Міністерство освіти і науки України
Вінницький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
при вивченні дисципліни
«СТЕГАНОГРАФІЯ»
для студентів
галузі знань 12 «Інформаційні технології»
спеціальності 125 «Кібербезпека»
(ОПП «Безпека інформаційних і комунікаційних систем»,
«Кібербезпека критичних систем»)

Вінниця 2019

УДК 004.056

Укладачі:

Віталій Володимирович Лукічов
Аліна Василівна Остапенко-Боженова

Рецензенти:

О. П. Войтович, кандидат технічних наук, доцент
А. С. Васюра, кандидат технічних наук, професор

Методичні вказівки до виконання лабораторних робіт при вивченні дисципліни «Стеганографія» / уклад. В. В. Лукічов, А. В. Остапенко-Боженова. – Вінниця: ВНТУ, 2019. – 78 с.

Методичні вказівки призначені для надання допомоги при виконанні лабораторних робіт під час вивчення дисципліни "Стеганографія". Лабораторний практикум містить перелік відповідних тем та теоретичні відомості по кожній темі, а також порядок виконання роботи та список контрольних запитань та завдань. Методичні вказівки призначені для студентів спеціальності 125 Кібербезпека, ОПП: «Безпека інформаційних і комунікаційних систем», «Кібербезпека критичних систем» денної та заочної форми навчання.

ЗМІСТ

ВСТУП.....6
Лабораторна робота №1.....7
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ТЕКСТОВІ ФАЙЛИ.....7
Методи перекручування формату текстового документа.....7
Синтаксичні методи.....8
Семантичні методи.....9
Опис програми TextHide2.....9
Порядок виконання роботи.....10
Зміст звіту.....11
Контрольні питання.....11
Лабораторна робота №2.....12
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ГРАФІЧНІ ТА АУДІО ФАЙЛИ.12
Вбудовування зображень в аудіо файли.....12
Вбудовування зображень в графічні файли.....13
Опис програми S-Tools.....13
Обробка графічних файлів.....16
Порядок виконання роботи.....17
Зміст звіту.....17
Контрольні питання.....19
Лабораторна робота №3.....20
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ГРАФІЧНИХ ФАЙЛАХ.....20
Алгоритм JPEG.....22
Опис роботи алгоритму.....22
Функціональні можливості програми.....23
Опис програми.....24
Порядок роботи з програмою.....24
Порядок виконання роботи.....27
Зміст звіту.....28
Контрольні питання.....28
Лабораторна робота №4.....29
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯ ТА АУДІО ФАЙЛИ.....29

Вбудовування зображень в аудіо файли.....29
Вбудовування зображень в графічні файли. 29
Опис програми. 31
Порядок виконання роботи.....33
Зміст звіту.....34
Контрольні питання.....34
Лабораторна робота №5.....35
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У BMP ЗОБРАЖЕННЯ. 35
Опис програми 35
Режими роботи з програмою пункт “Опції”.....37
Опції Командного рядка.....39
Робота з програмою.....40
Порядок виконання роботи.....43
Зміст звіту.....44
Контрольні питання.....44
Лабораторна робота №6.....45
ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ ПРОГРАМИ IMAGESPYER G2.....45
Програма ImageSpyer G2.....46
Налаштування програми 49
Порядок виконання роботи.....50
Зміст звіту.....50
Контрольні питання.....50
Лабораторна робота №7.....51
ПРИХОВУВАННЯ ДАНИХ ЗА ДОПОМОГОЮ ПРОГРАМИ “MSU STEGO VIDEO”. 51
Опис програми «MSU stego video».....51
Порядок виконання роботи.....57
Зміст звіту.....57
Контрольні питання.....57
Лабораторна робота №8.....58
ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЗВУКОВИХ ФАЙЛАХ.....58
Mp3Stego 58
Порядок виконання роботи.....60

Зміст звіту.....60
Контрольні питання.....60
Лабораторна робота №9.....61
ШИФРУВАННЯ ТЕКСТУ ТА ЗОБРАЖЕННЯ.....61
Функціональні можливості програми.....61
Порядок виконання роботи.....68
Зміст звіту.....68
Контрольні питання.....68
Лабораторна робота №10.....69
ПРИХОВУВАННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ ПРОГРАМИ OUR SECRET.....69
Опис програми:.....69
Приклад приховування інформації за допомогою програми OurSecret:.....70
Приклад діставання прихованої інформації з файлу за допомогою програми OurSecret:.....71
Порядок виконання роботи.....72
Зміст звіту.....74
Контрольні питання.....74
Лабораторна робота №11.....75
РОЗРОБКА СТЕГАНОГРАФІЧНОЇ ПРОГРАМИ.....75
Порядок виконання роботи.....75
Зміст звіту.....76
Контрольні питання.....76
ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....77
ГЛОССАРІЙ.....78

ВСТУП

З метою більш повного і всебічного засвоєння навчального матеріалу студентами у робочій навчальній програмі дисципліни «Стеганографія» передбачено вид занять – лабораторна робота. Даний вид засвоєння матеріалу передбачає опрацювання літературних джерел, вивчення публікацій фахових видань, присвячених різноманітним методам та способам стеганографії, а також застосування практичнх навичок для вирішення конкретних задач у галузі захисту інформації.

Метою викладання навчальної дисципліни «Стеганографія» є формування знань і навичок, необхідних для синтезу й аналізу алгоритмів приховування даних і вбудовування даних у різного виду інформацію, розробки програмних та апаратних засобів, що реалізують ці алгоритми, практичного застосування методів і засобів приховування інформації.

Основними завданнями вивчення дисципліни «Стеганографія» є ознайомлення студентів із сучасними стеганографічними методами та алгоритмами, відомими підходами до аналізу прихованої пропускної здатності стегоканалу та тенденціями розвитку стеганографії, навчити їх методам синтезу й аналізу стегоалгоритмів.

Згідно з вимогами освітньо-професійної програми студенти повинні:

знати: області застосування і вимоги, що висуваються до стеганографії; математичну модель стегосистеми і стеганографічні протоколи; види атак на стегосистеми і методи протидії їм; різновиди стегосистем; підхіди до оцінювання стійкості стеганографічних систем; методи приховування даних у нерухомих зображеннях; стегоалгоритми вбудовування інформації в зображення; методи приховування даних в аудіосигналах; методи приховування даних у відеопослідовностях; перспективи розвитку стеганографії; особливості практичного використання методів і алгоритмів приховування даних і вбудовування цифрових водяних знаків у різного виду інформацію;

вміти: самостійно оцінювати стійкість стеганографічної системи; здійснювати вибір стеганографічних методів та алгоритмів згідно з вимогами конкретного застосування; розробляти програмні та апаратні засоби, що реалізують стеганографічні методи та алгоритми; практично застосовувати відомі програмні засоби приховування даних і вбудовування цифрових водяних знаків у різного виду інформацію; користуватись науковою та довідковою літературою за напрямком дисципліни; знаходити раціональні методи розв'язання практичних задач.

Робота студентів розділена на відповідні лабораторні заняття, що включають перелік відповідних тем. По кожній з тем пропонується виконати конкретні завдання, а також відповісти на ряд контрольних запитань.

Лабораторна робота №1

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ТЕКСТОВІ ФАЙЛИ.

Мета роботи: Вивчення методів приховування інформації у текстових файлах, та набуття навичок приховування інформації за допомогою програми TextHide2.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Методи лінгвістичної стеганографії (приховування секретних повідомлень у текст) використовують або природну надлишковість мови, або формати представлення тексту. До цих методів належать:

- методи перекручування формату текстового документа;
- синтаксичні методи;
- семантичні методи.

Методи перекручування формату текстового документа

Такі методи маніпулюють інтервалами між словами і реченнями, або ж пропусками наприкінці текстових рядків. Використання пропусків для приховування даних обумовлено такими причинами. По-перше, введення додаткових пропусків не вносить великих змін у значення фрази або речення. По-друге, у випадкового читача навряд чи відразу виникне підозра щодо вставлених додаткових пропусків (у бітовому представленні).

Приховування таємного повідомлення можна проводити шляхом додавання одного або двох символів пропуску наприкінці речень після символу кінця. Наприклад, після крапки – для природної мови або після крапки з комою – для коду програми мовою С. При цьому один додатковий пропуск кодує значення біта 0, а два – 1. Цей метод є простим, але недостатньо ефективним, оскільки вимагає контейнер великого обсягу (швидкість передавання прихованих даних у даному випадку приблизно дорівнює одному бітові на 160 байтів тексту). Крім того, можливість приховування залежить від структури тексту (деякі тексти, типу “білі вірші”, не мають чітких знаків кінця).

Кодувати секретні дані можна додатковими пропусками наприкінці кожного рядка тексту (рис. 1.1). Тут два пропуски кодують 1, один пропуск – 0. Перевага такого методу приховування полягає в тому, що він може бути застосований до будь-якого тексту; зміни в форматі різко не впадають в око читачеві, забезпечується передавання більшої кількості прихованих даних у порівнянні з попереднім методом (1 біт на 80 байтів). Недолік методу полягає в тому, що деякі програми можуть необережно видаляти

додаткові пропуски. Крім цього, приховані в такий спосіб дані не завжди можуть бути відновлення з друкованої копії документу.

В	і	л	л	і		м	о	ж	е		п	е	р	е	х	о	п	и	т	и
с	т	е	г	а	н	о	г	р	а	м	у	,		п	о	с	л	а	н	у
в	і	д		А	л	і	с	и		д	о	б	о	б	а	.				

В	і	л	л	і		м	о	ж	е		п	е	р	е	х	о	п	и	т	и		
с	т	е	г	а	н	о	г	р	а	м	у	,		п	о	с	л	а	н	у		
в	і	д		А	л	і	с	и		д	о	б	о	б	а	,						

Рисунок 1.1 – Приклад приховування даних 101 пропусками наприкінці текстових рядків

Ще один метод приховування даних за допомогою пропусків маніпулює з текстами, які вирівняні з обох боків. У цьому методі дані кодуються шляхом керованого вибору місць для розміщення додаткових символів пропуску. Один символ між словами інтерпретується як “0”, а два – як “1”. Метод дозволяє вбудувати декілька бітів прихованої інформації в кожен рядок тексту (рис. 1.2).

Під_час__активних_атак,_коли_немає_можливості
__витягти__приховану_інформацію__або_довести_
_її__існування,_її__можна__знищити__простим__
додаванням_у__стеганограм_випадкових__даних.

Рисунок 1.2 – Приклад приховування бітового повідомлення 111100

У зв’язку з процедурою вирівнювання тексту з обох боків не кожен проміжок між словами може використовуватись для кодування прихованих даних. Для того, щоб визначити в якому з проміжків між словами приховано інформація, а які проміжки є частиною оригінального тексту, використовується такий метод декодування. Бітовий рядок який витягається із стеганограм, розбивається на пари. Пари бітів “01” інтерпретується як “1”; пари “10” – як “0”; а пари “00” і “11” є порожніми, тобто такими, що не несуть ніякої інформації. Наприклад, бітове повідомлення “1000101101” скорочується до “001”, а рядок “110011” – буде порожнім.

Синтаксичні методи

До синтаксичних методів лінгвістичної стеганографії відносять методи зміни пунктуації і методи зміни стилю і структури тексту.

В будь-якій мові існують випадки, коли правила пунктуації є неоднозначними і мають слабкий вплив на зміст тексту. Наприклад, обидві форми перерахування “хліб, олія і молоко” і “хліб, олія, молоко” є припустимими. Можна використовувати той факт, що вибір таких форм є довільним і використовувати альтернативний вибір для кодування даних у війковому вигляді. Наприклад, якщо з’являється форма перерахування з сполучником “і”, то кодується “1”, інакше “0”. Для приховування можна також застосовувати скорочення та абревіатури.

У будь-якій мові мається багато можливостей для синтаксичного приховування даних, але вони не часто зустрічаються в типових текстах. Середня швидкість передачі даних такими методами дорівнює декільком бітам на кілобайт тексту.

До синтаксичних методів відносяться методи зміни стилю або структури тексту без істотної зміни його значення або тону. Наприклад, пропозиція “До закінчення ночі я буду готовим” можна представити у виді “Я буду готовий швидше, ніж ніч закінчиться”. Такий підхід більш прозорий, але можливість його обмежена.

Семантичні методи

Семантичні методи стеганографії аналогічні синтаксичним. Для цих методів елементарними лінгвістичними компонентами вважаються окремі слова. Для такої заміни необхідні таблиці синонімів. Кодування секретного повідомлення проводиться вибором синоніма з необхідного місця таблиці. Наприклад першому слову-синоніму відповідає “1”, а другому “0” (табл.1.1). Якщо слову відповідає більша кількість синонімів, то можна кодувати більшу кількість бітів одночасно.

Таблиця 1.1 – Таблиця синонімів

“1”	“0”
Інформація	Дані
Дім	Хата
Захист	Оборона
Малюнок	Рисунок

Опис програми TextHide2

Ця програма наочно демонструє стеганографічне приховування інформації у текстовий файл. Інтерфейс програми дуже простий. Він зображений на рис.1.3.

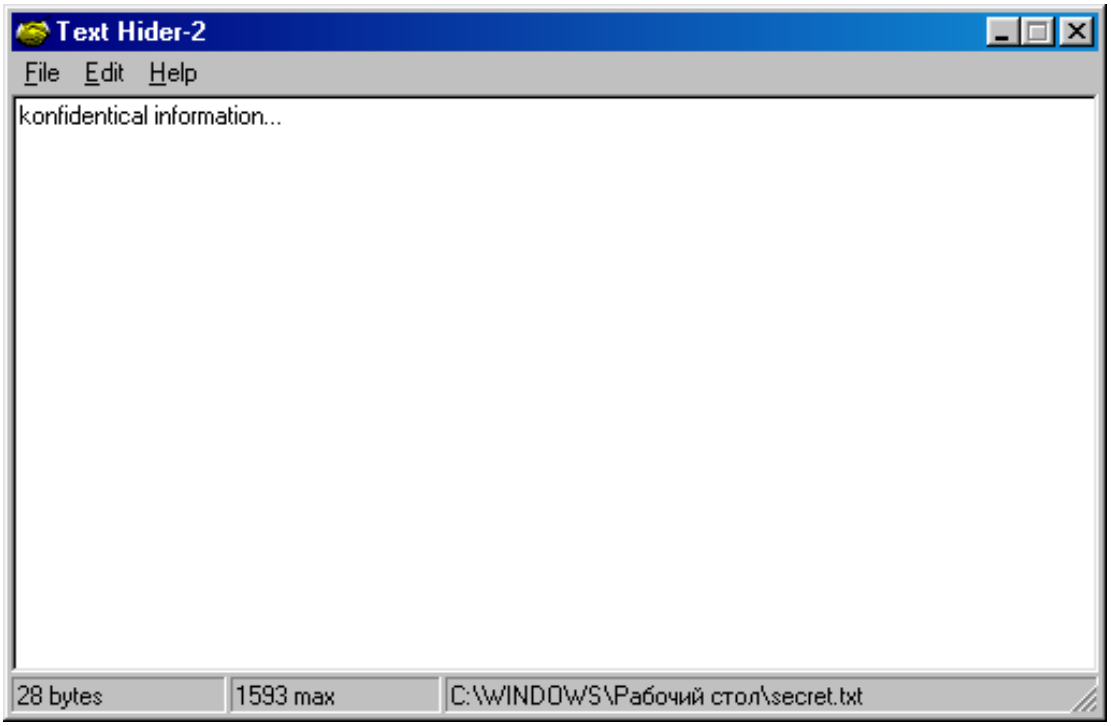


Рисунок 1.3 – Інтерфейс програми **TextHide2**

Після запуску програми в меню File вибираємо Open(text). Вибраний файл і буде контейнером. Далі, скориставшись Edit->Clear ми очистимо поле вводу інформації. Записавши необхідну секретну текстову інформацію, яка не повинна перевищувати максимальних розмірів (вони вказані в низу вікна), натискаємо File->Save As, і отримуємо заповнений контейнер. Щоб подивитись приховану інформацію треба виконати File->Open.

Оскільки, відкривши будь-який заповнений контейнер отримуємо доступ до прихованої інформації, в програмі організовано шифрування вмісту контейнера. Щоб його активізувати необхідно зайти File->Data protection, у діалоговому вікні поставити галочку і ввести пароль (не менше 5 символів). Потім зберегти контейнер. Для вилучення інформації, необхідно заповнити це діалогове вікно, а потім відкрити файл- контейнер.

Щоб вставити в контейнер довільний файл і вилучити його необхідно використовувати такі пункти меню: Import from, Export to відповідно.

Порядок виконання роботи

1. Вивчити короткі теоретичні відомості про методи лінгвістичної стеганографії.
2. Ознайомитися з функціональними можливостями програми TextHide2.

3. Вбудувати у будь-який файл типу *.txt текст, що складається з власного прізвища, ім’я та групи.
4. Візуально проаналізувати заповнений та незаповнений контейнери. Зробити висновки.
5. Вилучити текст із заповненого контейнера і порівняти його візуально з текстом, який вбудовувався. Зробити висновки.
6. Вбудувати текст у контейнер, використовуючи режим захисту інформації, та вилучити його. При цьому візуально проаналізувати заповнений та незаповнений контейнери та тексти до вбудовування і після вилучення.
7. Вбудувати у контейнер попередньо створений логотип, та вилучити його. При цьому візуально проаналізувати заповнений та незаповнений контейнери та логотипи до вбудовування і після вилучення.
8. Оформити звіт про виконання лабораторної роботи.

