НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут телекомунікаційних систем

Кафедра Інформаційно-телекомунікаційних мереж

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Лариса ГЛОБА
«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційно-комунікаційні технології»
зі спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Модифікований спосіб гібридного шифрування»

Виконав:
студент VI курсу, групи ТІ-91мп
Квашенко Ілля Валерійович _____

Керівник:
Доцент кафедри ІТМ ІТС, доцент, к.т.н.
Кононова Ірина Віталіївна _____

Рецензент:
Доцент кафедри ТК ІТС, доцент, к.т.н.
Явіся Валерій Сергійович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Інформаційно-телекомунікаційних мереж

Рівень вищої освіти – другий (магістерський)
Спеціальність – 172 «Телекомунікації та радіотехніка»
Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Лариса ГЛОБА
«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Квашенку Іллі Валерійовичу

1. Тема дисертації «Модифікований спосіб гібридного шифрування», науковий керівник дисертації доцент кафедри інформаційно-телекомунікаційних мереж ІТС Кононова Ірина Віталіївна, доцент, к.т.н., затверджені наказом по університету від «03» листопада 2020 р. № 3208-с

2. Термін подання студентом дисертації 10.12.2020 р.

3. Об’єкт дослідження - програмний засіб стеганографічного підпису електронного зображення.

4. Предмет дослідження - методи та засоби впровадження інформації в зображення.

5. Перелік завдань, які потрібно розробити:

– проаналізувати методи впровадження інформації в зображення різних форматів;

– дослідити існуючі програми для впровадження інформації в зображення;

– удосконалити метод впровадження інформації в зображення;

– розробити конкурентоспроможну програму на основі розробленого методу;

– протестувати розроблений програмний додаток.

6. Орієнтовний перелік ілюстративного матеріалу

6.1 Актуальність теми дослідження;

6.2 Мета та завдання дослідження;

6.3 Наукова новизна;

6.4 Аналіз предметної області;

6.5 Оцінка стікості стеганографії;

6.6 Розробка програмного засобу стеганографічного підпису електронного зображення

6.7 Результат роботи

7. Дата видачі завдання 03.09.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз впровадження методів впровадження інформації в зображення різних форматів	03.09.2020 – 21.09.2020	виконано
2	Дослідження існуючих програм для впровадження інформації в зображення;	21.09.2020 – 01.10.2020	виконано
3	Аналіз оцінки стійкості стеганографії	01.10.2020 – 21.10.2020	виконано
4	Розробка програмного засобу стеганографічного підпису електронного зображення, тестування та аналіз результатів роботи програми	21.10.2020 – 21.11.2020	виконано
5	Техніко–економічне обґрунтування витрат на виконання розробки засобу стеганографічного підпису електронного зображення	21.11.2020 – 28.11.2020	виконано
6	Розробка стартап-проекту	21.11.2020 – 28.11.2020	виконано
7	Підготування графічного матеріалу	28.11.2020 – 01.12.2020	виконано

Студент Ілля КВАШЕНКО

Науковий керівник дисертації Ірина КОНОНОВА

РЕФЕРАТ

Пояснювальна записка до дипломної роботи на тему «Модифікований спосіб гібридного шифрування» складається з 100 сторінок та містить 34 рисунків, 10 таблиць, 49 літературних джерел.

Об'єкт розробки – програмний засіб стеганографічного підпису електронного зображення.

Мета роботи – підвищення ефективності існуючого процесу ведення договірної діяльності, зниження витрат часу та впорядкування бази контрагентів підприємства шляхом впровадження розроблюваного програмного засобу.

Метод дослідження – розробка програми для приховування текстової інформації в зображенні за допомогою модифікованих алгоритмів впровадження.

В процесі роботи було виконано глибокий аналіз предметної області. В результаті чого виявилося, що використання криптографії залишається одним з найважливіших аспектів для подальших розробок і вдосконалень програми. Ідентичним за важливістю використання криптографії аспектом є розробка більш стійких до стегоаналізу алгоритмів впровадження інформації для приховування та використання в якості контейнерів і стегоповідомлень більшого числа форматів не тільки растрових зображень і текстової інформації, але і файлів абсолютно будь-яких форматів.

Результатами дипломної роботи є методика та програмний засіб, що можуть бути використані для збільшення ефективності приховування текстової інформації в зображенні за допомогою модифікованих алгоритмів впровадження.

Ключові слова: стеганографія, електронний підпис, програмне забезпечення, автоматизація, інформація.

ABSTRACT

Explanatory note to the thesis on the topic “Hybrid encryption modified method” consists of 100 pages and contains 34 pictures, 10 tables, 49 literary sources.

The object of the development is a program of the stenographic signature of the student's electronic report.

The purpose of the work is to increase the efficiency of the existing process of contractual activity, reduce the time costs and streamline the contractor's base of the company by introducing the developed software.

The research method is the development of a program for concealing textual information in the image using modified implementation algorithms.

In the process of work, a deep analysis of the subject area was performed. As a result, it turned out that the use of cryptography remains one of the most important aspects for further development and improvement of the program. Identical for the importance of the use of cryptography aspect is the development of more stable to steganalization algorithms for the introduction of information to conceal and use as containers and stencils of a large number of formats not only raster images and text information, but also files of absolutely any format.

The results of the thesis are a method and a software tool that can be used to increase the effectiveness of hiding text information in the image with the help of modified implementation algorithms.

The keywords: ELECTRONIC SIGNATURE, SOFTWARE, AUTOMATION, INFORMATION, STEGANOGRAPHY.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. Завдання захисту інформації в усі часи було актуальним в багатьох сферах життя. Це завдання умовно ділиться на два напрямки: стенографія і криптографія. Методи стеганографії спрямовані на приховування факту передачі або зберігання інформації. Метою криптографії є шифрування самої інформації. Ці напрямки застосовуються як разом, так і порізно [1].

Для захисту інформації, а точніше приховування самого факту захисту, в цифровий стенографії використовуються контейнери – цифрові об'єкти, куди впроваджується інформація, викликаючи при цьому деяке їх спотворення.

Найчастіше контейнерами є мультимедіа-об'єкти (зображення, відео, аудіо). Спотворення, що вносяться до контейнеру і які захищаються, знаходяться нижче порога чутливості людини, тому їх неможливо помітити без застосування спеціальних апаратних методів. У даній роботі в якості контейнерів використовуються формати зображень з втратою і без втрати якості.

Інформація, захищена таким чином, залишається захищеною тільки до моменту її виявлення. Зміст інформації у разі не захищений, тобто передані дані не зашифровані, що дозволяє без труднощів стати власником цих даних, але тільки за умови, що зловмисник виявляє присутність потенційно корисної інформації, яка приховується в контейнері, що вельми неочевидно і викладає занадто багато роботи.

Найчастіше стеганографічні і криптографічні методи застосовуються разом, що створює потужну систему захисту інформації [2].

Використання криптографії залишається одним з найважливіших аспектів для подальших розробок і вдосконалень програми.

Ідентичним за важливістю використання криптографії аспектом є розробка більш стійких до стегоаналізу алгоритмів впровадження інформації для приховування та використання в якості контейнерів і стегоповідомлень більшого числа форматів не тільки растрових зображень і текстової інформації, але і файлів абсолютно будь-яких форматів.

Мета та завдання дослідження. Виходячи з перерахованого вище була поставлена мета дослідження, а саме розробити програму для приховування текстової інформації в зображенні за допомогою модифікованих алгоритмів впровадження. Для досягнення цієї мети були сформовані наступні завдання:

– проаналізувати методи впровадження інформації в зображення різних форматів;

– дослідити існуючі програми для впровадження інформації в зображення;

– удосконалити метод впровадження інформації в зображення;

– розробити конкурентоспроможну програму на основі розробленого методу;

– протестувати розроблений програмний додаток.

Об'єкт дослідження – система впровадження інформації в зображення.

Предметом роботи є методи та засоби впровадження інформації в зображення.

Методи дослідження. Для вирішення поставлених завдань використовувалися методи і апарат теорії графів, теорії автоматів, математичної статистики, теорії множин.

Наукова новизна роботи полягає в наступному.

1. Вперше запропоновано показники ефективності для алгоритмів впровадження інформації в зображення, які дозволяють чисельно оцінити

вплив рівня надлишковості, створюваної алгоритмом, на характеристики отриманого зображення.

2. Запропоновано алгоритм нанесення інформації в зображення, що відрізняється підвищеною ефективністю, який дозволяє здійснювати операції з нанесення цифрового підпису з можливістю регулювання балансу основних характеристик.

Практична значущість результатів. У роботі досліджено засоби та методи для впровадження інформації в зображення для користувальницької експлуатації. Отримані результати можуть бути використані у межах організацій для розробки програми по впровадженню інформації в зображення.

Структура дипломної роботи. Робота складається зі вступу, 4 розділів, що включають в себе 14 підрозділів, висновків, списку літератури з 49 найменувань, загальний обсяг роботи 100 сторінок.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основи стеганографії

Стеганографія – це спосіб передачі або зберігання інформації таким чином, щоб безпосередньо передача або зберігання секретних даних залишалися засекреченими. [3,5].

Стеганографію, як засіб для передачі та зберігання даних, почали застосовувати давно. До головних первинних засобів можна віднести чорнило, що ставало помітним лише за певних умов. Наприклад: освітлення, нагрівання, вплив хімічних з'єднань або ж завдяки спеціальним трафаретам, які при з'єднанні з текстом відображали потрібну інформацію. Також так звані специфічні шифри, завдяки яким звичайні слова набували секретних значень, мікроточки, що являли собою зменшені у сотні разів фотознімки. Цифрова стеганографія виникла відносно нещодавно, в порівнянні з класичною, її вважають новою галуззю в науці. Цифрова стеганографія працює з прихованою передачею даних, цифровими відбитками та стеганографічними водяними знаками.

Загалом, цифрові відбитки використовують для захисту виключного права. Наприклад, при продажі електронних матеріалів. Для організації захисту в кожну копію додається спеціальна відмітка або цифровий відбиток. Якщо зловмиснику все ж таки вдасться забирати і підробляти цифрові відбитки, то викрити його буде неможливо. Але поки цього не відбулося, він не зможе поширювати дані, що захищені міткою, так званий «контейнер».[4].

Стеганографічні водяні використовуються для того, щоб захистити авторські права. При застосуванні водяних знаків передбачається наявність однакової стеганографічної мітки для кожної копії контейнера.

Класична стеганографія ставить собі за мету – приховану передачу інформації. Сутність полягає в тому, що захист контейнера грає не першочергову роль. Найважливішим є факт прихованої передачі інформації. Оскільки, безпосередньо, існування даних може не бути секретом, а факт передачі має бути таємним.

Для таких цілей потрібно розробити стегосистему згідно з певними принципами:

1) Потенційний шахрай знає про нюанси створення стеганографічної системи. Однак не володіє даними про ключ, за допомогою якого можна отримати інформацію про наявність таємних даних та їх зміст;

2) Оскільки шахрай не має ключа, то виявлення існування впровадженого повідомлення, не впливає на його конфіскацію та на викрадення таких самих повідомлень в інших даних;

3) Шахрай не повинен володіти технічними, програмними чи іншими перевагами у визначенні змісту впроваджених даних.

На рис. 1.1 [2] схематично показана узагальнена стеганографічна система.

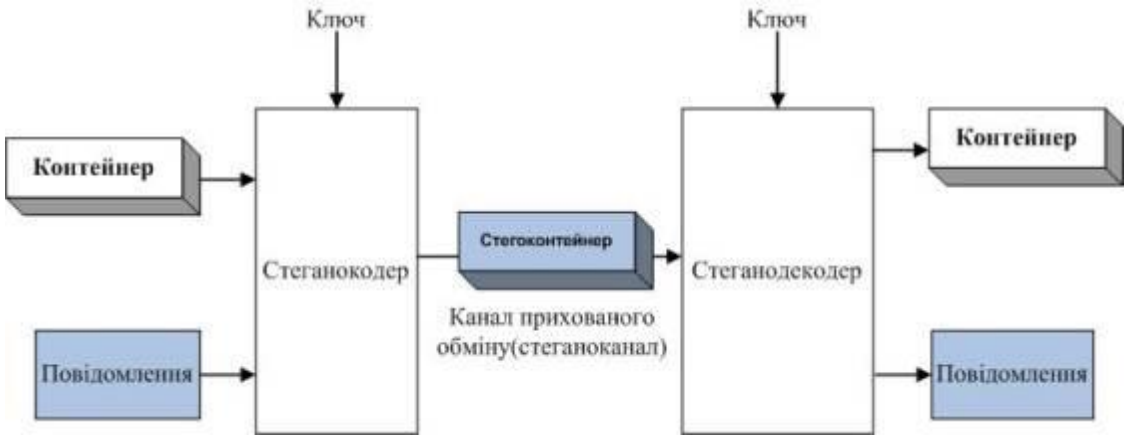


Рис. 1.1 Стеганографічна система

За видом стегоключа, стегосистеми поділяють на: стегосистему з відомим або відкритим ключем і з таємним або закритим.

Стегосистема з закритим ключем функціонує за таким принципом: для обміну таємними повідомленнями, що впроваджені в контейнер, таємний ключ має бути заздалегідь відкритим для всіх учасників обміну або переданий по захищеному каналу.

Стегосистема з відомим ключем містить 2 ключі, один з яких відкритий. Відкритий ключ можна надсилати за допомогою одного каналу зв'язку. Для впровадження та вилучення повідомлення використовуються різні ключи.

Якщо учасники обміну не мають довіри один до одного, то доцільно використовувати саме цю стегосистему.

Також існують стеганографічні системи взагалі без ключа, бо алгоритма перетворення повідомлення та контейнера достатньо для введення необхідних даних. На теперешній час такі системи є найактуальнішими.

1.2 Цифрове подання растрового зображення

Растрове зображення – це зображення, яке представляє собою сукупність точок (а інакше пікселів), які використовуються для його візуалізації на моніторі. У нього є дозвіл, який характеризується кількістю пікселів у фізичну довжину і ширину монітора.

Інформаційна вага (розмір, обсяг) растрового зображення дорівнює добутку кількості пікселів і інформаційної ваги одного пікселя [6].

Так, якщо N - це ширина растрового зображення, M - це довжина растрового зображення, а Q - це інформаційна вага одного пікселя, то інформаційний вага V всього зображення буде дорівнювати:

$$V = N * M * Q$$

На зорі виникнення растрове зображення було чорно-білим, тобто піксель або був підсвічений, або залишався чорним. Це означає, що для того, щоб закодувати інформацію про стан пікселя, необхідний був всього один біт машинного слова, який брав значення або одиниці 1, або нуля 0. Параметр, який характеризує кількість біт, які використовуються для опису одного пікселя, називається глибиною кольору, якістю передачі або бітністю зображення. Так, наприклад, для кодування реєстрового чорно-білого зображення з глибиною кольору k = 1, у якого N = 1000 пікселів, а M = 500 пікселів, був би необхідний обсяг пам'яті V = 1000 * 500 * 1 = 500000 біт або 62,5 кілобайт.

Такі цифри зараз неможливо уявити, так як технології давно зробили крок вперед, а зображення змінилися, і, отже, змінився їх розмір. Монохромні зображення бувають не тільки чорно-білими, де глибина кольору k = 1. При значенні k> 1 в зображенні з'являються градації сірого. Так, еволюція монохромного зображення почалася з глибини кольору k = 1, далі з'явилися зображення з k = 2, де було чотири можливих відтінки кольору, так як 2 2 = 4. Тобто колір одного пікселя можна було закодувати не одним бітом, який брав значення одного з двох кодів в двійковій системі (1 або 0), а двома бітами, які могли приймати значення одного з чотирьох кодів в двійковій системі (00, 01, 10, 11). Далі з'явилася можливість створення зображень з якістю передачі k = 3, k = 4, k = 8 і k = 16. У призначеному для користувача поданні буває досить використання зображень з бітністю k = 8 для правильного відображення.

Збільшення глибини кольору призвело до того, що кількість можливих кольорів стало непрактично велике. З'явилася колірна схема TrueColor, де для представлення елементів палітри почали використовуватися кодовані червона, зелена і синя складові кольору, тобто RGB-модель. Кольорові зображення з глибиною k = 8 біт могли використовувати 256 різних кольорів.

На червону складову кольору відводилося три біта, тобто всього 2 3 = 8 а червону складову кольору відводилося три біта, тобто всього 2 3 = 8 а червону складову кольору відводилося три біта, тобто всього 2 3 = 8 відтінків могло використовуватися в зображенні. Зелена складова мала також 8 відтінків. Синя складова мала два біта, тобто чотири можливих відтінки. Кольорові зображення з глибиною k = 12 біт могли використовувати 4096 різних кольорів, так як на кожну складову кольору відводилося по 4 біта, що відповідає 2 4 = 16 можливим відтінкам для кожної складової. Зображення з глибиною кольору 12 біт іноді використовуються в простих пристроях з кольоровим дисплеєм. Використання глибини кольору k = 15, k = 16, k = 18 біт в зображеннях називається технікою HighColor. Найбільш зручними для сприйняття і наближеними до кольорів реального світу є зображення з глибиною кольору k = 24, тобто у кожного каналу кольоровості довжина 8 біт і, відповідно, по 2 8 = 256 різних відтінків кожен канал може уявити. Такий спосіб подачі кольорів називається TrueColor і включає в себе можливість подання 16777216 кольорів. Так, наприклад, для кодування растрового зображення з глибиною кольору k = 24, у якого N = 1000 пікселів, а M = 500 пікселів, був би необхідний обсяг пам'яті V = 1000 * 500 * 24 = 12000000 біт або 1,5 мегабайта.

1.3 Формати растрових зображень

Форматів зображень дуже багато, використання одних більш придатне для знаходження оптимального співвідношення між якістю і розміром, використання інших більш придатне для відновлення зіпсованих знімків.

Головною особливістю формату є наявність можливості стиснути зображення. Стиснення можна проводити двома способами: з втратою якості і без втрати якості. Стисле зображення без втрати якості зберігає всю інформацію щодо компресії таким чином, що виникає можливість

відновити початкове зображення без стиснення. із стисненого зображення з втратою якості неможливо відновити вихідне зображення, зате воно має набагато менший розмір, ніж стисле без втрати якості.

Ось перелік деяких форматів зображень:

1) Формат BMP. Особливістю цього формату є той факт, що BMP - один з перших графічних форматів.

2) Формат RAW. Файли цього формату містять оригінальну інформацію, що надходить з матриці і необроблену процесором фотокамери. Цей формат дає величезні можливості по обробці зображення, тому що зберігає максимальну якість, і може бути стиснутий без втрати якості. Розширення у таких файлів залежить від фототехніки, на яку було знято зображення, так як у фототехніці різних виробників винайдено своє розширення під формат RAW.

3) Формат JPEG. Найголовнішою особливістю і визначальною рисою цього формату є його гнучкість, під якою мається на увазі широкий спектр стиснення зображення від максимального до мінімального розміру.

4) Формат JPEG 2000. Даний формат дуже новий і має масу переваг перед своїм попередником JPEG. При однаковій якості зображень розмір файлу в форматі JPEG набагато більший.

Головна відмінність цих форматів в методі стиснення. Саме метод стиснення JPEG при сильній компресії розбиває зображення на характерні квадрати, чого не відбувається при компресії методом JPEG 2000. На даний момент цей формат мало поширений.

5) Формат GIF. З появою мережі Інтернет популярність цього формату різко зросла, його почали використовувати для обміну зображеннями, так як розмір файлу цього формату був компактніших розмірів інших форматів того часу. Підтримує анімацію і компресію без

втрати якості. Недоліком є здатність зберігати зображення з дуже малою глибиною кольору (8 біт).

6) Формат PNG. Цей формат був створений для заміни формату GIF. Він також стискає без втрат, але явною перевагою цього формату є можливість зберігати необмежену кількість кольорів і підтримка прозорості.

7) Формат TIFF. Цей формат в більшості випадків використовується в поліграфії, так як має можливість зберігати зображення в різних колірних просторах і з великою глибиною кольору. При компресії не втрачає якості, через що розмір цих файлів дуже великий.

8) Формат PSD. При роботі в програмі Photoshop цей формат використовується для збереження проміжних результатів трудомісткої обробки зображення. Він має можливість зберігати зображення з шарами, з різною глибиною кольору і в різних колірних просторах. При компресії не втрачає якості, через що велика кількість інформації, що міститься у файлі такого формату, дуже збільшує його розмір.

1.4 Властивості системи людського зору

Людський зір має деякі властивості, які необхідно взяти до уваги при використанні в якості контейнера зображення. Фізіологічні особливості системи зору впливають на низькорівневі властивості. До них відносяться:

1) Відсутність реакції на невелику зміну яскравості. При використанні в якості контейнерів TrueColor-зображення з глибиною кольору 24 біта палітра відтінків дуже плавна, тому сусідні значення кольорів не відрізняються для системи людського зору.

2) Слабка реакція на невеликі зміни контрастності. У зображень реальних об'єктів контрастність або перепад яскравості між двома сусідніми пікселями дуже мала, що говорить про їх сильну корельованість.

3) Слабка реакція на невеликі зміни синього кольору. Це обумовлено тим, що компонента яскравості зображення залежить від червоної, зеленої і синьої компонент колірної моделі, що впливають з різною силою.

