює нулю.

Після кольорової таблиці слідують дані зображення, які зберігаються построчно, починаючи від низу до верху, а всередині кожного рядка - зліва направо. Через те, що на деяких системах неможливо обробляти дані менших розмірів, ніж 4 байти, довжина кожного рядка округлюється до найближчого значення, що кратне 4 байтам. Інакше кажучи, якщо довжина рядка не є кратною чотирьом, її заповнюють нулями. Це слід мати на увазі при читанні файлу, хоча доцільніше буде, якщо горизонтальні розміри зображень будуть кратні числу 4.

Формат файлу створений з врахуванням сумісності з різними платформами, тому не дивно, що зберігання кольорів палітри відрізняється від стандартів Video Graphics Array. У процесі читання файлу виконується відповідне перекодування [9].

2.2 Принцип роботи методу найменш значущих біт

Метод заміни молодшого біта (LSB), який використовується для стеганографічного захисту інформації, базується на зміні найменш значущих бітів. Через те, що молодший біт незначно впливає на візуалізацію байту в аудіо, фото та відео форматах, ці зміни не можна помітити людським оком [5]. На рис. 2.1 показано як молодший біт, позначений індексом 0, змінюється на 1.

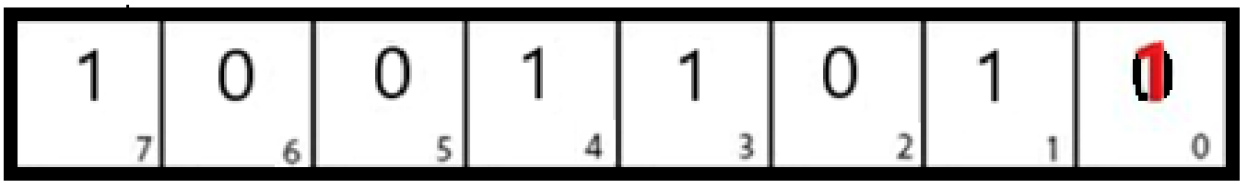


Рисунок 2.1 – Заміни молодшого біту

Цей метод популярний завдяки своїй простоті та здатності приховувати значний обсяг інформації у невеликих файлах прихований канал зв'язку може мати пропускну здатність від 12,5% до 30%. Найчастіше метод застосовується до растрових зображень у форматах, таких як GIF і BMP [3].

Суть методу полягає в тому, що найменш значущі біти пікселів зображення замінюються на біти секретного повідомлення. У простому випадку кожен послідовно розташований піксель контейнера змінюється послідовно. Щоб зробити приховування менш помітним, секретне повідомлення часто заповнюють випадковими бітами, щоб його довжина у бітах відповідала кількості пікселів в оригіналі. В такому випадку змінені пікселі будуть рівномірно розподілені, і статистичні тести не покажуть жодних відхилень [6].

Метод НЗБ характеризується низькою стеганографічною стійкістю до атак як пасивних, так і активних порушників. Основним його недоліком є висока чутливість до незначних спотворень контейнера, для зменшення якої часто застосовують завадостійке кодування.

2.3 Вибір методу стеганографічного аналізу

Для графічного контейнера та методу LSB можуть бути використані наступні методи стегоаналізу [10].

Метод хі-квадрат (χ2) – це статистичний метод, що використовується в стегоаналізі для виявлення прихованих даних у графічних зображеннях. Його основна мета – виявити відхилення у розподілі частот значень пікселів, які виникають при впровадженні стегоповідомлення, особливо з використанням методів найменш значущого біта (LSB).

RS-аналіз (Regular-Singular аналіз) – це статистичний метод стегоаналізу, спрямований на виявлення прихованої інформації в цифрових зображеннях. Він був розроблений спеціально для аналізу методів стеганографії, заснованих на зміні найменш значущого біта (LSB). Основна ідея методу – дослідження статистичних закономірностей, які порушуються при впровадженні стегоповідомлення.

Характеристичний аналіз (Feature-Based Analysis) – це метод стегоаналізу, який базується на вилученні набору характеристик із цифрового зображення та їх подальшому аналізі для виявлення прихованих даних. Ці характеристики відображають статистичні, текстурні, просторові та частотні властивості зображення.

SPA-аналіз (Sample Pair Analysis) – це метод стегоаналізу, спрямований на виявлення прихованих даних у графічних зображеннях шляхом аналізу змін у парах сусідніх пікселів. Цей метод особливо ефективний для виявлення стеганографії, заснованої на методі найменш значущого біта (LSB).

Зріз по молодшим бітам – це метод стегоаналізу, який використовується для виявлення прихованої інформації, що зберігається у найменш значущих бітах (LSB, Least Significant Bits) графічних або інших файлів. Цей підхід дозволяє візуалізувати зміни, внесені до LSB файлу, що часто є результатом стеганографічного вбудовування.

З наведених вище методів для подальшої реалізації в роботі обрано саме зріз

по молодших бітах.

Зріз по молодшим бітам є одним з найпопулярніших методів стегоаналізу і його використання в деяких ситуаціях може бути переважніше за інші методи, наведені вище.

Простота та ефективність:

– Зріз по молодших бітах є дуже простим в реалізації та використанні методом. Він дозволяє швидко та ефективно виявити приховану інформацію, вбудовану в зображення, без необхідності у складних математичних обчисленнях або додаткових перетвореннях.

– На відміну від характеристичний аналізу або spa-анализу, які потребують глибших обчислень та аналізу, зріз по молодших бітах може бути виконаний практично одразу, що робить його більш підходящим для швидкої перевірки на наявність прихованих даних.

Точність при виявленні змін у даних:

– Зріз по молодших бітах дозволяє точно виділити зміни, внесені в пікселі зображення при приховуванні даних. Це робить його високо ефективним для виявлення прихованих даних у зображеннях без необхідності у складних моделях чи глибоких обчисленнях.

– Метод Хі-квадрат та RS-аналіз можуть дати хибні спрацьовування, особливо якщо зображення було стиснене або піддане іншим операціям, які змінюють статистичні параметри або частотний розподіл даних.

Застосовність до великого обсягу даних:

– Метод зрізу по молодших бітах підходить для роботи з великими обсягами даних, оскільки його можна використовувати для аналізу всіх пікселів зображення та вилучення прихованих даних з них. Він не залежить від розміру даних, на відміну від характеристичний аналіз, який може бути застосований тільки до аудіо або відео потоків, та аналізу частотної області, який більше орієнтований на частотні зміни. Порівняльна характеристика методів стегоаналізу наведена в табл. 2.1.

Таблиця 2.1 - Порівняння методів стегоаналізу

Метод	Переваги	Обмеження
Метод Хі-квадрат (χ2)	Простий у реалізації, підходить для виявлення LSB-стеганографії та використовує статистичний підхід для аналізу змін у зображеннях.	Нечутливий до складних методів стеганографії і погано працює з обробленими або стиснутими зображеннями через зміни в статистичних властивостях.
RS-аналіз (Regular-Singular)	Високу точність для LSB-методів, не вимагає знання вихідного зображення і є простим у реалізації.	RS-аналіз залежить від якості зображення, чутливий до стиснення із втратами (JPEG) і обмежений для складних методів стеганографії, що знижує його ефективність у таких випадках.
Характеристичний аналіз	Відзначається універсальністю, використанням машинного навчання для підвищення точності та ефективністю для складних методів стеганографії.	Вимагає великих обчислювальних ресурсів, необхідності навчання на великому наборі даних і залежить від якості вхідних даних.
SPA-аналіз (Sample Pair Analysis)	Виявляє приховані дані без знання вихідного контейнера і використовує математичні моделі для виявлення аномалій.	Має високу обчислювальну складність і погано працює з зашумленими або сильно обробленими даними.
Зріз по молодшим бітам (LSB Slicing)	Простий і швидкий метод, який чітко візуалізує зміни в молодших бітах і не вимагає складних розрахунків.	Обмежений для складних стеганографічних методів, чутливий до шуму та обробки даних і працює тільки з LSB-методами.

2.4 Вимоги до функціонування розроблюваних засобів

В роботі передбачається розробити програмний комплекс засобів для демонстрації роботи стеганографії за методом LSB з використанням графічного контейнера формату BMP та стеганографічного аналізу методом зрізу по молодшим бітам контейнеру.

Основні функціональні вимоги:

– У якості контейнера використовується зображення з розширенням «*.BMP»;

– Дані, що вбудовуються можуть бути текстовим файлом або коротким повідомленням, введеним користувачем;

– Реалізація алгоритму LSB (заміна молодших бітів пікселів зображення на біти приховуваної інформації);

– Збереження стеганографічного зображення у форматі BMP;

– Реалізація зрізу по молодшим бітам.

Вимоги до візуалізації інформації:

– Порівняння вихідного та стеганографічного зображення;

– Відображення змін у бітовій структурі пікселів (наприклад, таблиця пікселів).

Вимоги до реалізації методу зрізу по молодшим бітам:

– Виділення та візуалізація молодших біт (LSB) кожного кольорового каналу (R, G, B).

Режими роботи розроблюваних засобів:

– приховування повідомлення у вибраному контейнері;

– отримання повідомлення з контейнеру;

– візуальний аналіз за допомогою зрізу по молодшим бітам.

В режимі приховування користувач повинен обрати файли з контейнером і таємним повідомленням. У результаті користувач отримує заповнений контейнер (див. рис. 2.2).

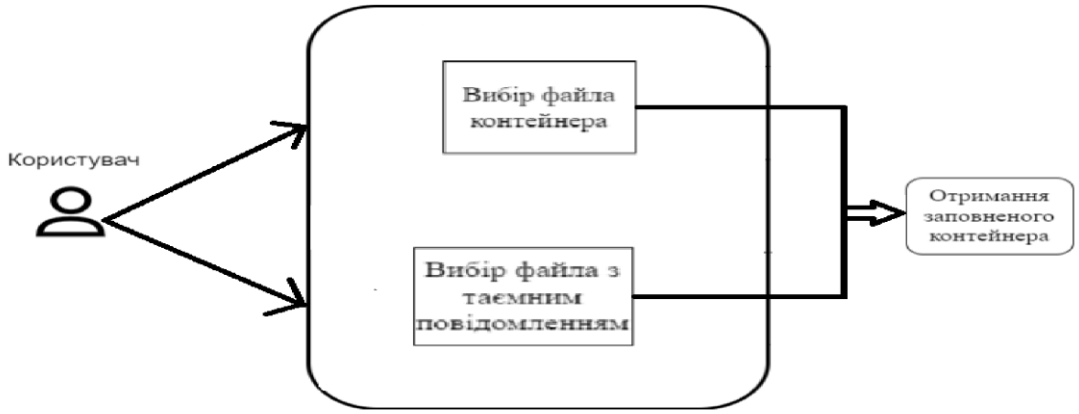


Рисунок 2.2 – Робота комплексу в режимі приховування повідомлення

В режимі отримання прихованого повідомлення користувач повинен обрати файл з заповненим контейнером. (див. рис. 2.3).

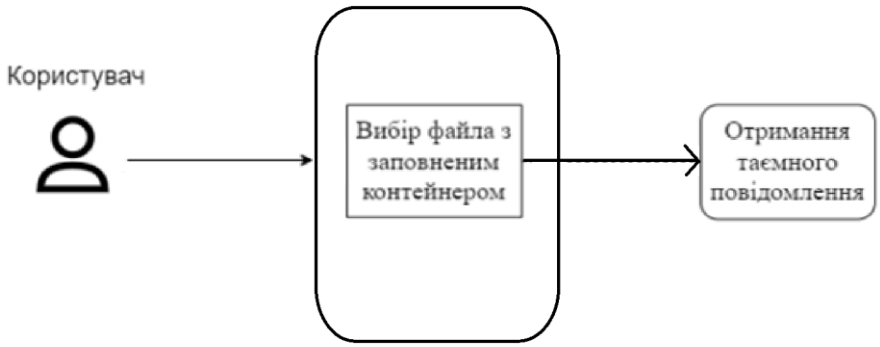


Рисунок 2.3 – Робота комплексу в режимі отримання повідомлення

2.5 Структура даних

Для передачі службової інформації у заповненому контейнері використовується стоп-секвенція довжиною 16 біт (11111111 11111110), що позначає кінець даних (див. рис. 2.4). Запис повідомлення виконується методом зміни молодших значущих бітів (LSB) у каналах кольорів (R, G, B) пікселів зображення. Для запису повідомлення використовується вся доступна ємність контейнера, за винятком 16 біт, відведених для стоп-секвенції, що розташована після основного повідомлення.