Примітка: усі маніпуляції проводити з файлами-контейнерами типу *.txt, у яких розмір більший ніж 40 кілобайт.

Зміст звіту

Звіт повинен містити:
- короткі відомості про методи лінгвістичної стеганографії;
- короткі відомості про програму TextHide2;
- інформацію про обсяги контейнеру, тексту і логотипу;
- висновки про результати експериментів.

Контрольні питання

1. Наведіть переваги та недоліки методів перекручування формату текстового документу.
2. Наведіть приклад синтаксичного стегано-методу.
3. Наведіть приклад семантичного стегано-методу.
4. Наведіть приклад лінгвістичного стегано-методу.

Лабораторна робота №2

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ГРАФІЧНІ ТА АУДІО ФАЙЛИ.

Мета роботи: Дослідження стеганографічної програми S-Tools.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Сьогодні з'явилося багато нових областей застосування стеганографії в комп'ютерній техніці. Повідомлення вбудовують тепер у цифрові дані, як правило ті, що мають аналогову природу. Це - мова, аудиозаписи, зображення, відео. Відомі також пропозиції по вбудовуванні інформації в текстові файли й у файли програм, що виконуються.

Існують два основних напрямки в комп'ютерної стеганографії: зв'язаний з цифровою обробкою сигналів і не зв'язаний. В останньому випадку повідомлення можуть бути вбудовані в заголовки файлів, заголовки пакетів даних. Цей напрямок має обмежене застосування в зв'язку з відносною легкістю розкриття і/або знищення схованої інформації. Більшість поточних досліджень в області стеганографії так чи інакше зв'язані з цифровою обробкою сигналів.

Вбудовування зображень в аудіо файли

За своїм характером звуки є неточними при оцінці правильного значення звукової хвилі в специфічний момент часу. Звуки в WAV файлах зберігаються у форматах по 8 або 16 бітів, і саме в такому вигляді вони попадають до DA- конвертера на звуковій карті. Оскільки число кодують 8 бітів, це означає, що значення можуть варіюватися між 0 і 255. При форматі в 16 бітів відповідно діапазон лежить між 0 і 65535.

S-Tools під час роботи розподіляє розрядний образ файлу, який ви хочете сховати, в найменш значимих бітах звукової послідовності.

Наприклад, припустимо, що звукова послідовність має восьми бітний формат:

132 134 137 141 121 101 74 38

У двійковій системі, це:

10000100 10000110 10001001 10001101 01111001 01100101 01001010 00100110

(LSB кожного байта, підкреслено)

Припустимо, що ми хочемо сховати подвійний байт 11010101 (213) всередині цієї послідовності. Ми просто замінюємо LSB (найменш

значимий біт) кожного типового байта з відповідним бітом від байта, що ми пробуємо ховати. Так що вищезгадана послідовність зміниться на:

133 135 136 141 120 101 74 39

У двійковій системі, це:

10000101 10000111 10001000 10001101 01111000 01100101 01001010 00100111

Ви можете ясно бачити, що значення елементів змінилися максимум на 1. Це не сприймається людським вухом, але все-таки ми сховали 8 бітів інформації в послідовності.

Вбудовування зображень в графічні файли

Усі комп'ютерні зображення складені з масиву точок, названих пікселями, що утворюють собою сітку. Кожний з цих пікселів має власний колір, представлений внутрішньо як окремі значення складових: червоного, зеленого і синього кольорів. У 24 бітному форматі, кожний з цих кольорових рівнів може мати значення між 0 (відсутність кольору) і 255 (максимальна кількість кольору). Піксель зі значенням RGB 0 0 0 є чорним, а зі значенням 255 255 255 - білим.

S-Tools під час роботи змінює молодші розряди файлу, що ви хочете сховати в найменш істотних бітах кольорових рівнів у зображенні.

Для 24 розрядних образів це просто, тому що оскільки числа в RGB потроюються, і усе, що ми повинні робити - розподіляти наші біти в молодші біти контейнера.

Значно складніше сховати що-небудь у зображеннях з максимумом в 256 кольорів(відтінки сірого). Тому що зображення-контейнер може вже мати більш ніж 200 кольорів, і наше втручання буде наближати його до абсолютного максимуму -255.

Просто побачити, що зображення з 32 або менш кольорами ніколи не перевищить 256 кольорів, незалежно від того наскільки ми в нього втручаємося. Щоб побачити це, представте номер RGB як номер на 3 біти. У процесі приховання, ми можемо змінити 3 розряди, а це - 8 можливих значень, один із яких - первісний зразок. Якщо один колір можна 'розгорнути' на 8 кольорів, то кількість кольорів, що ми можемо використовувати, перш ніж потрапимо в небезпеку перевищення межі в 256 кольорів дорівнює 256/8=32 кольорів.

Опис програми S-Tools

Розглянемо використання цифрової стеганографії на прикладі програми S-Tools (версія 4.00), написаної англійським програмістом Ендрю Брауном (Andrew Brown). Програма S-Tools може використовувати для контейнерів наступні мультимедійні формати (табл. 2.1).

Таблиця 2.1 – Формати мультимедійних файлів

Графічні	Аудіо
Bitmap (bmp) * GIF	Wave audio (wav) *

* - тільки непаковані файли

При вбудовуванні інформація, попередньо шифрується. Криптопротоколи, які дозволяє використовувати програма містяться в табл. 2.2.

Таблиця 2.2 – Підтримувані криптоалгоритми.

Назва криптопротоколу	Діапазон розміру ключів (символів)
IDEA	1 - 256
DES	1 - 256
Triple DES	1 - 256
MDC	1 - 256

Головне вікно програми представлено на рис.2.1.

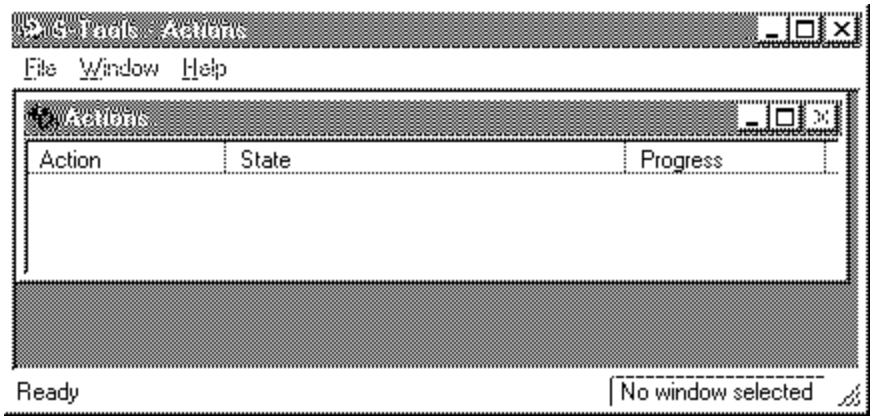


Рисунок 2.1- Головне вікно програми

Програма S-Tools має зручний інтерфейс побудований на принципі перетягування і контекстних меню які з’являються при натисненні правою кнопкою миші в потрібну область програми. Імпорт контейнерів здійснюється перетягуванням їх у область вікна програми, результати перетягування відображені на рис.2.2 і рис.2.3.

Рисунок 2.2 – Імпорт графічного файлу

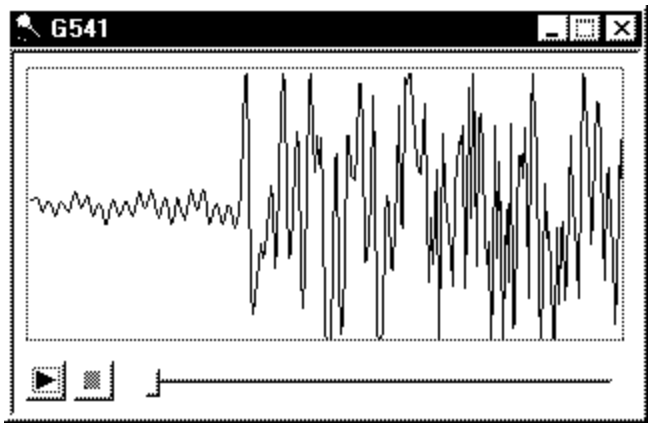


Рисунок 2.3 – Імпорт аудіо файлу

В інформаційній стрічці при цьому буде відображена максимальна кількість інформації, яку обраний контейнер може містити. Далі в область контейнера перетягуємо файл який підлягає схову. В наслідок цих дій з'явиться вікно шифрування, яке дозволить вибрати пароль і криптоалгоритм для шифрування (рис.2.4).

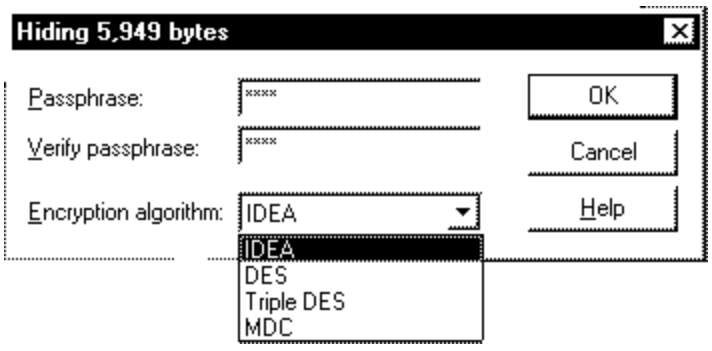


Рисунок 2.4 – Вікно шифрування

Отриманий після процедур шифрування і вбудовування файл можна зберегти через контекстне меню натисненням правої клавіші миші в області контейнера.

Для отримання вбудованої інформації файл-контейнер аналогічно до вищезгаданого імпортують в робочу зону програми і вибирають пункт меню видобути (reveal) з контекстного меню контейнера. З'явиться вікно розшифрування, в якому необхідно вибрати шифр і пароль з якими було зашифровано дані. Після розшифрування з'явиться вікно перегляду архіву (рис2.5), в якому буде показано сховані в контейнері файли.

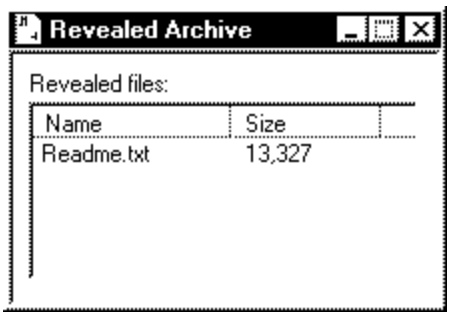


Рисунок 2.5- Відтворення файлу

Потрібний файл можна зберегти за допомогою контекстного меню вікна.

Обробка графічних файлів

Як було сказано вище для підвищення ефективності приховування інформації і захисту від можливих помилок графічні контейнери попередньо проходять обробку. S-tools пробує зменшувати число кольорів в зображенні із максимальним збереженням деталей наскільки це можливо, або збільшує діапазон кольорів (для чорно – білих зображень).

З відповідного меню обробки файлів ви зможете здійснити:

1. Перетворення зображення в 24-бітний формат (Convert to a 24-bit image).
2. Зменшення кольорів зображення (Attempt color reduction):
   а) зональне усереднення кольорів (Median cut box color):
      – центральної області (Center);
      – окремих кольорів (Average colors);
      – окремих пікселів (Average pixels);
   б) розширення області (Dimension choice):
      – кольорів (Large RGB distance);
      – яскравості (Large luminosity distance).

Також після перетворення можна використати алгоритм розмиву Флойда-Штейнберга для знищення слідів переходу (Enable Floyd-Steinberg dithering).

Порядок виконання роботи

1. Вивчити надані теоретичні відомості.
2. Ознайомитися з функціональними можливостями програми S-tools.
3. За номером варіанту в таблиці 3 виконайте наступні завдання
   - Завдання 1
     - Виберіть файл контейнер заданого типу і файл для схову (розмір контейнера має бути достатній для вміщення приховуваних даних).
     - Виконайте вбудовування даних.
     - Порівняйте контейнери до і після вбудовування.
     - Отримайте сховані в контейнері дані.
   - Завдання 2
     - Виберіть файл контейнер заданого типу і файл для схову (розмір контейнера має бути достатній для вміщення приховуваних даних).
     - Виконайте вбудовування даних, попередньо перетворивши контейнер.
     - Порівняйте контейнери до і після вбудовування.
     - Отримайте сховані в контейнері дані.

Для отримання чорно білого контейнера збережіть зображення в режимі bitmap за допомогою графічного редактора.

Для отримання кольорового контейнера (256 кольорів) створіть зображення в режимі RGB використавши для його заповнення інструмент gradient fill з використанням не менше п’яти кольорів за допомогою графічного редактора.

Зміст звіту

Звіт повинен містити:

- короткі відомості про використані методи приховування інформації;
- короткі відомості про програму S-tools;
- висновки про результати експериментів.

Таблиця 2.3 – Варіанти завдань.

варіант	Завдання 1		Завдання 2		
	Контейнер:	Алгоритм шифрування	Контейнер	Тип перетворення	Алгоритм шифрування
1	BMP	IDEA	Кольорове зображення	Усереднення центральної області зображення з розширенням області кольорів	Triple DES
2		DES		Усереднення центральної області зображення з розширенням діапазону яскравості	IDEA
3		Triple DES		Усереднення окремих кольорів зображення з розширенням області кольорів	MDC
4		MDC		Усереднення окремих кольорів зображення з розширенням діапазону яскравості	DES
5		IDEA		Усереднення окремих пікселів зображення з розширенням області кольорів	Triple DES
6		DES	Чорно-біле зображення	Перетворення в 24 – бітне зображення	IDEA
7		Triple DES			MDC
8	WAV	MDC			DES
9		IDEA			Triple DES
10		DES			IDEA
11		Triple DES	Кольорове зображення	Усереднення окремих пікселів зображення з розширенням діапазону яскравості	MDC
12		MDC		Усереднення центральної області зображення з розширенням діапазону яскравості	DES
13		IDEA		Усереднення окремих кольорів зображення з розширенням області кольорів	Triple DES
14	GIF	DES		Усереднення окремих кольорів зображення з розширенням діапазону яскравості	IDEA
15		Triple DES		Усереднення окремих пікселів зображення з розширенням області кольорів	MDC

Контрольні питання

1. Яка суть методів стеганографії без застосування цифрової обробки сигналів.
2. Якими є особливості методів із застосуванням цифрової обробки сигналів.
3. Охарактеризуйте метод вбудовування стегоінформації в найменш значимий біт (LSB).
4. Які існують відмінності при застосуванні метода вбудовування в стего інформації в найменш значимий біт при обробці графічних файлів від обробки аудіо файлів.
5. Яким чином можливе вбудовування інформації в чорно-білі зображення.
6. Яким чином можливе вбудовування інформації в зображення з великою кількістю кольорів
7. Для чого при приховувані інформації застосовується шифрування.
8. Які особливості приховування інформації в чорно-біле та кольорове зображення.

Лабораторна робота №3

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ГРАФІЧНИХ ФАЙЛАХ.

Мета роботи: Вивчити методи вбудовування інформації в графічні файли. Набути навичок вбудовування інформації в графічні файли за допомогою програми “JPHS For Windows ”.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Статичні растрові зображення являють собою двовимірний масив чисел. Елементи цього масиву називають *пікселями* (від англійського pixel – picture element). Усі зображення можна підрозділити на двох груп – з палітрою і без неї. У зображень з палітрою в пікселі зберігається число – індекс у деякому одномірному векторі кольорів , названих *палітрою.* Найчастіше зустрічаються палітри з 16 і 256 кольорів.

Зображення без палітри бувають у якій-небудь системі кольоропредставлення і *у градаціях сірого* . Для останніх значення кожного пікселі інтерпретується як яскравість відповідної точки .

Протягом останніх 15 років у рамках комп'ютерної графіки бурхливо розвивається така область як алгоритми архівації зображень. Існування цієї області обумовлене тим, що зображення – це своєрідний тип даних, який характеризується трьома особливостями:

1. Зображення (як і відео) займають набагато більше місця в пам'яті, ніж текст.
2. Другою особливістю зображень є те, що людський зір при аналізі зображення оперує контурами, загальним переходом квітів і порівняно невідчутно до малих змін у зображенні. Таким чином, ми можемо створити ефективні алгоритми архівації зображень, у яких декомпресія зображення не буде збігатися з оригіналом, однак людина цього не помітить.
3. Ми можемо легко помітити, що зображення, у відмінність, наприклад, від тексту, має надмірність у 2-х вимірах. Тобто як правило, сусідні точки , як по горизонталі, так і по вертикалі, у зображенні близькі по кольору. Крім того, ми можемо використовувати подібність між колірними площинами R, G і B у наших алгоритмах, що дає можливість створити ще більш ефективні алгоритми.

Усього на даний момент відомо мінімум три сімейства алгоритмів, що розроблені винятково для стиску зображень, і застосовувані в них методи практично неможливо застосувати до архівації ще яких-небудь видів даних.