4) Різна реакція на різні діапазони частот. Відомо що зміни в низькочастотній складовій зображення будуть сильніше помітні оку, ніж зміни у високочастотній складовій.

З'ясувати ступінь реакції на зміну яскравості і частоти можна простим прикладом з однотонним зображенням як показано на рисунку 1.2. [7]

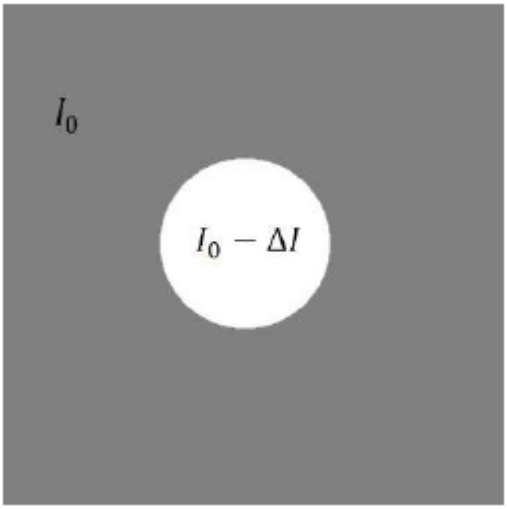


Рис. 1.2 Поріг нерозрізненості

Піддослідному представляють деяке однотонне зображення, на яке він дивиться деякий час, поки його очі не адаптуються до яскравості і зображення. Далі навколо центральної плями зображення починають збільшувати яскравість так, щоб ця зміна Δ I стала помітною випробуваному.

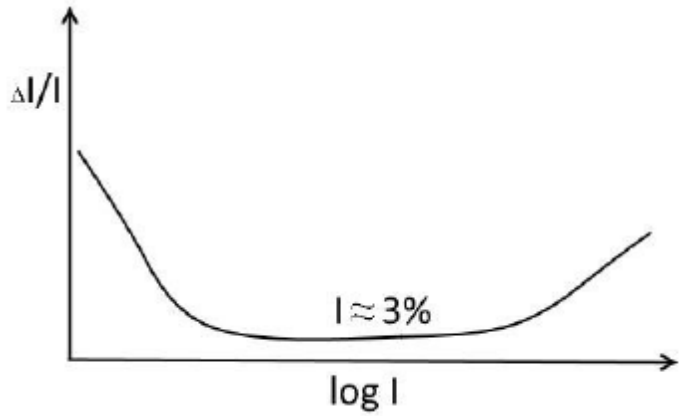


Рис. 1.3 Залежність мінімального контрасту від яскравості

На рис. 1.3 [8] зображена залежність мінімального контрасту зображення від яскравості I / ΔI. Контраст залишається постійним для зміни яскравості середнього діапазону. Коли яскравість занадто мала або велика, поріг нерозрізненості стає зовсім іншим – його значення зростає.

Для середніх значень яскравості $\Delta I \approx 0.01 - 0.03$, як показують дослідження.

Аналогічний досвід проводиться для виявлення ступеня реакції на зміну частоти, тільки навколо центральної плями зображення змінюють частоту. Реакція на зміну частоти відбивається в тому, що вона більш сильна на низькочастотний шум, ніж на високочастотний, що відбувається з-за нерівномірності амплітудно-частотної характеристики системи людського зору.

Особливості будови зорового апарату і особливості роботи нервової системи впливають на високорівневі властивості системи людського зору. До них відносяться:

1) Реакція на колір. Звертають на себе набагато більше уваги яскравіі строкаті деталі в зображенні, ніж тьмяні.

2) Реакція на форму. Менш примітні предмети однорідні і округлі, ніж різкі і довгасті.

3) Реакція на розмір. Звернуть на себе увагу набагато сильніше великі об'єкти.

4) Реакція на місце розташування. Більш примітні об'єкти на передньому плані, ніж на задньому.

5) Реакція на різкість. Чіткі об'єкти звернуть на себе увагу набагато сильніше, ніж розмиті.

6) Реакція на зовнішні подразники. При впливі на людину якогось зовнішнього подразника людина стає необ'єктивною.

Високорівневі властивості більш перспективні для застосування в стеганографії, однак, практично всі алгоритми з вбудовування засновані на низькорівневих властивостях, тому що застосування високорівневих властивостей тягне за собою розробку інтелектуальних методів впровадження даних. [3]

Низькорівневі особливості, зокрема мала чутливість на зміну частоти і контрастності, лягли в основу математичної моделі алгоритму стиснення JPEG і методів вбудовування інформації в зображення з компресією. Відсутність реакції на зміну яскравості лягло в основу групи стеганографічних алгоритмів, які впроваджують інформацію в зображення за допомогою зміни найменш значущого біта.

Висновки

У рамках першого розділу розкрито теоретичну сторону питання стеганографії, визначено основні формати зображень, запропоновано низку методів здатних здійснити стеганографічний запис. Для подальшого розгляду обрано метод LSB у якості формату зображень застосовано bmp файл.

Суть цього методу полягає в заміні останніх значущих бітів в контейнері (зображення, аудіо або відеозапису) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.

Також будемо використовувати для шифрування / дешифрування bmp файл, який не містить палітру. В такому bmp файлі кожні 3 байта визначають 3 кольори пікселя.

РОЗДІЛ 2
ОЦІНКА СТІЙКОСТІ СТЕГАНОГРАФІЇ

2.1 Поняття стійкості стеганографії

Під стійкістю стеганографії звичайно розуміють число («гіпотетичних», «елементарних») операцій, які слід виконати для обчислення секретного ключа з відкритого ключа або, якщо досліджується симетричний алгоритм шифрування, для розшифровки повідомлення без знання ключа. Від числа і характеру «елементарних» операцій безпосередньо залежить час, необхідний для їх виконання. Наприклад, «елементарною операцією» в задачі дешифрування тексту шляхом повного перебору ключів можна вважати процедуру розшифрування тексту одним ключем. Тоді складність всього дешифрування оцінюється як $2^{(\text{довжина ключа})}$, що відповідає «класичній» оцінці складності повного перебору. Як приклад розглянемо схему Ель-Гамаля, та відзначимо, що у згаданій схемі секретним ключем є випадкове число x, а відкритим – пара (g, Y), де [18]

$$Y = g^x \ (\mathrm{mod}\ p) \qquad (2.1)$$

Пошук числа x по відомим Y, g і p називають дискретним логарифмуванням. « Логарифмування » - тому, що треба знайти показник ступеня, а « дискретне » - тому, що завдання вирішується в цілих числах (більше того, в простому полі).

Після опублікування робіт Діффі, Хеллмана, Ель-Гамаля та інших дослідників, завдання дискретного логарифмування опинилося в центрі уваги багатьох математиків і криптоаналітиків. Стійкість підпису безпосередньо залежить від ефективності знаходження дискретного логарифму. Пояснимо алгоритм, на рис. 2.1 « зображено » кінцеве поле[18]. Елементи поля розміщені по «колу», що, власне, відбиває

структуру поля. Десь на «колі» розміщується точка a і точка Y з координатами ax. Десь розташоване і невідоме нам число x.

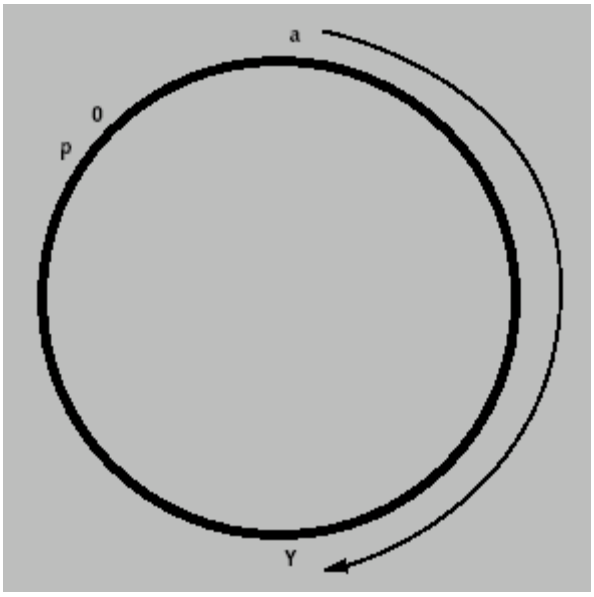


Рис. 2.1 Кінцеве поле

Можна сказати, що завдання дискретного логарифмування еквівалентне задачі пошуку «числа обертів » при обході по колу в «шляху» від точки a до точки Y. Неважко здогадатися, що «великий похід» по колу еквівалентний повному перебору ключів. Існують методи, що оптимізують перебір (зокрема, метод Полларда або метод « стрибаючого кенгуру»). Є ще кілька східних методів, але всі вони мають евристичний характер і вірогідну оцінку складності: при довжині ключа в 512 біт складність дискретного логарифмування буде порядку 2^{256}.

Крім методів «великого походу» розроблений і активно вдосконалюється метод, що дає субекспоненціальну по p оцінку складності. Метод отримав назву «метод відображень» (у літературі можна зустріти назву «метод індексів»). Його модифікований алгоритм для полів многочленів отримав назву по імені автора – алгоритм Копперсміта [22].

Оригінальний метод відображень застосуємо для будь-якого числового поля GF (pn), тоді складність завдання дискретного логарифмування має порядок 2^{180}.

Стверджується, що алгоритм Копперсміта ефективний у полях GF (2^k), де $k < 520$.

Також розглянемо еліптичні криві і новий стандарт на електронний підпис. Так як до цих пір ми говорили в основному про числа перейдемо до розгляду «об'єктів», їх «властивостей» та операцій. Прикладами можуть служити безліч комплексних чисел, поле многочленів з раціональними коефіцієнтами і т.д. Так від, групи точок еліптичної кривої, будучи спочатку абстрактним алгебраїчним об'єктом, виявилася зручною для побудови алгоритмів стеганографії [16].

«Еліптичною кривою» називають безліч пар точок (X, Y), які задовольняють рівняння:

$$y^2 = ax^3 + bx + c \tag{2.2}$$

Можна накласти обмеження на безліч значень змінних x, y, і коефіцієнтів a, b, c. Обмежуючи область визначення рівняння значущим для додатків числовою безліччю (полем) отримаємо еліптичну криву, задану над аналізованим полем. На рис. 2.2 зображено загальний вигляд еліптичної кривої, визначеної на множині дійсних чисел[18].

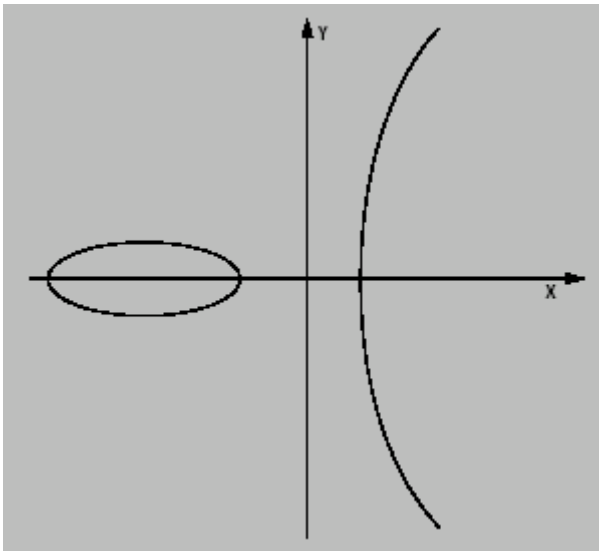


Рис. 2.2 Загальний вид еліптичної кривої

У додатку до криптографії (і в новому стандарті про цифровий підпис) еліптична крива над кінцевим простим полем GF (p) визначається як безліч пар (x, y), які задовольняють рівняння:

$$y^2 = x^3 + ax + b \ (\mathrm{mod}\ p) \qquad (2.3)$$

Пари (x, y) будемо називати «точкою». Точки еліптичної кривої можна «складати». «Сума» двох точок, в свою чергу, теж «лежить» на еліптичній кривій.

Крім точок, що лежать на еліптичній кривій, розглядається також «нульова точка». Вважається, що сума двох точок A з координатами (X_A, Y_A) і B з координатами (X_B, Y_B) дорівнює O, якщо $X_A = X_B$, $Y_A = б\text{-}Y_B$ (mod p). Нульова точка не лежить на еліптичній кривій, але, тим не менш, бере участь в обчисленнях; її можна розглядати як нескінченно віддалену від кривої.

Безліч точок еліптичної кривої разом з нульовою точкою і з введеною операцією складання будемо називати «групою» [23]. Для кожної еліптичної кривої число точок у групі кінцеве, але досить велике. Оцінка порядку (числа елементів) групи точок еліптичної кривої m така:

$$p+1-2\sqrt{p} \le m \le p+1+2\sqrt{p} \qquad (2.4)$$

де p – порядок поля, над яким визначена крива. Якщо в схемі Ель-Гамаля рекомендується використовувати число p порядку 2^{512}, то у випадку еліптичної кривої досить взяти $p > 2^{255}$.

Важливу роль в алгоритмах підпису з використанням еліптичних кривих грають «кратні» точки. Точка Q називається точкою кратності k, якщо для деякої точки P k раз виконано рівність:

$$P = Q + Q + Q + . + Q = kQ \qquad (2.5)$$

Якщо для деякої точки P існує таке число k, що kP = 0, це число називають порядком точки P.

Кратні точки еліптичної кривої є аналогом ступенів чисел в простому полі. Задача обчислення кратності точки еквівалентна задачі обчислення дискретного логарифму. Власне, на складності обчислення «кратності» точки еліптичної кривої і заснована стійкість стеганографії. Хоча еквівалентність задачі дискретного логарифмування і задачі обчислення кратності і доведена, друга має велику складність. Саме тому при побудові алгоритмів підпису в групі точок еліптичної кривої виявилося можливим обійтися більш короткими ключами в порівнянні з простим полем при забезпеченні більшої стійкості.

Секретним ключем, як і раніше, покладемо деяке випадкове число x. Відкритим ключем будемо вважати координати точки на еліптичній кривій P, обумовлену як P = xQ, де Q – спеціальним чином вибрана точка еліптичної кривої («базова точка»). Координати точки Q разом з коефіцієнтами рівняння, що задає криву, є параметрами схеми підпису і повинні бути відомі всім учасникам обміну повідомленнями.

Вибір точки Q залежить від використовуваних алгоритмів і досить непростий. Так, згідно стандарту точка Q повинна мати порядок q, де q – просте число з «хорошими алгебраїчними властивостями». Число q досить велике ($2^{254} < q < 2^{256}$). При побудові конкретного алгоритму, що реалізує обчислення стеганографії, американський стандарт передбачає використання алгоритму DSA. Виявилося, що і українські і американські стандарти добре підходять для реалізації в групі точок еліптичної кривої без особливих модифікацій [24]. Деякі фахівці відзначають навіть, що опис алгоритму стеганографії Ель-Гамаля на еліптичній кривій «простіше і природніше».

2.2 Класифікація атак порушника

Криптографія становить основу сформульованих вище послуг безпеки і є найбільш потужним засобом забезпечення конфіденційності, контролю цілісності та аутентифікації.

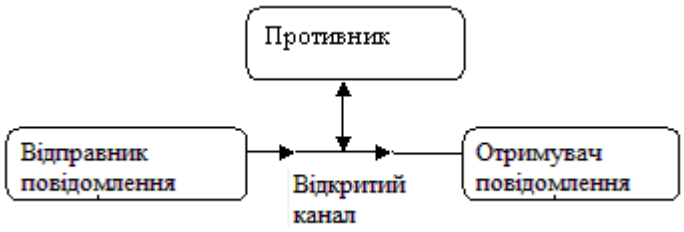


Рис. 2.3 Базова модель [25]

Базова модель (рис 2.3) припускає існування противника, що має доступ до відкритого каналу зв'язку і перехоплює шляхом прослуховування всі повідомлення, передані від відправника до одержувача [26]. Прослуховування з боку противника називається пасивним перехопленням повідомлень. Крім того противник може активно втручатися в процес передачі інформації – модифікувати передані повідомлення і навіть вилучати повідомлення з каналу. Такі дії називаються активним перехопленням повідомлень[27].

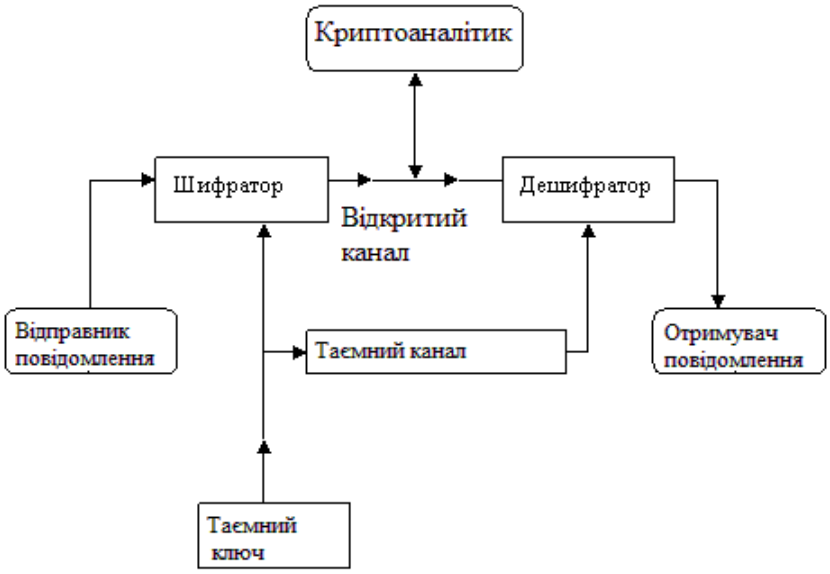


Рис. 2.4 Класична симетрична криптосистема [26]

Способи протидії атакам порушника засновані на застосуванні криптографічних методів. Класична одноключева (або симетрична) криптосистема представлена на рис. 2.4.

Перш ніж передати повідомлення у відкритий канал, відправник з метою приховування істинного змісту піддає вихідну інформацію

спеціальному перетворенню. Для отримання вихідної інформації на приймальному кінці необхідно виконати зворотне перетворення. Процедура прямого перетворення на передавальному кінці називається шифруванням, процедура зворотного перетворення називається дешифруванням. У одноключевій криптосистемі для виконання процедур шифрування і дешифрування необхідно знати загальний для відправника і одержувача секретний компонент – секретний ключ.

Зловмисник, спостерігаючи шифротекст, не може прочитати відкритий текст. Для здійснення своєї мети криптоаналітик застосовує атаки. Відомі такі класичні атаки (у порядку зростання ефективності).

Атака на основі відомого шифротексту. Криптоаналітик у своєму розпорядженні має тільки шифротекст, який завжди можна отримати з каналу і на його основі розкрити секретний ключ.

Атака на основі відомого відкритого тексту. Криптоаналітик у своєму розпорядженні володіє потрібною кількістю пар «відкритий текст / шифротекст». На практиці кількість ключів, як правило, істотно менше числа переданих повідомлень. Це означає, що один ключ використовується для шифрування серії відкритих текстів. Таким чином розкриття ключа дозволяє прочитати всі повідомлення, зашифровані на цьому ключі (наприклад, всі повідомлення одного сеансу зв'язку).

Атака на основі вибіркового відкритого тексту. Криптоаналітик має можливість нав'язувати відправнику потрібний йому відкритий текст і отримувати його в зашифрованому вигляді [28]. Всі відкриті тексти повинні бути обрані заздалегідь – до отримання відповідних шифротекстів. У зарубіжних джерелах таку атаку часто називають «опівнічною», або атакою «короткого перепочинку» - як жартівливе нагадування про те, що зловмисник може скористатися криптографічним пристроєм і зашифрувати підготовлені заздалегідь відкриті тексти в той

момент, коли криптограф залишить своє робоче місце для короткого відпочинку. Особливо ефективною атака може бути, наприклад, у випадку якщо криптоаналітик заволодів пристроєм, що містить секретний ключ. При розробці більшості криптографічних пристроїв використовується спеціальна технологія (TEMPEST), що не дозволяє зчитувати секретну інформацію за допомогою зовнішнього впливу. Однак криптоаналітик може спробувати розкрити ключ, працюючи з пристроєм як з «чорним ящиком», тобто подаючи на нього певний відкритий текст, і аналізуючи отриманий шифротекст [18].

Адаптивна атака на основі вибіркового відкритого тексту. Окремий випадок атаки на основі вибіркового відкритого тексту. Криптоаналітик може не тільки вибирати текст, що шифрується, а й здійснювати свій подальший вибір на основі отриманих результатів шифрування.

Атака на основі вибіркового шифротексту. Це окремий випадок атаки на основі вибіркового шифротексту. Вибираючи черговий шифротекст, криптоаналітик вже знає всі відкриті тексти, відповідні всім раніше вибраними шифротекстам.

Атака на основі вибіркового тексту. Криптоаналітик має можливість атакувати криптосистему як з боку відправника, так і з боку одержувача – вибирати відкриті тексти і шифротекст, шифрувати і дешифрувати їх. Дана атака може бути адаптована з будь-якого боку [29].

Практична криптосистема повинна витримувати всі різновиди описаних вище атак.

Для вирішення в першу чергу завдання розподілу ключів була висунута концепція двохключової (або асиметричної) криптографії (рис. 2.5).