Максимальна довжина таємного повідомлення залежить від розміру зображення. Наприклад, для зображення розміром 1920x1080 пікселів можна вбудувати до 6 220 784 біт даних. Оскільки кожен піксель містить 3 канали (R, G, B), і в кожному каналі можна зберігати 1 біт, це обчислюється як: $1920 \times 1080 \times 3 - 16 = 6\,220\,784$ біт. У байтах це становить $6\,220\,784 \div 8 = 777\,598$ байт, що приблизно дорівнює 759 кБ.

Таким чином, дана реалізація не потребує додаткових бітів для заголовка, який містить довжину повідомлення, що не обмежує корисний об’єм контейнера.

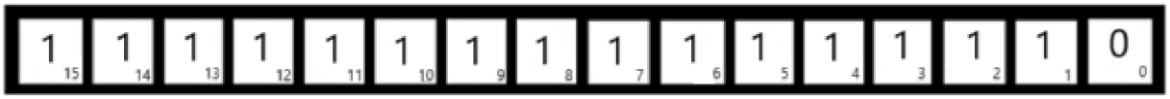


Рисунок 2.4 – Структура стоп-послідовності прихованого повідомлення

2.6 Висновки за розділом

Наведено структуру стеганографічного контейнера (BMP-файла). У якості методу стеганографічного аналізу, який використовується для виявлення прихованої інформації, що зберігається у найменш значущих бітах, обрано зріз по молодшим бітам. Описано режими роботи розроблюваного комплексу засобів. Розроблено структуру стоп-послідовності прихованого повідомлення.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір засобів реалізації

Для створення графічного додатку було обрано інтегроване середовище розробки Visual Studio 2022. Ця версія Visual Studio є однією з найшвидших, ефективних та зручних, підходить як для студентів, так і для розробників, які працюють над масштабними проєктами. Visual Studio 2022 для Windows працює як 64-розрядний додаток, що дозволяє комфортно відкривати, редагувати, запускати та налагоджувати навіть найскладніші та об'ємні проєкти без обмежень, пов’язаних із використанням пам’яті [11].

Для розробки способу стеганографічного захисту інформації, обрано мова C#. Мова програмування C# має чіткий синтаксис, який поєднує в собі кращі риси інших мов, таких як C++ та Java. Це дозволяє швидше навчатися і розробляти застосунки. C# є об'єктно-орієнтованою мовою, що допомагає розробляти масштабовані, підтримувані та повторно використовувані програми [12]. Принципи ООП, такі як інкапсуляція, успадкування і поліморфізм, значно поліпшать структуру коду. Вибір мови C# зумовлений її широкими можливостями, які цілком достатні для успішної реалізації комплексу.

Для створення графічного інтерфейсу програми було обрано технологію Windows Forms.NET. Windows Forms — це технологія інтерфейсу користувача для .NET, набір керованих бібліотек, які спрощують загальні завдання застосунків, такі як читання і запис у файлову систему [13]. Використовуючи конструктор Windows Forms з функцією перетягування у Visual Studio, можна легко створювати застосунки Windows Forms.

Для реалізації шифрування алгоритмом приховування по всіх пікселях був розроблений самописний клас Steganography.

3.2 Розробка підпрограми приховування повідомлення

Як зазначено в попередніх розділах, система реалізує єдиний метод приховування по всіх пікселях. На рисунку 3.1 представлена блок-схема узагальненого алгоритму цього методу в режимі приховування повідомлень. Вихідний код програми наведено в додатку А.

Блок 1 – Початок процесу.

Блок 2 – Зчитування з файлу або текстового поля.

Блок 3 – Підготування заголовка.

Блок 4 – i < image.Height, умова, що перевіряє, чи індекс і менший за висоту зображення.

Блок 5 – Заповнення молодших бітів в RGB підготовленими бітами в повідомленні.

Блок 6 – Записування нових бітів у масив пікселів.

Блок 7 – Завершення процесу.

3.3 Розробка підпрограми отримання повідомлення

Блок-схема узагальненого алгоритму методу приховування по всіх пікселях у режимі отримання повідомлення наведена на рисунку 3.2. Вихідний код програми див. в додатку А.

Блок 1 – Початок процесу.

Блок 2 – Отримання масиву пікселів з зображення.

Блок 3 – i < image.Height: Умова, що перевіряє, чи індекс і менший за висоту зображення.

Блок 4 – Зчитування частини повідомлення з молодших бітів.

Блок 5 – Записування нових бітів у масив пікселів.

Блок 6 – Завершення процесу.

3.4 Розробка підпрограми зрізу по молодшим бітам

Блок-схема узагальненого алгоритму методом зрізу по молодших бітах у режимі роботи візуального аналізу за допомогою зрізу по молодших бітах наведена на рисунку 3.3. Вихідний код програми див. в додатку А.

Блок 1 – Початок процесу.

Блок 2 – Отримуємо масив пікселів з зображення.

Блок 3 – i < image.Height: Умова, що перевіряє, чи індекс i менший за висоту зображення.

Блок 4 – В залежності від кольору вибирається, який колір максимізується

Блок 5 – Завершення процесу.

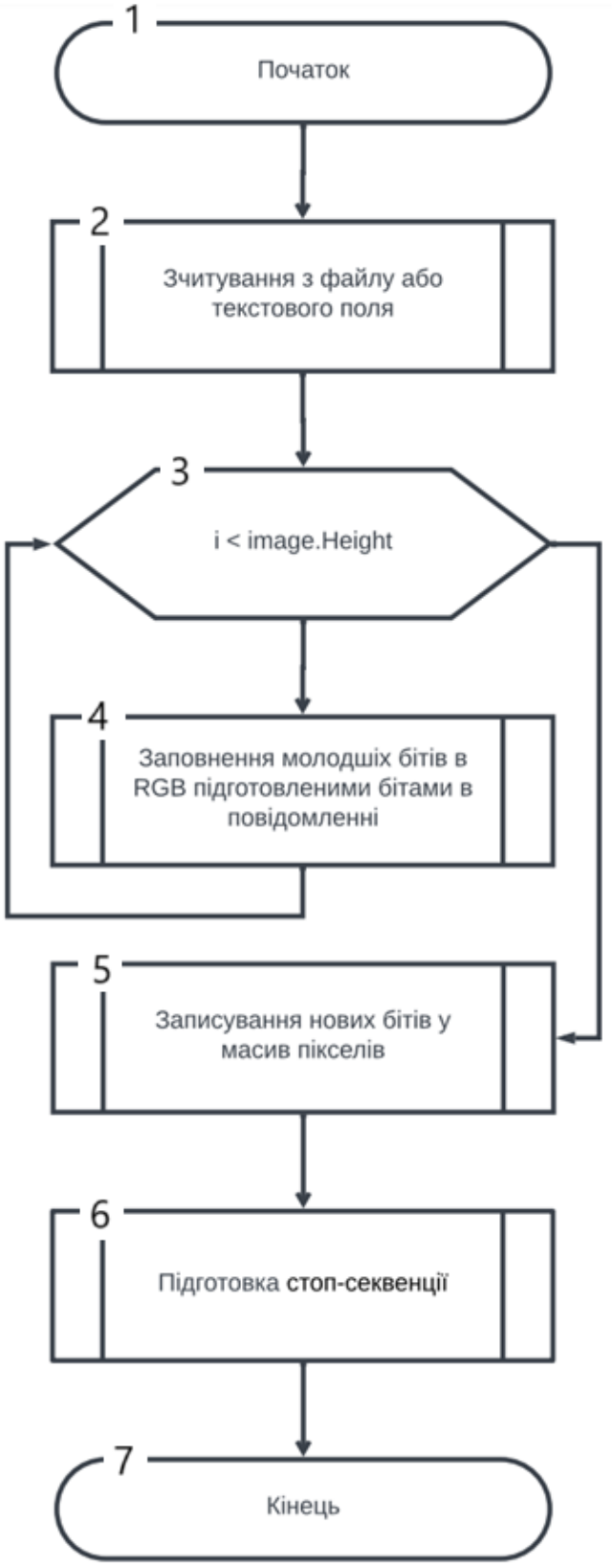


Рисунок 3.1 – Блок-схема узагальненого алгоритму приховування повідомлення

Рисунок 3.2 – Блок-схема узагальненого алгоритму отримання повідомлення

Рисунок 3.3 – Блок-схема узагальненого алгоритму зрізу по молодшим бітам

3.5 Перевірка працездатності

Для перевірки працездатності приховування інформації обране порожній контейнер без будь-яких вбудованих даних, символ 'A' (див.рис. 3.4).

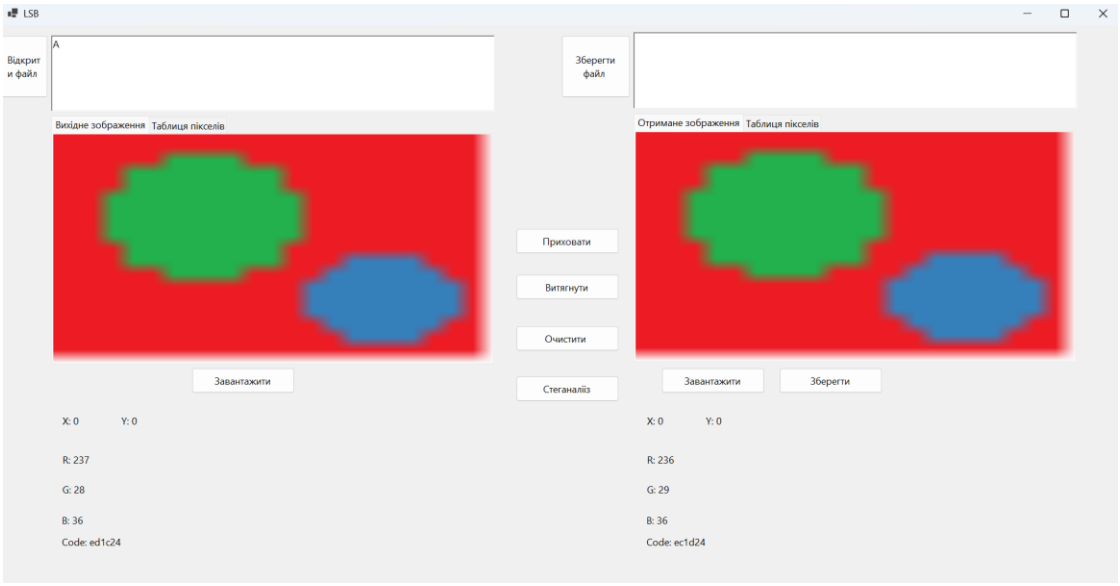


Рисунок 3.4 – Отримання заповненого контейнера методом LSB

У результаті вибору файлу із заповненим контейнером отримано таємне повідомлення (див.рис.3.5).