Всі ці алгоритми можна поділити на два типи – алгоритми з втратами і алгоритми без втрат при ущільненні . Алгоритми без втрат досить універсальні і покривають усі типи зображень, але – у них, по сьогоднішніх мірках, занадто маленький коефіцієнт архівації. Використовуючи один з алгоритмів ущільнення без втрат, можна забезпечити архівацію зображення приблизно в два рази. У той же час алгоритми ущільнення з втратами оперують з коефіцієнтами 10-200 разів. Крім можливості модифікації зображення, одна з основних причин подібної різниці полягає в тім, що традиційні алгоритми орієнтовані на роботу з ланцюжком. Вони не враховують, так звану, "когерентність областей" у зображеннях. Ідея когерентності областей полягає в малій зміні кольору і структури на невеликій ділянці зображення.

Розвиток теорії і практики алгоритмів ущільнення призвів до виникнення різних ідей приховування корисної інформації в зображення. Справа в тім, що зображення є дуже зручними для використання їх як стегоконтейнерів. Це обумовлено багатьма причинами:

1. заздалегідь відомим розміром контейнера;
2. відсутністю обмежень, обумовлених вимогами реального часу;
3. наявністю в більшості реальних зображень текстурних зон, що мають шумову структуру і добре придатні для вбудовування інформації;
4. слабкою чутливістю людського ока до незначних змін кольорів зображення, його яскравості, контрастності, змісту в ньому шуму, викривленням поблизу контурів;
5. існуванням практично значимої задачі захисту фотографій, картин, відео від незаконного тиражування і поширення;
6. відносно великим обсягом цифрового представлення зображень, що дозволяє упроваджувати ЦВЗ великого обсягу або підвищувати робастність впровадження;

При впровадженні прихованої інформації в зображення, раніше користувались методом вкладання в незначні біти для зменшення візуальної помітності. Але більш сучасний підхід полягає у вбудовуванні інформації у найбільш істотні зони зображення , руйнування яких призведе до повної деградації всього зображення. Стегоалгоритми враховують властивості людського зору аналогічно алгоритмам ущільнення. Зазвичай використовуються такі самі перетворення: наприклад дискретне косинусне перетворення як в JPEG, чи вейвлет-перетворення як в JPEG2000). Причому існують три можливості вкладення інформації:

1. у вихідне зображення,
2. одночасно з процесом ущільнення зображення-контейнера
3. у вже ущільнене алгоритмом JPEG зображення.

Алгоритм JPEG

JPEG – один із самих популярних і досить могутніх алгоритмів. Практично він є стандартом де-факто для повнокольорових зображень. Оперує алгоритм областями 8x8, на яких яскравість і колір міняються порівняно плавно. Унаслідок цього, при розкладанні матриці такої області в подвійний ряд по косинусах значимими виявляються тільки перші коефіцієнти. Таким чином, стиск у JPEG здійснюється за рахунок плавності зміни квітів у зображенні.

Алгоритм розроблений групою експертів в області фотографії спеціально для стиску 24-бітних зображень. JPEG – Joint Photographic Expert Group – підрозділ у рамках ISO – Міжнародної організації по стандартизації. У цілому алгоритм заснований на дискретному косинусному перетворенні (надалі ДКП), застосовуваному до матриці зображення для одержання деякої нової матриці коефіцієнтів. Для одержання вихідного зображення застосовується зворотне перетворення.

ДКП розкладає зображення по амплітудах деяких частот. Таким чином, при перетворенні ми одержуємо матрицю, у якій багато коефіцієнтів або близькі, або дорівнюють нулеві. Крім того, завдяки недосконалості людського зору, можна апроксимувати коефіцієнти більш грубо без помітної втрати якості зображення.

Для цього використовується квантування коефіцієнтів. У найпростішому випадку – це арифметичне побітове зрушення вправо. При цьому перетворенні губиться частина інформації, але можуть досягатися великі коефіцієнти стиску.

Опис роботи алгоритму

Отже, розглянемо алгоритм докладніше. Нехай ми стискаємо 24-бітне зображення.

Крок 1. Переводимо зображення з колірного простору RGB, з компонентами, що відповідають за червоний (Red), зелений (Green) і синій (Blue) складові кольори точки , у колірний простір YCrCb (іноді називають YUV). У ньому Y – складова яскравості , а Cr, Cb – компоненти, що відповідають за колір (хроматичний червоний і хроматичний синій). За рахунок того, що людське око менш чуттєве до кольору, чим до яскравості, з'являється можливість архивувати масиви для Cr і Cb компонент із великими втратами і, відповідно, великими коефіцієнтами стиску.

Крок 2. Етап дискретизації. Розбиваємо вихідне зображення на матриці 8x8. Формуємо з кожної три робочі матриці ДКП – по 8 біт окремо для кожного компонента. На цьому етапі ми втрачаємо 3/4 корисної інформації про колірні складові зображення й одержуємо відразу ущільнення у два рази. Ми можемо вчиняти так завдяки роботі в просторі

YCrCb. На результуючому RGB зображенні, як показала практика, це позначається несильно.

Крок 3. Застосовуємо ДКП до кожної робочої матриці. При цьому ми одержуємо матрицю, у якій коефіцієнти в лівому верхньому куті відповідають низькочастотній складовій зображення, а в правому нижньому – високочастотній.

Крок 4. Робимо квантування. У принципі, це просто розподіл робочої матриці на матрицю квантування поелементно. Для кожного компонента (Y, U і V), у загальному випадку, задається своя матриця квантування q[u,v]. На цьому кроці здійснюється керування ступенем ущільнення, і відбуваються самі великі втрати.

Крок 5 . Переводимо матрицю 8x8 у 64-елементний вектор за допомогою "зигзаг"-сканування, тобто беремо елементи з індексами (0,0), (0,1), (1,0), (2,0)... Таким чином, на початку вектора ми одержуємо коефіцієнти матриці, що відповідають низьким частотам, а наприкінці – високим.

Крок 6. Згортаємо вектор за допомогою алгоритму групового кодування - RLE.

Крок 7. Згортаємо отримані пари кодуванням по Хаффману з фіксованою таблицею. Процес відновлення зображення в цьому алгоритмі цілком симетричний. Метод дозволяє стискати деякі зображення в 10-15 разів без серйозних втрат.

Істотними позитивними сторонами алгоритму є те, що:

1. Задається ступінь ущільнення.
2. Вихідне кольорове зображення може мати 24 біта на точку.

Негативними сторонами алгоритму є те, що:

1. При підвищенні ступеня стиску зображення розпадається на окремі квадрати (8x8). Це зв'язано з тим, що відбуваються великі втрати в низьких частотах при квантуванні, і відновити вихідні дані стає неможливо.
2. Виявляється ефект Гіббса – ореоли по границях різких переходів кольорів.

Даний алгоритм дозволяє отримувати коефіцієнт компресії від 2 до 200 разів і придатний однаково для ущільнення як повнокольорових так і зображень в градаціях сірого без різких переходів кольорів.

Функціональні можливості програми

Програма "JPHS For Windows " є бета версією і не потребує інсталяції. Вона дозволяє приховувати конфіденційну інформацію в Jpeg – файлах. Для цього вам потрібно мати файл з даними, які ви бажаєте приховати , а також Jpeg - файл контейнер, в якому можна здійснити приховування.

Опис програми

Програма “JPHS For Windows ” використовує алгоритм “BlowFish” як базу для генератора псевдо випадкових чисел . Ключ береться з введеної вами парольної фрази. Даний алгоритм буде генерувати однакові послідовності випадкових чисел, лише коли ключі будуть однаковими , тобто , коли парольні фрази будуть однакові. Випадкові числа потрібні для того, щоб визначити , де розташована прихована інформація, серед інформацією картинки-контейнера. В результаті певний шум додається до візуальної інформації. А потім визначити , чи це є справжній шум, чи прихована інформація дуже важко. За відсутності оригінального Jpeg – файла та при малому відсотку приховування даних – це неможливо.

Порядок роботи з програмою

Проведемо ближче знайомство з роботою програми. Дана програма є бета версією і не потребує інсталяції. Вона написана на англійській мові, і тому всі позначення в ній будуть англійською. Для запуску програми потрібно запустити файл “Jphswin.exe”, і, після ліцензійного попередження, відкриється головне вікно програми. Робоча область програми розбита на 4 умовні частини : інформацію про файл контейнер, інформацію про файл для приховування, інформацію про вже заповнений контейнер, і інформацію про поточний стан програми – це найнижча область. Виконувати різні операції дозволяється через меню програми

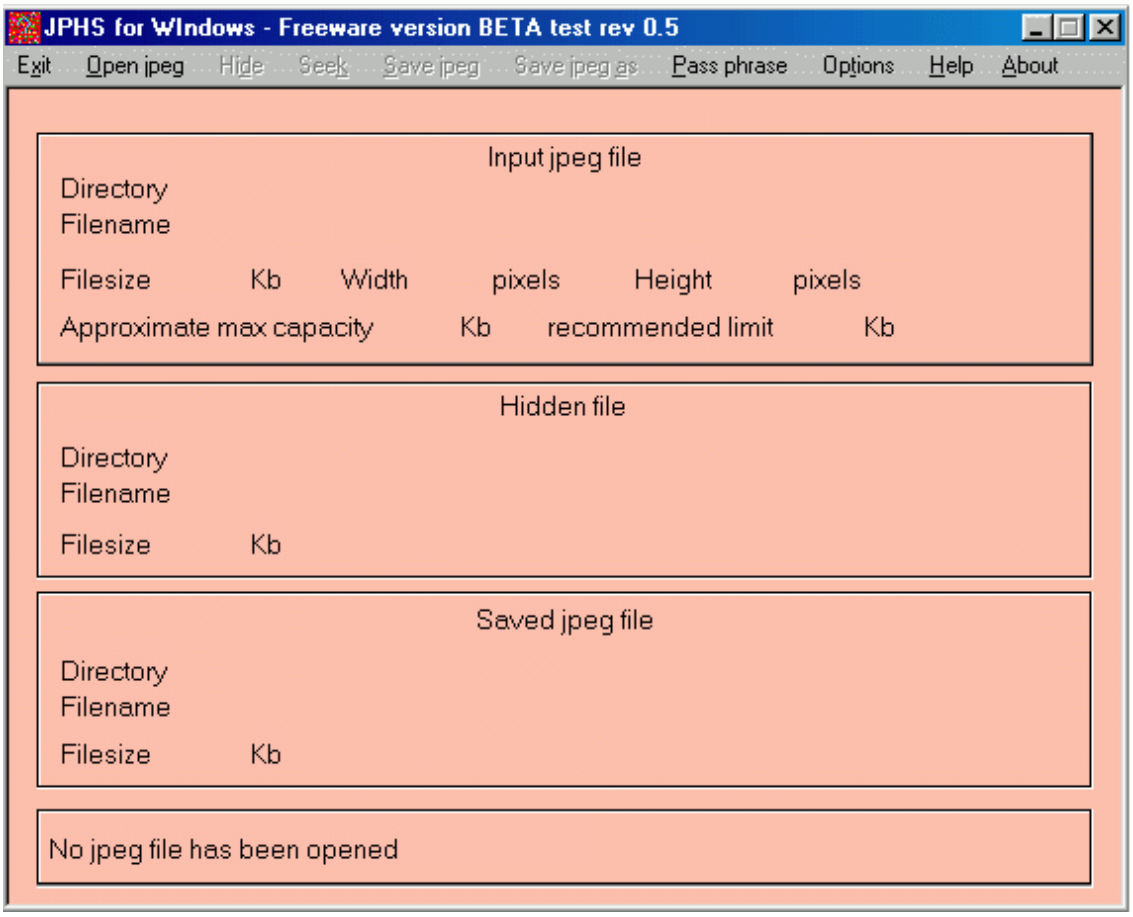
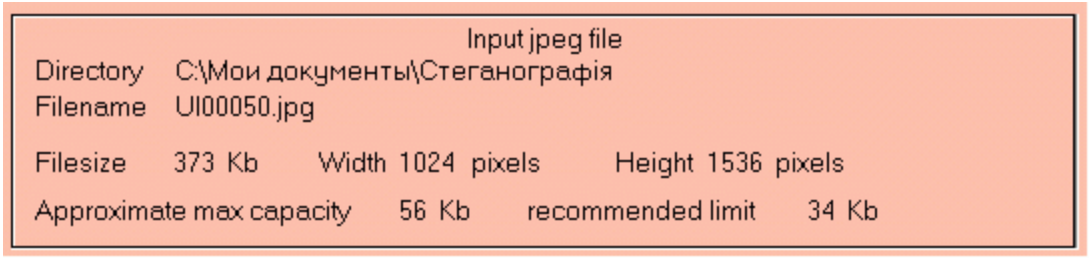


Рисунок 3.1 – Головне вікно програми

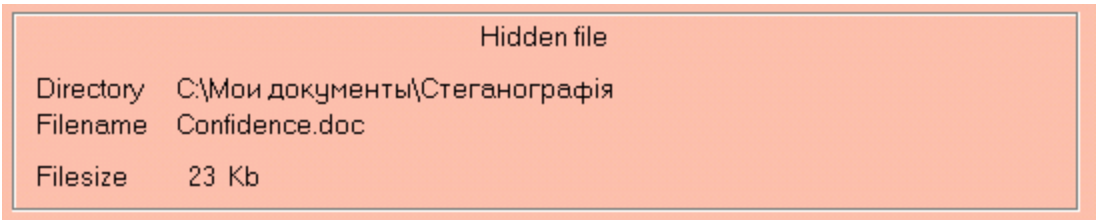
Перелічимо основні можливості програми:

1. Приховування даних

В стрічці меню натисніть кнопку “Open jpeg”. Вам запропонується знайти файл-контейнер, тобто графічний файл, який стиснутий за технологією jpeg чи jpg і має відповідне розширення. Після вибору в файлу у верхній частині програми з'явиться необхідна інформація про цей файл: назва файлу , місцезнаходження, розмір, ширина , висота , а також максимальна і рекомендована ємності для приховування.



Далі натисніть кнопку “Hide”. Програма запитає у вас парольну фразу, яку ви повинні ввести і підтвердити, після чого зможете вказати файл з інформацією для приховування. В головному вікні програми одразу добавиться інформація про приховуваний файл



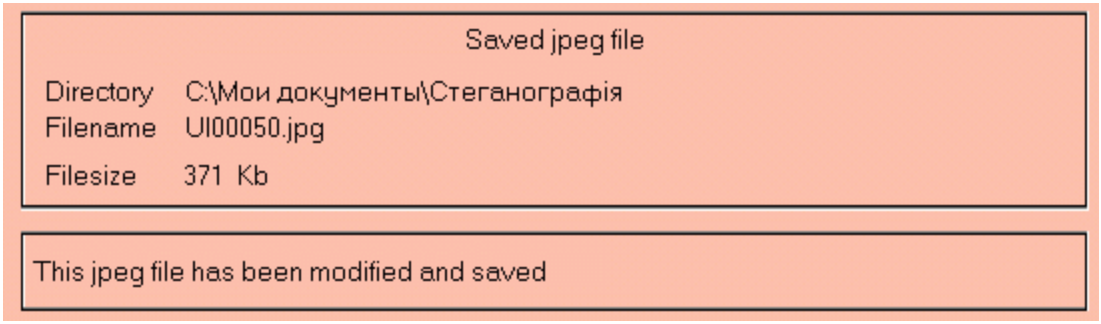
2. Зберігання заповненого контейнеру

Програма пропонує два методи зберігання заповненого контейнера:

Перший – це перезапис оригінального пустого контейнера – для цього потрібно натиснути кнопку “Save jpeg”.

Другий – зберігання заповненого контейнера як нового файлу з іншим ім'ям “Save jpeg as”

Після завершення операції зберігання, в головне вікно буде добавлена інформація про збережений файл-контейнер.



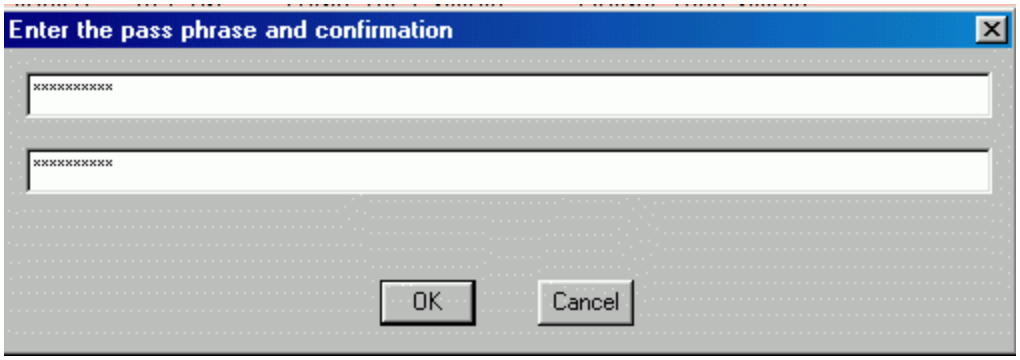
,

3. Пошук прихованої інформації

Для пошуку прихованої інформації знов натисніть пункт меню “Open jpeg”. Виберіть вже заповнений контейнер. А потім натисніть пункт меню “Seek”. Програма знов запитає вашу парольну фразу , після чого вам запропонується створити файл для зберігання витягнутої інформації. Якщо файл-контейнер є пустим, або парольна фраза не співпадає, то створений файл буде пустим і видасться попередження про невідповідність парольної фрази.