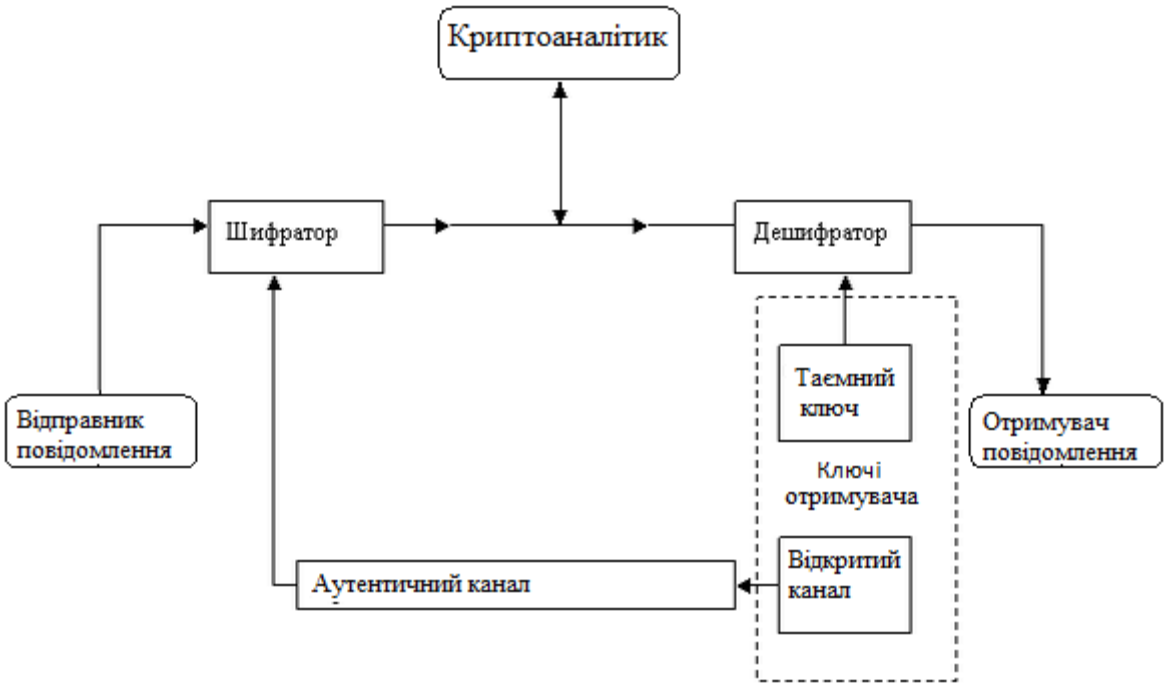


Рис. 2.5 Двохключова криптографія [30]

У такій схемі для шифрування і дешифрування використовуються різні ключі. Для шифрування інформації, призначеної конкретному одержувачу, використовують унікальний відкритий ключ одержувача-адресата. Відповідно для дешифрування одержувач використовує парний секретний ключ. Для передачі відкритого ключа від одержувача до відправника секретний канал не потрібен. Замість секретного каналу використовують автентичний канал, що гарантує справжність джерела переданої інформації (відкритого ключа відправника). Підкреслимо, що автентичний канал є відкритим і доступним криптоаналітику противника [31]. Однак механізм аутентифікації дозволяє виявляти спроби порушення цілісності та автентичності переданої інформації (у цьому сенсі аутентифікація має ряд аналогій з методами завадостійкого кодування, зокрема з кодами, які виявляють помилки). Відсутність аутентифікації дозволило б противнику замінити відкритий ключ одержувача на свій власний відкритий ключ. У цій ситуації противник отримує доступ до всієї адресованої одержувачу інформації.

Інша унікальна властивість двохключової криптосистеми полягає у можливості доказу приналежності у випадку відмови відправника / одержувача від раніше переданого / прийнятого повідомлення і досягається застосуванням стеганографії (рис. 2.6.).

Рис. 2.6 Властивість двохключової криптосистеми [32]

Стеганографія забезпечує також аутентифікацію і контроль цілісності переданої інформації.

Пред'явивши автентичний відкритий ключ відправника, завжди можна довести, що прийняте повідомлення було зашифровано на парному секретному ключі, тобто належить відправнику. Одержувач знає тільки відкритий ключ, яким користується для перевірки підпису, і тому не може підписати повідомлення від особи відправника.

Асиметричні криптосистеми можуть бути атаковані тими ж способами, що й симетричні. Проте слід мати на увазі, що в асиметричній криптосистемі криптоаналітик знає відкритий ключ з визначення і, отже, атака на відкритому тексті завжди можлива. Існує специфічна атака на основі перевірки шифротексту, коли криптоаналітик, знаючи відкритий

ключ, може заздалегідь зашифрувати достатню кількість відкритих текстів (за умови, що їх не дуже багато) і потім порівнюючи отриманий шифротекст з перехопленим, розкрити переданий відкритий текст.

Гольдвассер, Мікалі і Ривест запропонували класифікацію атак для схем стеганографії. Наведемо ці атаки в порядку зростання ефективності.

Атака на основі відомого відкритого ключа. Криптоаналітик знає тільки відкритий ключ для перевірки стеганографії. Атака завжди можлива.

Атака на основі відомого повідомлення. Криптоаналітик знає відкритий ключ і може отримати деяку кількість підписаних повідомлень. Однак на вибір цих повідомлень він вплинути не може.

Атака на основі вибіркового повідомлення. Криптоаналітик може вибрати необхідну кількість повідомлень і отримати їх підписи. Передбачається, що вибір повідомлень виконується до того, як відкритий ключ буде опублікований.

Спрямована атака на основі вибіркового повідомлення. Атака аналогічна попередній з тією різницею, що повідомлення вибираються, коли відкритий ключ вже відомий.

Адаптивна атака на основі вибіркового повідомлення. Атака аналогічна попередній, але криптоаналітик вибирає повідомлення послідовно, виходячи з обчислених підписів для раніше вибраних повідомлень.

2.3 Приховування даних в нерухомих зображеннях

Велика кількість досліджень розглядає зображення як стегоконтейнери. Дане твердження обгрунтовано причинами:

- Існуванням практично значущим завданням захисту фотографій, картин, відео від протизаконного використання;

- Відносно великим обсягом цифрового представлення зображень, що дозволяє впроваджувати ЦВЗ великого обсягу або підвищувати працездатність впровадження;

- Заздалегідь відомим розміром контейнера, відсутністю обмежень;

- Наявністю текстурних областей в великій кількості зображень, що підходять для розташування інформації;

- Слабкою чутливістю людського ока до незначних змін кольорів зображення, його яскравості, контрастності, вмісту в ньому шуму, спотворень поблизу контурів;

- Методами цифрової обробки зображень [14].

Необхідно зазначити, що митоди цифрової обробки зображень провокує і значні труднощі в забезпеченні працездатності ЦВЗ: чим кращими стають методи стиснення зображень, тим менше можливостей є доступними для розміщення додаткової інформації. Розвиток методів та алгоритмів стиснення зображень призвів до зміни уявлень про техніку впровадження ЦВЗ. Спочатку пропонувалося вбудовувати інформацію в додаткові порожні біти, але тепер інформація поміщається в найбільш впливові ділянки зображення, і пошкодження даної ділянки бітів приведе до деградації самого зображення. Тому не випадково стегоалгоритми враховують властивості системи людського зору (СЛЗ), аналогічно алгоритмам стиснення зображень [33]. У стегоалгоритмах найчастіше використовуються ті ж перетворення, що і в сучасних алгоритмах стиснення. При цьому існують, очевидно, три можливості. Вкладення інформації може проводитися у вихідне зображення, або одночасно зі здійсненням стиснення зображення-контейнера, або у вже стислий алгоритмом JPEG зображення. Тому розглянуті властивості людського зору і їх облік в алгоритмах стиснення зображень [13].

Виконання лінійних ортогональних перетворень зображень – обчислювально трудомісткий процес, незважаючи на наявність швидких

алгоритмів. Тому, в деяких випадках можна обмежитися будуванням інформації в просторовій області зображення.

2.4 Алгоритми впровадження тексту у зображення

2.4.1 Алгоритм фрактального нанесення

Фрактальна архівація заснована на тому, що ми представляємо зображення в більш компактній формі - за допомогою коефіцієнтів системи функцій (Iterated Function System - далі по тексту як IFS). Перш, ніж розглядати сам процес архівації, розберемо, як IFS будує зображення, тобто процес декомпресії.

Строго кажучи, IFS являє собою набір тривимірних афінних перетворень, які в нашому випадку перетворюють зображення. Перетворюються точки в тривимірному просторі (x_координата, y_координата, яскравість) [39].

Лінзи можуть проектувати частину зображення довільної форми в будь-яке інше місце отриманого зображення. Області, що були спроектовані, не перетинаються. Лінза має можливість дзеркально відображати і повертати свій фрагмент зображення, змінювати яскравість і зменшувати контрастність та повинна масштабувати свій фрагмент зображення.

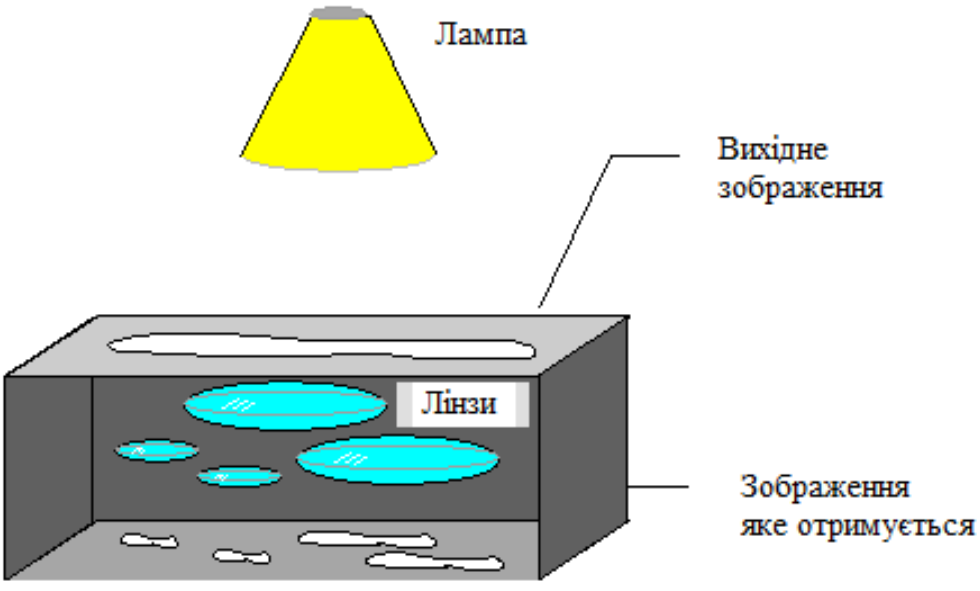


Рис. 2.10 Фотокопіювальна машина[10]

Побудова алгоритму

Основним завданням при використанні фрактального алгоритму є пошук афінних перетворень. У найзагальнішому випадку ми можемо перекладати будь-які за розміром і формою області зображення, однак у цьому випадку виходить астрономічне число варіантів, що перебираються різних фрагментів, яке неможливо обробити на поточний момент навіть на суперкомп'ютері.

У навчальному варіанті алгоритму , викладеному далі, зроблені наступні обмеження на області:

Області – квадрати, сторони яких знаходяться паралельно до зображення.

При перекладі доменної області в рангове зменшення розмірів проводиться рівно в два рази. Це істотно спрощує як компресор, так і декомпрессор, так як задача масштабування невеликих областей є нетривіальною.

При перекладі доменної області в ранговий поворот куба є можливим лише на 0^0, 90^0, 180^0 або 270^0. Також допускається дзеркальне відображення. Загальна кількість потенціальних перетворень (рахуючи пусте) - 8.

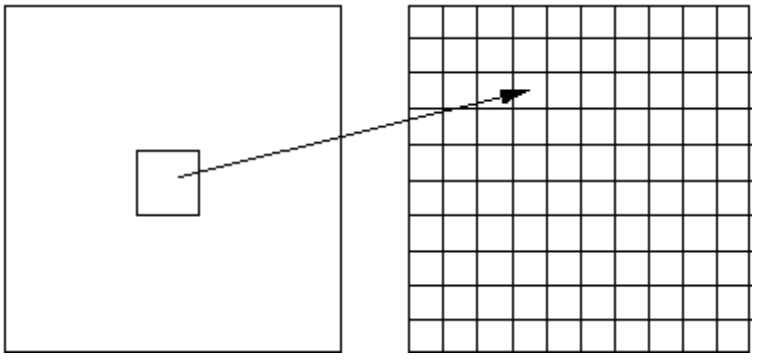


Рис. 2.11 Приклад масштабування [40]

Масштабування по вертикалі здійснюється в 0,75 разів, так як це дає можливість зробити алгоритм, якому потрібно менше число операцій навіть у випадку великих зображеннях. Також це дозволяє компактніше обробити дані для вбудовування. Для кожного перетворення в IFS нам необхідно:

два числа задання зсуву доменного блоку. При обмеженні зображень до 512x512 пікселів буде досить 8 біт на число;

три біта для того, щоб задати перетворення симетрії при перекладі блоку;

7-9 біт для задання зсуву по яскравості при перекладі.

Інформацію про розмір блоків можна зберігати в заголовку файлу. Таким чином, було витрачено менше 4 байт на одне афінне перетворення.

Негативні сторони запропонованих обмежень.

Оскільки всі області є квадратами, неможливо скористатися подобою об'єктів, що не є квадратами.

Аналогічно ми не зможемо скористатися подобою об'єктів в зображенні, що мають велику різницю коефіцієнта подібності від 2.

Також алгоритм не зможе скористатися подобою об'єктів в зображенні, кут між якими не кратний 90 0 .

Така плата за швидкість компресії і за простоту упаковки коефіцієнтів у файл.

Сам алгоритм упаковки зводиться до перебору всіх доменних блоків і підбору для кожного відповідного йому рангового блоку. Нижче наводиться схема цього алгоритму.

for (all range blocks) {

min_distance = MaximumDistance;

R IJ = image-> CopyBlock (i, j);

for (all domain blocks) {/ / З поворотами і отр.

current = Координати перетворення;

D = image-> CopyBlock (current);

current_distance = R IJ . L2distance (D);

if (current_distance <min_distance) {

/ / Якщо коефіцієнти best гірше:

best = current;

}

} / / Next Range

Save_Coefficients_to_file (best);

} / / Next domain

Як видно з наведеного алгоритму, для кожного рангового блоку робимо його перевірку з усіма можливими доменними блоками (у тому числі з минулими перетворення симетрії), знаходимо варіант з найменшою мірою L 2 (найменшим середньоквадратичним відхиленням) і зберігаємо коефіцієнти цього перетворення у файл. Коефіцієнти - це (1) координати знайденого блоку, (2) число від 0 до 7, що характеризує перетворення симетрії (поворот, віддзеркалення блоку), і (3) зсув по яскравості для цієї пари блоків. Зрушення по яскравості обчислюється як [44]:

$$q = \left[\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij} \right] / n^2$$

де R IJ - значення пікселів рангового блоку (R), а D IJ - значення пікселів доменного блоку (D). При цьому міра вважається як:

$$d(R, D) = \sum_{i=1}^{n} \sum_{j=1}^{n} (0{,}75 r_{ij} + q - d_{ij})^2$$

Не обчислюємо квадратного кореня з L 2 заходів і не ділимо її на n, оскільки дані перетворення монотонні і не завадять нам знайти

екстремум, проте ми зможемо виконувати на дві операції менше для кожного блоку.

Порахуємо кількість операцій, необхідних нам для стиснення зображення 512x512 пікселів при заданому розмірі блока 8 пікселів, градацією якого є 256 кольорів:

Таблиця 2.1

Кількість необхідних операцій для стиску зображення

Частина програми	Число операцій
for (all domain blocks)	4096 (= 512/8 x 512/8)
for (all range blocks) + symmetry transformation	492032 (= (512/2-8) * (512/2-8) * 8)
Обчислення q і d (R, D)	> 3 * 64 операцій "+" > 2 * 64 операцій "*"
Підсумок:	> 3 * 128.983.236.608 операцій "+" > 2 * 128.983.236.608 операцій "*"

Таким чином, нам вдалося зменшити число операцій алгоритму компресії до цілком обчислюваних (нехай і за кілька годин) величин.

Схема алгоритму декомпресії зображень

Декомпресія алгоритму фрактального стиснення є простою. Використовуючи коефіцієнти тривимірних афінних перетворень, треба провести невелику кількість ітерацій даних перетворень.

В якості базового зображення може бути використано повністю чорне зображення, оскільки відповідний математичний апарат гарантує нам збіжність послідовності зображень, одержуваних у ході виконання IFS, до нерухомого зображення. Зазвичай 16 ітерацій є достатньою кількістю.

Прочитаємо з файлу коефіцієнти всіх блоків;

Створимо чорне зображення потрібного розміру;

Until (зображення не стане нерухомим) {

For (every range (R)) {

D = image-> CopyBlock (D_coord_for_R);

For (every pixel (i, j) in the block {

R IJ = 0.75 D IJ + O R ;

} / / Next pixel

} / / Next block

} / / Until end

Оскільки ми записували коефіцієнти для блоків R IJ (які, як ми обумовили, в нашому окремому випадку є квадратами однакового розміру) послідовно , то виходить, що ми послідовно заповнюємо зображення по квадратах сітки розбиття використанням афінної перетворення [13].

Як можна підрахувати, кількість операцій на один піксель зображення в градаціях сірого при відновленні надзвичайно мало (N операцій "+", 1 операцій "*", де N - кількість ітерацій, тобто 7-16). Завдяки цьому, декомпресія зображень для фрактального алгоритму проходить швидше декомпресії, наприклад, для алгоритму JPEG, в якому на точку доводиться (при оптимальної реалізації операцій зворотного ДКП і квантування) 64 операції "+" і 64 операції "? "(без урахування кроків RLE і кодування по Хаффману!). При цьому для фрактального алгоритму множення відбувається на раціональне число, одне для кожного блоку. Це означає, що ми можемо, по-перше, використовувати раціональну арифметику, яка істотно швидше арифметики з плаваючою точкою. По-друге, множення вектора на число - більш проста і швидка операція, часто закладається в архітектуру процесора (процесори SGI, Intel MMX ...), ніж скалярний добуток двох векторів, необхідне в разі JPEG. Для повнокольорового зображення ситуація якісно не змінюється,

оскільки переведення в інший колірний простір використовують обидва алгоритму. Блок-схема алгоритму фрактального нанесення зображена на рис 2.11.

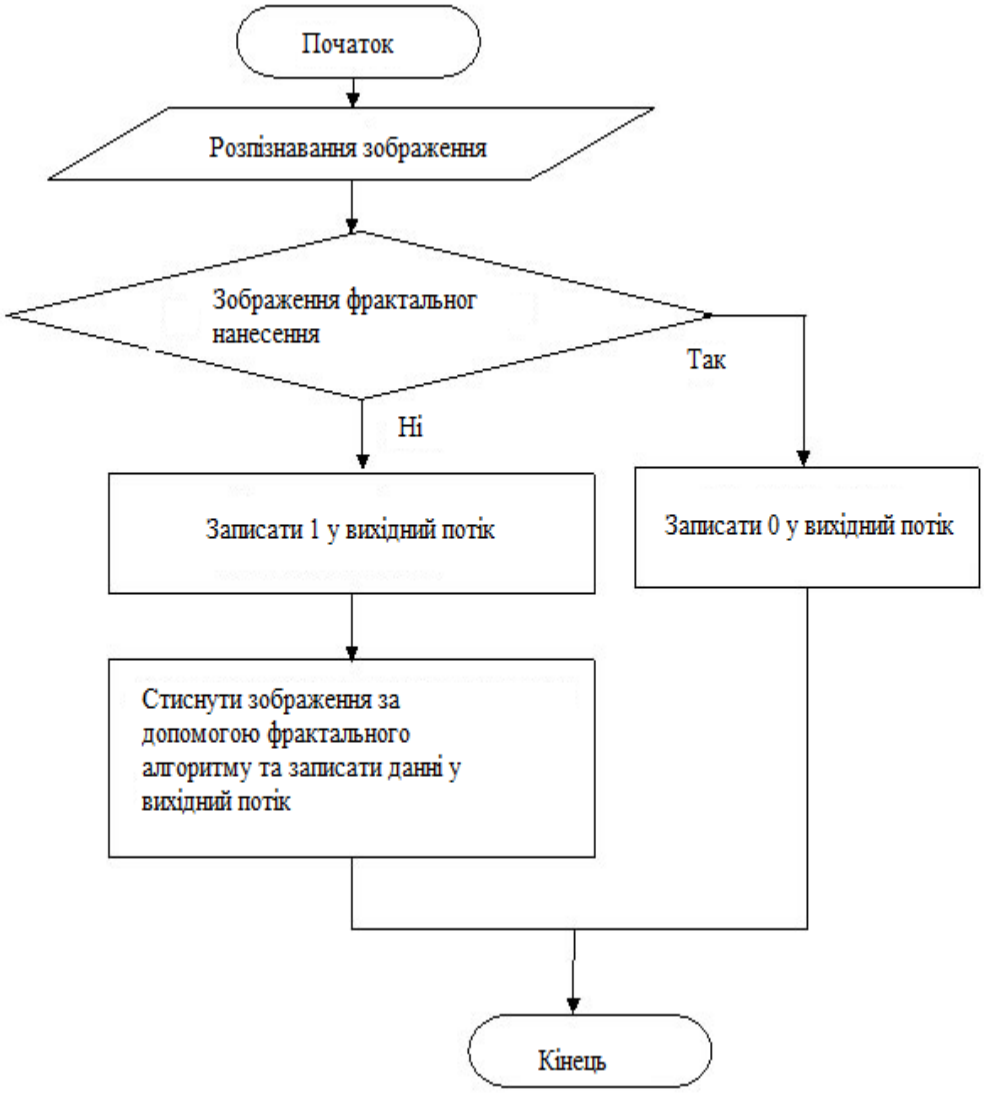


Рис. 2.12 Блок-схема алгоритму фрактального нанесення даних

Характеристики фрактального алгоритму :

Коефіцієнти компресії: 2-2000 (За визначенням користувача).