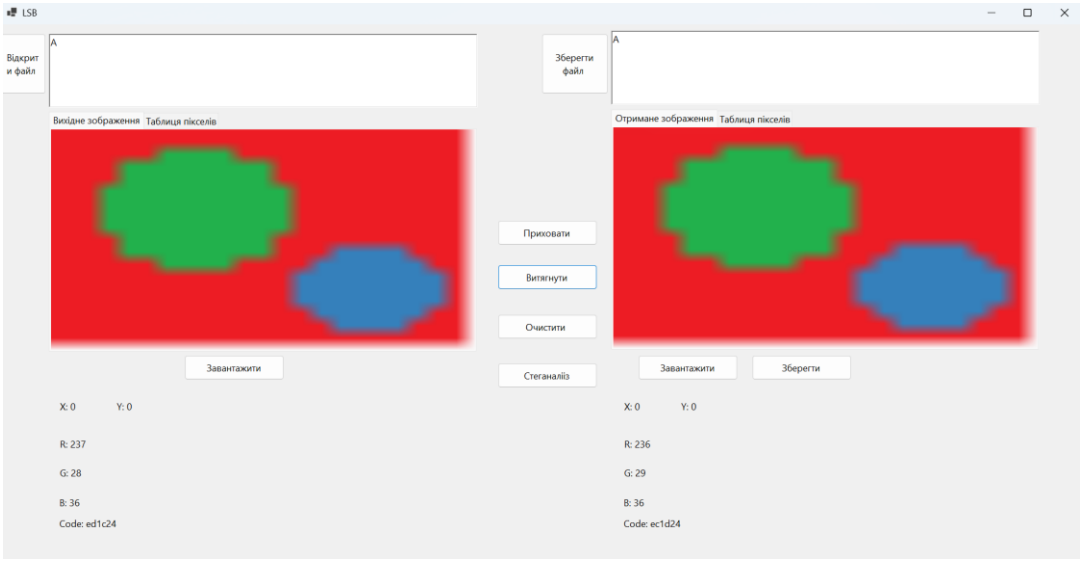


Рисунок 3.5 – Отримання таємного повідомлення методом LSB

Для запису повідомлення використовується вся доступна ємність контейнера, за винятком 16 бітів, відведених для стоп-секвенції, що розташована після основного повідомлення. Представлення повідомлення у двійковій системі числення - 010000011111111111111110.

На рисунках 3.6 – 3.7 виділені червоним квадратом довжина повідомлення та

стоп-секвенції. 0100 0001 – основне повідомлення (див.рис.3.6), символ 'A' в двійковому коді, у десятковій системі дорівнює 65, а у шістнадцятковій системі - 41H. 1111111111111110 – стоп-секвенція, яка сигналізує про завершення прихованих даних (див.рис.3.7).

При визначенні кожного біта прихованого повідомлення можна використати або шістнадцяткову інтерпретацію коду пікселя з таблиці, або десяткові значення колірних каналів R, G, B (див. рис. 3.6, 3.7). В першому випадку беремо до уваги молодші біти кожного байта коду пікселя. В другому випадку, якщо значення парне, то наступний біт повідомлення дорівнює «1», інакше – «0».

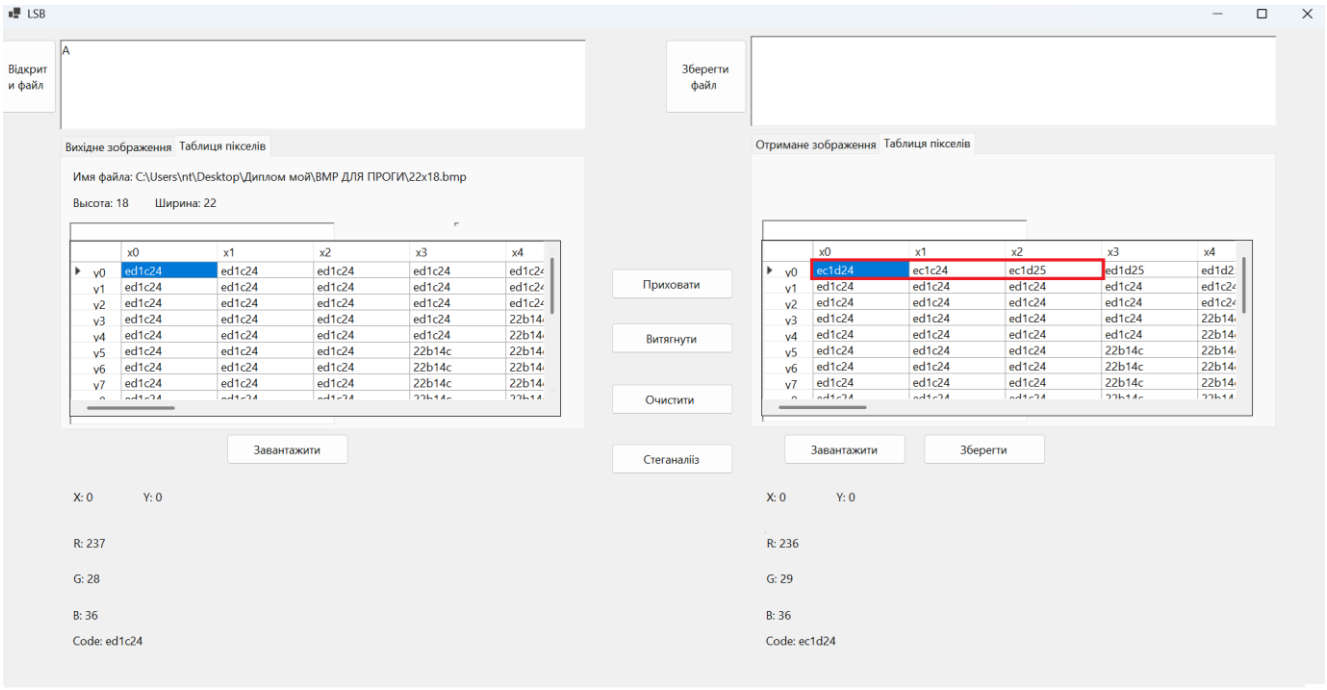


Рисунок 3.6 – Пікселі з прихованим повідомленням

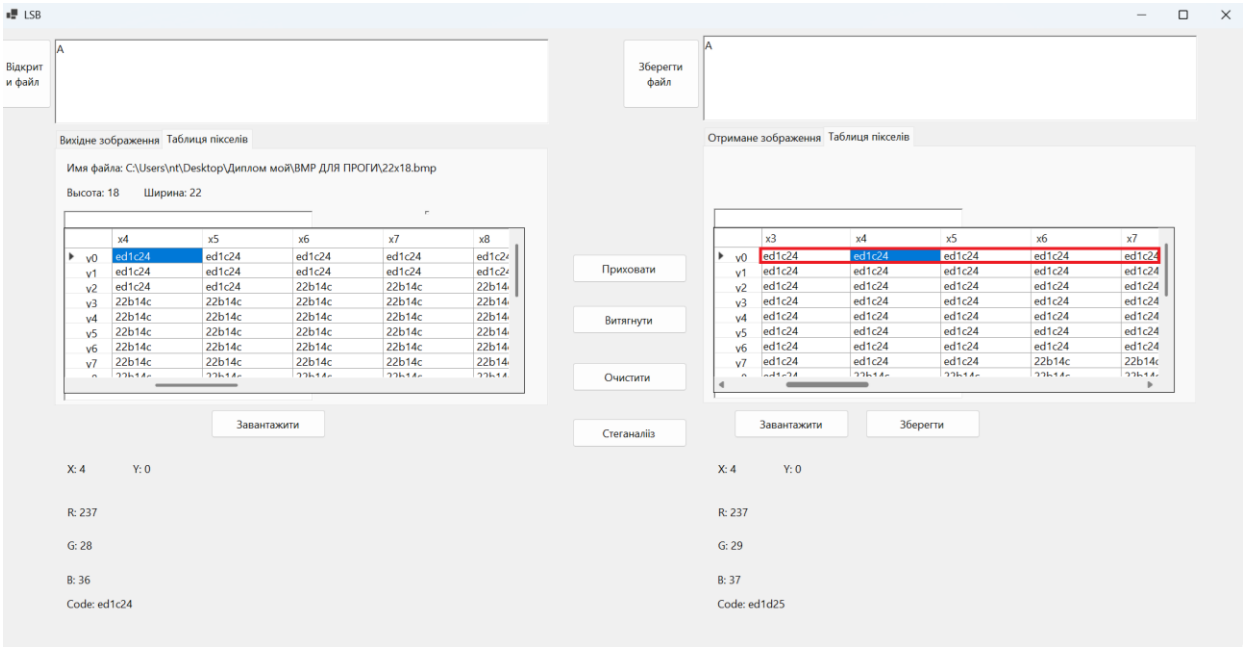


Рисунок 3.7 – Пікселі з прихованою стоп-секвенцією

3.6 Інструкція з використання розроблених засобів

Розроблений комплекс засобів може слугувати як інструмент для стеганографічного захисту інформації, стеганографічного аналізу, а також виконувати функцію навчальної програми. Основний інтерфейс програми складається з елементів для приховування, вилучення секретного повідомлення та таблиці пікселів (див. рис. 3.8).

Основний інтерфейс складається з полів та кнопок:

а) Відкрити файл – дозволяє вибрати файл з розширенням «*.TXT» із секретним повідомленням.

б) Початкове поле тексту може бути обраним текстовим файлом або повідомленням, введеним користувачем, яке потрібно для приховування в контейнері.

в) Зберегти файл – дозволяє зберегти приховане повідомлення, отримане з заповненого контейнера (див. п. № е, г).

г) Поле вмісту прихованого повідомлення

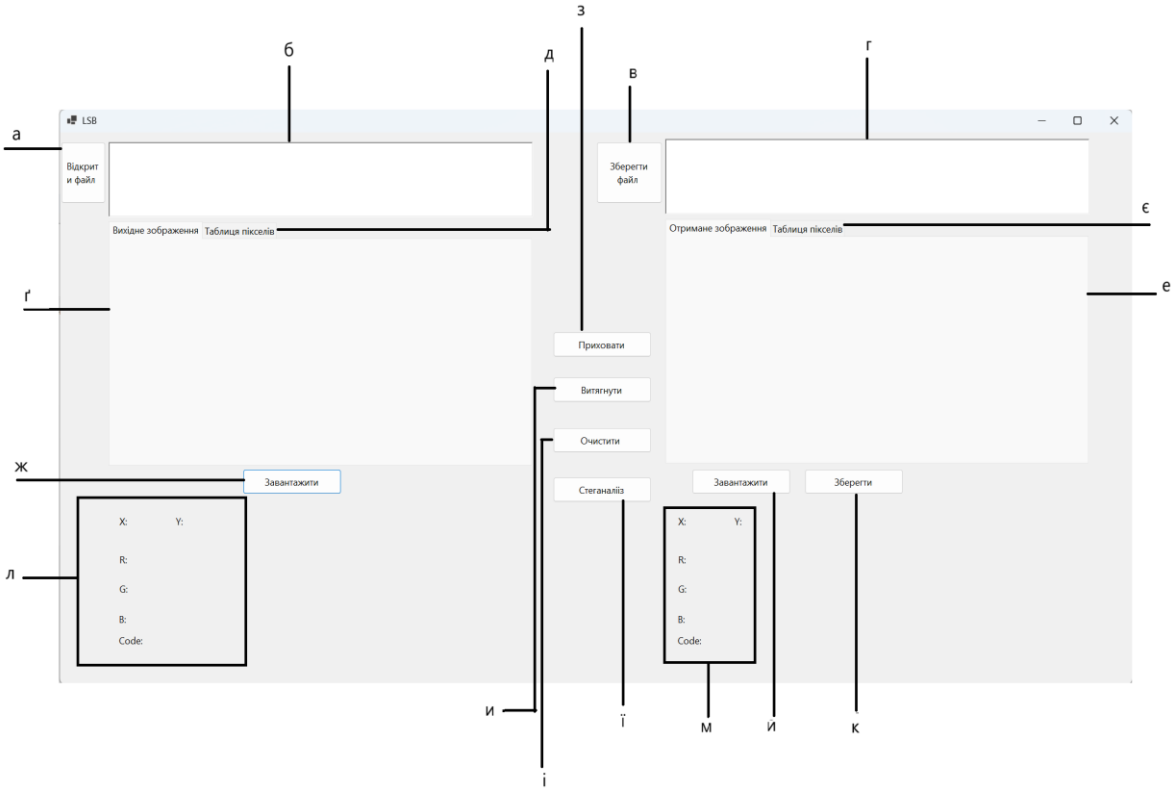


Рисунок 3.8 – Основний інтерфейс програми

ґ) Поле вихідного (порожнього) контейнера.