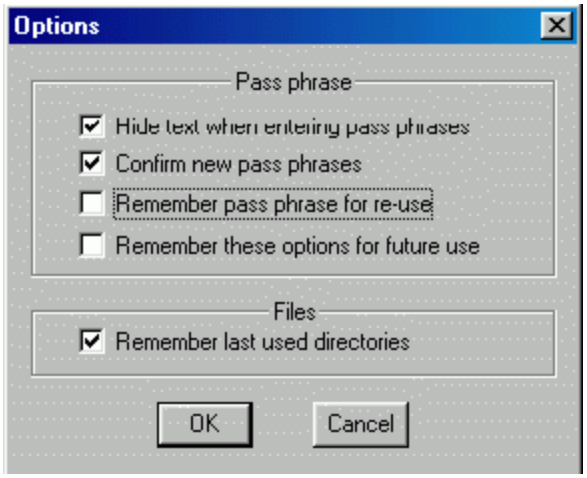
4. Парольна фраза

Пункт меню “Pass phrase” дозволить змінити поточну парольну фразу



5. Опції

Даний пункт меню дозволяє уточнювати деякі настройки програми під конкретного користувача. Вікно опцій має вигляд:



Перше поле дає змогу визначити спосіб введення парольної фрази – у закритому чи відкритому вигляді (без зірочок).

Друге поле запитує необхідність підтвердження паролю , тобто його повторне введення для перевірки.

Третє поле дає змогу запам'ятати один раз введену парольну фразу як поточну, без необхідності її постійного введення при роботі з черговим зображенням.

Четверте поле запитує чи запам'ятати змінені вами настройки для майбутнього використання.

П'яте поле запитує чи потрібно запам'ятовувати останню використовувану директорію знаходження файлів

Порядок виконання роботи

1. Вивчити надані теоретичні відомості.
2. Ознайомитися з функціональними можливостями програми "JPHS For Windows ".
3. Візьміть зображення-контейнер і виконайте наступні дії:
   - в один контейнер вбудуйте інформацію рекомендованої величини (recommended limit),
   - в другий – максимально безпечної (approximate max capacity),
   - в третій – значно більшої за максимальну, але менше половини розміру контейнеру.
   - спробуйте прочитати приховану інформацію з заповнених контейнерів.

- порівняйте три заповнені контейнери з незаповненим і зробіть відповідні висновки.

4. Візьміть зображення-контейнери і вбудуйте інформацію рекомендованої довжини. Далі виконайте наступні дії:
   - Спробуйте видобути дані по одній парольній фразі з з заповненого і пустого контейнерів.
   - Спробуйте видобути дані по різним парольним фразам з заповненого контейнеру.
   - Прокоментуйте результати.
5. Візьміть зображення-контейнери і виконайте натупні операції:
   - в один контейнер вбудуйте дві різні інформації
   - спробуйте прочитати вбудовану інформацію
   - результати прокоментуйте.

Зміст звіту

Звіт повинен містити:
- короткі відомості про використані методи приховування інформації;
- короткі відомості про програму “JPHS For Windows ”;
- висновки про результати експериментів.

Контрольні питання

1. Які бувають зображення ?
2. Назвіть причини необхідності ущільнення зображень.
3. Порівняйте алгоритми ущільнення з втратами і без втрат.
4. Назвіть причини використання зображень як стегоконтейнерів.
5. Опишіть стандарт ущільнення “JPEG”.
6. Як працює алгоритм “JPEG” ?
7. Які він має недоліки і переваги.
8. Який рівень ущільнення надає стандарт “JPEG”.

Лабораторна робота №4

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯ ТА АУДІО ФАЙЛИ.

Мета роботи: Вивчення методів приховування інформації у зображеннях та аудіо файлах, та набуття навичок з приховування інформації за допомогою програми GiviTools.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з бурхливим розвитком технологій мультимедіа гостро стало питання захисту авторських прав і інтелектуальної власності, представленої в цифровому виді. Переваги представлення і передачі інформації у цифровому вигляді можуть бути перекреслені легкістю, з якою можливе їхнє викрадення і модифікація. Тому в всьому світі розробляються методи захисту інформації організаційного, методологічного і технічного характеру, серед них - методи стеганографії.

Вбудовування зображень в аудіо файли.

За своїм характером звуки э неточними при оцінці правильного значення звукової хвилі в специфічний момент часу. Звуки в WAV файлах зберігаються, у форматах по 8 або 16 бітів, і саме в такому виді вони потрапляють до DA конвертера на вашій звуковій карті. Оскільки число кодують 8 бітів, це означає, що значення можуть варіюватися між 0 і 255. При форматі в 16 бітів відповідно діапазон лежить між 0 і 65535.

Вбудовування зображень в графічні файли.

Якщо контейнером є зображення, то інформацію, що вбудовується, простіше представити у виді двовимірного масиву біт. З метою підвищення секретності вбудовування повідомлення в контейнер може здійснюватися за допомогою ключа. Для підвищення стійкості до перекручувань використовують широполосні сигнали або завадостійке кодування. Значно підвищується стійкість до перекручувань при вбудовуванні інформації в спектральну області сигналу. Саме тому важливим етапом попередньої обробки сигналу є обчислення його узагальненого перетворення Фур'є.

Загальна схема вбудовування інформації в зображення представлена на мал. 4.1.

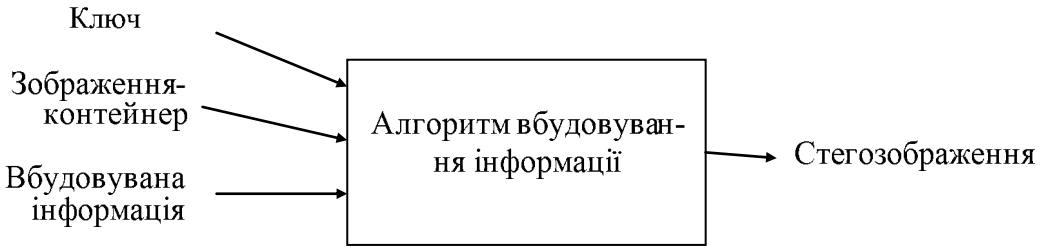


Рисунок 4.1. – Загальна схема вбудовування інформації в зображення

Загальна схема витягу інформації з зображення представлена на мал. 2.

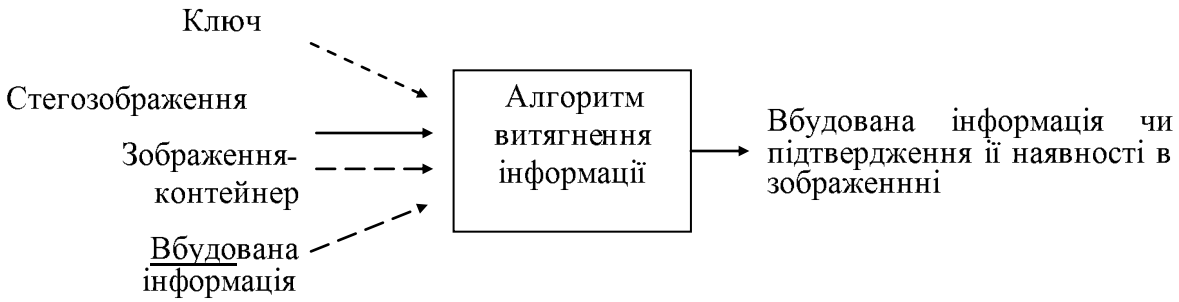


Рисунок 4.2. – Загальна схема витягу інформації

Стрілка, показана курсивом означає, що компонент може бути відсутнім. Це залежить від використовуваного алгоритму вбудовування інформації і поставленої прикладної задачі.

Зображення володіють великою психовізуальною надмірністю. Око людини подібне низькочастотному фільтрові, якому непомітні перекручування у високочастотній області зображень. Стеганографія заснована на використанні наявної в зображеннях психовізуальної надмірності .

Класифікація властивостей системи людського зору приведена в табл.4.1.

Таблиця 4.1. Властивості системи людського зору.

Низькорівневі	**Високо рівневі**
1. Чутливість до зміни яскравості зображення 2. Частотна чутливість 3. Ефект маскування	1. Чутливість до: - контрасту - розміру - місця розташування - кольору - форми - зовнішніх подразників 2. Підвищена увага до зображень переднього плану

Причини поширеності використання для контейнера нерухомого зображення:

- Розмір контейнера заздалегідь відомий
- Відсутні обмеження режиму передачі в реальному часі
- Можливість впровадження інформації великого обсягу
- Слабка чутливість ока людини до деяких змін характеристик зображення (див. табл. 1)

Узагальнена схема впровадження даних у зображення ґрунтується на наступних процедурах:

1. Фільтрації зображення на основі орієнтованих смугових фільтрів.
2. Обчисленні порога маскування.
3. Приведенні енергії впроваджуваного сигналу до значення, меншому, чим поріг маскування для кожного компонента.

До дійсного часу реалізована велика кількість методів впровадження інформації в зображення, що можуть бути розділені на невелику кількість груп, подібних по використовуваній ідеї.

Приховування даних у нерухомих зображеннях здійснюється на підставі наступних двох груп методів:

1. Прямі методи модифікації зображення в просторовій області.
2. Методи, що модифікують зображення, попередньо перетворене в іншу форму.

Прямі методи вбудовують інформацію безпосередньо в підмножину пікселів зображення. Вона впроваджується за рахунок маніпуляцій яскравістю і колірними складовими без обрахунково громіздких лінійних перетворень зображення.

Методи даного класу розрізняються тільки вибором підмножини пікселів, що модифікуються, і стратегією зміни значень пікселів.

У методах, що використовують попереднє перетворення, інформація впроваджується за рахунок декомпозиції зображення-контейнера. Ці методи використовують переваги, яким володіє представлення зображення кінцевим набором коефіцієнтів. Як правило такі методи мають гарні характеристики робосності.

Опис програми.

Розглянемо використання цифрової стеганографії на прикладі програми GiviTools. Програма GiviTools може використовувати для контейнерів наступні формати (табл. 4.2).

Таблиця 4.2. Формати файлів.

Графічні	**Аудіо**
Bitmap (bmp)	Wave audio (wav)

Головне вікно програми представлено на рис. 4.3.

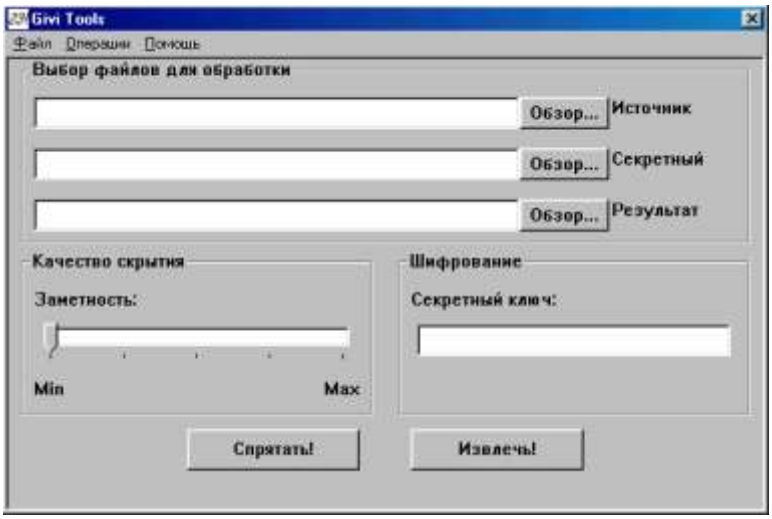


Рисунок 4.3. Головне вікно програми.

Керування програмою здійснюється наступним чином:

В залежності від операції, яку ми бажаємо виконати, потрібно, щоб були заповнені всі три поля (при приховуванні інформації), або тільки два верхніх (при витягуванні).

При прихованні інформації поле "Источник" повинне містити ім'я існуючого файлу зі звуком (розширення wav), або файлу-зображення (розширення bmp). Цей файл не змінюється.

Поле "Секретный" повинне містити файл, що ми хочемо сховати. Це може бути файл будь-якого типу (що завгодно). Для коректного відновлення інформації розмір даного файлу повинний бути менше 16 Мб. На перший погляд може здатися, що цього мало. Врахуємо, однак, що розмір файлу-джерела повинен бути в декілька разів більше, ніж секретного, а bmp розміром у 80 Мб явно викличе підозру і незручність. Цей файл також не змінюється.

У полі "Результат" знаходиться ім'я файлу, що створюється програмою. Це буде файл того ж типу, що і "Источник". Тобто , якщо " Источник " - wav, то "Результат" також буде wav'ом. Аналогічно і з bmp.

Увага! Якщо файл із таким ім'ям вже існує, він буде перезаписаний без підтвердження. Будьте уважні.

При витязі інформації поле " Источник" повинне містити ім'я існуючого файлу зі звуком (розширення wav), або файлу-зображення (розширення bmp), у якому попередньо була схована інформація. Цей файл не змінюється.

У полі "Секретный" знаходиться ім'я файлу, що відтворюється з " Источник". Це буде файл того ж типу, що і файл, що був захований раніше.

Увага! Програма не зберігає тип приховуваного файлу. Це означає, що розширення його Ви повинні вибрати самі. Наприклад, якщо Ви сховали rar-архів, то в поле "Секретный" при відновленні Ви повинні самостійно вказати розширення rar.

Увага! Якщо файл із таким ім'ям вже існує, він буде перезаписаний без підтвердження. Будьте уважні.

Поле "Результат" при даній операції ігнорується.

Ступінь помітності вибирається в залежності від співвідношення розміру файлу-джерела і секретного файлу. Сама дія виконується за допомогою трэкбара, розташованого в лівому нижньому куті вікна. Чим менше помітність, тим ближче файл-результат по своїй якості до файлу-джерела.

На даний момент шифрування в програмі не працює, тому поле "Секретний ключ" ігнорується.

Коли всі необхідні поля заповнені, можна приступати до операцій приховування/витягу інформації. Отут усе просто: для приховування натискаємо "Спрятать", для відновлення натискаємо "Извлечь". Після виконання відповідної операції, програма видає повідомлення про успішне її закінчення. Можна приступати до наступних приховувань/витягів.

Усі зазначені дії можна робити за допомогою відповідних пунктів меню, але користуватися основним вікном набагато зручніше.

Порядок виконання роботи

1. Вивчити короткі теоретичні.
2. Ознайомитися з функціональними можливостями програми GiviTools.
3. Вибрати файл-контейнер і файл для приховування згідно до свого варіанта (розмір контейнера має бути достатній для вміщення приховуваних даних).
4. Змінюючи якість приховування, виконати вбудовування даних.
5. Порівняти контейнери до і після вбудовування. Зробити висновки.
6. Отримати сховані в контейнері дані. Перевірити приховувані дані на втрату інформації. Зробити висновки.
7. Оформити звіт про виконання лабораторної роботи:
   - короткі відомості про методи приховування інформації у зображеннях та аудіо файлах;
   - короткі відомості про програму GiviTools;
   - інформацію про обсяги контейнеру, файлу для приховування;
   - висновки, зроблені під час виконання лабораторної роботи.

Варіант	Контейнер	Файл для схову
1	BMP	*.doc
2		*.txt
3		*.avi
4		*.jpeg
5		*.bmp
6		*.wav
7		*.gif
8		*.exe
9		*.com
10		*.rar
1	WAV	*.doc
2		*.txt
3		*.avi
4		*.jpeg
5		*.bmp
6		*.wav
7		*.gif
8		*.exe
9		*.com
10		*.rar

** - файли з даними розширеннями ви можете знайти на вашому комп'ютері або створити власноруч, записавши в них будь-яку свою інформацію.*

Зміст звіту

- короткі відомості про використані методи приховування інформації;
- короткі відомості про програму “ GiviTools ”;
- висновки про результати експериментів.

Контрольні питання

1. Чому найбільшу популярність як контейнер здобуло нерухоме зображення.
2. Яка існує класифікація властивостей системи людського зору.
3. На підставі яких груп методів здійснюється приховування даних у нерухомі зображення.
4. Які методи мають гарні характеристики робосності.
5. У яких форматах зберігаються звуки у *.wav файлах.

Лабораторна робота №5

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У BMP ЗОБРАЖЕННЯ.

Мета роботи: Ознайомитися з роботою програми по стеганографії bmpPacker версії V1.2a (Build2). Отримати навички роботи з програмою. Ознайомитися з режимами роботи програми через пункт меню “Опції”.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Опис програми

Дана програма по стеганографії має назву bmpPacker версія V1.2a (Build2) і для користування не потребу ліцензії.