Симетричність: 100-100000

Характерні особливості: Може вільно масштабувати зображення при розархівації, збільшуючи його в 2-4 рази без появи "сходового

ефекту". При збільшенні ступеня компресії з'являється "блочний" ефект на кордонах блоків в зображенні.

2.4.2 Алгоритм частотного нанесення

Ідея алгоритму полягає в збереженні різници значень між сусідніми блоками зображення, що зазвичай приймає значення близьке до 0.

Так два числа a 2 i і a 2 i +1 завжди можна представити у вигляді b 1 i = (a 2 i + a 2 i +1) / 2 і b 2i = (a 2 i - a 2 i +1) / 2. Аналогічно послідовність A I може бути попарно переведена в послідовність b 1,2 i .

Розберемо конкретний приклад: нехай ми стискаємо рядок з 8 значень яскравості пікселів (AI): (220, 211, 212, 218, 217, 214, 210, 202). Ми отримаємо такі послідовності b 1 i , і b 2 i : (215.5, 215, 215.5, 206) і (4.5, -3, 1.5, 4). Зауважимо, що значення b 2 i досить близькі до 0. Повторимо операцію, розглядаючи b 1 i як A I . Дана дія виконується як би рекурсивно, звідки і назва алгоритму. Ми отримаємо з (215.5, 215, 215.5, 206): (215.25, 210.75) (0.25, 4.75). Отримані коефіцієнти, округливши до цілих і стиснувши, наприклад, за допомогою алгоритму Хаффмана з фіксованими таблицями, ми можемо помістити у файл.

Зауважимо, що ми застосовували наше перетворення до ланцюжку тільки два рази. Реально ми можемо дозволити собі застосування wavelet-перетворення 4-6 разів. Більш того, додаткове стиснення можна отримати, використовуючи таблиці алгоритму Гоффмана з нерівномірним кроком (тобто нам доведеться зберігати код Хаффмана для найближчого в таблиці значення). Ці прийоми дозволяють досягти помітних коефіцієнтів стиску.

Характеристики частотного алгоритму :

Коефіцієнти компресії: 2-200 (За визначенням користувача).

Клас зображень: Як у фрактального і просторового.

Симетричність: ~ 1.5

Характерні особливості: Крім того, при високому ступені стиснення зображення розпадається на окремі блоки.

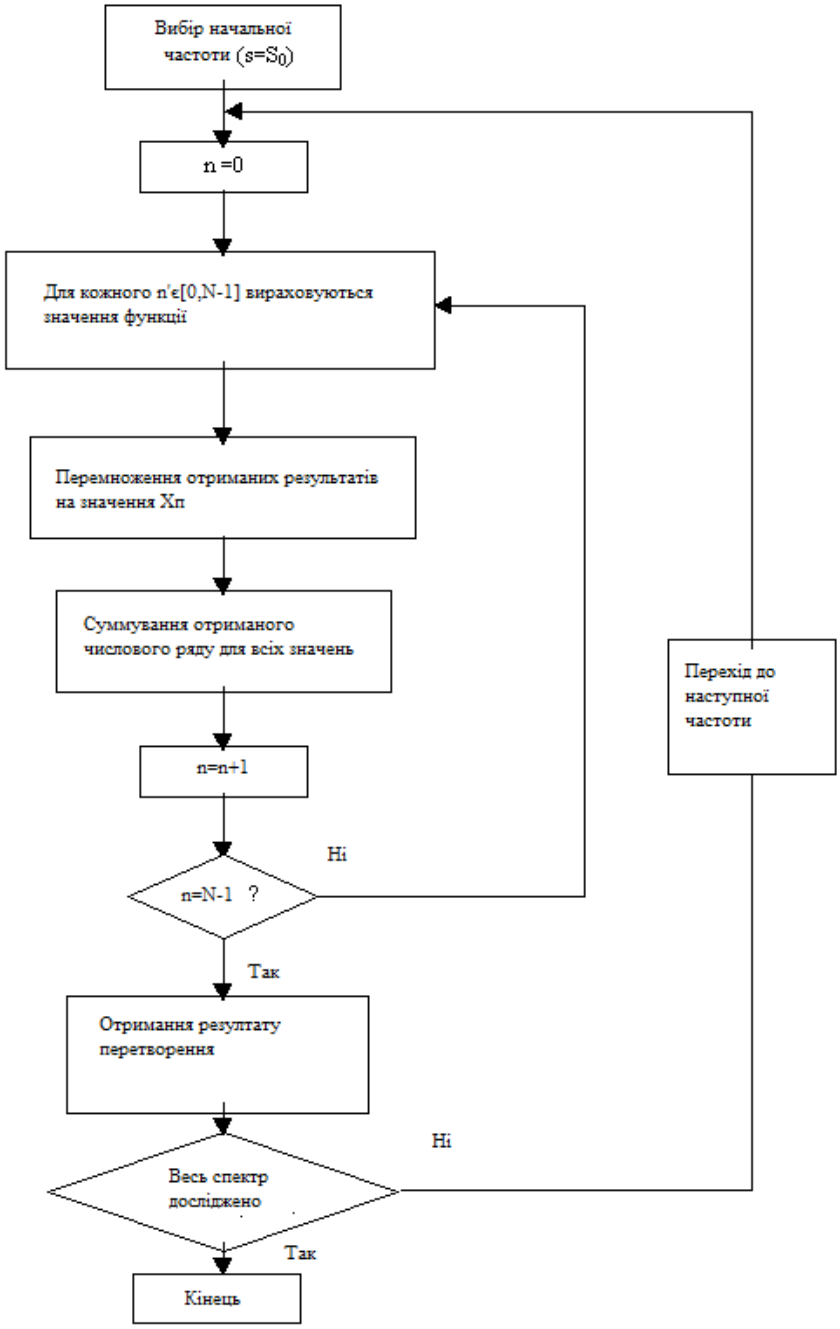


Рис. 2.13 Блок-схема алгоритму частотного нанесення

Висновки

Другий розділ дипломної роботи наводить оцінку стійкості стеганографії. Розкривається поняття стійкості стеганографії. Так під стійкістю стеганографії звичайно розуміють число («гіпотетичних», «елементарних») операцій, які слід виконати для обчислення секретного ключа з відкритого ключа або, якщо досліджується симетричний алгоритм шифрування, для розшифровки повідомлення без знання ключа.

Проводиться класифікація атак порушника та розкриваються моделі приховування даних у нерухомих зображеннях. Описані алгоритми нанесення стеганографії на нерухомі зображення, до кожного з алгоритмів наведено блок-схеми для більш повного донесення викладеного матеріалу у межах розділу.

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ СТЕГАНОГРАФІЧНОГО ПІДПИСУ ЕЛЕКТРОННОГО ЗОБРАЖЕННЯ

3.1 Вибір програмного інструментарію

В якості програмного інструментарія буде використано мову C# через корисні переваги при розробці даного проекту:

Кросплатформеність. Дана мова має поширений список систем, з якими вона повністю сумісна. Особливо це стосується систем сімества Windows, з якими працює більша частина населення та на якій даний додаток розробляється.

Бібліотека класів. .Net пропонує велику кількість вбудованих методів роботи та імпорт додаткових бібліотек для реалізації проекту.

Велика кількість різних технологій. В залежності від цільової аудиторії, .Net пропонує велику кількість рішень для роботи з додатковими програмами, таким чином при роботі з базами даних буде використана технологія ADO.NET, а при необхідності веб-інтерфейсу додатку – ASP.NET.

Автоматичне прибирання сміття. Звільнення невикористовуванних об’єктів пам’яті дозволить не концентрувати увагу на використаних ресурсах, на відмінно від мов-аналогів.

В якості середовища програмування буде використано программу Microsoft Visual Studio через зручний інтерфейс та можливість встановлювати необхідні бібліотеки всередині додатку.

3.2 Розробка програмного засобу стеганографічного підпису зображення

Для зображень формату PNG реалізується метод заміни найменш значущого біта. Використання такого методу не є самостійним через

високий ризик втрати інформації в ході спотворення контейнера, а також в ході атак.

Для більш надійного способу передачі даних необхідно зашифрувати початкову інформацію перед вбудовуванням в зображення, таким чином факт наявності повідомлення буде приховано та при подальшому виявлені повідомлення не зможе бути розшифровано.

Також для забезпечення неможливості виявлення алгоритму впровадження повідомлення буде застосовано алгоритм, що обирає піксель для вбудовування інформації згідно до конкретного правила. Такий спосіб вбудовування інформації забезпечить більш високий рівень захисту повідомлення від спроб розшифрування.

Після завантаження зображення виконується розрахунок корисної ємності контейнера.

```
// завантаження зображення
container.load (fileName);
// висота зображення
int height = container.height ();
// ширина зображення
int width = container.width ();
// кількість біт для впровадження
int quantityBitForChange = height * width * 3;
// корисна ємність контейнера
int quantitySymbolForInjection = quantityBitForChange / 8;
```

Текст, що буде вбудовано, зберігається в бінарному вигляді згідно UTF-8, в якій символи латинського алфавіту, розділові знаки та цифри займають 1 байт, а літери кирилиці – 2 байтами.

```
// рядок введеного повідомлення
QString stegomessage = dataInj-> toPlainText ();
```

// переклад з стринг в масив біт

```
QByteArray data = stegomessage.toUtf8 ();
```

Зображення, яке є контейнером, розбивається на R, G і B складові.

```
int index = 0;
```

// вектор значень колірних компонент

```
QVector < unsigned char > Colors (colorCount);
```

// запис в вектор послідовно кожної компоненти кожного пікселя в порядку проходження компонент rgb

```
for ( int y = 0; y <height; y ++) {
QRgb rgb = container.pixel (x, y); colors [index ++] = Rgb rgb =
container.pixel (x, y); colors [index ++] =
qRed (rgb); colors [index ++] = qGreen (rgb); colors [index ++]
= qBlue (rgb); }}
```

// вектор, в якому зберігаються всі порядкові номери компонент пікселів у рандомному порядку

```
QVector < int > Order (colorCount);
```

// заповнення вектора числами по порядку від 0 до максимальної кількості компонент всіх пікселів

```
for ( int n = 0; n <colorCount; n ++)  {
order [n] = n; }
```

// рандомізация

```
std :: srand (seed);
std :: random_shuffle (order.begin (), order.end ());
```

Інформація для впровадження перекладається з байт в біти згідно функції toBitArray ().

```
QBitArray FormGeneralWindow :: toBitArray
( const QByteArray & byteArr ) {
```

// число біт в масиві байт, який необхідно перетворити

```
const int quantityBit = byteArr .size () * 8;
int index = 0;
```

// якщо довжина масиву байт дорівнює 0, то повертаємо порожній масив біт

```
if (QuantityBit <= 0) {
return QBitArray (); }
```

// масив біт довжиною, обчисленою від масиву байт для перетворення

```
QBitArray bitArr (quantityBit);
```

// вкладені цикли де по кожному елементу масиву байт (по байту) обчислюється 8 біт

```
for ( unsigned char byte: byteArr ) {
for ( int bit = 0; bit <8; bit ++) {
```

// якщо наймолодший біт першого байту дорівнює 1, то виконується умова

```
if (Byte & 0x01) {
// в масив біт записується 1
bitArr.setBit (index, true );
}
```

// побітове зрушення байта для того, щоб на наступному повторенні циклу перевіряти знову останній біт

```
byte = byte >> 1; index ++; }
return bitArr; }
// лічильник змінених біт
int countChangeBit = 0;
// цикл впровадження даних
for ( int n = 0; n <quantityBit; n ++) {
```

// рандомне число зі створеного вектора рандомних чисел, яке буде використовуватися як індекс

```
const int randNum = randVec [n];
// якщо впроваджуваний біт дорівнює 0, то
if (! (BitArr [n])) {
// якщо молодший біт компоненти дорівнює одиниці, то
if (Colors [randNum] & 0x0001) {
// збільшуємо лічильник змінених біт
countChangeBit ++; }
// Обнуляємо молодший біт
colors [randNum] & = 0x00FE; }
// якщо впроваджуваний біт дорівнює 1, то
if (BitArr [n]) {
// якщо молодший біт компоненти дорівнює нулю, то
if (! (Colors [randNum] & 0x0001)) {
// збільшуємо лічильник
countChangeBit ++; }
// молодший біт встановлюємо в одиницю
colors [randNum] | = 0x0001; }
// обчислення відносини пікового сигналу до шуму
int gridPSNR = 20 * log10 (height * width * 255 / countChangeBit);
```

Новим зображенням є копія поточного зображення з подальшим впровадженням змін.

```
// нове зображення - копія завантаженого
newImage = QImage (Width, height, QImage :: Format  RGB888 );
// цикли заповнення зображення модифікованими компонентами
```

```
for ( int y = 0; y <height; y ++) {
for ( int x = 0; x <width; x ++) {
const int r = colors [index ++];
const int g = colors [index ++];
const int b = colors [index ++];
const QRgb rgb = qRgb (r, g, b);
// заповнення зображення по координатам значеннями rgb
 newImage.setPixel (x, y, rgb); }}

void FormGeneralWindow :: saveInj () {
QString fileName = QFileDialog :: getSaveFileName
( this , Tr ( "Зберегти зображення" ),
"/ Home" , Tr ( "Images (* .png * .jpg)" )); /
qDebug () << fileName; newImage.save
(fileName);
}
```

Витяг даних зі стегоконтейнера відбувається тими ж етапами, які застосовуються при впровадженні даних, тільки у зворотному порядку.

Завантажується зображення для вилучення з нього даних при натисканні на кнопку Load. При натисканні на кнопку Ok викликається функція витягання даних extr ().

Обчислюються розміри зображення, генерується псевдовипадкова послідовність від значення функції рандомізації.

// значення, що рандомізують функції, щодо якого буде складатися рандомна послідовність індексів порядкових номерів пікселів зображення

```
 const int seed = 123;
 int index = 0;
// ширина завантаженого зображення
const int width = container.width ();
```

```
// висота завантаженого зображення
const int height = container.height ();
// кількість пікселів в зображенні
const int quantityPixel = width * height;
// кількість компонент всіх пікселів
const int quantityColors = quantityPixel * 3;
const int quantityPixel = width * height;
```

// вектор для запису рандомних чисел, які будуть використовуватися в якості порядкового номера пікселя при виборі пікселя для впровадження

```
QVector < int > NumRand (quantityColors);
// заповнюємо вектор числами по порядку
for ( int n = 0; n <quantityColors; n ++) {
 numRand [n] = n; }
// рандомізация
std :: srand (seed);
std :: random_shuffle (numRand.begin (), numRand.end ());
```

Записуються компоненти всіх пікселів зображення в порядку слідування R, G, B в один вектор.

// вектор для запису в нього значень компонент rgb по порядку проходження

```
QVector < unsigned char >
```

// в цих вкладених циклах відбувається запис значень rgb по прядку проходження

```
for ( int row = 0; row <height; row ++) {
for ( int column = 0; column <width; column ++) {
QRgb rgb = container.pixel (column, row); rgbVector [index ++] =
qRed (rgb); rgbVector [index ++] = qGreen (rgb); rgbVector [index ++] =
qBlue (rgb); }}
```

Обчислюється довжина повідомлення.

```
QBitArray num (32);
for ( int n = 0; n <32; n ++) {
// отримуємо рандомний номер компоненти пікселя
 int numRandom = numRand [n];
// умова виконується якщо молодший біт 1
 if (RgbVector [numRandom] & 0x0001) {
 num.setBit (n, true );
 }
// умова виконується якщо молодший біт 0
else
{
num.setBit (n, false );
}
QByteArray arr = toByteArray (num);
// розмір введених даних
 int arrg = * ( int *) Arr.data ();
 qDebug () << arrg;
// масив з розміром даних і самими даними
QByteArray dataWithSize;
int rt = arrg * 8;
QBitArray mess (rt);
```

Після отримання довжини повідомлення можна витягти все повідомлення.

```
int numRandom;
for ( int n = 32, i = 0; n <quantityColors, i <rt; n ++, i ++) {{
// отримуємо рандомний номер компоненти пікселя
numRandom = numRand [n];
// цикл виконується при впровадженні 1
if (! (RgbVector [numRandom] & 0x0001)) {
mess.setBit (i, false ); }
// цикл Виконуємо при впровадженні 0
if (RgbVector [numRandom] & 0x0001) {
mess.setBit (i, true );
}
QByteArray Message = toByteArray (mess);
dataExtr-> appendPlainText ( QString :: fromUtf8 (Message.constData ()))
```

Для перерахованих вище дій була використана функція перекладу з біт в байти toByteArray ().

```
QByteArray FormGeneralWindow ::
toByteArray ( const QBitArray & bitArr )
// кількість біт у вхідному масиві біт
const int quantityBit = bitArr .count ();
// обчислення кількості байт
const int byteCount = (quantityBit + 7) / 8;
```

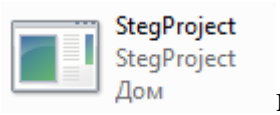
// якщо кількість байт менше або дорівнює 0, то повертаємо порожній масив

```
if (ByteCount <= 0) {
return QByteArray (); }
int index = 0;              ;
```

```
// обнулений масив байт
QByteArray byteArr (byteCount, 0);
// до кількості біт у вхідному масиві біт
for ( int n = 0; n <quantityBit;) {
unsigned char byte = 0;
// побітово
 for ( int bit = 0; bit <8; bit ++) {
 // побітове зрушення вправо
byte = byte >> 1;
if (N <quantityBit) if ( bitArr [N ++]) byte | = 0x80; }
 byteArr [index ++] = byte; }
return byteArr; }
```

Проведемо тестування розробленого програмного додатку стеганографічного підпису електронного зображення.

3.3 Тестування програмного засобу стеганографічного підпису електронного зображення

Для початку роботи із засобом стеганографічного підпису електронного зображення необхідно натиснути на 

на екрані з’явиться головне вікно програми рис.3.1.

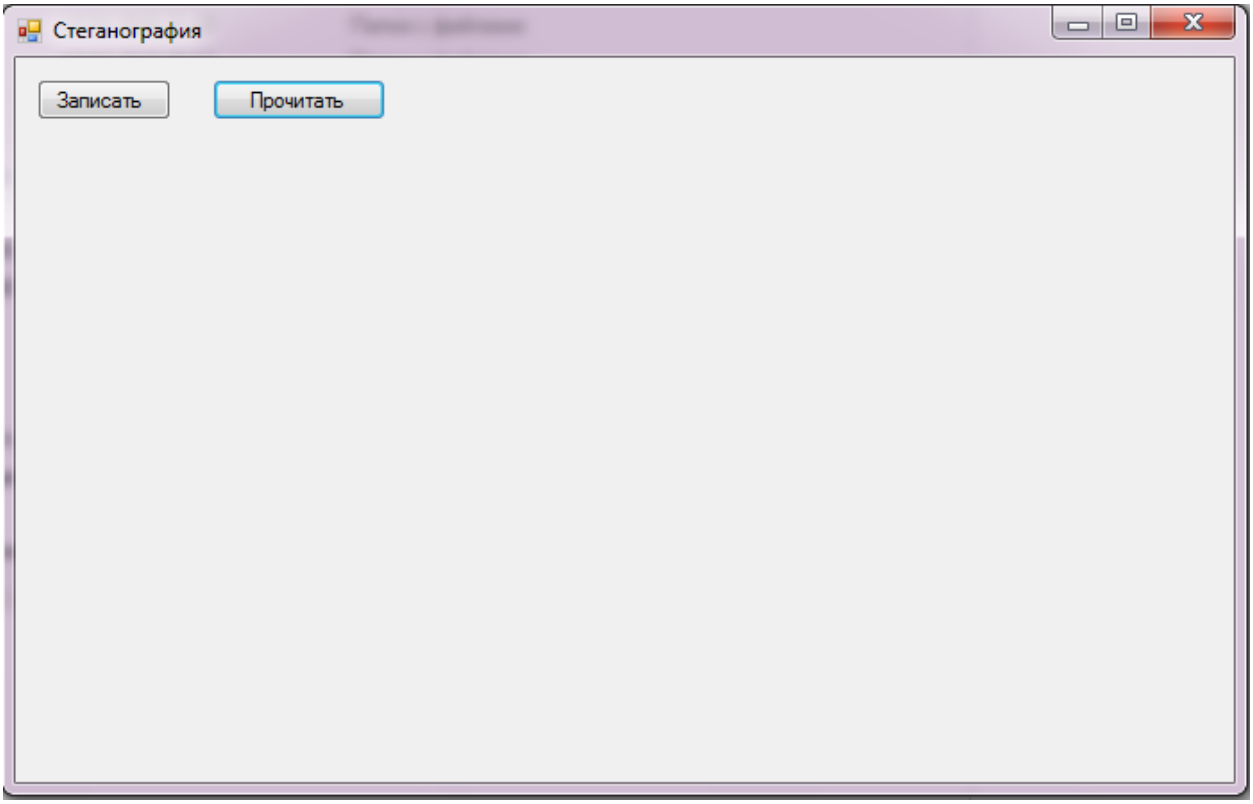
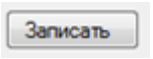


Рис. 3.1 Головне вікно програми

Далі натискаємо на активній кнопці 

на екрані з’явиться екран з вибором зображення на яке варто необхідно нанести повідомлення (текст) (рис. 3.2).