д) Таблиця пікселів – відображає координати X, Y, шлях до файлу, висоту, ширину та бітову структуру пікселів вихідного (порожнього) контейнера (див. рис. 3.9).

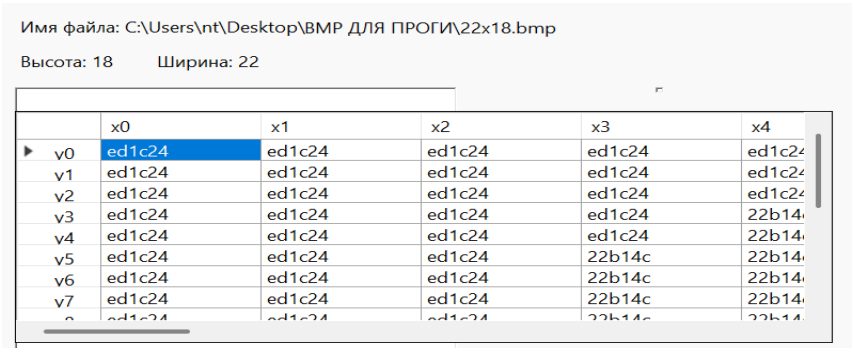
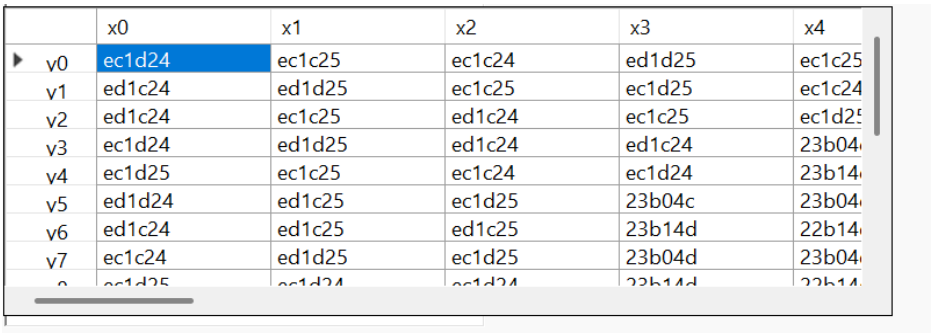


Рисунок 3.9 – Бітова структура пікселів порожнього контейнера

е) Поле отриманого (заповненого) контейнера.

є) Таблиця пікселів – відображає координати X, Y та зміни в бітовій структурі пікселів отриманого (заповненого) контейнера (див. рис. 3.10).



	x0	x1	x2	x3	x4
v0	ec1d24	ec1c25	ec1c24	ed1d25	ec1c25
v1	ed1c24	ed1d25	ec1c25	ec1d25	ec1c24
v2	ed1c24	ec1c25	ed1c24	ec1c25	ec1d25
v3	ec1d24	ed1d25	ed1c24	ed1c24	23b04
v4	ec1d25	ec1c25	ec1c24	ec1d24	23b14
v5	ed1d24	ed1c25	ec1d25	23b04c	23b04
v6	ed1c24	ed1c25	ed1c25	23b14d	22b14
v7	ec1c24	ed1d25	ec1d25	23b04d	23b04

Рисунок 3.10 – Бітова структура пікселів заповненого контейнера

ж) Завантажити – дозволяє вибрати графічний контейнер розширенням «*.BMP».

з) Приховати – приховує секретне повідомлення в графічному контейнері та відображає його (див. п. № е).

и) Витягнути – витягає секретне повідомлення із заповненого контейнера (див. п. № г, м).

і) Очистити – очищає текстові поля та обидва контейнери (див. п. № б, г, е, ґ)

ї) Стегоаналіз – відкриває вікно візуального стегоаналізу. Використовується для візуального аналізу контейнера за трьома компонентами (Red, Green, Blue) – див. рис. 3.11.

й) Завантажити – завантажує попередньо заповнений контейнер.

к) Зберегти – зберігає заповнений контейнер.

л) Показує координати X, Y та код пікселю (R, G, B) порожнього контейнера.

м) Показує координати X, Y та код пікселю (R, G, B) заповненого контейнера.

Кнопка «Стегоаналіз» веде на інший екран, де відбувається візуальний аналіз кожного каналу контейнера (Red, Green, Blue) – див. рис. 3.11.

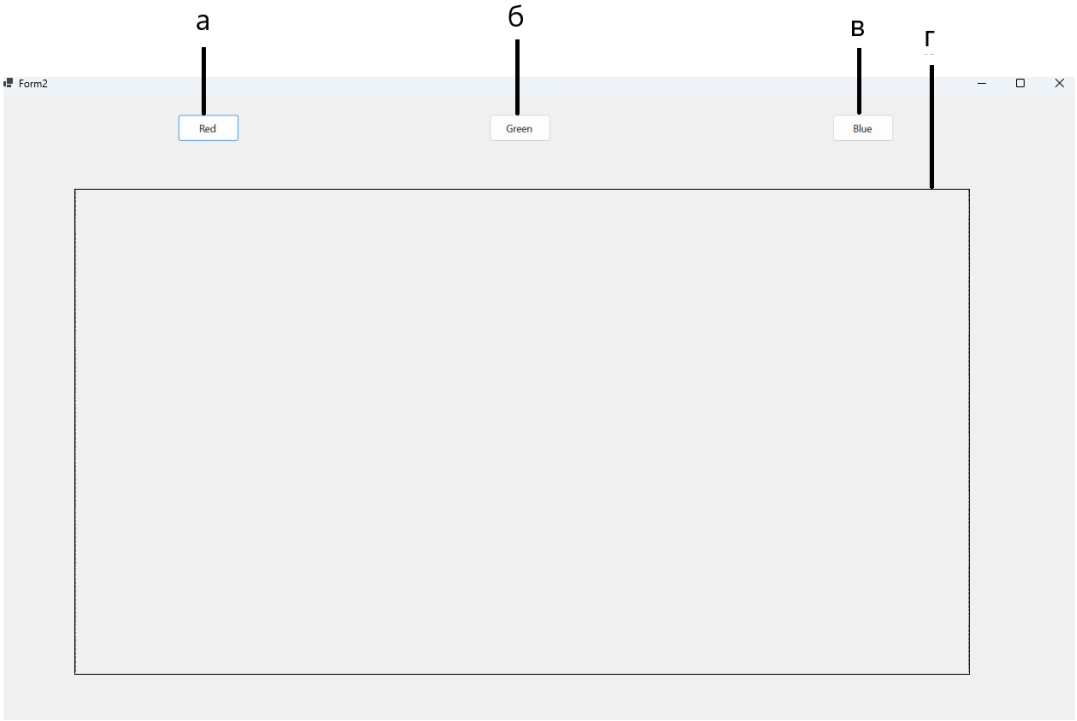


Рисунок 3.11 – Інтерфейс вікна стегоаналізу

а) Red - показує червоний канал заповненого контейнера.

б) Green - показує зелений канал заповненого контейнера.

в) Blue - показує синій канал заповненого контейнера.

г) Поле візуального аналізу контейнера.

3.7 Висновки за розділом

Для розробки засобів було обрано середовище Visual Studio 2022 та мову програмування C#. Для створення графічного інтерфейсу використано технологію Windows Forms .NET. Розроблено блок-схеми узагальнених алгоритмів та програмне забезпечення для роботи в режимах приховування, отримання повідомлення та візуального аналізу за допомогою зрізу по молодшим бітам. Проведена перевірка працездатності програми. Складена інструкція з використання цього програмного комплексу.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ ЗРІЗУ ПО МОЛОДШИМ БІТАМ

Для проведення експериментів було взято 5 зображень. Зображення під номером 1 розміром 22x18 намальовано самостійно в редакторі Paint (див. рис. 4.1).

Рисунок 4.1 – Зображення номер 1

Зображення під номерами 2-5 були взяті з [14] (див. рис. 4.2 – 4.3). Усі зображення вибиралися з урахуванням їхніх характеристик, щоб мінімізувати помітність прихованих даних. Фотографії з великою кількістю деталей і текстур наприклад (трава, міські пейзажі) краще підходять, оскільки зміни у найменш значущих бітах стають менш помітними. Однорідні області, такі як небо або однотонні стіни, більш схильні до візуального виявлення змін. Зображення з високою роздільною здатністю та глибиною кольору (наприклад, 24-бітні RGB) забезпечують більше місця для вбудовування даних.

а

б

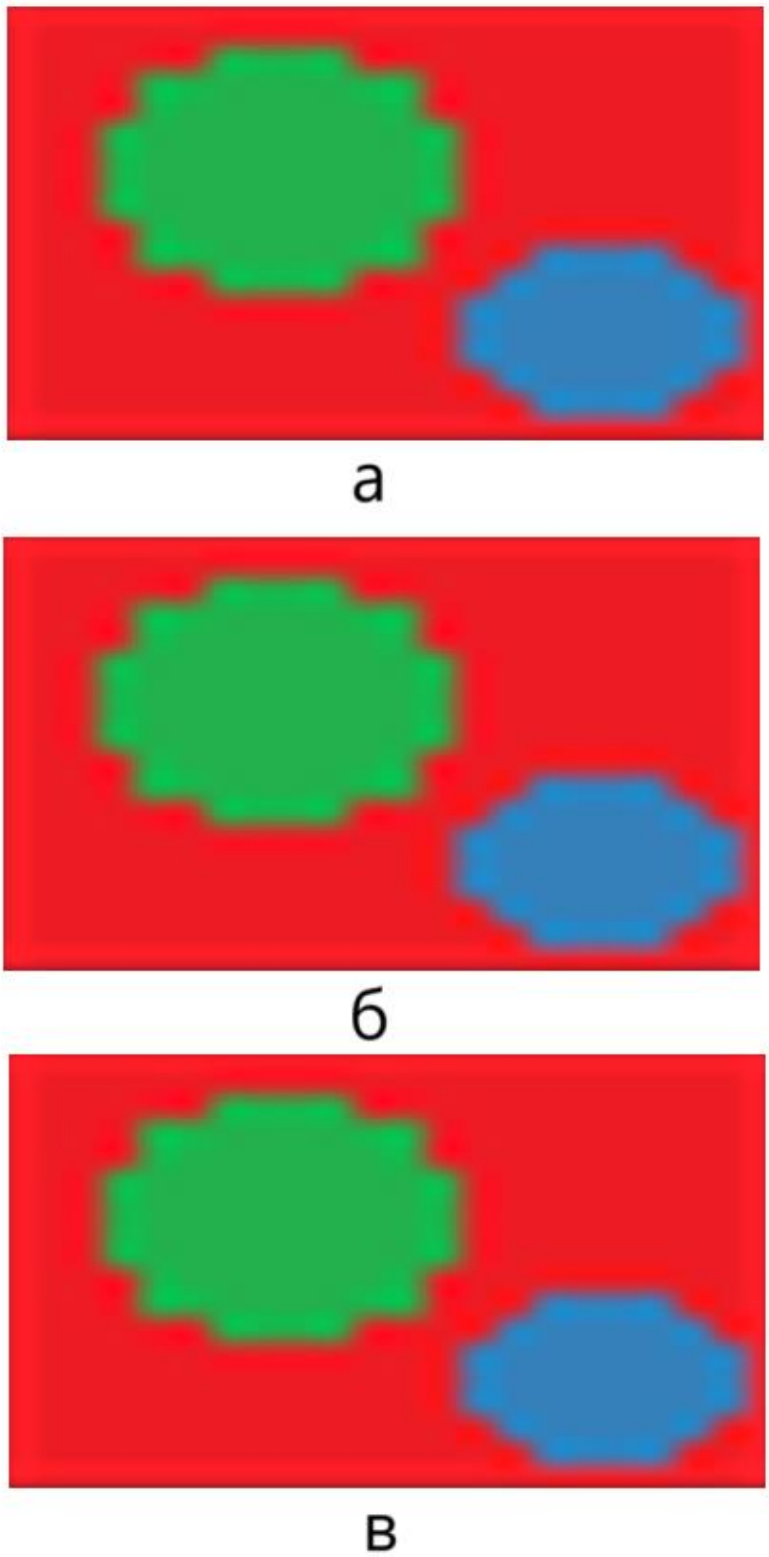
в

зображення 360х240 (а); зображення 400х183 (б); зображення 434х289(в)