Програма за допомогою алгоритмів шифрування скриває текстові, графічні файли в графічному файлі типу .bmp. Головне вікно програми зображено на рис. 5.1

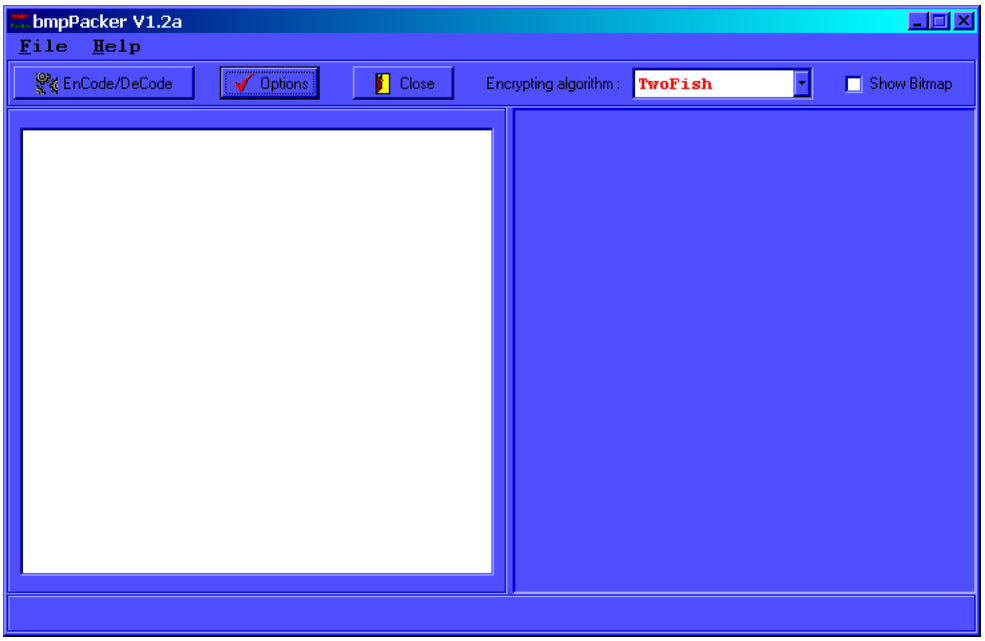
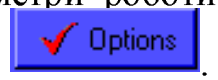


Рисунок 5.1 – Головне вікно програми.

Програма містить вкладку за допомогою якої можна обрати алгоритм для шифрування файлу (рис. 5.2).

Також параметри роботи з програмою можна задати через кнопку “Options” - > 

. Вікно роботи з параметрами програми зображено на рис. 3 і містить:

- вибір методу шифрування;
- вибір використання стиснення (архівації) файлів;
- рівні стиснення файлів;
- вибір режиму роботи програми (вийти з програми одразу після шифрування і показати результат роботи програми у спеціальному вікні);
- режим роботи з паролями.

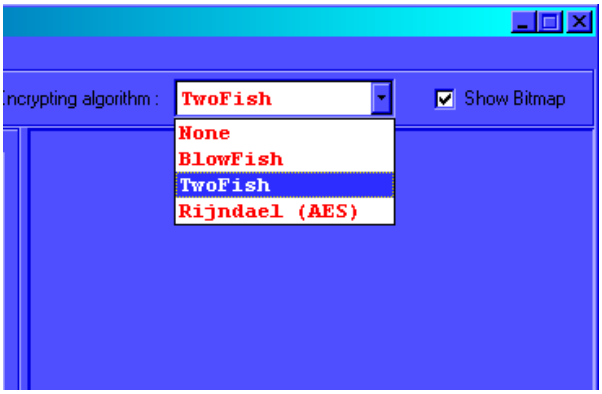


Рис. 5.2 – Вибір алгоритмів шифрування

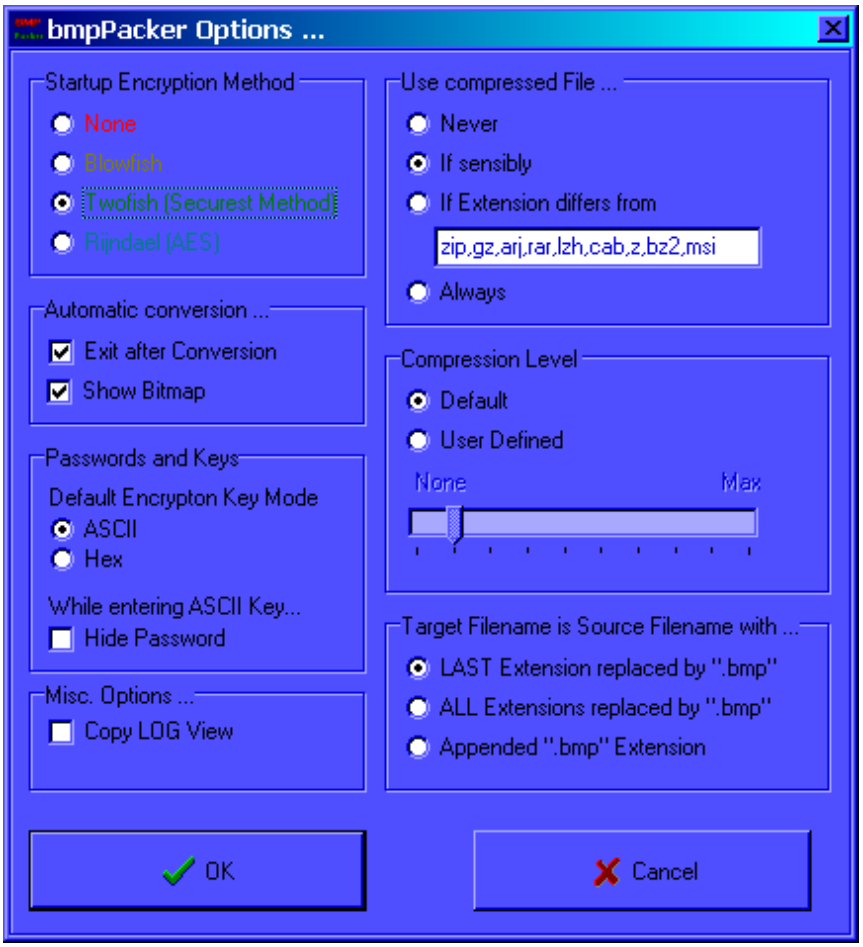


Рисунок 5.3 – Вікно “Options”

Режими роботи з програмою пункт “Опції”

Деякі зміни до опцій діють одразу - інші набирають сили тільки після наступного запуску програми. Зміни зберігаються перед виходом із програми.

1 Метод шифрування

Це - метод шифрування, що використовується за замовчуванням. Всі обрані алгоритми кодування в цій програмі набагато більше безпечні, ніж два найкраще відомих алгоритми кодування DES і потрійний DES.

– Без шифрування
– BlowFish

Висока гарантія шифрування блоку розміром 64 біти, з ключем змінної довжини. Blowfish був розроблений у 1993 Брюсом Шнеіером. Основний вихідний текст був написаний у С Брюсом Шнеіером

– TwoFish

Також розроблений Брюсом Шнеіером. Не так широко використовується, але більш стійкий за BlowFish. Розмір його блоку

шифрування залежить від довжини пароля від 128 до 256 біта. На 2004 рік ще ніхто не міг зламати TwoFish.

– Rijndael

Rijndael – AES новинка серез алгоритмів шифрування в 2002 року. Rijndael був розроблений Вінсентом Ріймeном і Джоан Даемен. Розмір його блоку шифрування залежить від довжини пароля від 128 до 256 біта. На 2004 рік ще ніхто не міг зламати Rijndael.

2 Автоматичне Перетворення

• Вихід після Перетворення

Після роботи з перетвореннями над файлом, програма автоматично виключається.

• Показувати крапковий малюнок

Дана опція також присутня й у головному вікні програми. Полягає у тому, що в правій частині вікна програми показує малюнок, який отримали шляхом шифрування.

3 Паролі і ключі

• *Default Encryption Key Method*

Заданий за замовчуванням метод ключа шифрування

• *Hide Password*

Прихований пароль.

Дана опція ховає символи пароля, що набираються вами. Ця опція діє тільки для паролів в режимі ASCII.

4 Використання стиснення файлів

bmpPacker здатний стискати файли перед кодуванням. Ця група опцій

Визначає, яким методом повинний бути оброблений файл.

- *Never* – Функція стиснення заблокована.
- *If sensibly (Default)* – Значення за замовчуванням. BmpPacker стискає вихідний файл автоматичним шляхом і вирішує після цього який з двох файлів повинен використовуватися.
  - якщо стиснутий файл менший чим вихідний файл, стиснутий файл кодується.
  - якщо стиснутий файл дорівнює або більший чим вихідний файл, тоді кодується вихідний файл.
- *If Extension differs from...* – Якщо розширення відрізняється від ... Файл стиснутий, якщо розширення вихідного файлу не відповідає одному з даного розширень, що приведено нижче. Регістр при цьому не має ніякого значення. "ZIP" або "Zip", або "zip" приймається як одне і теж. Задані за замовчуванням розширення:
  - Zip – стиснутий файл, керований архівами PKZIP або WINZIP
  - Gz – стиснутий файл, керований GNU ZIP (gzip)
  - Arj – стиснутий файл, керований ARJ
  - Rar – стиснутий файл, керований RAR
  - Lzh – стиснутий файл, керований Lharc
  - Cab – стиснутий файл, керований Windows

- Z – стиснутий файл, керований UNIX/LINUX/BSD compress
- Bz2 – стиснутий файл, використовуваний великою кількістю UNIX/LINUX/BSD систем.
- Msi – даний формат стиску має Microsofts.

- *Always* – Стискає і шифрує файл у будь-якому випадку.

5 Рівень Стиснення

Якщо файл стиснутий, ця опція визначає параметри настроювання стиску.

• *Значення за замовчуванням*

Використовується заданий за замовчуванням рівень стисків.

• *Обумовлений користувачем*

Використовуйте один з 10 рівнів стиснення . Якщо крайній лівий, то ніякого стиснення не відбувається. При крайньому правому – максимальне стиснення.

6 Misc- опції

• *Copy LOG View* – Представлення ФАЙЛУ РЕЄСТРАЦІЇ

7 Цільове Ім'я файлу - Вихідне Ім'я файлу з...

• *Last Extension replaced by „.bmp“* – Останнє Розширення, замінене на „.bmp”

Цільове ім'я файлу – той же саме як вихідне ім'я файлу, за винятком розширення зміненого на „.bmp”.

Приклад:

Вихідне ім'я файлу = xyz.txt.tar.*gz* = > ім'я файлу після шифрування = xyz.txt.tar.*bmp*

• *All Extensions replaced by „.bmp“* – Усі розширення, замінені на „.bmp”

Цільове ім'я файлу - те ж самий як вихідне ім'я файлу, за винятком того, що всі розширення замінені на „.bmp”.

Приклад:

Вихідне ім'я файлу = xyz.*txt.tar.gz* = > ім'я файлу після шифрации = xyz.*bmp*

• *Appended „.bmp“ Extension*

Доданий у кінець „.bmp” розширення.

Цільове ім'я файлу - те ж самий як вихідне ім'я файлу, за винятком того, що „.bmp” доданий у кінець.

Приклад:

Вихідне ім'я файлу = xyz.txt.tar.gz = > ім'я файлу Target = xyz.txt.tar.gz.*bmp*

Опції Командного рядка

Синтаксис командного рядка:

BmpPacker [-K:Key | -k:Key | -B | -b | -T | -t | -R | -r | -N | -n | -S | -s] [ім'я файлу]

Значення:

- *-K: oder -k:* Клавіша, що повинна використовуватися для стиску командного рядка.
- *-B або –b* Використовують Blowfish для кодування
- *-T або -t* Використовують Twofish для кодування
- *-R або -r* Використовують Rijndael для кодування
- *-N або -n* Не використовують ніяке кодування
- *-S або -s* Показати результат роботи програми.
- *Ім'я файлу*, який повинний зашифровуватися або розшифровуватися

Ці опції командного рядка скасовують опції, що встановлювалися в пункту меню “Опції”, але вони не змінюють (заміняють) їх.

Наприклад:

Якщо вибрали Rijndael як ваш улюблений алгоритм шифрування, і при цьому викоритовуєте: BmpPacker -T myfile.doc, то файл кодується, використ-товуючи Twofish алгоритм. Після того, як запустити bmpPacker знову (без опцій командного рядка) використовується обраний алгоритм Rijndael.

Робота з програмою

У головному вікні програми натиснемо кнопку 

- з'явиться вікно для вибору документа для шифрування і дешифрування даних. Після того як для шифрування обрали текстовий документ, з'явиться вікно для задання пароля (ключа шифрування) (Рис.4). Довжина пароля при цьому має бути не менше, ніж 8 символів.

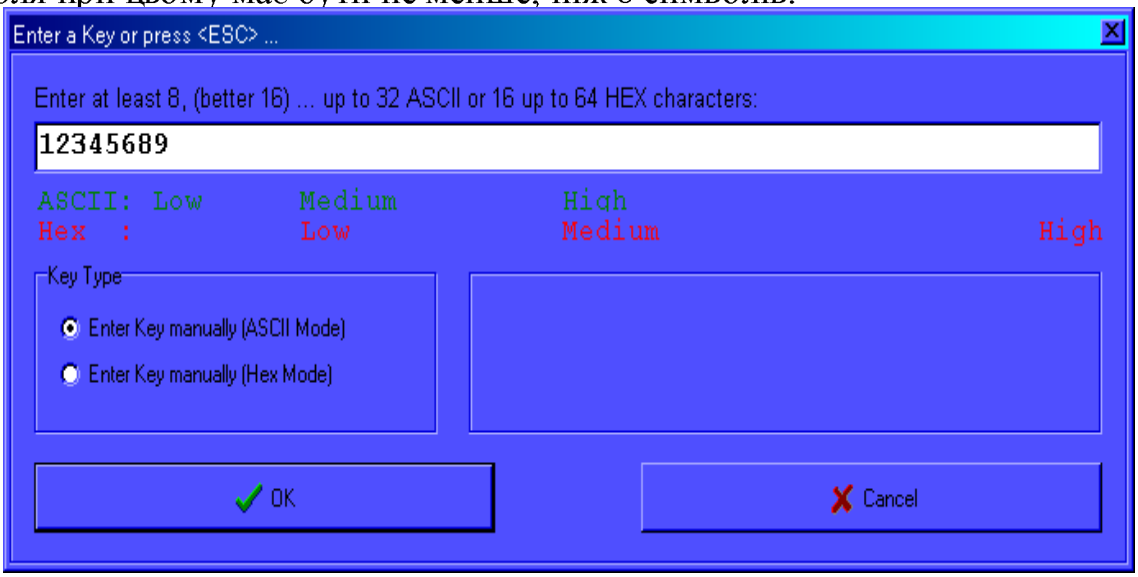


Рисунок 5.4 – вікно шифрування тексту

Якщо маємо справу з скритим паролем, то вікно виглядатиме таким чином:

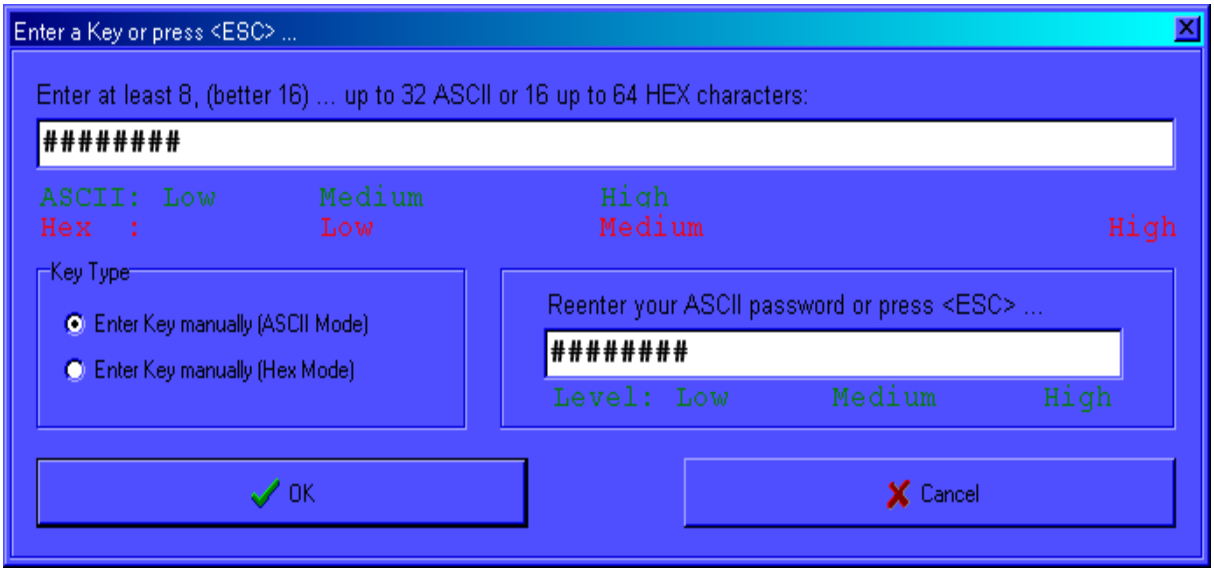


Рисунок 5.5 – введення скритого пароля

Після того як ввели пароль в головному вікні програми буде показано процедуру укриття текстового файлу в малюнок – рис. 5.6

Зашифрований файл буде знаходитись в тій же папці, де оригінал.

Оскільки приховувати можна і графічні файли, процедура шифрування проводиться аналогічно – рис.5.7

Для розшифрування програма вимагає введення пароля. Оскільки графічний файл більший за об'ємом, ніж текстовий, то в результаті зашифрований графічний файл теж матиме більший об'єм, ніж зашифрований текстовий. Також об'єм зашифрованого файлу залежить від ступеня стиснення, що обирає користувач.