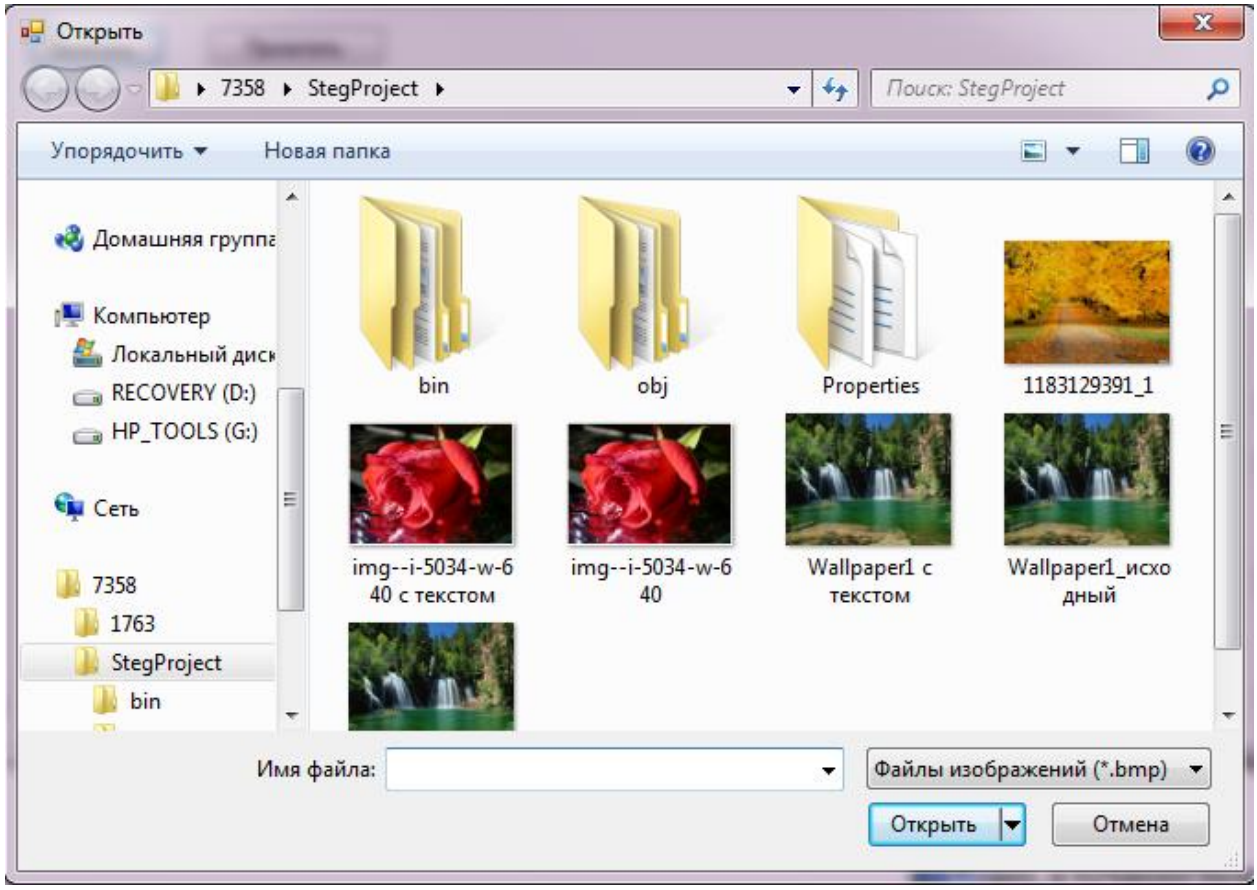


Рис. 3.2 Вибір зображення

Обираємо зображення на яке необхідно нанести текст (рис. 3.3) та натискаємо на кнопці 

.

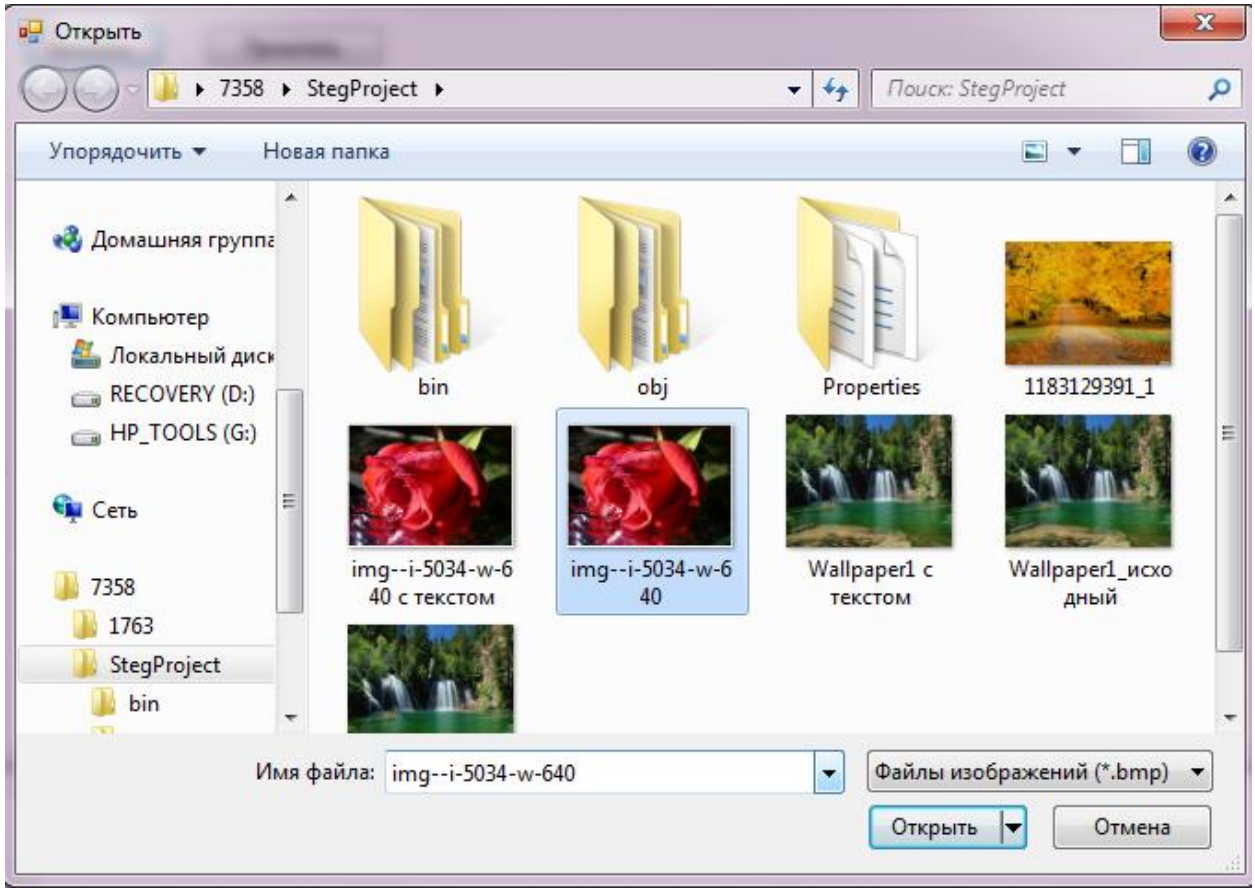


Рис. 3.3 Вибір зображення

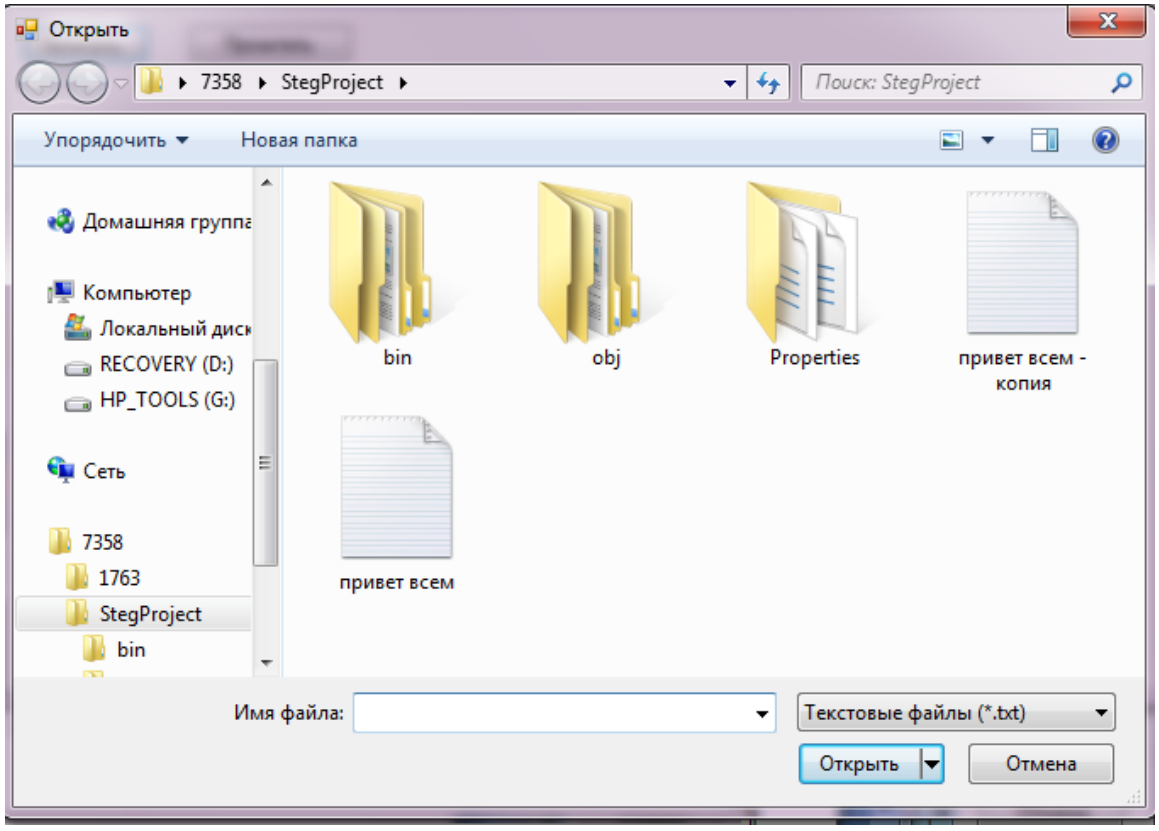


Рис. 3.4 Вибір тексту

Наступним кроком на екрану з’явиться вікно вибору файлу формату .txt, даний файл повинен мати текст, який необхідно нанести на зображення.

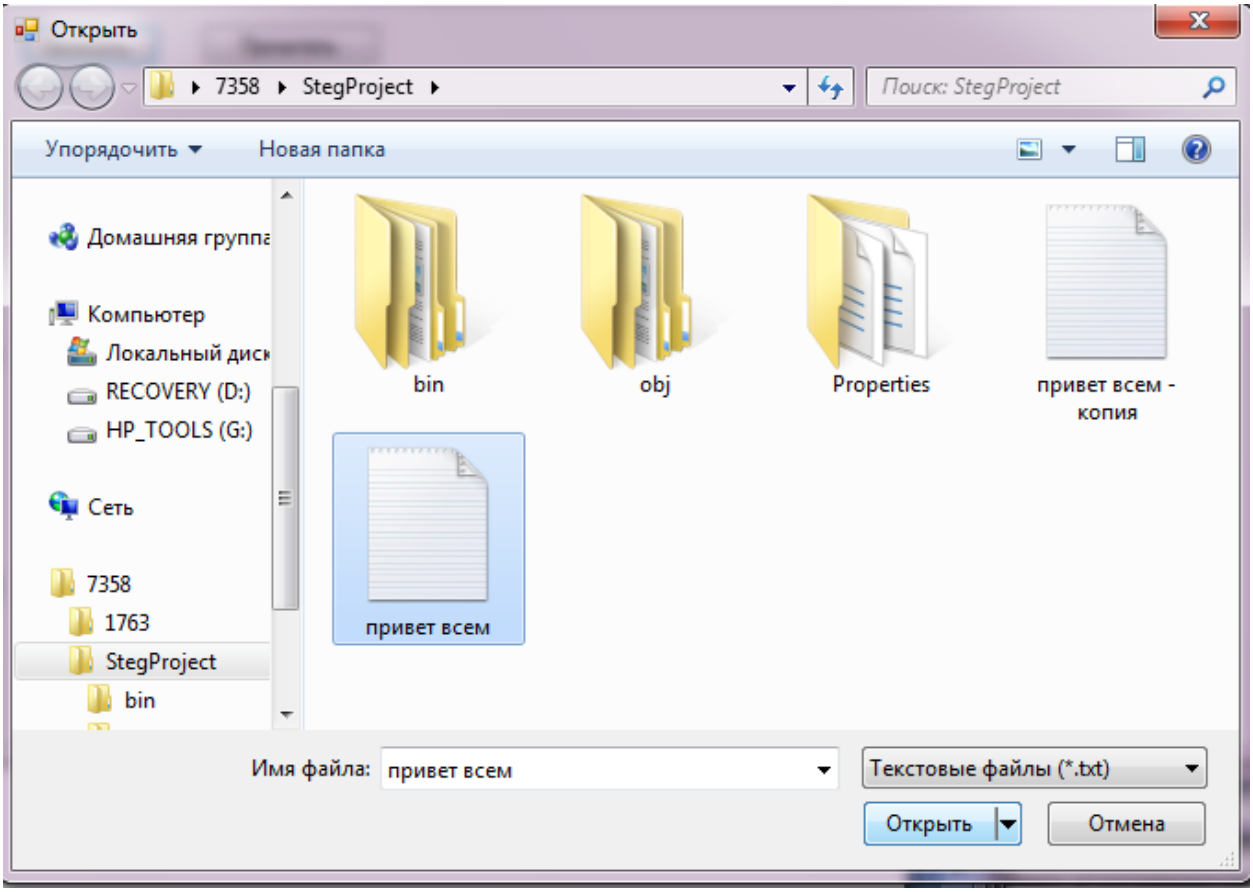


Рис. 3.5 Вибір тексту

Обираємо файл з текстом який необхідно нанести на зображення (рис. 3.5) та натискаємо на кнопці 

. На екрані з’являється зображення з нанесеним текстом (рис. 3.6).

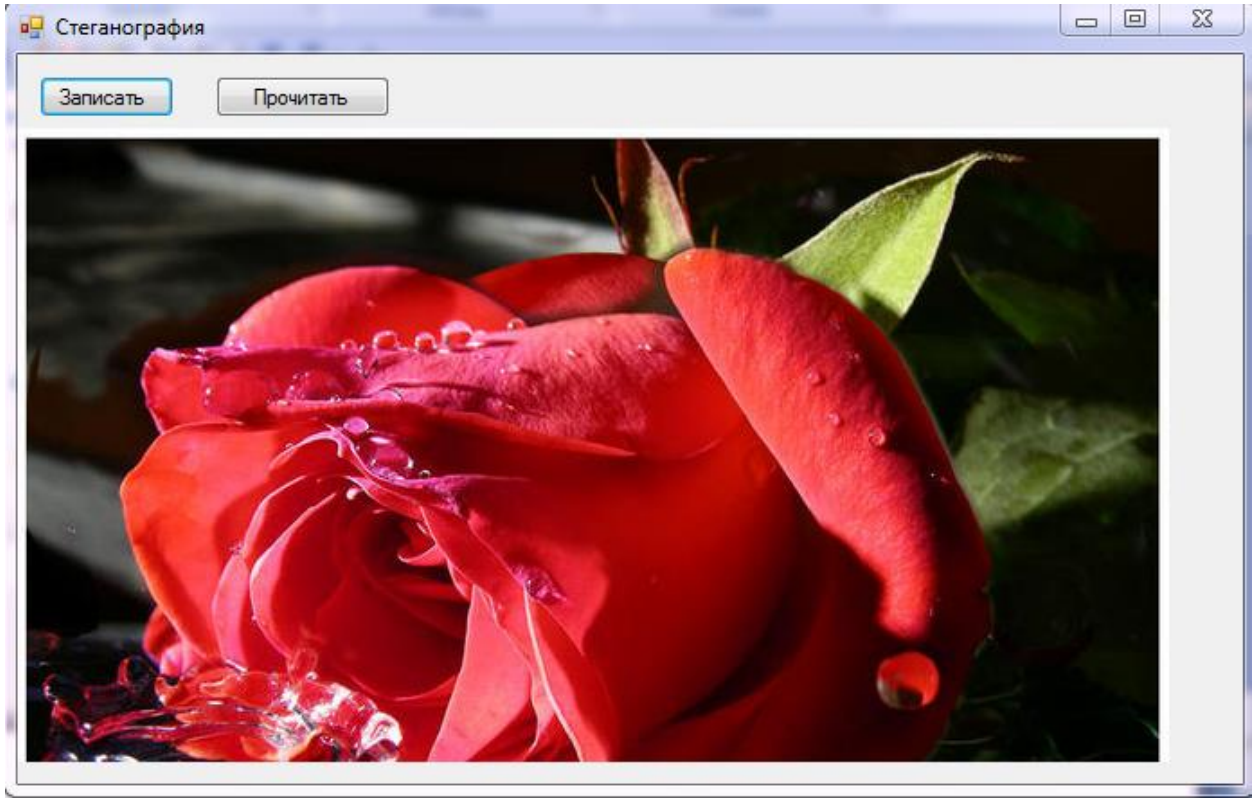


Рис. 3.6 Зображення з текстом

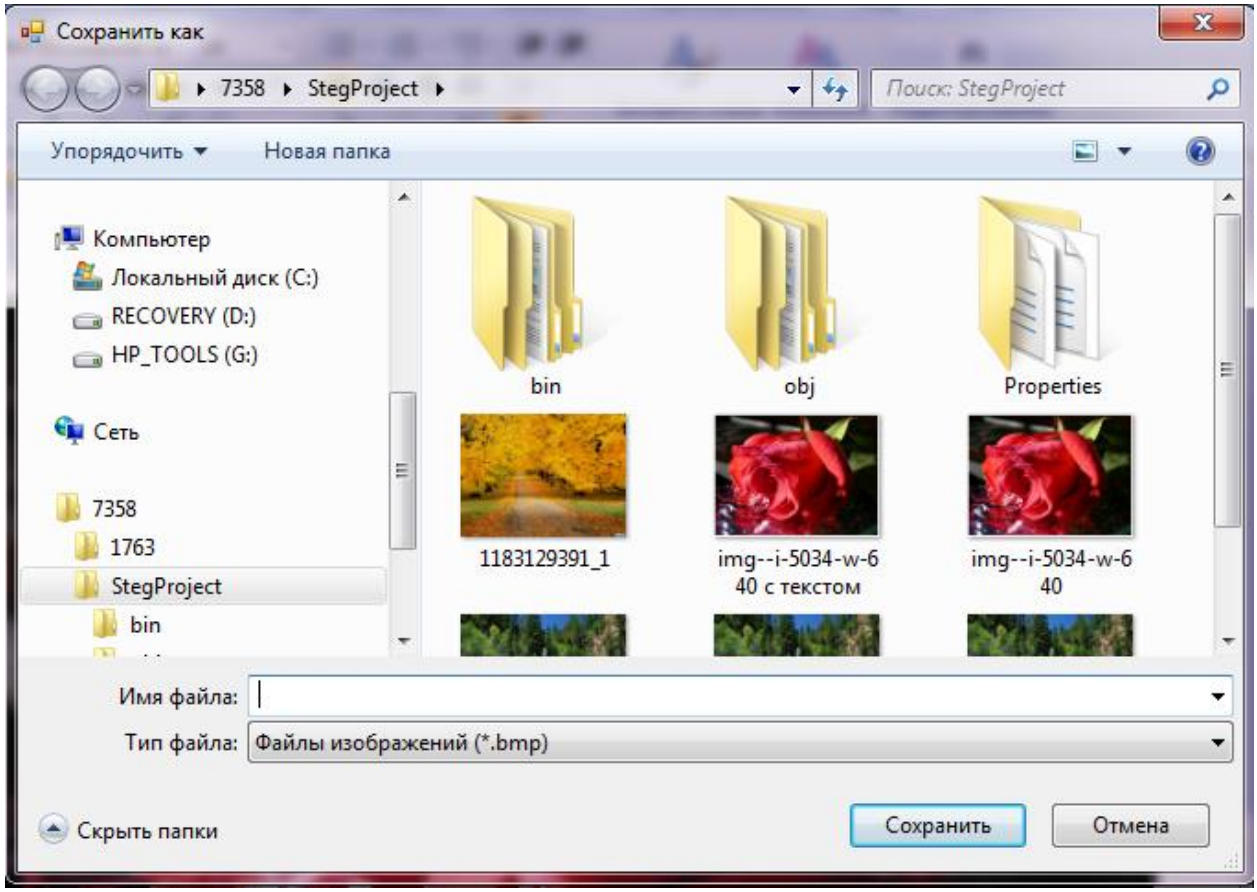


Рис. 3.7 Процес збереження отриманого зображення

Обираємо як зберегти отримане зображення, для цього у полі Ім’я файлу водимо назву для зображення з текстом (рис. 3.8)