Рисунок 4.2 – Зображення номер 2, 3, 4

Рисунок 4.3 – Зображення номер 5 600x400

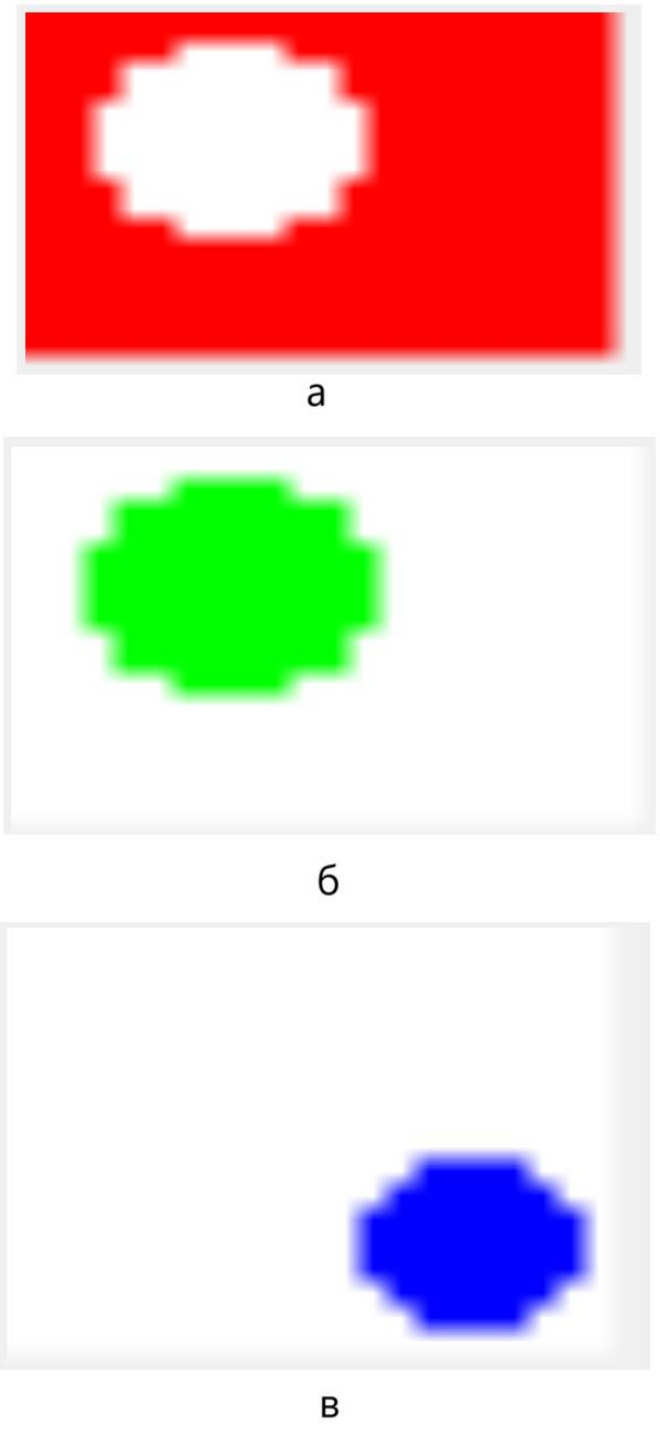
Кожен контейнер зображення №1 буде заповнено чвертю максимальної довжини, половиною максимальної довжини та максимально можливою довжиною повідомлення. Для зображення №1, результати наведено на рис. 4.4.



заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.4 – Заповнення контейнера

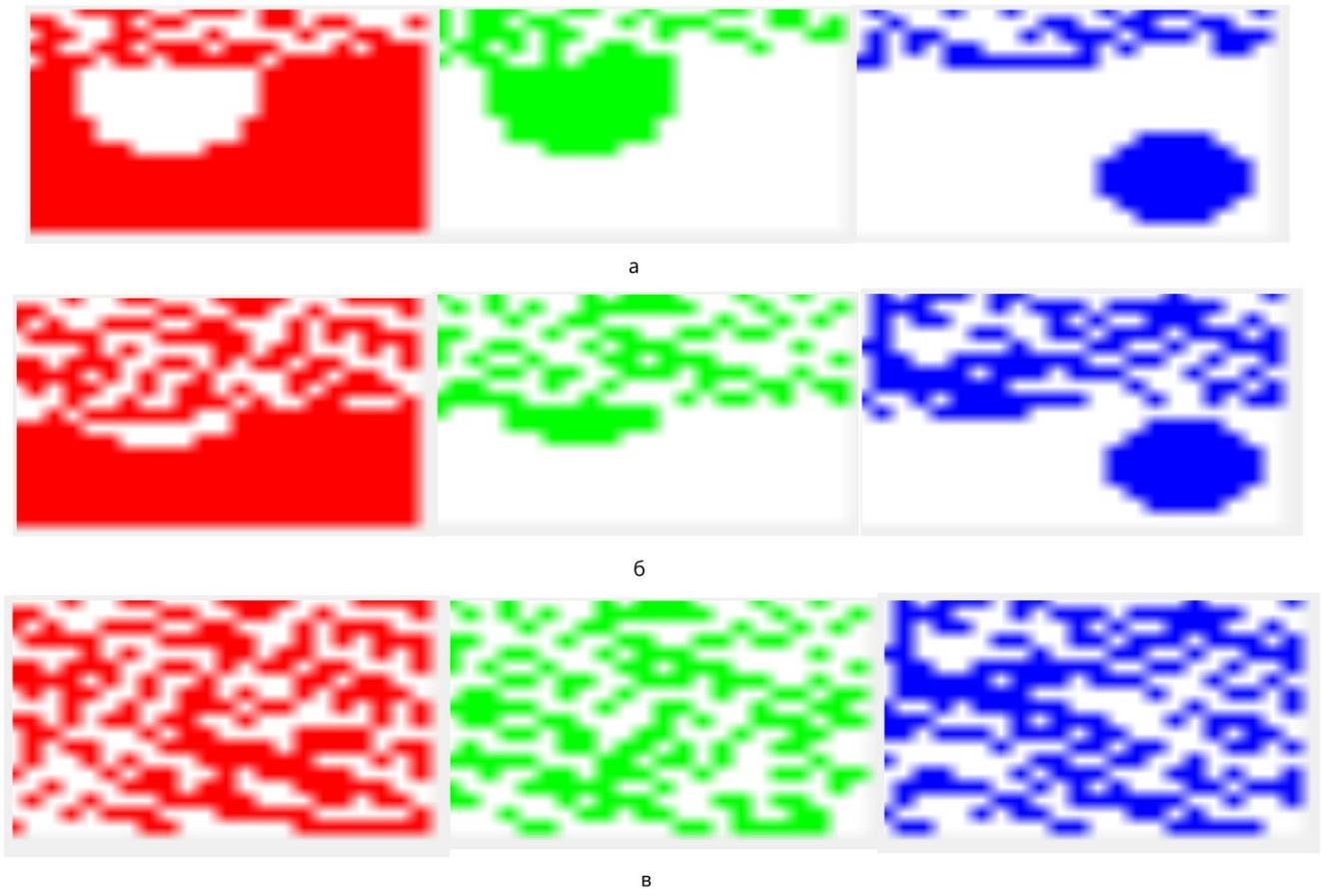
Результат зрізу за молодшими бітами зображення №1 без повідомлення наведено на рис. 4.5.



червоний канал (а); зелений канал (б); синій канал (в)

Рисунок 4.5 – Зріз за молодшими бітами контейнера без повідомлення

Результат зрізу за молодшими бітами з різною довжиною прихованого повідомлення у кольорових каналах. Результати наведено на рис. 4.6.



заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.6 – Зріз за молодшими бітами контейнера з повідомлення

В зображенні №1 (без напівтонів) з інтенсивними складовими по (R, G, B) факт наявності повідомлення можна визначити за тим, що зріз по молодших бітах дає хаотичні зображення.

Кожен контейнер зображення №2 буде заповнено чвертю максимальної довжини, половиною максимальної довжини та максимально можливою довжиною повідомлення. Для зображення №2, результати наведено на рис. 4.7.

а

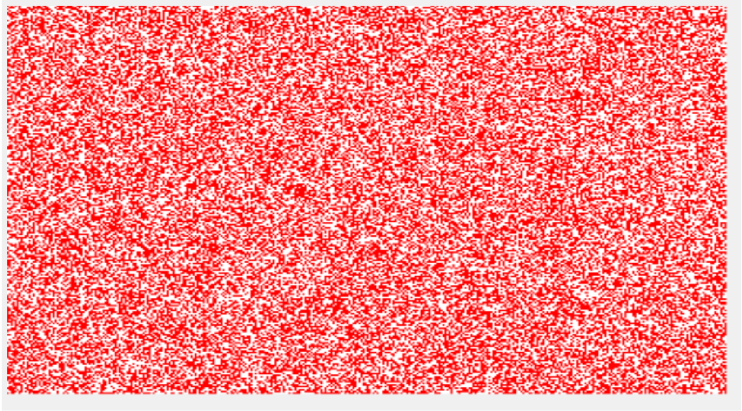
б

в

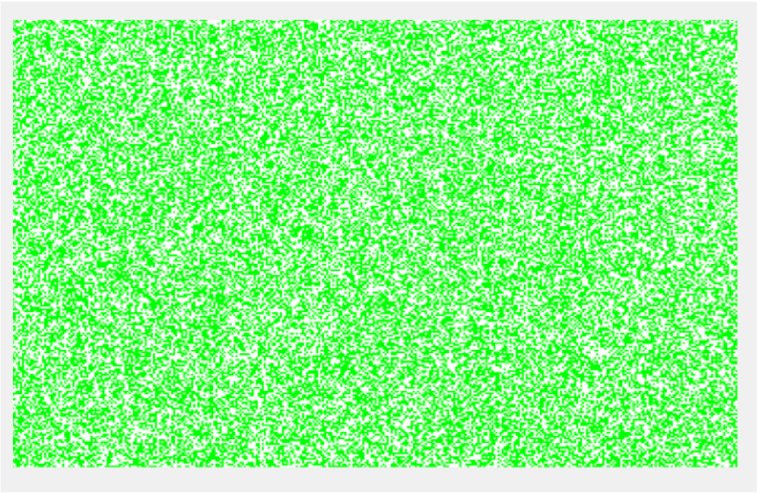
заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.7 – Заповнення контейнера

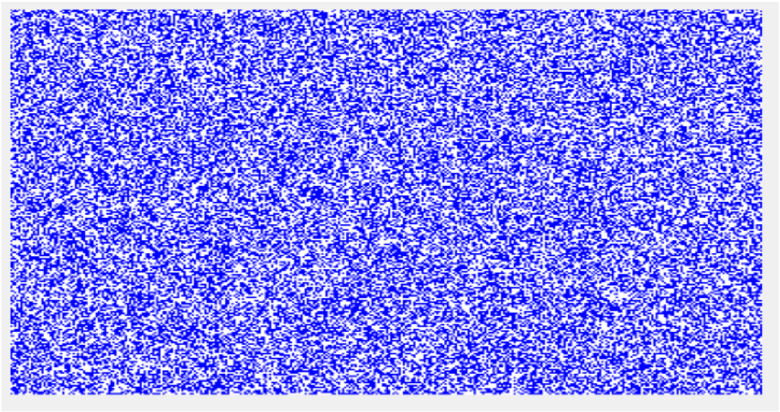
Результат зрізу за молодшими бітами зображення №2 без повідомлення наведено на рис. 4.8.