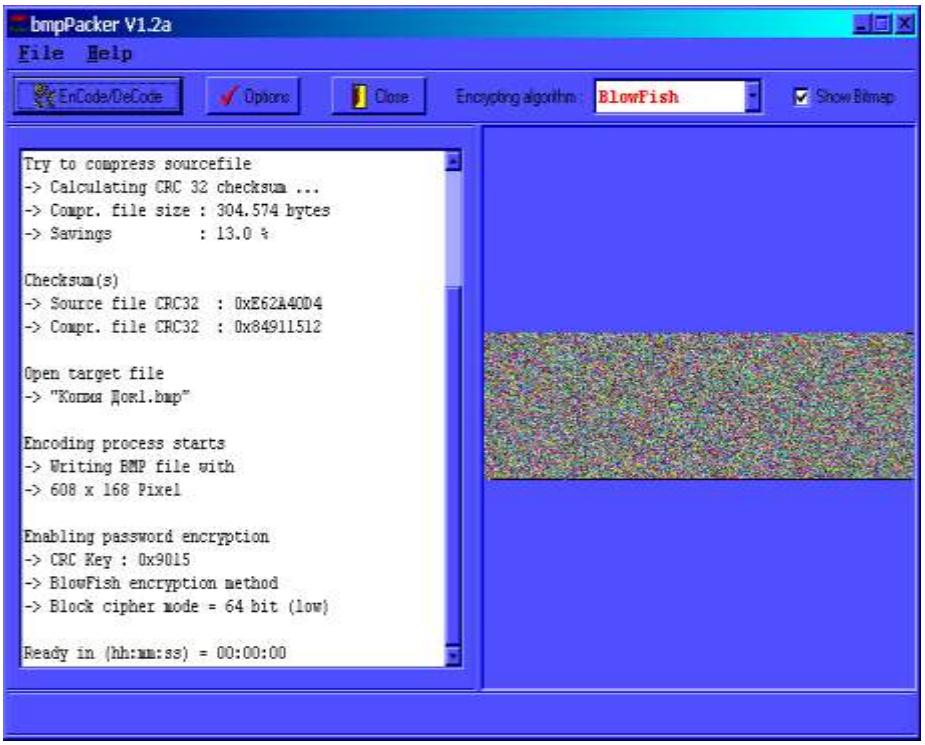


Рисунок 5.6 – шифрування текстового файлу.

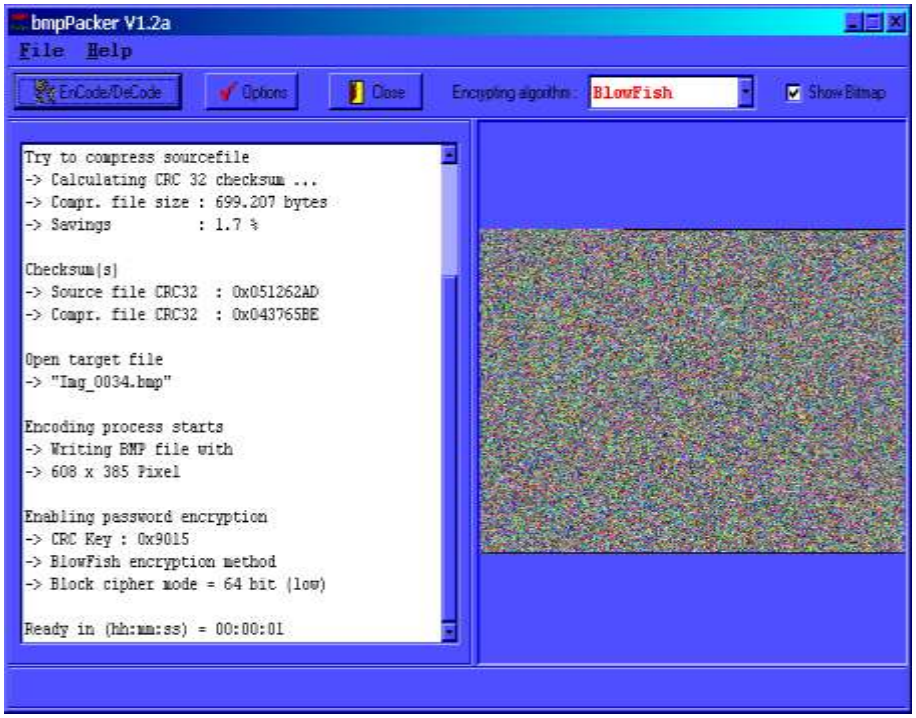


Рисунок 5.7 – шифрування малюнка.

Порядок виконання роботи

1. За допомогою програми спробуйте сховати файл (відповідно до варіанта) у зображенні.
2. Виконайте декілька процедур шифрування, при цьому використовуючи різні варіанти роботи з програмою через Options. Порівняйте створені зашифровані зображення за об'ємом і зробіть висновки.
3. Проведіть процедуру розшифрування використовуючи для початку невірний пароль, а потім вірний.

Варіант	Завдання
1	За допомогою прихованого введення пароля приховати текстовий файл. Використати: в 1-му випадку відсутність стиснення, в 2-му максимальне стиснення.
2	Приховати графічний файл. Використати: в 1-му випадку метод шифрування BlowFish без стиснення, в 2-му аналогічний метод шифрування з будь-яким ступенем стиснення. В обох випадках приховане введення пароля.
3	Приховати текстовий файл. Використати: в 1-му випадку метод шифрування TwoFish без стиснення, в 2-му аналогічний метод шифрування з будь-яким ступенем стиснення. В одному з випадків застосувати приховане введення пароля.
4	Приховати графічний файл. Використати: в 1-му випадку алгоритм шифрування BlowFish, в 2-му без шифрування. При цьому без стиснення.
5	Приховати текстовий файл. Використати: в 1-му випадку алгоритм шифрування Rijndael, в 2-му без шифрування. При цьому без стиснення.
6	Приховати текстовий файл. Використати: в 1-му випадку алгоритм шифрування Rijndael, в 2-му без шифрування. При цьому використати однаковий ступінь стиснення.
7	За допомогою прихованого введення пароля приховати графічний файл. Використати: в 1-му випадку малий ступінь стиснення, в 2-му максимальне стиснення.
8	Приховати текстовий файл, використовуючи 2 різні алгоритми шифрування і однаковий ступінь стиснення.
9	Приховати графічний файл, використовуючи 2 різні алгоритми шифрування і однаковий ступінь стиснення.
10	Приховати графічний файл, використовуючи будь-який метод архівації з різним ступенем стиснення.
11	Приховати графічний файл, використовуючи різні методи архівації і однаковий ступінь стиснення.
12	Приховати графічний файл, використовуючи лише різні методи зміни розширення в результаті роботи з програмою. При цьому обрати однаковий метод шифрування без ступеня стиснення.

Зміст звіту

- короткі відомості про використані методи приховування інформації;
- короткі відомості про програму “ bmpPacker ”;
- висновки про результати експериментів.

Контрольні питання

1. Які режими роботи з паролем використовуються а програмі?
2. З якими видами файлів можлива робота програми?
3. Які методи архівації використовується в програмі? Як впливає ступінь стиснення на результат переховування?
4. Яка різниця в результатах переховування графічних і текстових видів файлів?
5. Які методи зміни розширення результуючого файлу використовуються в опціях програми?

Лабораторна робота №6

ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ ПРОГРАМИ IMAGESPYER G2.

Мета роботи: Вивчення методів приховування інформації у зображеннях, та набуття навичок з приховування інформації за допомогою програми ImageSpyer G2.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

LSB (Least Significant Bit, найменший значущий біт) – суть цього методу полягає в заміні останніх значущих бітів у контейнері (зображення , аудіо або відеозапису) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.

Принцип цього методу полягає в наступному: Припустимо, є 8-бітне зображення в градаціях сірого. 00h (00000000b) позначає чорний колір, FFh (11111111b) – білий. Усього є 256 градацій (2 ^ 8). Також припустимо, що повідомлення складається з 1 байта – наприклад, 01101011b. При використанні 2 молодших біт в описах пікселів, нам буде потрібно 4 пікселя. Припустимо, вони чорного кольору. Тоді пікселі, що містять приховане повідомлення, будуть виглядати наступним чином: 00000001 00000010 00000010 00000011. Тоді колір пікселів зміниться: першого — на 1/255, другого і третього – на 2/255 і четвертого – на 3/255. Такі градації, мало того що непомітні для людини, можуть взагалі не відобразитися при використанні низькоякісних пристроїв виведення. В ролі базового контейнера пропонується використовувати файли BMP-зображень високої роздільності з глибиною кольору 24 та 32 біти, таємне зображення може мати розширення .BMP, .GIF, .PNG, .JPEG.

Недоліком методу LSB є чутливість до розміру зображення, тобто чим менший розмір зображення, тим більше будуть відрізнятися два сусідні пікселі, тому пропонується використовувати зображення з великою роздільністю. Також метод «видає себе» при побітовому перегляді зображення, де чітко видно області зображення в які «вбудовано» таємну інформацію. Попри це, метод запису Least Significant Bit є досить популярним, стійким та простим в реалізації.

У програмі використовується авторська реалізація цього алгоритму. Простіше кажучи, в результаті підсумкове зображення поміщає обсяг інформації, рівний числу піскелей вихідного зображення. За вирахуванням мізерного обсягу службової інформації. Наприклад, зображення розміром

450x340 пікселів дозволяє включити приблизно 149 КБ довільної інформації. Виходячи з розмірів файлів, це приблизно 1/3 від файлу-контейнера.

Крім того, що саме зображення важко запідозрити в наявності в ньому будь-якої сторонньої інформації, програма захищає впроваджений файл одним з 40 стійких криптоалгоритмів. У налаштуваннях можна вибрати алгоритм і режим шифрування. ImageSpyer може виробляти зашифрування будь-яким з 40 доступних алгоритмів шифрування: ГОСТ 28147-89, Cast128, Cast256, Blowfish, IDEA, Mars, Misty 1, RC2, RC4, RC5, RC6, FROG, Rijndael, SAFER, SAFER-K40, SAFER-SK40 , SAFER-K64, SAFER-SK64, SAFER-K128, SAFER-SK128, TEA, TEAN, Skipjack, SCOP, Q128, 3Way, Twofish, Shark, Square, Single DES, Double DES, Triple DES, Double DES16, Triple DES16, TripleDES24, DESX, NewDES, Diamond II, Diamond II Lite, SapphireII. Для ускладнення розпізнавання даних можливе завдання довільного набору порядку біт, таким чином, не знаючи даного набору не вийде коректно вилучити приховану інформацію.

Для стискання прихованих файлів використовується алгоритм стискання LZMA (*Lempel-Ziv-Markov chain-Algorithm*) - алгоритм стискання даних, що розробляється з 2001 року. Алгоритм заснований на схемі стискання даних за словником і забезпечує високий коефіцієнт стискання, а також дозволяє використовувати словники різного розміру (до 4 Гб).

Програма ImageSpyer G2

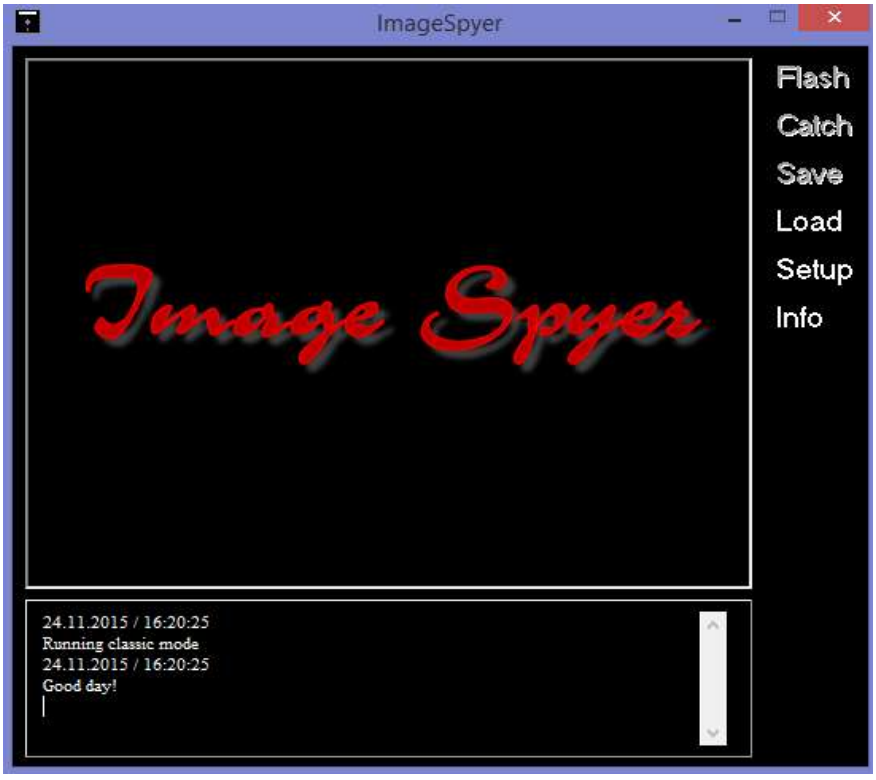


Рисунок 6.1 – Головне вікно програми.

Меню програми знаходиться з правої сторони вікна. Призначення пунктів меню: Flash – приховування файл у зображення, Catch – вилучити файл із зображення, Save – зберегти файл-контейнер, Load – завантажити файл-контейнер, Setup – налаштування програми, Info – інформація про програму.

Для приховування файлу у зображенні потрібно зробити наступні дії:

1. Вибрати файл-контейнер. Натиснувши Load у меню програми.

Рисунок 6.2 – Завантажений файл-контейнер. Max hidden data size – означає найбільший розмір файла, який можна приховати в цьому файлі.

2. Вибрати файл для приховування. Натиснувши Flash у меню програми.
3. Після вибору файлу з’явиться вікно у якому потрібно ввести пароль для файлу-контейнера та для прихованого файлу.

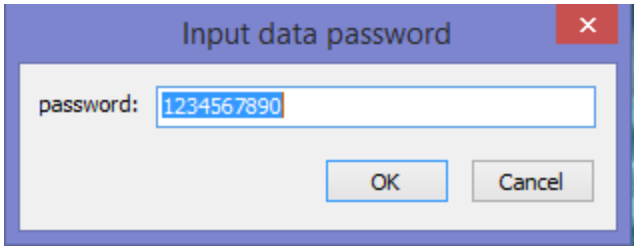


Рисунок 6.3 – Вікно введення паролю для шифрування прихованого файлу.

3. Після приховування файлу з’явиться вікно підтвердження приховування файлу і програма запропонує зберегти файл. Файл можна зберегти у форматах *.bmp, *.tif або *.tiff.

Для вилучення прихованого файлу потрібно зробити наступні дії:

1. Обрати файл-контейнер із повідомленням, натиснувши на пункті меню Load.
2. Натиснути на пункт меню Catch. З’явиться вікно для введення пароля для вилучення файлу.
3. Після введення паролю необхідно обрати місце для збереження вилученого файлу. За замовчуванням вилучений файл буде мати назву restored_<ім’я прихованого файлу>.
4. Після обрання місця для збереження з’явиться ще одне вікно для введення пароля у якому необхідно вказати пароль для розшифрування файлу. Якщо пароль було введено правильно з’явиться наступне вікно.

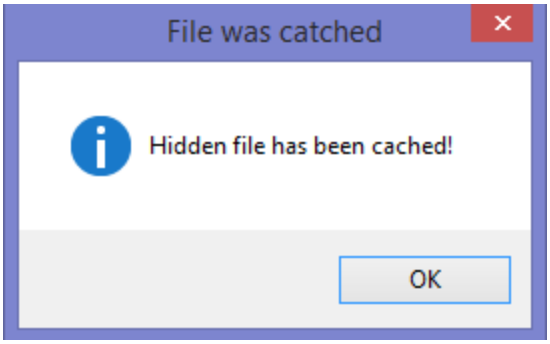


Рисунок 6.4 – Повідомлення про успішне вилучення файлу.

Налаштування програми

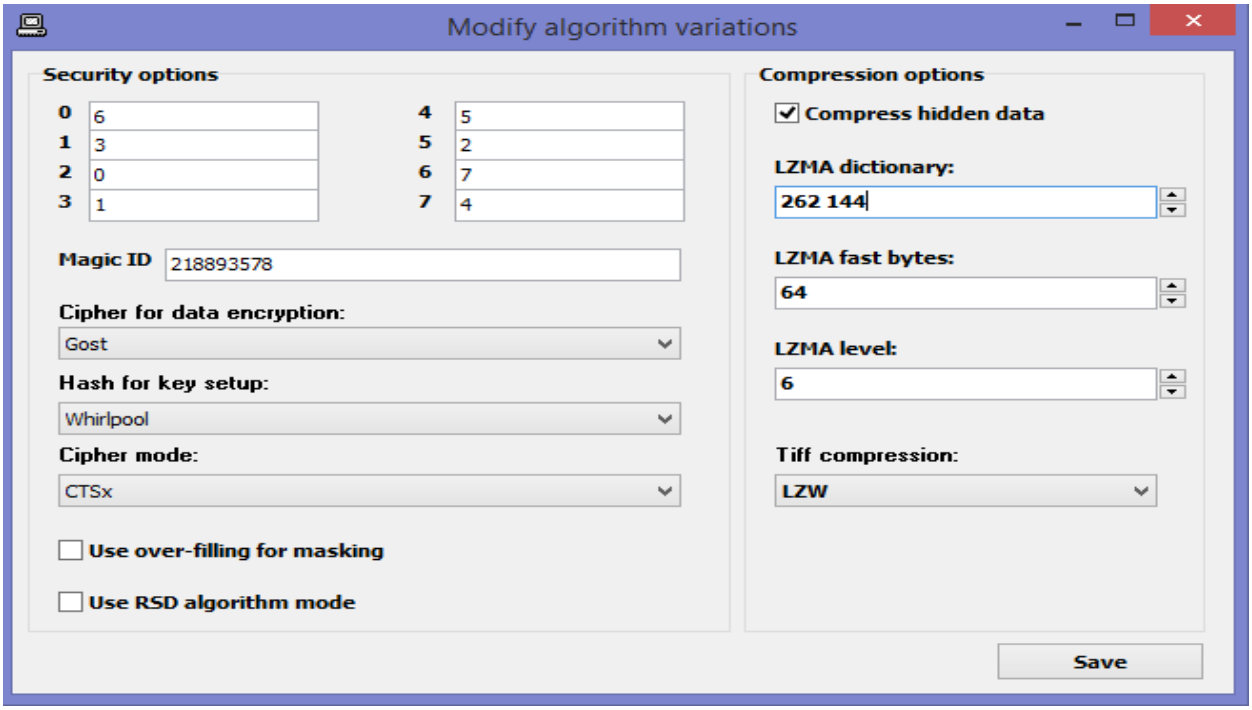


Рисунок 6.5 – Вікно налаштувань програми. Вікно складається з двох частин Security options та Compression options.