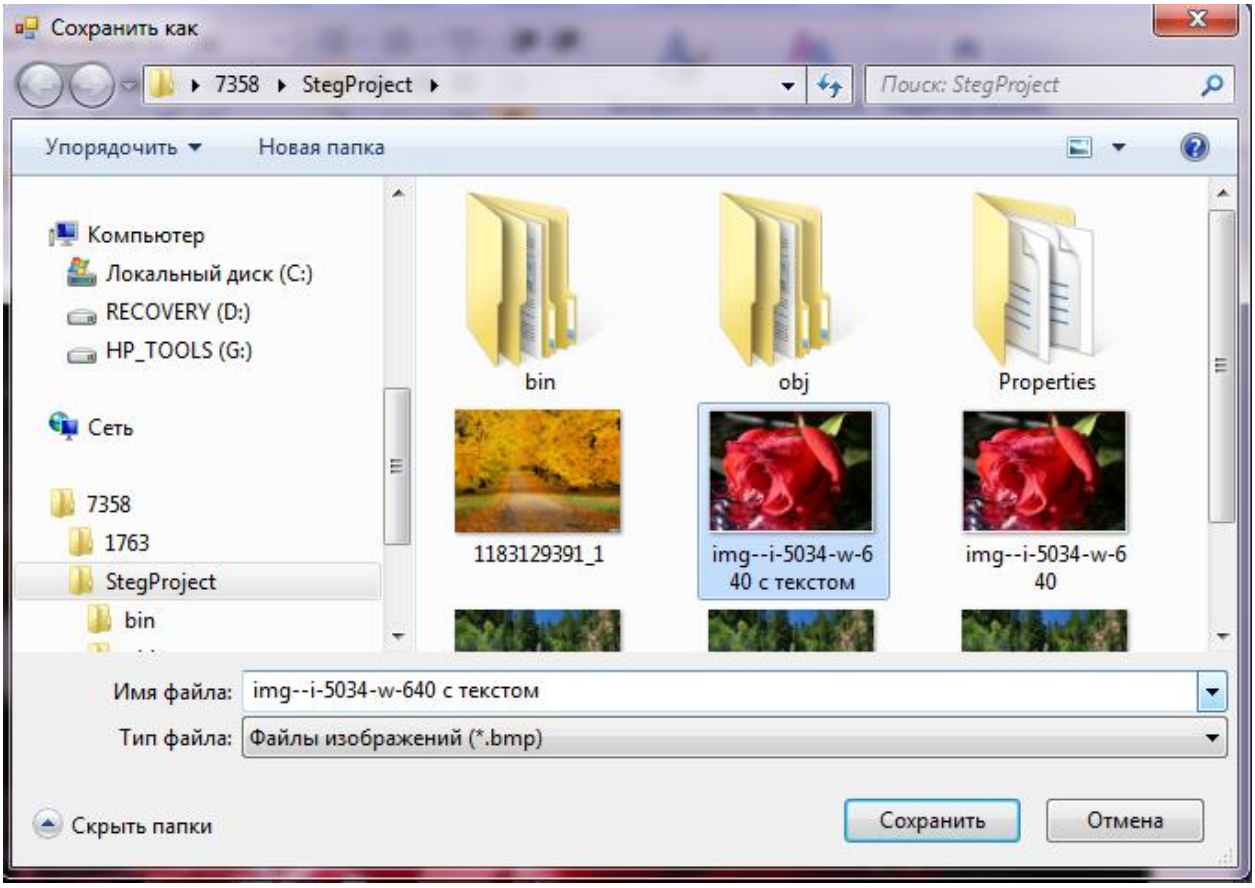


Рис. 3.8 Процес збереження отриманого зображення

Тепер здійснимо розшифрування отриманого зображення, для цього виконаємо всі кроки у зворотному напрямку, результати розшифрування представимо на скріншитах (рис. 3.9 – 3.12)

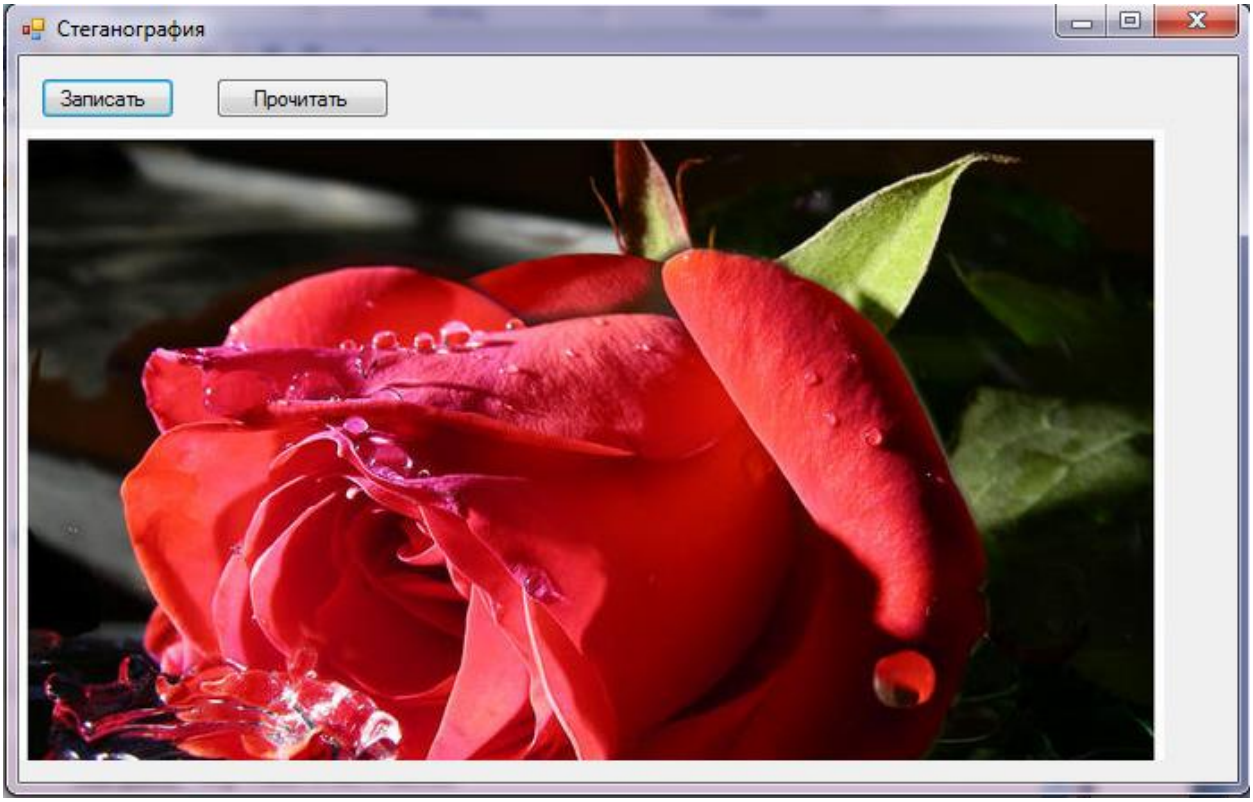


Рис. 3.9 Процес розшифрування отриманого зображення

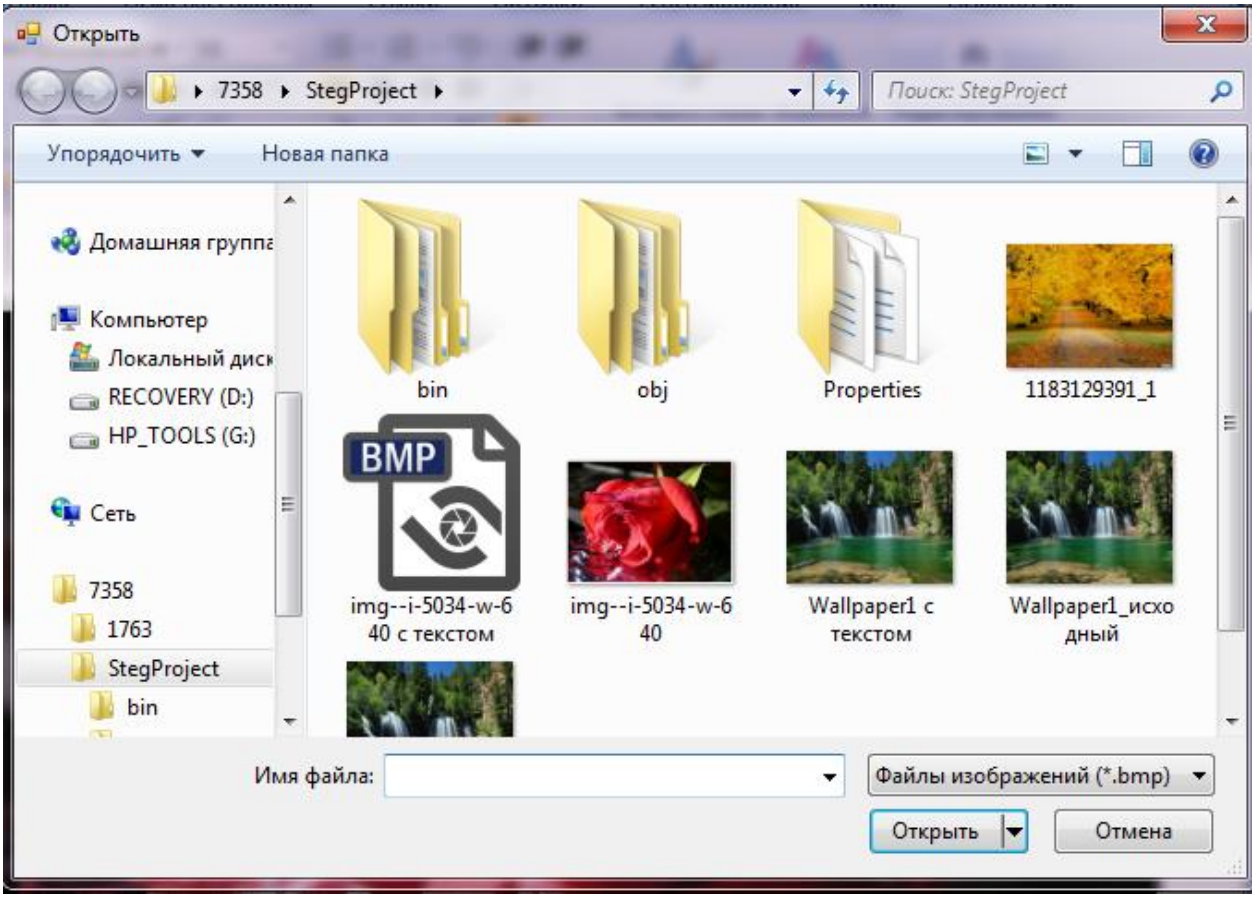


Рис. 3.10 Процес розшифрування отриманого зображення

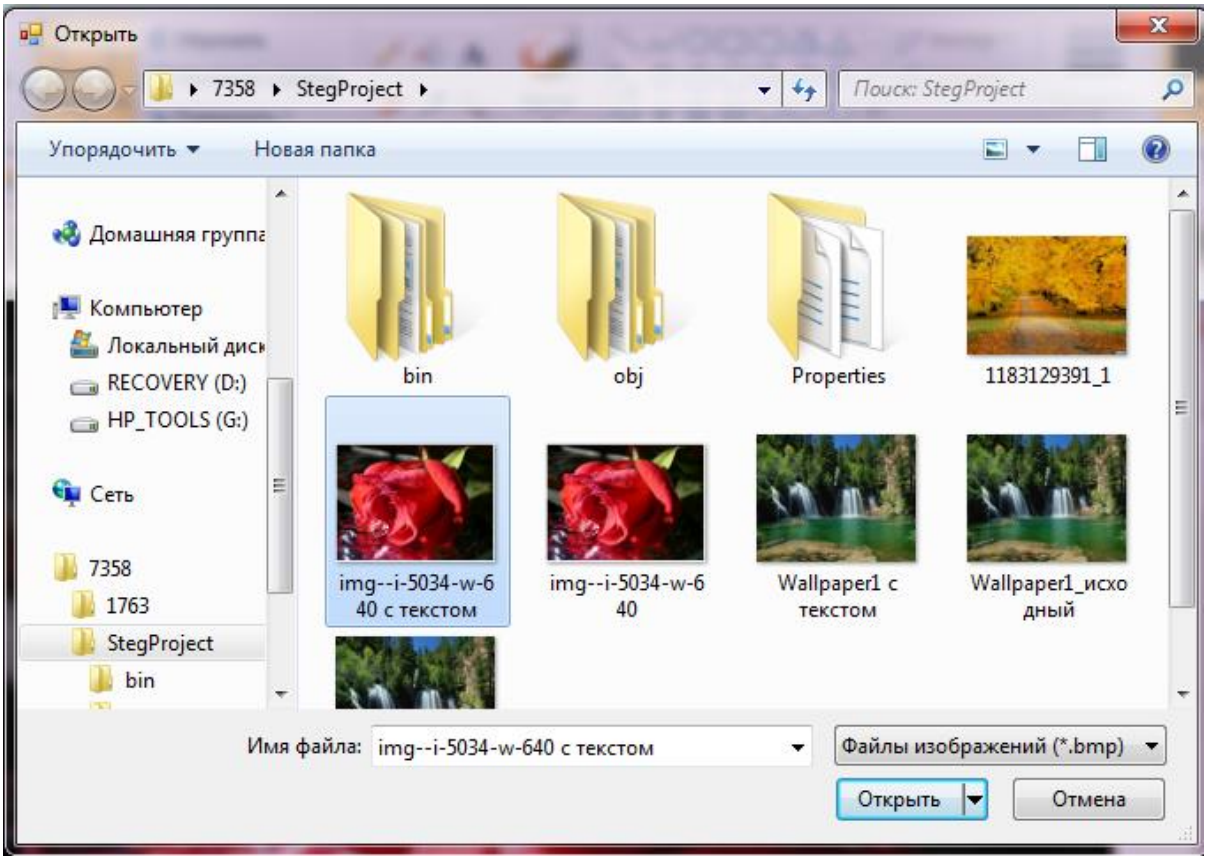


Рис. 3.11 Процес розшифрування отриманого зображення

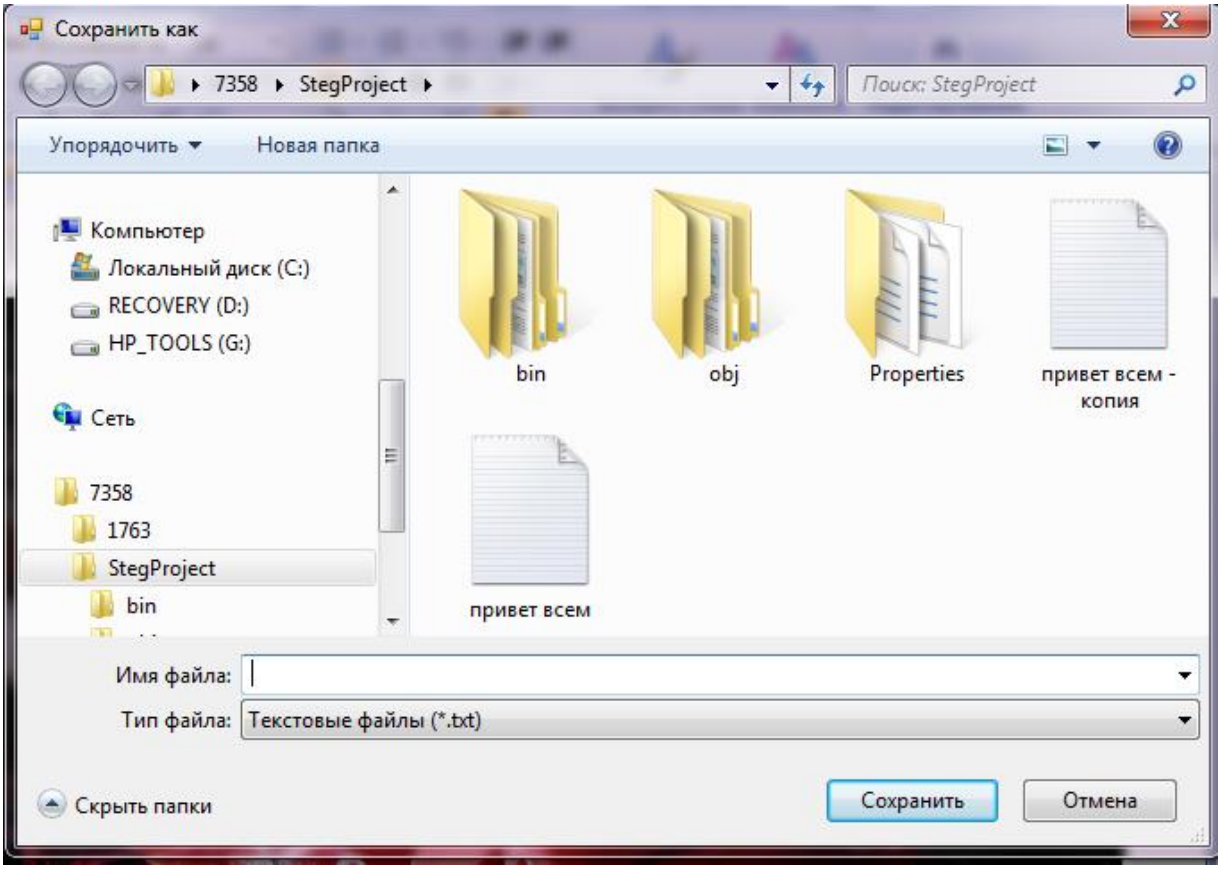


Рис. 3.12 Процес розшифрування отриманого зображення

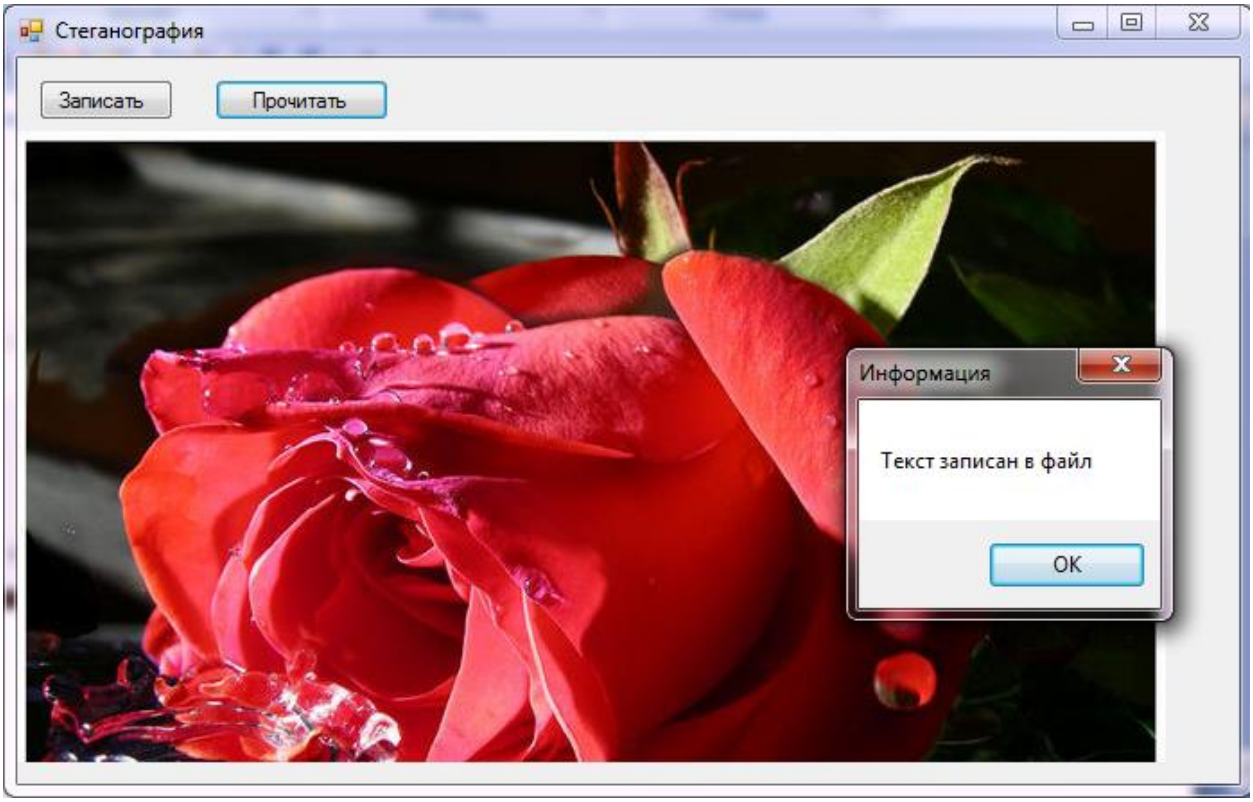


Рис. 3.13 Процес розшифрування отриманого зображення

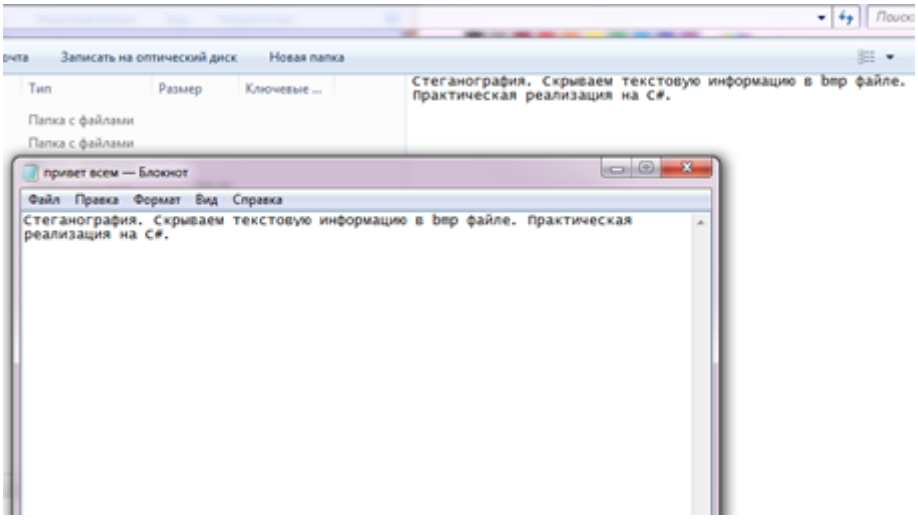


Рис. 3.14 Процес розшифрування отриманого зображення

Під час тестування збоїв та недоліків у роботі засобу стеганографічного підпису електронного зображення не виявлено, що говорить про високу якість розробки та можливість впровадження у роботу за потребою.