а



б

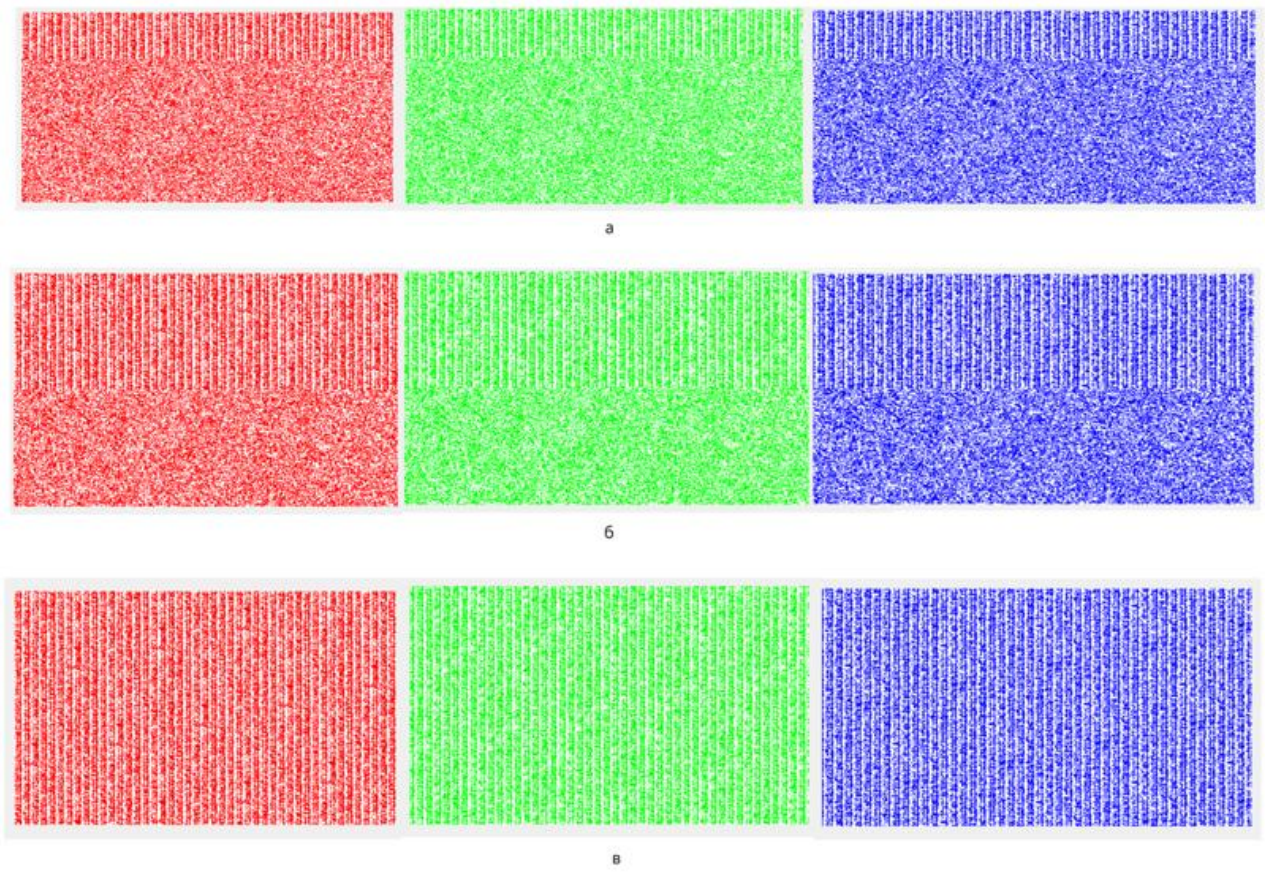


в

червоний канал (а); зелений канал (б); синій канал (в)

Рисунок 4.8 – Зріз за молодшими бітами контейнера без повідомлення

Результат зрізу за молодшими бітами з різною довжиною прихованого повідомлення у кольорових каналах. Результати наведено на рис. 4.9.



заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.9 – Зріз за молодшими бітами контейнера з повідомлення

У зображенні №2, при використанні зрізу за молодшими бітами для заповнення контейнера повідомленням, спостерігається наступне: у варіанті (а) 75% точок утворюють випадкову послідовність, тоді як решта 25% є невипадковими. У варіанті (б) 50% точок формують випадкову послідовність, а інші 50% – невипадкові. У варіанті (в) 100% точок створюють невипадкову послідовність.

Кожен контейнер зображення №3 буде заповнено чвертю максимальної довжини, половиною максимальної довжини та максимально можливою довжиною повідомлення. Для зображення №3, результати наведено на рис. 4.10.

а

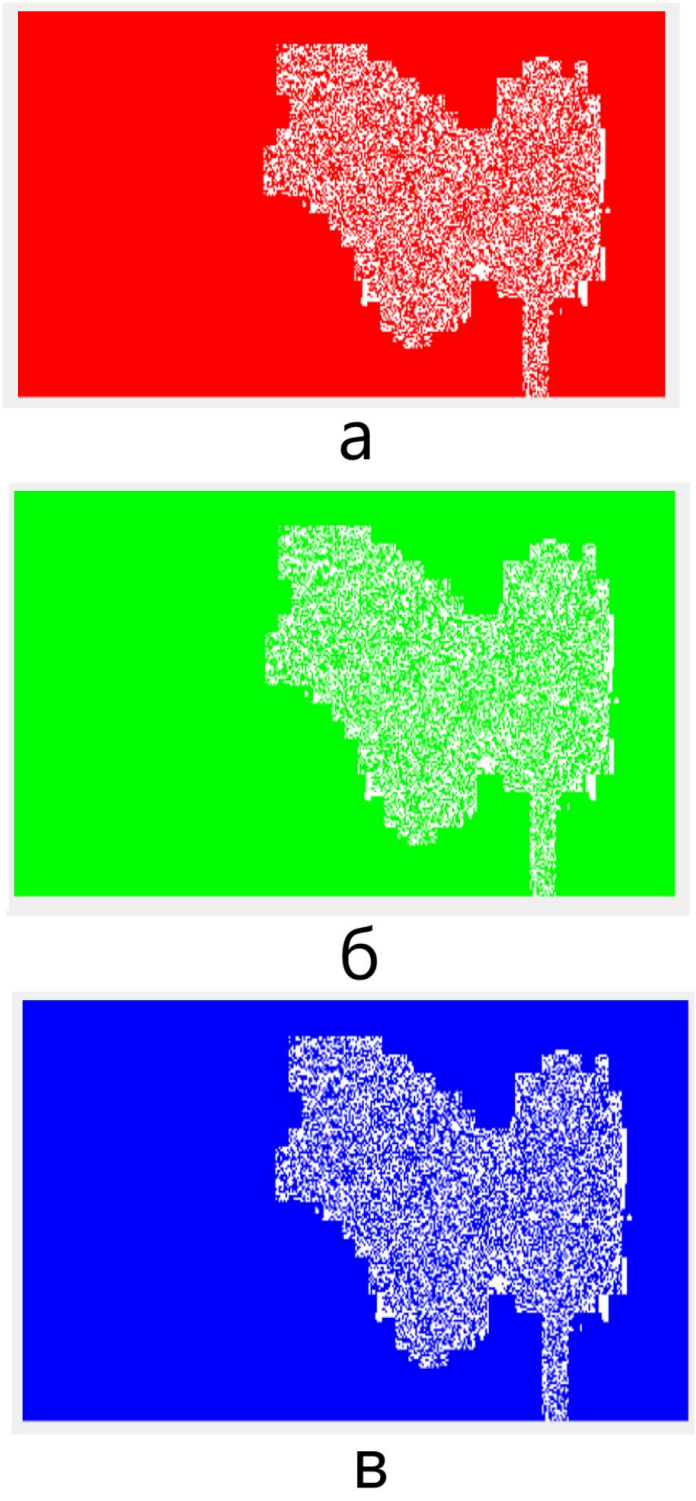
б

в

заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.10 – Заповнення контейнера

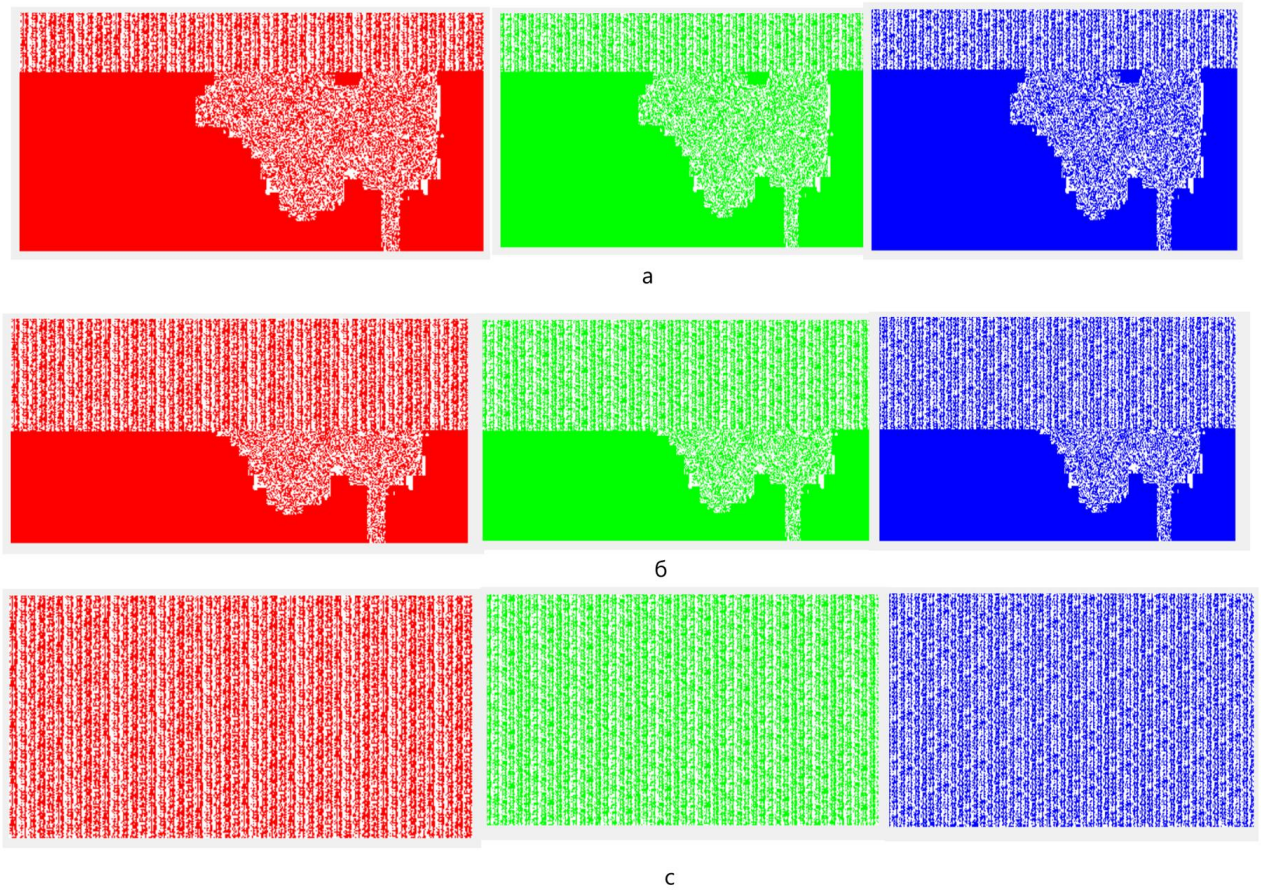
Результат зрізу за молодшими бітами зображення №1 без повідомлення наведено на рис. 4.11.