Security Options

Поля Magic ID та ті що знаходяться над ним використовуються для налаштування алгоритму секретного випадкового розподілу.

У пункті Cipher for data encryption можна вказати алгоритм шифрування приховуваного файлу.

Пункт Hash for key setup вказує алгоритм розрахунку хеша ключа.

У пункті можна обрати режим шифрування Cipher mode.

Use over-filling for masking – використання надлишковості

Use RSD algorithm mode – ввімкнення алгоритму секретного випадкового розподілу для приховування файлу.

Compression options

Compress hidden data – вказує чи будуть стискатися приховувані дані.

LZMA dictionary, LZMA fast bytes та LZMA level використовуються для налаштування якості та швидкості стискання. Чим вище значення LZMA dictionary та LZMA level тим якісніше, але повільніше стискається приховуваний файл. Значення ж LZMA fast bytes чим вище тим швидше стискається приховуваний файл.

Tiff compression – використовується при збереженні файлу формата *.tiff.

Порядок виконання роботи

Завдання 1

1. Вбудуйте інформацію в контейнер.
2. Порівняйте контейнер до і після вбудовування.
3. Вилучіть інформацію із контейнера.

Завдання 2

1. Вбудуйте в контейнер файл більше допустимого розміру.
2. Опишіть результат.
3. Вилучіть файл із контейнера.

Зміст звіту

– короткі відомості про використані методи приховування інформації;
– короткі відомості про програму “ImageSpyer G2”;
– висновки про результати експериментів.

Контрольні питання

1. Для чого призначена програма ImageSpyer G2.
2. Опишіть її функціональні можливості.
3. Які контейнери дозволяє використовувати програма.
4. Для чого використовується шифрування даних?
5. Для чого гешується ключ?
6. Опишіть алгоритм LSB.
7. Опишіть принцип роботи LSB.
8. Назвіть недоліки LSB.
9. Розкрийте суть алгоритму LZMA.

Лабораторна робота №7

ПРИХОВУВАННЯ ДАНИХ ЗА ДОПОМОГОЮ ПРОГРАМИ “MSU STEGO VIDEO”.

Мета роботи: Дослідження роботи програми “MSU stego video”.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Стеганографія - це метод організації зв'язку, що властиво приховує саму наявність зв'язку. На відміну від криптографії, де ворог точно може визначити чи є передане повідомлення зашифрованим текстом, методи стеганографії дозволяють вбудовувати секретні повідомлення в необразливі послання так, щоб неможливо було запідозрити існування вбудованого таємного послання.

Слово "стеганографія" в перекладі із грецького буквально означає "тайнопис" (steganos - секрет, таємниця; graphy - запис). До неї ставиться величезна безліч секретних засобів зв'язку, таких як невидиме чорнило, мікрофотознімки, умовне розташування знаків, таємні канали й засоби зв'язку на плаваючих частотах і т.д.

Стеганографія займає свою нішу в забезпеченні безпеки: вона не заміняє, а доповнює криптографію. Приховання повідомлення методами стеганографії значно знижує ймовірність виявлення самого факту передачі повідомлення. А якщо це повідомлення до того ж зашифроване, то воно має ще один, додатковий, рівень захисту.

У цей час у зв'язку з бурхливим розвитком обчислювальної техніки й нових каналів передачі інформації з'явилися нові стеганографічні методи, в основі яких лежать особливості подання інформації в комп'ютерних файлах, обчислювальних мережах і т.п. Це дає нам можливість говорити про становлення нового напрямку - комп'ютерної стеганографії.

Існує багато програм за допомогою яких є можливість приховувати дані в інших даних, і приховані дані передавати по Інтернету, мережі, через дискети. Розглянута в даній лабораторній роботі програма «MSU stego video» є однією з таких програм.

Опис програми «MSU stego video»

«MSU stego video» дає можливість приховувати дані відеопослідовності файлів *.avi. «MSU stego video» також дає можливість витягувати приховувані дані з зашифрованих файлів. Дана програма є простою в використанні , має дружній інтерфейс, і підказки при роботі.

Для встановлення програми потрібно розпакувати папку msu_stegovideo потім , слідуючи інструкціям які будуть з'являтись на екрані, проінсталювати її.

Після інсталяції запускаємо програму «MSU stego video», на екрані з'являється головне вікно програми, в яке дає можливість обирати можливі методи роботи в програмі «MSU stego video»:

Рисунок 7.1 Вибір напрямку роботи програми.

1. **Hide file in video –** робота програми в напрямку приховування даних.
2. **Extract file from video –** робота програми в напрямку отримування даних, прихованих раніше.

Вибираємо потрібний нам пункт (1) і натискаємо кнопку «Next». Далі, у слідуючому діалоговому вікні (Рис. 7.2.), у відповідних стрічках за допомогою кнопок «Browse» вибираємо слідуючі дані:

1. **Choose source video –** вибрати *.avi-файл, який буде використовуватись як стегоконтейнер.
2. **Choose file for video with hidden info –** вибрати *.avi-файл, який буде містити приховану в ньому інформацію.
3. **Choose file that you want to hide –** вибрати файл, який потрібно приховати.

Рисунок 7.2 Ініціалізація програми

Після введення всіх даних натискаємо кнопку «Next». У наступному діалоговому вікні (рис. 7.3.) пропонується стискання вихідного файлу і попереджується про можливу втрату частини інформації після стиснення.

Рисунок 7.3 Запит на стиснення стего

Далі пропонуються налаштування роботи програми, де можна самому вказати наскільки помітною буде прихована інформація, а також надійність (рис. 7.4).

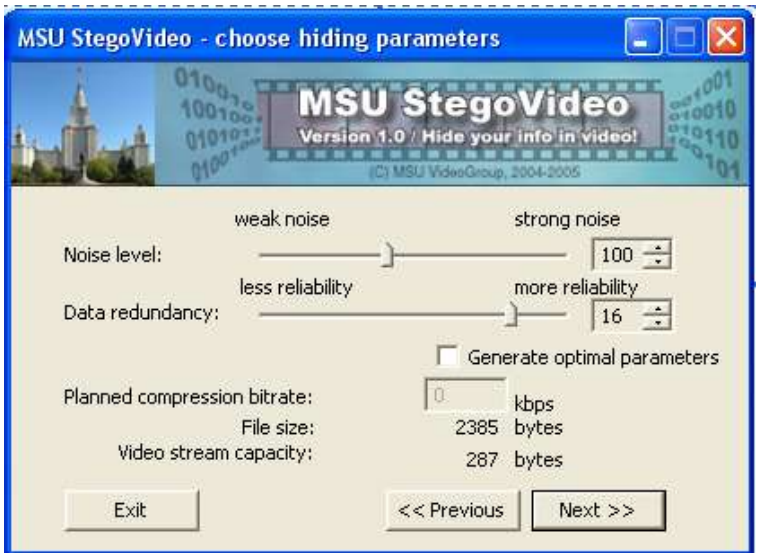


Рисунок 7.4 Параметри приховування

- **strong noise** – сильный шум
- **weak noise** – невеликий шум
- **less reliability** – зменшення надійності
- **more reliability** – збільшення надійногсті
- **planned compression bitrate** – запланована компресійна швидкість передачі данних
- **video stream capacity** – ємність (пропускна здатність) потоку відеосигналу

Після встановлення всіх установок, программа автоматично генерує пароль, який буде використовуватись для отримування даних із стего. Важливо зберегти або запам'ятати цей пароль, оскільки без нього дані із стего отримати буде неможливо. Натискаючи кнопку, Ви підтверджуєте той факт, що ви зберегли пароль (рис 7.5).

Рисунок 7.5 – Зберігання пароля

Одразу після натискання кнопки, розпочинається вбудова інформації у відео послідовність avi-файлу (рис 7.6):

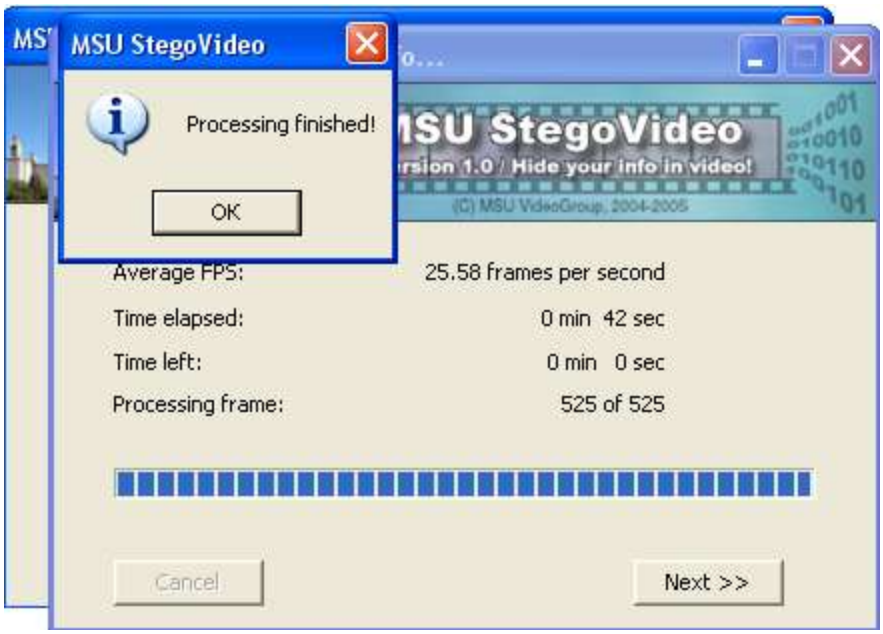


Рис. 7.6 Завершення приховування.

Після виводу повідомлення про закінчення процесу, робота програми в напрямку приховування вважається закінченою.

Щоб отримати приховані дані із стего, потрібно після запуску програми вибрати пункт «**Extract file from video».** Далі вказується шлях до стего та файл для запису витягнутої інформації (Рис 7.7.)

Рисунок 7.7 Видобування файлу із стенограми.

Після натиснення кнопки «Next» потрібно ввести пароль, який збережено раніше при процесі приховування інформації (рис 7.8.):

Рис. 7.8 Введення паролю

Після введення правильного паролю і натиснення кнопки «Next» програма автоматично витягує приховану інформацію із стего до вказаного файлу(рис. 7.9):

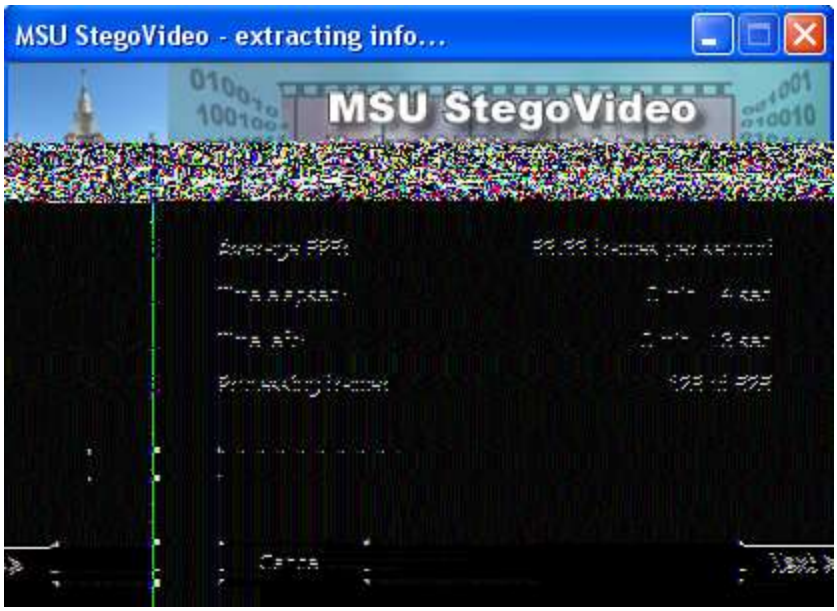


Рисунок 7.9 Процес видобування прихованої інформації

Порядок виконання роботи

1. Вивчити короткі теоретичні відомості.
2. Ознайомитися з функціональними можливостями програми “MSU stego video”
3. Вбудувати у будь-який файл типу *.avi.
4. Візуально проаналізувати заповнений та незаповнений контейнери. Зробити висновки.
5. Вилучити файл із заповненого контейнера і порівняти його.
6. Оформити звіт про виконання лабораторної роботи.

Зміст звіту

– короткі відомості про використані методи приховування інформації;
– короткі відомості про програму “MSU stego video”;
– висновки про результати експериментів.

Контрольні питання

1. Які функціональні можливості програми “MSU stego video”.
2. Які параметри приховування дозволяє налаштувати програма.
3. Які вимоги до контейнера.
4. Які дані дозволяє приховувати програма.
5. Як змінюється якість відео після приховування у нього інформації.

Лабораторна робота №8

ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЗВУКОВИХ ФАЙЛАХ

Мета роботи: Вивчити методи вбудовування інформації у звукові файли формату MP3. Дослідити програму Mp3Stego, яка дозволяє вбудовувати текстові файли в звукові файли формату MP3

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Mp3Stego

На сьогодні існує на так вже й багато стеганографічних інструментів, доступних в мережі, які могли б приховати інформацію в MP3-файли, стиснувши при цьому звукові доріжки за допомогою у формат MPEG Audio III. Із звукових файлів найцікавішими є MP3 та WMA файли, тому що вони пропонують близьку до CD якість за ступенем стиснення від 11 до 1 (128 кілобіт на секунду). Це дає дуже гарну можливість для приховування інформації. Незважаючи на те, що WMA має кращу можливість для приховування в нього файлів, доступ до коду цих файлів зазвичай закритий, тому як контейнери використовуватимуться саме MP3 - файли .

MP3Stego приховує інформацію в MP3-файлах під час процесу стиснення. Спочатку дані стискаються, шифруються, а потім ховаються в бітовий потік. MP3Stego була написана з розрахунком на те, що вона могла б бути використаною в якості системи маркування авторських прав на MP3-файли (слабкої, але все ще набагато кращої, ніж у формату MPEG). Будь-хто може розпакувати бітовий потік, та стиснути його знову, проте це видалить приховану в ньому інформацію.

Процес відбувається на основі процесу кодування Layer III, який називається inner_loop. Внутрішній цикл квантування вхідних даних збільшує розмір квантовального кроку, поки неквантовані дані не зможуть бути закодовані з наявною кількістю бітів. Інший цикл перевіряє спотворення, що вносяться квантуванням і які не перевищують поріг, визначений психоакустичною моделлю. Змінна part2_3_length містить деяку кількість main_data бітів, використовуваних для даних коду Хаффмана в MP3-бітовому потоці. Ми кодуємо біти парності шляхом зміни стану кінця внутрішнього циклу. Лише випадково обрані значення part2_3_length будуть змінені; вибір робиться з використанням псевдовипадкового генератора бітів, заснованого на SHA-1.