Висновки

У межах третього розділу здійснено розробку програмного додатку стеганографічного підпису електронного звіту студента. Описано основні модулі програми та здійснено тестування засобу.

Під час тестування збоїв та недоліків у роботі засобу стеганографічного підпису електронного звіту студента не виявлено, що говорить про високу якість розробки та можливість впровадження у роботу за потребою.

РОЗДІЛ 4
ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

4.1 Техніко–економічне обґрунтування витрат на виконання розробки засобу стеганографічного підпису електронного зображення

Метою економічної частини даної дипломної роботи є розрахунок витрат на створення та впровадження засобу стеганографічного підпису електронного зображення. Рішення поставленої задачі неможливо без оцінки техніко–економічної ефективності розроблюваної системи.

У процесі розрахунку зіставляються дані по витратах на створення системи і дані, що відображають зростання ефективності системи. Система буде економічно ефективною, якщо показники ефективності (термін окупності, розрахунковий коефіцієнт ефективності капітальних вкладень) відповідають галузевим нормам ефективності капіталовкладень.

Оцінка економічної ефективності системи – складна і трудомістка робота, що вимагає не тільки технічних, але і економічних навичок. І лише в поєднанні цих двох складових може призвести до достовірного результату проведеного аналізу.

Економічні оцінки впровадження системи:

Ефективність – виконання необхідних функцій при мінімальних витратах.

Економічний ефект – результат впровадження системи або технології, виражений у вартісній формі, а так само економія від впровадження.

Термін окупності – період часу, протягом якого окупаються витрати.

Джерела економії – економічна оцінка результатів впливу впроваджуваного на технологічні процеси обробки і використання даних:

поліпшення показників діяльності організації; збільшення обсягів і скорочення термінів переробки інформації; зменшення чисельності

персоналу; поява нових можливостей; підвищення продуктивності праці.

Для розрахунку економічної ефективності розробки використовуються наступні основні показники:

– зниження витрат на обробку інформації, річний приріст прибутку (річна економія);

– річна економічна ефективність;

– термін окупності [17, с. 89].

Розроблювана системи дозволяє знизити трудомісткість і витрати часу на обробку даних. Виходячи з цього річна економія (зростання прибутку) розраховується за рахунок економії живої праці. А інші показники ефективності (підвищення точності, оперативності розрахунків) враховуються на якісному рівні. Економічна доцільність розробки системи полягає в економії трудовитрат в порівнянні з ручною обробкою та отриманні більш достовірної інформації за більш короткий час. У нашому проекті задіяні такі фахівці: керівник проекту, фахівець з інформаційного забезпечення (ІЗ) і проектувальник. Матрична організаційна структура демонструє принцип організації роботи над автоматизованою системою (рис. 4.1).

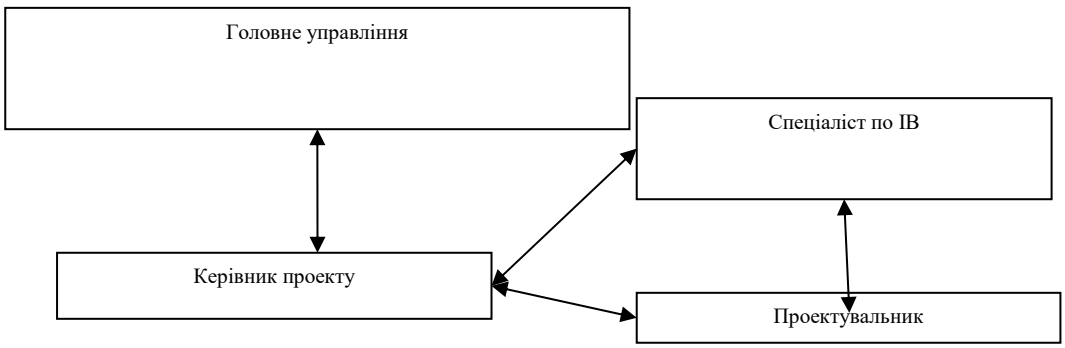


Рис. 4.1 Організаційна структура проектної групи

Розглянемо календарний план роботи над ІС.

Життєвим циклом програми вважається період з моменту прийняття рішення про проведення розробки до повного вилучення з експлуатації.

Етапи життєвого циклу:

Етап роботи над проектом склав близько трьох місяців.

Етап впровадження ІС склав один місяць.

Етап зрілості: перехід до автоматизованої системи – один місяць.

Етап занепаду: висновок недоліків, необхідність удосконалення, поява нових технологій і моральне старіння на даний момент не встановлено.

Складемо календарний графік робіт над проектом (рис.4.2).

Рис. 4.2 Календарний графік роботи над проектом

Детально опишемо дані в таблиці: Витрати часу на розробку програми (табл.4.1).

Таблиця 4.1

Витрати часу на розробку програми

Найменування етапів	Виконавці	Період (дні)
Передпроектне дослідження (аналіз предметної області)	керівник проекту спеціаліст з ІВ	7
Технічне завдання (формулювання вимог до програми)	керівник проекту спеціаліст з ІВ	9
Розробка і налагодження ІС (розробка архітектури та структури про грами, алгоритму, розробка інтерфейсу користувача)	керівник проекту спеціаліст з ІВ проектувальник	30
Тестування (перевірка та аналіз результатів при реальних можливостях зовнішнього середовища)	керівник проекту спеціаліст з ІВ проектувальник	7
Технічна звітність (оформлення програмної документації)	керівник проекту спеціаліст з ІВ проектувальник	10
Впровадження системи	керівник проекту спеціаліст з ІВ	3

За вище вказаним підрахунками термін розробки системи становить 66 днів (трохи більше 2–х місяців).

Теоретична спрямованість – є характерною рисою проведеної роботи. Кінцевим результатом може бути інформаційна система, що демонструє можливість застосування теоретичних розробок і не пропонує вихід на ринок науково–технічної продукції. Отже, основними джерелами витрат при роботі над темою, як частини етапу проектування життєвого циклу цілеспрямованої ІС можна вважати капітальні витрати, які можуть бути враховані і мінімізовані.

Таблиця 4.2

Витрати на витратні матеріали

Найменування матеріалу	Витрати, шт.	Ціна, грн. /шт.	Сума, грн.
флеш – накопичувач	1	60,0	60,0
допоміжна література	4	90	360
канцтовари	–	–	20,0
Разом:	440		

Розрахунок заробітної плати розробників представлений в таблиці 4.3. Враховуючи 22–х робочих днів на місяць.

Таблиця 4.3

Заробітна плата розробників

Найменування етапів	Виконавці	Трудомісткість чол. год.	Денна з/п, грн./год
Передпроектне дослідження (аналіз предметної області)	керівник проекту	1	38,235
	Спеціаліст з ІВ	3	49,2
	проектувальник	8	23,667
Технічне завдання (формулювання вимог до програми)	керівник проекту	2	38,235
	Спеціаліст з ІВ	5	49,2
Розробка і налагодження ІС (розробка архітектури та структури про грами, алгоритму, розробка інтерфейсу користувача)	керівник проекту	5	38,235
	Спеціаліст з ІВ	6	49,2
	проектувальник	10	23,667
Тестування (перевірка та аналіз результатів при реальних можливостях зовнішнього середовища)	керівник проекту	2	38,235
	Спеціаліст з ІВ	3	49,2
	проектуваль	2	23,667
Технічна звітність (оформлення програмної документації)	керівник проекту	6	38,235
	Спеціаліст з ІВ	6	49,2
	проектувальник	10	23,667
Впровадження системи	керівник проекту	1	38,235
	Спеціаліст з ІВ	2	49,2

Скомпонувавши дані, що знаходяться в таблиці, в результаті отримуємо (табл.4.4):

Таблиця 4.4

Основна заробітна плата розробників

Виконавці	Робочі години	Денна з/п, грн./год	Оклад, грн./год
Керівник проекту	17	38,235	650,00
Спеціаліст з ІВ	25	49,2	1 230,00
Проектувальник	30	23,667	710,00
Всього:			2 590,00

Додаткова заробітна плата розробників ІС становить 20% від основної заробітної плати: 2590,00 * 0,2 = 518,00 грн.

Фонд заробітної плати являє собою суму основної та додаткової заробітної плати:

$$\Sigma фзп = 2590,00 + 518,00 = 3108,00 \text{ (грн)}$$

Отже, витрати з оплати праці розробникам склали 3108,0 гривен.

Відрахування на соціальні потреби становлять 22% від фонду оплати праці: 3108,00 * 0,22 = 1087,80 (грн)

Накладні витрати становлять 250% від величини основної заробітної плати: 2590,00 * 2,5 = 6475,00 (грн)

Інші витрати складаються з показників на машинний час (близько 2 – місяців на розробку, налагодження і тестування програми: враховуючи 8 годинний робочий день – 576 годин вартістю 5 грн. / год): 576 * 5 = 2880 (грн)

На підставі даних результатів можна скласти таблицю: "Калькуляції теми" (табл.4.5).

Таблиця 4.5

Калькуляції теми

Найменування статей витрат	Витрати, грн.
Витратні матеріали	440
Основна заробітна плата розробників	2590,00
Додаткова заробітна плата розробників	518,00
Відрахування на соціальне страхування	1087,80
накладні витрати	6475,00
Інші витрати	2880,00
Разом витрат:	13990,80

У результаті ми отримуємо, витрати з оплати праці розробникам в сумі:

13990,80 гривен.

Розрахуємо загальногосподарські витрати.

Фонд часу, що виділяється на ремонт обладнання:

$$Ф_{ремо.\ об.} = Ф_{дн} + Ф_{міс} \qquad (4.1)$$

де $Ф_{дн}$ – денний фонд часу на ремонт обладнання

Фонд робочого часу, що виділяється на щоденні профілактичні роботи:

$$Ф_{дн} = t_{смпроф} Ч (Д_к – Д_в – Д_с), \qquad (4.2)$$

де

$t_{смпроф}$ – це кількість годин, що виділяються на щоденні профілактичні роботи:

$Д_к = 62$ – дні календарні (2 місяці);

$Д_в = 18$ – дні вихідні;

$Д_с = 2$ – дні святкові

$Ф_{дн} = 0,5 Ч (62 – 18 – 2) = 21$ (година)

Фонд робочого часу, що виділяється на місячні профілактичні роботи:

$$Ф_{міс} = t_{міспроф} * 12, \quad (4.3)$$

де

$t_{міспроф}$ = 2:00 (кількість годин виділене на щомісячні профілактичні роботи)

$Ф_{міс} = 2 * 2 = 4$ (годину)

Дійсний фонд часу роботи обладнання:

$$Ф_{дійс} = (Д_к – Д_в – Д_с) * t_{см} * С, \quad (4.4)$$

де

С = 1 (кількість змін)

$t_{см}$ = 8 (годин в день)

$Ф_{дійс} = (62 – 18 – 2) * 8 * 1 = 336$ (годину)

Витрати на електроенергію розраховується в основному на споживання енергії комп'ютером.

Витрати на освітлення одного робочого місця:

$$З_{осв} = Ц_{1КВт} * N_л * N_{осв} * Ф_{дійс}, \quad (4.5)$$

де

$Ц_{1КВт / ч}$ = 0,99 грн (ціна 1 КВт / год в)

$N_л$ = 2 (кількість освітлювальних приладів)

$N_{осв}$ = 0,06 (потужність одного освітлювального приладу)

$З_{осв}$ = 0,99 * 2 Ч* 0,06 * 336 = 124,2 (грн)

Витрати на електроенергію для монітора:

$$З_{елмон} = Ц_{КВт / ч} * N_{мон} * Ф_{дійс}, \quad (4.6)$$

де

$N_{мон}$ = 0,4 (потужність монітора)

$З_{елмон}$ = 0,99 * 0,4 * 336 = 414 (грн)

Витрати на електроенергію системного блоку:

$$З_{елсист. бл} = Ц_{КВт / ч} * N_{сист. бл} * Ф_{дійс}, \quad (4.7)$$

де

$N_{сист. бл}$ = 0,3 (потужність системного блоку)

$З_{елсист. бл}$ = 0,99 * 0,3 * 336 = 310,5 (грн)

Загальна сума витрат на електроенергію

$$ΣЗ_{ел} = З_{осв} + З_{елмон} + З_{елсист. бл} \quad (4.8)$$

$ΣЗ_{ел}$ = 124,2 + 414 + 310,5 = 848,7 (грн)

Отже, витрати на електроенергію складає сума в 848,7 гривен

Не можна не відзначити і витрати на послуги Інтернет – ресурсу. Щомісячна оплата становить 140 грн. А так як розробка тривала 1 місяць, то витрати склали 140 грн.

Розрахуємо амортизацію обладнання:

$$А_{об} = Ц_{об} * Н_{п. в / об}, \quad (4.9)$$

Де $Н_{п. в / об}$ = 24% (норма амортизації по устаткуванню)

$Ц_{об}$ = 10 000 грн. (Вартість використаного обладнання)

$А_{об}$ = 10 000 * 0,24 = 2400 (грн)

Отримана сума в 2 400 гривен – амортизація обладнання на протязі року.

$$А_{об1} = А_{об} / Ч_{рд}, \quad (4.10)$$

де $Ч_{рд} = 251$ (робочі дні в плині року)

$А_{об1} = 2\ 400/251 = 9{,}56 \approx 10$ (грн) – денна сума амортизації.

За весь період розробки проекту витрати на амортизацію обладнання склало 620 гривен.

Проведемо розрахунки по обслуговуванню обладнання, вони становлять 25% амортизаційних відрахувань. Отримуємо 620 * 0,25 = 155 гривен.

Складемо кошторис витрат на розробку програмного продукту.

Таблиця 4.6

Загальні витрати на розробку програми

Статті витрат	Числове значення (грн)
Витрати з оплати праці розробника програми	3263,40
Витрати на витратні матеріали	440,00
Витрати на електроенергію	848,70
Витрати на інтернет–трафік	140,00
Витрати на амортизацію устаткування	620,00
Витрати на обслуговування обладнання	155,00
Разом:	5467,10

Таким чином на розробку проекту знадобиться затратити 5467,10 гривні. Отже, на підставі отриманих даних розрахуємо економічну ефективність розробки програмного продукту (ПП).

Всі найкращі результати, показуваний ПП і є показник ефекту. Економічний ефект від використання ПП за розрахунковий період Т визначається за формулою, грн.:

$$ЕТ = РТ – ЗТ \quad (4.11)$$

де РТ – вартісна оцінка результатів застосування ПП протягом періоду Т, грн.;

ЗТ – вартісна оцінка витрат на створення і супровід ПП, грн. (Використовуємо ЗК).

Вартісна оцінка результатів застосування ПП за розрахунковий період Т визначається за формулою:

$$P_t = \sum_{t=0}^{T} P_t * \alpha_t \quad (4.12)$$

Т – розрахунковий період;

P_t – вартісна оцінка результатів року t розрахункового періоду, грн.;

α_t – функція, що дисконтуються, яка вводиться з метою приведення всіх витрат і результатів до одного моменту часу.

Функція, що дисконтуються має вигляд:

$$\alpha_t = 1 / (1 + p)\, t, \quad (4.13)$$

р – коефіцієнт дисконтування (р = Ен = 0.2, Ен – нормативний коефіцієнт ефективності капітальних вкладень).

$$P_t = \sum_{t=0}^{T} P_t / 1{,}2^t \quad (4.14)$$

Таким чином, розроблена нами ІС замінює ручну працю, отже, набір корисних результатів у ній не змінюється. В якості оцінки результатів застосування ІС в рік береться різниця (економія) витрат, що виникає в результаті використання ІС, тобто Pt = Ey.

Економія від заміни ручної обробки інформації на автоматизовану утворюється в результаті зниження витрат на обробку інформації і визначається за формулою, грн.:

$$E_y = B_p - B_a \quad (4.15)$$

B_p – витрати на ручну обробку інформації, грн.;

B_a – витрати на автоматизовану обробку інформації, гнр.

Витрати на ручну обробку інформації визначаються за формулою:

$$B_p = O_i * B * Г_д / H_в \quad (4.16)$$

O_i – обсяг інформації, оброблюваної вручну, Мбайт;

В – вартість однієї години роботи, грн. / годину;

$Г_д$ – коефіцієнт, що враховує додаткові витрати часу на логічні операції при ручній обробці інформації;

$H_в$ – норма виробітку, Мбайт / год.

В даному випадку: O_i = 50 Мбайт (загальний розмір оброблюваних даних, які вводяться для реєстрації за рік з подальшим підрахунком статистики),

Ц = 450/22/8 * 2,55 грн. / годину,

$Г_д$ = 2,5 (встановлений експериментально),

$H_в$ = 0,004 Мбайт / год.

Виходить, витрати на ручну обробку інформації будуть рівні:

B_p = 50 * 2,55 * 2,5 / 0,004 = 79 687,5 грн.

Витрати на автоматизовану обробку інформації розраховуються за наступною формулою:

$$В_а = t_а * В_м + t_о * (В_м + В_о) \quad (4.17)$$

$t_а$ – час автоматичної обробки, год.;

$В_м$ – вартість однієї години машинного часу, грн. / годину;

$t_о$ – час роботи оператора, год.;

$В_о$ – вартість однієї години роботи оператора, грн. / годину.

Для нашої автоматизованої системи:

$В_о = 450/22/8 * 2,55$ грн.

(Для введення даних оператором в систему знадобиться: (1000 випадків) * (5хв. реєстрації 1 випадку) = 5000 хв. = 83.3 години; Для автоматичної обробки введених даних, знадобиться 1080 хв. = 18 годин на рік).

Отже, витрати на автоматизовану обробку інформації будуть рівні:

$В_а = 18 * 2 + 83,3 * (2 + 2,55) = 415,02$ грн.

Таким чином, річна економія від впровадження АС дорівнює:

$Е_у = 79\ 687,5 – 415,02 = 79\ 272,48$ грн.

Економічний ефект від використання ІС за рік визначається за формулою, грн.:

$Е_Г = Е_у – Е_н * З_к.$

$E_г = 79\ 272{,}48 – 0.2 * 37\ 610{,}2 \approx 71\ 750{,}44$ грн.

Ефективність розробки може бути оцінена за формулою:

$E_р = E_г * 0.4/Вк.$

$E_р = 71\ 750{,}44 * 0.4/37\ 610{,}2 \approx 0.76$

Оскільки $E_р > 0.20$, розробка інформаційної системи є економічно доцільною.

Здійснимо вартісну оцінку результатів застосування АС (економія) за розрахунковий період Т = 3 роки (за умови відсутності внесених у програму доопрацювань і змін):

$$P_З = \sum_{t=0}^{3} 79272{,}48/1{,}2^t = 79272{,}48 + 79272{,}48/1{,}2 + 79272{,}48/1{,}2^2 =$$
$$79272{,}48 + 66060{,}4 + 55050{,}3 = 200383{,}18$$

Економічний ефект від використання АС за розрахунковий період Т = 5 років складе:

ЕТ = 200 383.18 – 37 610,2 = 162 772,98 грн.

Таким чином, поставлена задача доказу економічної ефективності розробленої АС повністю підтвердилася.

Створювана система дає економічний ефект за рахунок, перш за все, скорочення часу і сил співробітника. Розширення сфери застосування АС і її подальше вдосконалення дозволить ще більше підвищити економічний ефект від застосування інформаційної системи.