червоний канал (а); зелений канал (б); синій канал (в)

Рисунок 4.11 – Зріз за молодшими бітами контейнера без повідомлення

Результат зрізу за молодшими бітами з різною довжиною прихованого повідомлення у кольорових каналах. Результати наведено на рис. 4.12.



а

б

с

заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.12 – Зріз за молодшими бітами контейнера з повідомлення

У зображенні №3, при використанні зрізу за молодшими бітами для заповнення контейнера повідомленням, спостерігається таке: у варіанті (а) 75% точок утворюють випадкову послідовність, а решта 25% є невипадковими. У варіанті (б) 50% точок формують випадкову послідовність, тоді як інші 50% є невипадковими. У варіанті (в) 100% точок створюють невипадкову послідовність.

Кожен контейнер зображення №4 буде заповнено чвертю максимальної довжини, половиною максимальної довжини та максимально можливою довжиною повідомлення. Для зображення №4, результати наведено на рис. 4.13.

а

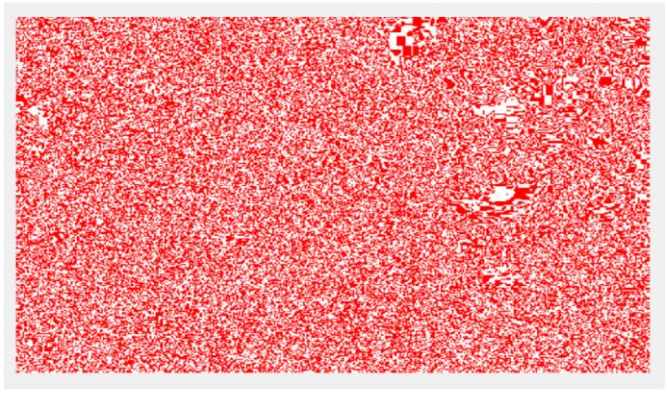
б

в

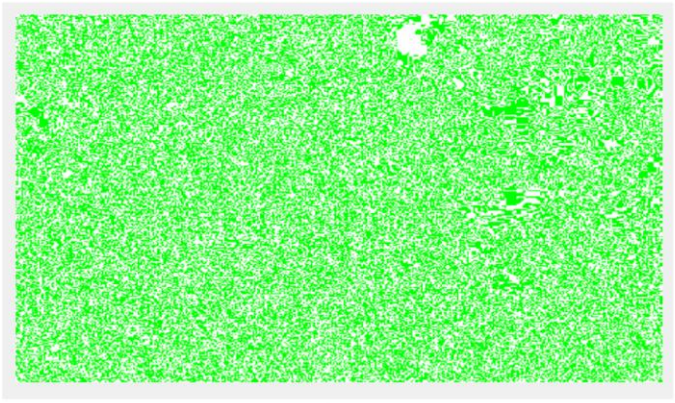
заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.13 – Заповнення контейнера

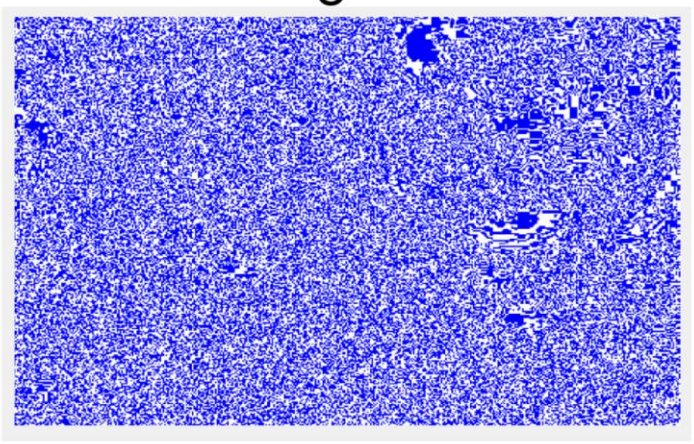
Результат зрізу за молодшими бітами зображення №5 без повідомлення наведено на рис. 4.14.



а



б

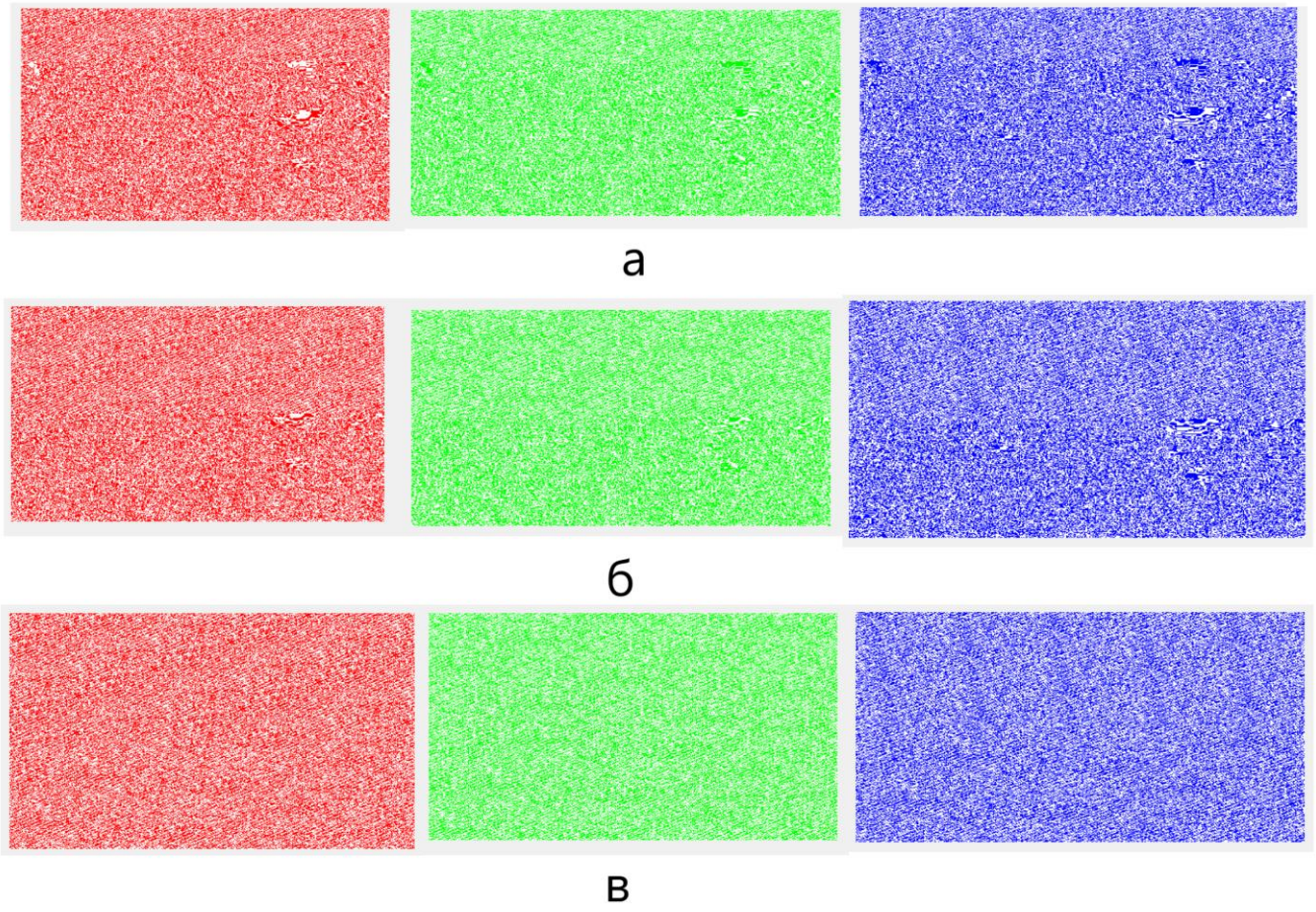


в

червоний канал (а); зелений канал (б); синій канал (в)

Рисунок 4.14 – Зріз за молодшими бітами контейнера без повідомлення

Результат зрізу за молодшими бітами з різною довжиною прихованого повідомлення у кольорових каналах. Результати наведено на рис. 4.15.

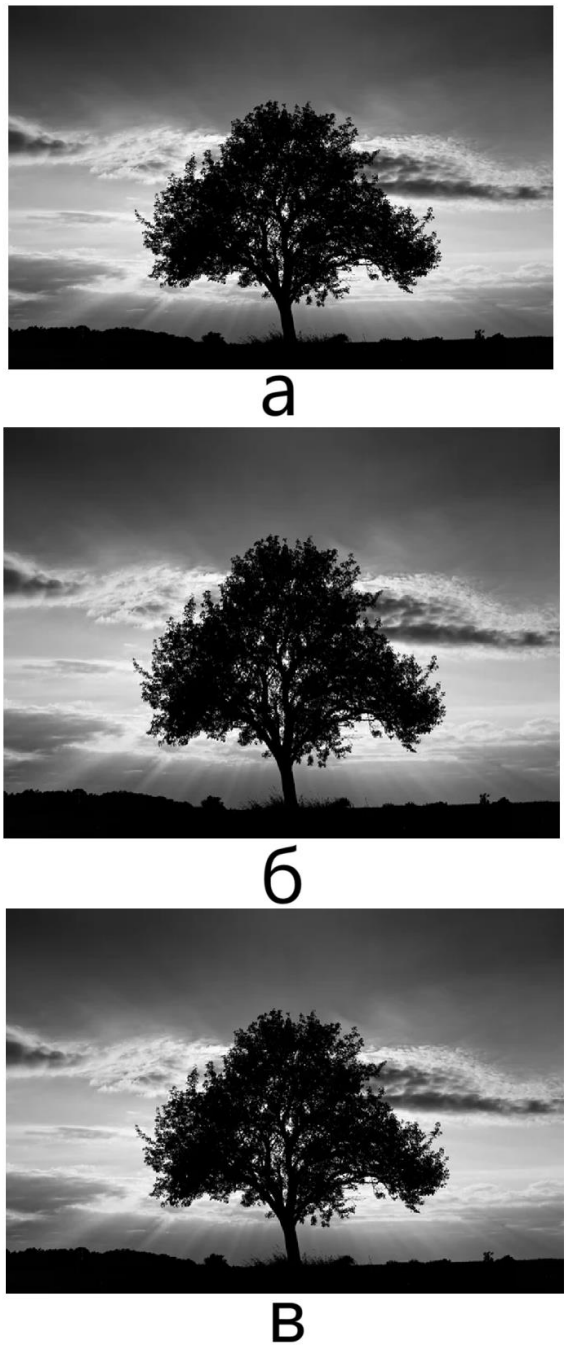


заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.15 – Зріз за молодшими бітами контейнера з повідомлення

У зображенні №4, при використанні зрізу за молодшими бітами для заповнення контейнера повідомленням, спостерігається таке: у варіанті (а) 75% точок утворюють випадкову послідовність, а решта 25% є невипадковими. У варіанті (б) 50% точок формують випадкову послідовність, а інші 50% є невипадковими. У варіанті (в) 100% точок складають невипадкову послідовність.

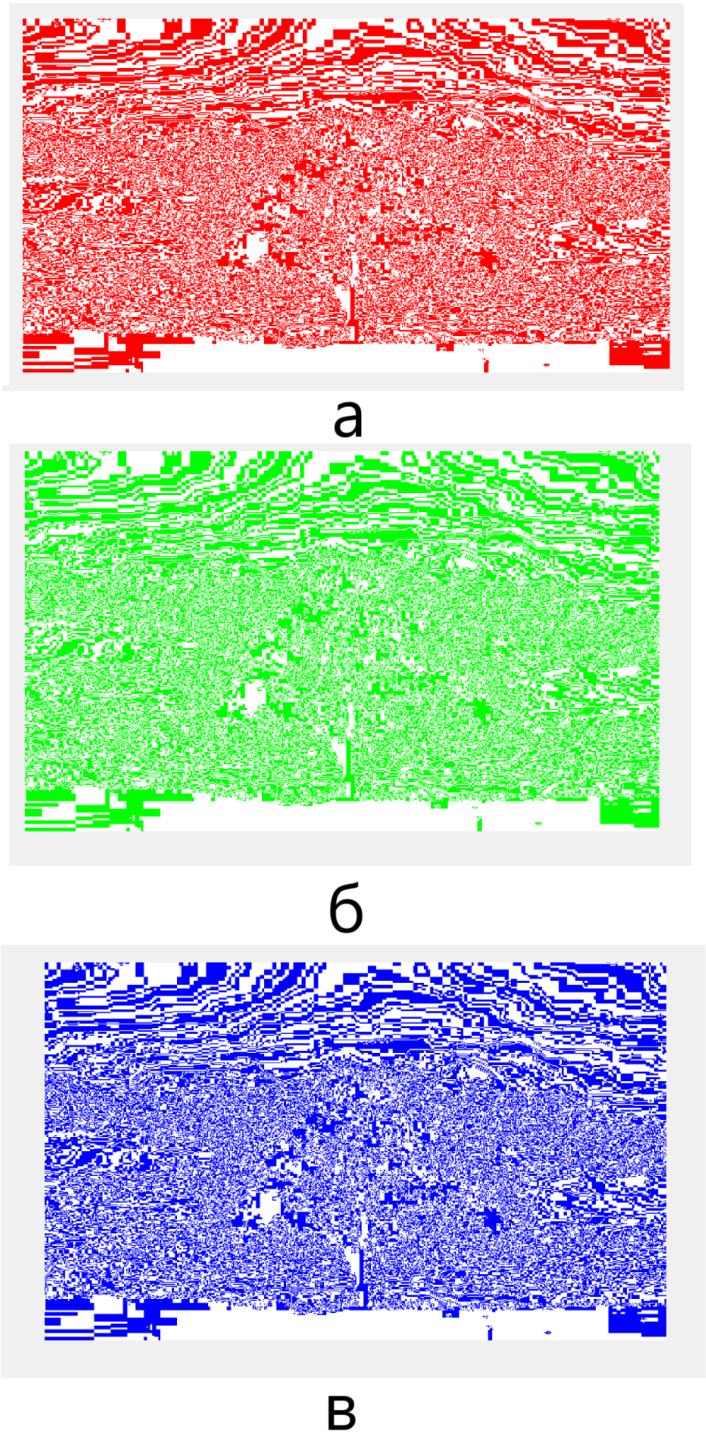
Кожен контейнер зображення №5 буде заповнено чвертю максимальної довжини, половиною максимальної довжини та максимально можливою довжиною повідомлення. Для зображення №5, результати наведено на рис. 4.16.



заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.16 – Заповнення контейнера

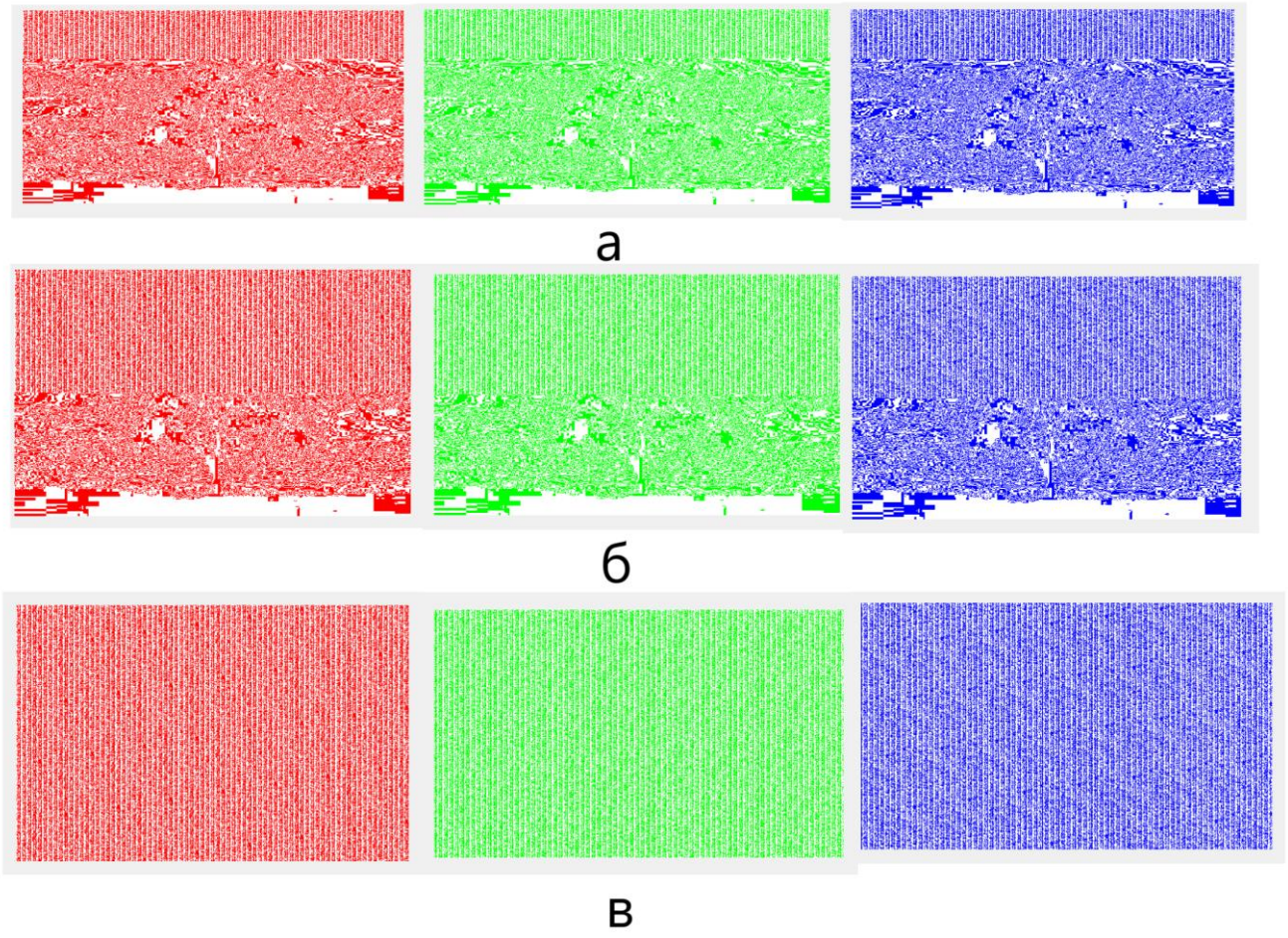
Результат зрізу за молодшими бітами зображення №5 без повідомлення наведено на рис. 4.17



червоний канал (а); зелений канал (б); синій канал (в)

Рисунок 4.17 – Зріз за молодшими бітами контейнера без повідомлення

Результат зрізу за молодшими бітами з різною довжиною прихованого повідомлення у кольорових каналах. Результати наведено на рис. 4.18



заповнення контейнера на 25% (а); заповнення контейнера на 50% (б); заповнення контейнера на 100% (в)

Рисунок 4.18 – Зріз за молодшими бітами контейнера з повідомлення

У зображенні №5, при використанні зрізу за молодшими бітами для заповнення контейнера повідомленням, спостерігається таке: у варіанті (а) 75% точок утворюють випадкову послідовність, а решта 25% є невипадковими. У варіанті (б) 50% точок формують випадкову послідовність, а інші 50% є невипадковими. У варіанті (в) 100% точок складають невипадкову послідовність.

4.1 Висновки за розділом

Сформовано набір зображень для проведення експериментів. У контейнери було вкладено приховані повідомлення різної довжини, які займають 25%, 50% та 100% пропускної здатності контейнера. Проведено серію експериментів із

визначення ефективності використання зрізу за молодшими бітами для заданих умов.

ВИСНОВКИ

У цій роботі розроблено програмні засоби, що забезпечують реалізацію та демонстрацію процесів стеганографічного приховування / вилучення даних та стеганографічного аналізу.

Розглянуто основи стеганографії, описано та проведено порівняння різних методів стеганографії та стегоаналізу. Для реалізації в роботі обрано методи LSB та LSB slicing.

Описано функціонування комплексу засобів у трьох режимах: приховування, отримання прихованого повідомлення та зрізу по молодших бітах. Розроблено структури даних, приведено структуру стоп-секвенції прихованого повідомлення.

Обрано інструменти розробки, мову програмування та технології для створення графічного інтерфейсу комплексу засобів.

Розроблено блок-схеми алгоритмів роботи комплексу засобів для режимів приховування, отримання повідомлень та зрізу по молодших бітах. Проведено тестування функціональності та підготовлено інструкцію з використання комплексу.

Розроблені програмні засоби можуть використовуватися на практиці для приховування повідомлень у графічні контейнери BMP або їх стеганографічного аналізу, а також в цілях навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дверіс О.Е. Дослідження та розробка засобів демонстрації стеганографії та стегоаналізу. *Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті*: Тези XVI Міжнародної науково-практичної конференції, м. Дніпро, 14-15 груд. 2022 р. Дніпро: ДІІТ, 2022. С. 144.

2. Дверіс О.Е., Остапець Д.О. Засоби демонстрації стеганографії та стегоаналізу. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті: Тези XVIII Міжнародної науково-практичної конференції, м. Дніпро, 12-13 грудня 2024р. Дніпро: УДУНТ, 2024. С. 179.

3. Хорошко В.О., Яремчук Ю.Є., Карпінець В.В. Комп'ютерна стеганографія – В: ВНТУ,2017. – 155 с.

4. Колобова А.К., Колобов Д.Г., Герасимов А.С. Стеганография от древности до наших дней. *Безпека інформаційних технологій.* 2015. №4. С.76-79.

5. КонаховичГ.Ф., КлимчукВ.П., Паук С.М., Потапов В.Г. Защита информации в телекоммуникационных системах. – К.: "МК-Пресс", 2005. – 288 с.

6. Кузнецов О.О. Стеганографія : навчальний посібник / О.О. Кузнецов, С.П. Євсеєв, О.Г. Король. – Х. : Вид. ХНЕУ, 2011. – 232 с.

7. Стасюк О.І., Гнатюк С.О., Довгич Н.І., Літош М.С. Сучасні стеганографічні методи захисту інформації. Науково-технічний журнал «Захист інформації», 2011 №1 (50).С. 56-63.

8. КонаховичГ.Ф.,ПрогоновД.О.,ПузиренкоО.Ю.Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: підручник – К: «Центр учбової літератури», 2018. – 558 с.

9. Маценко В.Г. Комп’ютерна графіка: Навчальний посібник. – Ч: Рута, 2009 – 343 с.

10. Смірнов О.А., Мелешко Є.В. Дослідження методів стегоаналізу цифрових зображень. *Збірник наукових праць Центру воєнно-стратегічних досліджень*

*НУОУ імені Івана Черняховського. 2023. №1 (77).С. 92-98.*URL: звернення: 28.12.2024).

11. What's new in Visual Studio 2022. *Microsoft Learn*. URL: https://learn.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2022?view=vs-2022 (дата звернення 30.11.2024).

12. A tour of the C# language. *Microsoft Learn*. URL: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/ (дата звернення 30.11.2024).

13. Desktop Guide (Windows Forms .NET). *Microsoft Learn*. URL: https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-6.0 (дата звернення 30.11.2024).

14. *Depositphotos* URL: https://depositphotos.com/ua/ (дата звернення: 20.11.2024).

ДОДАТОК А

Вихідний код програми