Витрати на розробку, отримані методом калькуляції, складають 5467,10 грн.

Економічний ефект від використання даного ПП за розрахунковий період (3 роки) складе 162 772,98 грн.

Отже, можна зробити висновок, що отримані показники дають достатньо підстав для запровадження системи.

4.2 Аналіз впровадження розробленого додатку

Впровадження сучасних інформаційних технологій – справа дорога. Функціонування компаній в торговому середовищі потребує, щонайменше аналізу економічних наслідків, а в ідеалі – оцінки економічної ефективності кожного кроку перетворення системи.

У минулому параграфі ми розрахували економічну ефективність і виявили абсолютну окупність витрат на розробку і впровадження. Тепер ми опишемо сам процес впровадження інформаційних систем і проаналізуємо його з точки зору інформаційної системи.

Впровадження системи залежить від купи обставин і є складним завданням. При впровадженні застосовується багатофакторна операція, де на кожному етапі відбувається тісна співпраця працівників замовника і виконавця, навчання персоналу та контроль проведення заявлених вимог.

Необхідно визначити три кроки впровадження інформаційної системи:

Дослідження. Необхідно провести дослідження предметної галузі і бізнес факторів.

Доопрацювання системи. Програмісти забезпечують потрібну працездатність системи.

Запуск системи. Етап використання системи, що містить процеси навчання працівників.

На етап дослідження бізнес факторів надається більше часу. Оскільки, потрібно як можна точно описати, які процеси потрібно покращити.

В більшості випадків працездатність інформаційної системи дещо ширше, ніж в реальності. Тому, потрібно виявити як наявність тих чи інших опцій вплине на остаточну вартість системи, час введення, і головне, чи відповідає запропонована функціональність заданим цілям.

Корисною практикою є своєчасне подання звітів у вигляді документа, де описана вивчена предметна область з результатами проведених опитувань та досліджень.

Після етапу дослідження потрібно точно вирішити остаточну вартість і терміни впровадження інформаційної системи.

На етапі доопрацювання системи необхідний контроль розробки зазначених функцій та методів додатку. Розумно, щоб з боку замовника виступала людина, добре знайома із завданнями компанії та її бізнес процесами.

На фазі запуску системи необхідно сконцентрувати уваги на зміні бізнес моделі в сторону підвищення обізнаності функціоналу з боку споживачів та персоналу. Доступне пояснення та короткий опис функціоналу може допомогти з прискоренням вивчення функцій додатку для його повноцінного використання.

Впровадження можна вважати успішним у випадку коректного використання системи користувачами, відсутність негативних показників в роботі, несправностей тощо. Необхідно згодом регулярно аналізувати потреби користувачів та персоналу відносно даної системи для подальшого розвитку продукту, орієнтуючись на сучасні проблеми персоналу та клієнтів. Даний підхід дозволить прискорити процес використання продукту, вихідну якість та підвищення показників роботи,

що може вплинути на подальший розвиток продукту та його вихід на вищий рівень ринку.

Процес впровадження інформаційної системи має займати найменшу можливу кількість часу. Таким чином користувачі та персонал швидко зможуть проаналізувати вразливості та потенціальні покращення системи. Логічно пам'ятати про можливі ризики і фінансові витрати.

Для успішної роботи з інформаційною системою необхідно мати у розпорядженні наступні пристрої [23]. Системні вимоги:

операційна система: Windows 7/10;

процесор: Pentium 6, 5000 МГц;

оперативна пам'ять: не менше 3 Гб;

вільне місце на жорсткому диску: не менше 1 Гб;

маніпулятор типу "миша";

РОЗДІЛ 5
РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

Технологія стеганографічного підпису електронного зображення може мати місце в будь-якій галузі, що пов’язана з захистом інформації, впровадженням додаткового рівня захисту облікових записів користувачів тощо. Дана технологія не потребує великих обчислювальних ресурсів та може бути застосована в додатках різного масштабу.

Однією з реалізацій даної технології є реалізація двохфакторної аутентифікації користувача за допомогою зображення, що містить в собі повідомленння, яке буде відомо на стороні клієнта та сервера. Такий засіб захисту може завадити викрадачам облікових записів застосовувати знімок екрану в якості повідомлення та підвищити рівень надійності облікових записів користувачів.

Таблиця 5.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Двохфакторна аутентифікація за допомогою стеганографічного підпису електронного зображення	Підвищення рівня захисту облікових записів користувачів	Можливість гарантування підвищенного рівня безпеки користувачів
	Відмова від використання сторонніх сервісів шляхом розгортання власної платформи	Можливість розгорнути власний сервіс на базі шаблону та відмовитись від щомісячної підписки або раптових сбоїв в роботі сервісу
	Відокремлення потенціально важливих сегментів даних	Можливість надавати окремим файлам доступ по двохфакторній аутентифікації

Визначений перелік переваг ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності. Як видно з таблиці 5.1 перевагами є підвищенний рівень захисту, самостійність та гнучкість.

5.2 Можливості запуску проекту

Таблиця 5.2

Попередня характеристика потенційного ринку стартап-проекту

	Показники стану ринку	Характеристика
	Кількість головних конкурентів	>30
	Динаміка ринку	Ринок рішень двохфакторної аутентифікації досить великий ті постійно зростає
	Наявність та характер обмежень для входу	Велика кількість конкурентів може буди компенсована унікальним рішенням використовувати стеганографічний підпис
	Специфічні вимоги до стандартизації та сертифікації	Для використання двохфакторної аутентифікації в системі користувачів, що виконують операції з платіжними картками, має виконуватися стандарт PCI DSS

Таблиця 5.3

Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба в захисті облікових записів	Самостійні та корпоративні клієнти	Корпоративні клієнти вимагають оптових тарифів, або розгортання власного сервісу	Ідеальне співвідношення ціна-якість

Таблиця 5.4

Фактори загроз

	Фактор	Зміст загрози	Можлива реакція компанії
	Зависока конкуренція	Багато різних рішень провокують постійні зміни в політиці конкурентів	Необхідність слідкування за конкурентами та проводження регулярніх опитувань користувачів
	Обмежений функціонал	Націленність на конкретну задачу може бути недоліком	Сертифікувати усіма можливими засобами, щоб зробити продукт більш надійним порівняно з конкурентами

5.3 Технологічний аудит

Таблиця 5.5

Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технології
1	Двохфакторна аутентифікація за допомогою стеганографічного підпису електронного зображення	Microsoft Visual Studio	Є в наявності	Доступні по тріальній ліцензії
2		MongoDB нереляційна база даних	Є в наявності	Доступні безкоштовно
3		Бібілотеки C#	Є в наявності	Доступні безкоштовно
4		Microsoft .Net Framework 4.8	Є в наявності	Доступні безкоштовно

5.4 Розроблення ринкової стратегії продукту

Таблиця 5.6

Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті
1	Провайдери засобів захисту	Не всі готові прийняти	Є в наявності	Висока
2	Приватні компанії	Не всі готові прийняти	Є в наявності	Висока
3	Державні установи	Не готові прийняти	Є в наявності	Низька
4	Навчальні та дослідницькі програми	Не всі готові прийняти	Є в наявності	Низька

Таблиця 5.7

Визначення базової стратегії розвитку

	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкуренто-спроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
	Демонстрація можливостей продукту шляхом публікації науково-популярних статей та відеоматеріалів в соцмережах	Співпраця з цільовими групами	Вільне розповсюдження для некомерційних користувачів	Оновлення продукту відповідно до побажань клієнтів та відповідно до оновлень конкурентів

Таблиця 5.8

Визначення базової стратегії розвитку

№	Чи є проект першим в своєму роді на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде продукт копіювати основні характеристики конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Пошук нових користувачів та заохочення існуючих	Основний функціонал буде схожим	Відстеження нових технологій та адаптація продукту

5.5 Розроблення маркетингової стратегії стартап-проекту

Таблиця 5.9

Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Часте оновлення програми	Врахування побажань клієнтів та виправлення помилок в найкоротші терміни	Створення відділу підтримки для аналізування реагування на відгуки клієнтів
2	Формування звітів з результатами моделювання	Можливість формувати звіти по роботі програми з детальним описом процесу моделювання та оцінкою ефективності мережі, що тестувалася	Модуль для формування звітів з результатами роботи програми

Таблиця 5.10

Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="2">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="2">Двохфакторна аутентифікація за допомогою стеганографічного підпису електронного зображення</td></tr>
<tr><td rowspan="7">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>Розмір</td></tr>
<tr><td>1. Модуль впровадження стеганографічного підпису</td><td>5МБ</td></tr>
<tr><td>2. Модуль графічного інтерфейсу</td><td>20МБ</td></tr>
<tr><td>3. Модуль аналізу результатів та статистики</td><td>5МБ</td></tr>
<tr><td colspan="2">Якість: внутрішнє тестування програми, логування збоїв та система зворотнього зв’язку для повідомлення про неналежну роботу програми</td></tr>
<tr><td colspan="2">Пакування: продаж електронних ключів-ліцензій та інструкцією, що надсилається на електронну адресу замовника</td></tr>
<tr><td colspan="2">Марка: назва організації-розробника + назва товару</td></tr>
<tr><td rowspan="2">III. Товар із підкріпленням</td><td colspan="2">До продажу: реалізація системи</td></tr>
<tr><td colspan="2">Після продажу: електронна версія додатку з ключем</td></tr>
<tr><td colspan="3">За рахунок чого потенційний товар буде захищено від копіювання: прив’язка копії програмного продукту до конкретного ПК та активація програми шляхом введення ліцензійованого ключа, або шляхом надання послуг без передачі програмного продукту замовнику</td></tr>
</table>

Таблиця 5.11

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі становлення ціни на товар/послугу
1	0$ - 100$	0-100	>100000$/місяць	20-100

Таблиця 5.12

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Покупка ліцензії на продукт або договір про надання послуг без передачі ПО до замовника	Розміщення на власному сайті для електронної дистрибуції	Канал збуту одноріневий (через роздрібну дистрибуцію)	Вертикальна (право власності залишається у розробника ПЗ)

Таблиця 5.13 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Купують товар на вимогу	Тематичні зустрічі, конференції, презентації тощо	Надання послуг тестування мереж розробленим продуктом, продаж ПЗ	Демонстрація основних функцій розробленого продукту	Охоплення аудиторії, пояснення функцій та можливостей розробленого продукту та його переваг

Висновки

П’ятий розділ присвячений розробці стартап-проекту для розроблюваного продукту, зроблено опис ідеї проекту, проведено детальний аналіз конкурентів та потенційних можливостей для реалізації продукту на даному ринку. Для дослідження в рамках розробки стартап-проекту було виконано наступні завдання:

1. Наведено загальний опис ідеї проекту для виходу на ринок, описаний приблизний функціонал проекту, визначено конкурентів та проведено порівняння функціоналу продуктів від конкурентів для визначення переваг та недоліків розроблюваного продукту в порівнянні з вищезгаданими конкурентами.
2. Проведено технічний аудит проекту і визначено можливості реалізації даного проекту.
3. Проаналізовано ринкові можливості для запуску проекту, наведено порівняльну характеристику перспектив запуску з конкурентами, створено план та ринкову стратегію розвитку продукту, визначено цільові групи та способи просування проекту в маси.
4. Розроблено маркетингову програму для просування продукту на ринку, описано способи та основні напрямки збуту, визначно першочергові цільові групи та маркетингові стратегії для розширення клієнтської бази.

В підсумку, було отримано стартап-проект для запуску розробленого програмного продукту на ринок, отримано навички створення стартап-проектів, побудови маркетингової стратегії та аналізу обраного ринку.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У рамках даної дипломної роботи розроблено програму для приховування текстової інформації в зображенні за допомогою модифікованих алгоритмів впровадження. Для досягнення цієї мети були виконані наступні завдання:

– проаналізовано методи впровадження інформації в зображення різних форматів;

– досліджено існуючі програми для впровадження інформації в зображення;

– удосконалено метод впровадження інформації в зображення;

– розроблено конкурентоспроможну програму на основі розробленого методу;

– протестувано розроблений програмний додаток.

У роботі обрано метод LSB у якості формату зображень застосовано bmp файлу.

Суть цього методу полягає в заміні останніх значущих бітів в контейнері (зображення, аудіо або відеозапису) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.

Також використано для шифрування / дешифрування bmp файл, який не містить палітру. В такому bmp файлі кожні 3 байта визначають 3 кольори пікселя.

Під стійкістю стеганографії звичайно розуміють число («гіпотетичних», «елементарних») операцій, які слід виконати для обчислення секретного ключа з відкритого ключа або, якщо досліджується симетричний алгоритм шифрування, для розшифровки повідомлення без знання ключа.

Проводиться класифікація атак порушника та розкриваються моделі приховування даних у нерухомих зображеннях. Описані алгоритми

нанесення стенографії на нерухомі зображення, до кожного з алгоритмів наведено блок-схеми для більш повного донесення викладеного матеріалу у межах розділу.

Під час тестування збоїв та недоліків у роботі засобу стеганографічного підпису електронного зображення не виявлено, що говорить про високу якість розробки та можливість впровадження у роботу за потребою

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Овчарук І. В. Аналіз чутливості зорового сприйняття інформації людиною на основі стеганографічного методу LSB / І. В. Овчарук, А. А. Пристінська // Водний транспорт. – 2019. – Вип. 1. – С. 151-158. – Режим доступу: http://nbuv.gov.ua/UJRN/Vodt_2019_1_26.
2. Губенко Н. Е., Сипаков Д. С. Анализ особенностей методов цифровой стеганографии для защиты информации, передаваемой по открытым каналам / Н. Е. Губенко, Д. С. Сипаков // Информатика и кибернетика. – Донецк: ДонНТУ, 2015. – № 2. С. 28-37.
3. Антощук С.Г. Прогнозирование количества ошибок на этапе эксплуатации адаптируемых учетных информационных систем. / С.Г. Антощук, Д.А. Маевский, С. А. Яремчук // Радиоэлектронные и компьютерные системы, НТЖ, Харьков: ХАИ. – 2017. – № 6(47). – С. 204-209.
4. Гуда А.Н., Калинин Т.С., Чернов А.В. Реализация надежного программного обеспечения задач технической диагностики информационно-управляющих систем // Известия высших учебных заведений. Северо-Кавказскийрегион. Технические науки, 2011. – №4. – С.26-31.
5. Систематизація та класифікація наявних стеганографічних методів приховування інформації [Електронний ресурс] / О. В. Весельська, Р. В. Зюбіна, О. В. Фролов // Наукоємні технології. – 2016. – № 2. – С. 187-194.
6. Юдін О. К. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів / О. К. Юдін, Р. В. Зюбіна, О. В. Фролов // Радиоэлектроника и информатика. – Х. : НХНУРЕ, 2015. – № 3. – С. 24–31.

7. Пескова О. Ю. Применение сетевой стеганографии для защиты данных, передаваемых по открытым каналам Интернет / О. Ю. Пескова, Г. Ю. Халабурда // Материалы научной конференции» Интернет и современное общество». – 2012. – С. 348–354.

8. Yamada S., Ohba M., Osaki S.. S-shaped Reliability Growth Modeling for Software Fault Detection. IEEE Transactions on Reliability, 1983. – vol. 32. – no. 5. – pp. 475–484.

9. Yamada S., Osaki S.. Software Reliability Growth Modeling: Models and Applications. IEEE Transactions on Software Engineering, 1985. – vol. 11. – no. 12. – pp. 1431–1437.

10. Kuo S. Y., Hung C. Y. and Lyu M. R., “Framework for modeling software reliability, using various testing efforts and fault detection rates”, IEEE Transactions on Reliability, 2007. – Vol. 54. – №.2. – pp 198-211.

11. J. McCall, P. Richards, G. Walters. Factors in Software Quality. three volumes, NTIS AD-A049-014, AD-A049-015, AD-A049-055, November 1977.

12. B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. MacLeod, and M. J. Merritt. Characteristics of Software Quality. North Holland, 1978.

13. B. Boehm. Software Risk Management. IEEE Computer Society Press, CA, 1989.

14. International Standard ISO/IEC 9126. Information technology – Software product evaluation – Quality characteristics and guidelines for their use. International Organization for Standardization, International Electrotechnical Commission, Geneva, 1991.

15. В. В. Липаев. Обеспечение качества программных средств. Методы и стандарты. Синтег, Москва, 2001.

16. ISO/IEC 9126-1. 2001. Software engineering – Software product quality – Part 1: Quality model. Geneva, Switzerland: International Organization for Standardization.

17. ISO/IEC DTR 9126-2. 2001. Software engineering – Software product quality – Part 2: External metrics. Geneva, Switzerland: International Organization for Standardization.

18. ISO/IEC DTR 9126-3. 2000. Software engineering – Software product quality – Part 3: Internal metric. Geneva, Switzerland: International Organization for Standardization.

19. Харченко В.С., Тарасюк О.М., Гордеев А.А. Мамутов С.С. Инструментальные средства оценки качества програмного обеспечения // Материалы конф. «Информационные технологии – в науку и образование». – Харьков: Харьк. нац. ун-т радіоєлектроники, 2005. – С. 98-99.

20. ДСТУ 3918–1999 (ISO/IEC 12207:1995). Інформаційні технології. Процеси життєвого циклу програмного забезпечення. – Введ. 01.07.2000. – К.: Держстандарт України, 1995. – 55 с.

21. Лайза Криспин, Джанет Грегори Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. – М.: «Вильямс», 2010. – 464 с.

22. Кент Бек. Экстремальное программирование: разработка через тестирование. – «Питер», 2003.

23. Juran, Joseph M., Frank M. Gryna, Juran's Quality Control Handbook. Mcgraw-Hill.

24. Walter A. Shewhart, Economic Control of Quality of Manufactured Product // 50th Anniversary Commemorative Issue. American Society for Quality December 1980, 501 p.

25. Майерс Г. Искусство тестирования программ. – М: Финансы и статистика. – 1982. – 176 с.

26. Нельсона Э. Обеспечение качества программных средств. Методы и стандарты. – М.: СИНТЕГ, 2001. – 380 с.

27. Канер С., Фолк Д., Нгуен Е.К. Тестирование программного обеспечения: Пер с англ. - К.: DiaSoft. – 2000. – 544 с.

28. James Martin: Rapid Application Development, Macmillan Coll Div.

29. Beck, K., Extreme Programming Explained: Embrace Change, Addison-Wesley, 1999.

30. Manifesto for Agile Software Development, Agile Alliance, 2001, webpage: Manifesto-for-Agile-Software-Dev.

31. Agilian - All-in-one Modeling Tool supporting Agile Modeling. Support latest UML, BPMN, ERD, DFD and Textual Analysis, webpage: http://www.visual-paradigm.com/product/ag/

32. Beck, K. Test-Driven Development by Example, Addison Wesley, 2003

33. Jones E.L. Integrating testing into the curriculum – arsenic in small doses. / Jones E.L. // In Proc. 32nd SIGCSE Technical Symp. Computer Science Education, ACM, 2001, p. 337-341.

34. Бернет С., Пейн С.: Криптография. Официальное руководство RSA Security – М. «Бином», 2012. – 325 с.

35. Венбо Мао Современная криптография: теория и практика = Modern Cryptography: Theory and Practice. – М.: «Вильямс», 2005. – С. 768.

36. Воробьев В.И., Грибунин В.Г. Теория и практика вейвлет-преобразования. – СПб: ВУС, 2009. – 325 с.

37. Зима В.: Безопасность глобальных сетевых технологий – «БХВ-Петербург», 2011. – 344 с.

38. Нильс Фергюсон, Брюс Шнайер Практическая криптография : Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – М.: «Диалектика», 2012. – С. 432.

39. Павлов К.А Компьютерная безопасность. Криптографические методы защиты. ДМК Москва, 2010. – 233 с.

40. Ростовцев А.Г. , Михайлова Н.В. Методи криптоаналізу класичних шифрів. – К.: «Наука», 2012. – С. 142.

41. Саломан А. Криптографія з відкритим ключем. – К.: «Наука», 2013. – 342 с.

42. Серов Р.Е., Гончаров В.В., Основы современной криптографии – Москва, Горячая линия – Телеком, 2011. – 443 с.

43. Столлингс В. Криптография и защита сетей: теория и практика. М: Вильямс. 2001. Пер. с англ. – 235 с.

44. Чмора А.Л. Сучасна прикладна криптографія. 2-е вид., Стер. – М.: Геліос АРВ, 2012. – 256 с.

45. Шнайер, Брюс. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си – М.: Издательство ТРИУМФ, 2002 – 816 с.

46. Mallat S. A Theory For Multiresolution Signal Decomposition: The Wavelet Representation / / IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. – Vol. 11. – P. 674-693.

47. Shapiro J. Embedded Image Coding Using Zerotrees Of Wavelet Coefficients / / IEEE Transactions on Signal Processing, 1993. – Vol. 41, No. 12.

48. Said A., Pearlman W. A New Fast And Efficient Image Codec Based On Set Partitioning in Hierarchical Trees / / IEEE Transactions on Circuits and Systems for Video Technology, 1996. – Vol. 6. – P. 243-250.

49. Antonini M., Barlaud M., Mathieu P., Daubechies I. Image Coding Using Wavelet transform / / IEEE Transactions On Image Processing, 1992. – Vol. 1, № 2. – P. 205-220