Рисунок 11.1 – Зображення головного меню програми Mp3Stego (із графічним інтерфейсом)

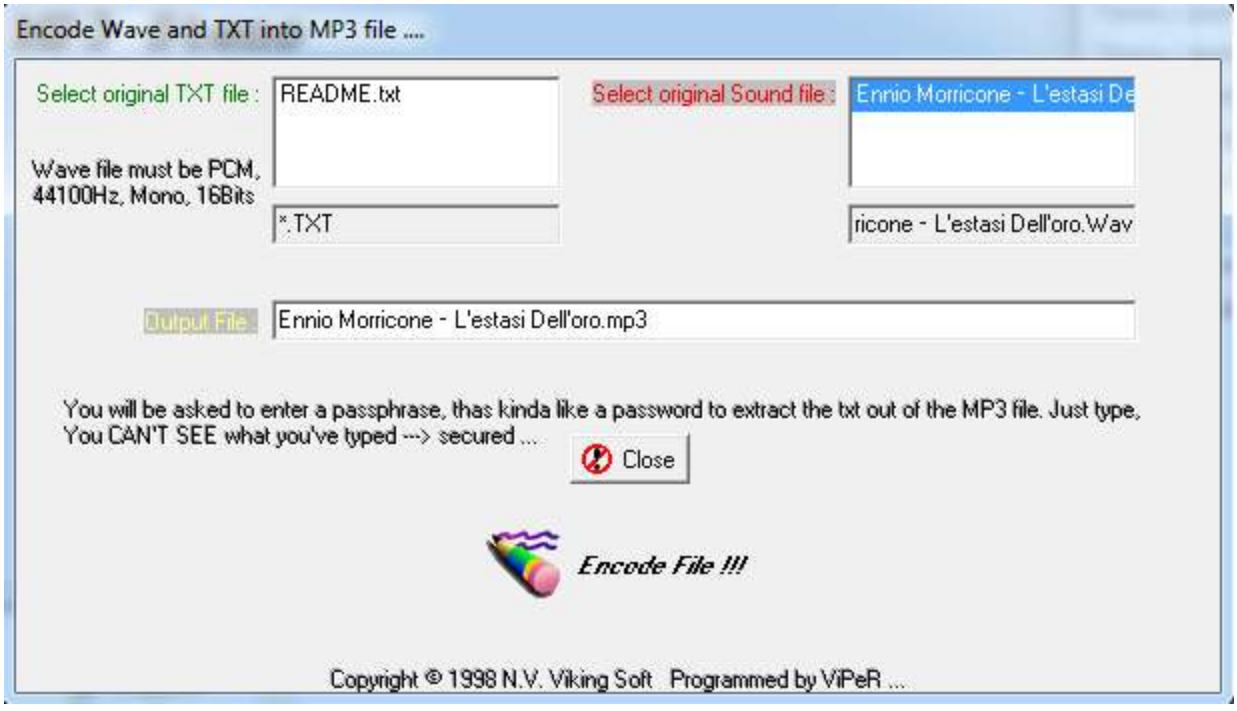


Рисунок 11.2 – Зображення вікна кодування

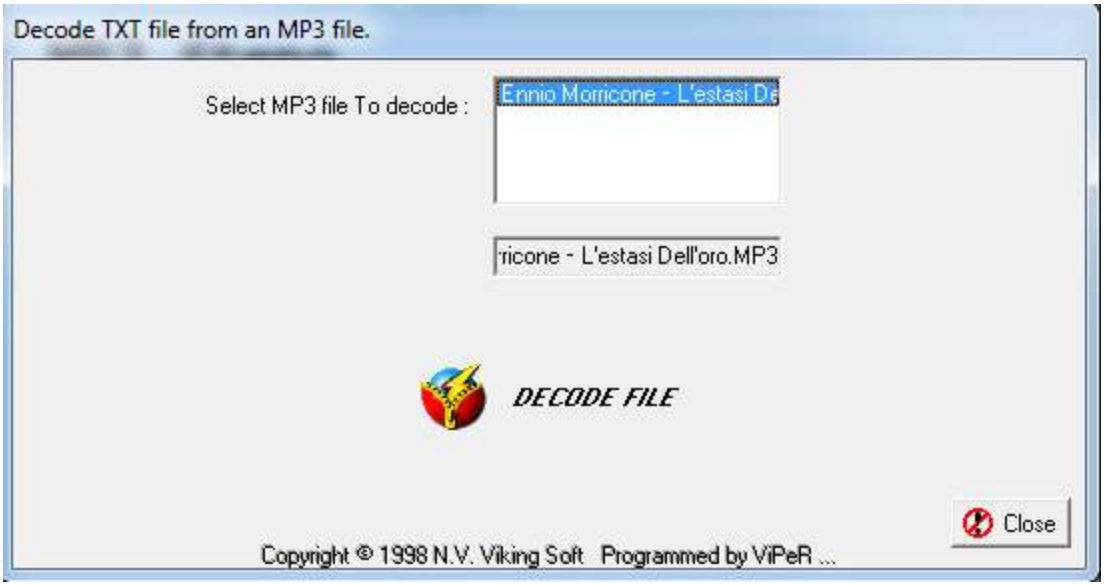


Рисунок 11.3 – Зображення вікна декодування

Порядок виконання роботи

1. Закодувати невеликий TXT-файл у звуковий файл
2. Спробувати закодувати великий (150-200кб) файл у попередньо створену копію звукового файлу, який використовувався у завданні 1
3. Перевірити обидва контейнери на наявність підозрілих шумів, або яких-небудь відхилень у звучанні (можна порівнювати із немодифікованим звуковим файлом)
4. Витягнути інформацію із обох контейнерів, перевірити її стан.
5. До кожного із пунктів зробити короткі пояснення та скриншоти.

Зміст звіту

– короткі відомості про використані методи приховування інформації;
– короткі відомості про програму Mp3Stego;
– висновки про результати експериментів.

Контрольні питання

1. Яким чином Mp3Stego приховує txt - файли у mp3 – контейнери?
2. Які ще звукові файли окрім mp3 можуть служити контейнерами?
3. Оцініть загальну якість та працездатність програми Mp3Stego.

Лабораторна робота №9

ШИФРУВАННЯ ТЕКСТУ ТА ЗОБРАЖЕННЯ

Мета роботи: Набути навички шифрування зображень і тексту, приховування інформації у зображення за допомогою програми «Cipher image».

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Шифрува́ння - оборотне перетворення даних, з метою приховання інформації. Шифрування з'явилось близько 4 тис. років назад.

Розшифрування - процес санкціонованого перетворення зашифрованих даних у придатні для читання.

Дешифрува́ння - процес несанкціонованого отримання інформації з зашифрованих даних. При цьому ключ дешифрування зазвичай невідомий.

Є два методи шифрування: симетричне та асиметричне.

1. в симетричному шифруванні один і той самий ключ (що зберігається в секреті) використовується як для шифрування, так і для розшифрування. Розроблено ефективні (швидкі й надійні) методи шифрування.
2. в асиметричному шифруванні є два пов'язаних ключа - пара ключів. Відкритий ключ - публічний, до нього повинні мати доступ всі ті, хто матиме потребу зашифрувати інформацію. Тоді як закритий ключ — приватний ключ, повинен бути доступним лише тому хто має право розшифрувати інформацію, за своїм розміром він значно більший від секретного ключа симетричного шифрування.

При впровадженні прихованої інформації в зображення, раніше користувались методом вкладання в незначні біти для зменшення візуальної помітності. Але більш сучасний підхід полягає у вбудовуванні інформації у найбільш істотні зони зображення , руйнування яких призведе до повної деградації всього зображення . Стегоалгоритми враховують властивості людського зору аналогічно алгоритмам ущільнення. Зазвичай використовуються такі самі перетворення.

Функціональні можливості програми

Програмний засіб «Cipher image» дає можливість приховувати цінну інформацію від несанкціонованого доступу. Використовувати дане ПЗ для кодування зображень і приховування тексту у зображення легко. У зашифрованому вигляді можуть бути збережені кілька зображень, разом із прихованим текстом, у один файл.

Ця невелика утиліта може запросто перетворити звичайне графічне зображення у приховану інформацію. Таке зображення із зашифрованим у ньому текстом можна розмітити у себе на сайті і ніхто, крім користувачів, для яких він призначений, не зможе її прочитати. Усі інші користувачі будуть вважати його звичайною картинкою. Безкоштовна програма «Cipher image free» дозволяє зберігати відразу кілька зашифрованих зображень в одному багатосторінковому TIFF файлі. Вона підтримує більше 21 форматів графічних документів, має простий і інтуїтивно зрозумілий інтерфейс.

Особливості програми

- 7 форматів для збереження файлів;
- Прихований текст повідомлення може бути до 64 Кбайт;
- 128-бітове шифрування зображень з випадковим ключовим словом;
- Вбудований генератор ключових слів;
- Підказки при роботі «Magic Help»;
- Підтримка всіх версій Windows.

Порядок роботи з програмою

Проведемо ближче знайомство з роботою програми. Оскільки програма потребує інсталяції, знайдіть у теці лабораторної роботи виконуваний файл під назвою «Setup_cipher_image_free.exe». Далі крок за кроком виконайте послідовність дій для встановлення програми на ваш персональний комп'ютер. Програма написана на англійській мові, тому всі позначення в ній будуть англійською.

Інтерфейс даної програми зручний та лаконічний (рис.9.1). Програма поділена на 3 діалогових вікна, меню та допоміжну стрічку.

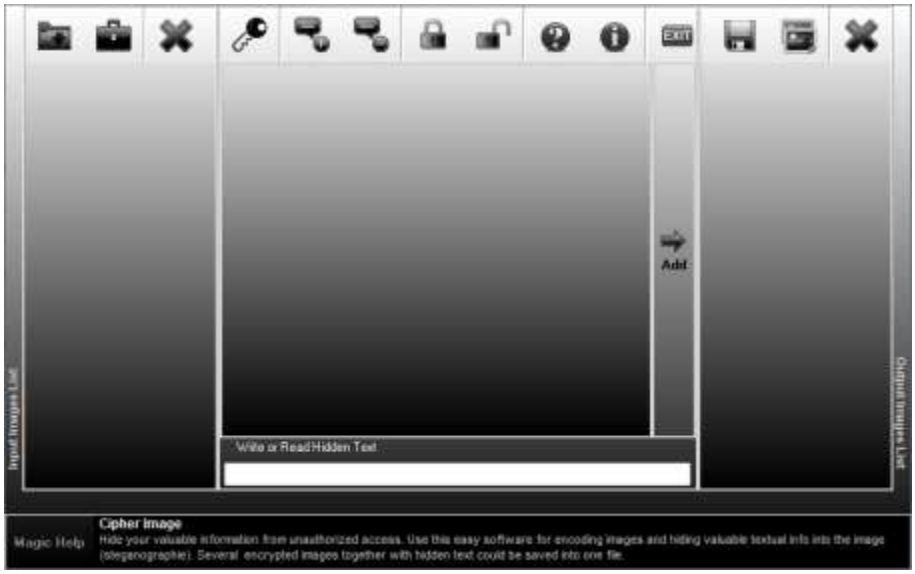


Рисунок 9.1 – Вигляд вікна програми

Перше вікно під назвою «Список вхідних зображень» містить зображення для редагування. Натисніть «**відкрити зображення…**» щоб додати зображення до «**списку вхідних зображень**» з файлу (рис. 2). Або натисніть кнопку «**вставити зображення з буферу обміну**», щоб додати зображення до «**списку вхідних зображень**» з буферу обміну.

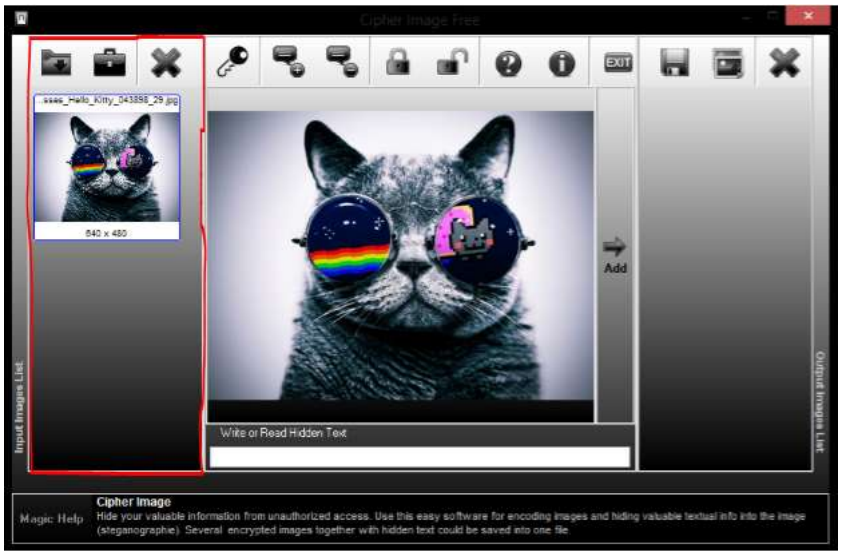


Рисунок 9.2 – Вигляд вікна програми після додавання зображення до «списку вхідних зображень»

Друге вікно «вікно для перегляду» має робоче поле для редагування зображення. Натисніть на картинку в «**списку вхідних зображень**» (рис. 3) щоб додати у вікно. Також має кнопку «ADD» для додавання зображення до списку вихідних зображень та стрічку для писання та читання прихованого тексту.

Рисунок 9.3 – Вигляд головного вікна з робочою зоною

Третє вікно «список вихідних зображень» містить зображення для збереження у файл. Натисніть кнопку «**додати зображення**». Зображення додасться до «**списку вихідних зображень**» (рис. 9.4).

Рисунок 9.4 – Вигляд вікна програми після додавання зображення до «списку вихідних зображень»

У магічній стрічці пишуться коментарі до об’єкту на який наведений курсор.

Кожне вікно має своє відповідне меню (рис. 9.5). Воно складається із кнопок які находяться вверху програми. Нижче описані всі кнопки відповідно.

Рисунок 9.5 – Вигляд меню

- Відкрити картинку…
- натисніть цю кнопку, щоб додати зображення до «списку вхідних зображень» з файлу.
- Вставити картинку з буферу обміну
- натисніть цю кнопку, щоб додати зображення до «списку вхідних зображень» з буферу обміну.
- Видалити вибране (вибрані) зображення
- натисніть цю кнопку, щоб видалити вибране (вибрані) зображення зі «списку вхідних зображень».
- Новий ключ

Натисніть цю клавішу. В діалоговому вікні під назвою «**Новий ключ**» введіть ключове слово у поле для розшифровки і зашифровки зображень (≤ 16 літерів) , наприклад «Моє кохання». Ви можете створити ключове слово автоматично. Щоб зробити це, потрібно натиснути «**генерувати ключ**» (рис. 6). Додатково є можливість обрати ключ з файлу, а потім зберегти його на вашому комп'ютері під довільним ім'ям в файлі з розширенням txt.

Рисунок 9.6 – Вигляд вікна для введення генерованого ключа

Додати прихований текст

Введення тексту в введений текст («**писати або читати прихований текст**» панель) (рис. 9.7). Натисніть «**додати прихований текст**». Текст буде приховано в зображення.

Рисунок 9.7 – Вигляд вікна з текстом, що буде приховано

Видалити прихований текст

видалити прихований текст натисніть «**видалити прихований текст**».

Шифрувати зображення

в діалоговому вікні «**новий ключ**» (натисніть кнопку «**новий ключ**») введіть ключове слово (наприклад «Моє кохання»). Ви зможете створити ключове слово автоматично. Щоб зробити це, ви повинні натиснути «**генерувати ключ**». Натисніть «**шифрувати зображення**» і зображення буде зашифроване (рис. 9.8).

Рисунок 9.8 – Вигляд вікна програми після кодування зображення

Розшифрувати зображення

в діалоговому вікні «**новий ключ**» (натисніть кнопку «**новий ключ**») введіть ключове слово (наприклад «Моє кохання»). Натисніть «**розшифрувати зображення**» і зображення буде розшифроване (рис. 9.9).

Рисунок 9.9 – Вигляд вікна програми після розшифрування зображення

- **Допомога**
  натиснувши цю кнопку побачите документовану допомогу в PDF форматі.
- **Інформація**
  натисніть цю кнопку щоб переглянути інформацію про цю програму.
- **Вихід**
  натисніть цю кнопку щоб вийти з програми. Всі не збережені дані будуть втрачені!
- **Зберегти зображення як…**
  натисніть цю кнопку. В діалоговому вікні вкажіть ім’я файлу, встановіть формат зображення і натисніть «**зберегти**». **Важливо! Зашифроване зображення потрібно зберігати тільки в BMP, TIFF, PNG!**
- **Копіювати вибране зображення**
  натисніть цю кнопку щоб скопіювати дане зображення зі «**списку виведення зображення**» в буфер обміну.
- **Видалити вибране (вибрані) зображення**
  натисніть цю клавішу щоб видалити вибране (вибрані) зображення зі «**списку виведення зображення**».

Порядок виконання роботи

1. Ознайомитися з функціональними можливостями програми «Cipher image».
2. Вибрати графічний файл для шифрування.
3. Виконати шифрування зображення з прихованим текстом та без прихованого тексту.
4. При шифруванні надати ключ (генерування).
5. Отримати зашифровані дані з новоствореного зображення. При розшифруванні вказати неправильне зашифроване слово, а потім правильне.

Зміст звіту

- короткі відомості про програму «Cipher image»;
- покроковий опис виконання роботи зі скрінами;
- висновки, зроблені під час виконання лабораторної роботи.

Контрольні питання

1. У яких форматах потрібно зберігати зашифроване зображення?
2. Як змінився розмір зашифрованого зображення (без тексту, з текстом)?
3. Чи змінився вигляд зашифрованого зображення?
4. Які методи існують шифрування? Які між ними різниця?
5. У які контейнери краще вбудовувати інформацію і чому?

Лабораторна робота №10

ПРИХОВУВАННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ ПРОГРАМИ OUR SECRET

Мета роботи: Вивчення методів приховування інформації у зображеннях та аудіо файлах, та набуття навичок з приховування інформації за допомогою програми Our Secret.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Опис програми:

OurSecret дозволяє приховати і зашифрувати файли усередині інших файлів (файлів носіїв), таких, як фотографії або звукові файли. Це дозволяє шифрувати конфіденційну інформацію, в той же час приховуючи його у файл, який не буде виглядати підозрілим, таким чином ніхто навіть не знає, що є зашифрована інформація.

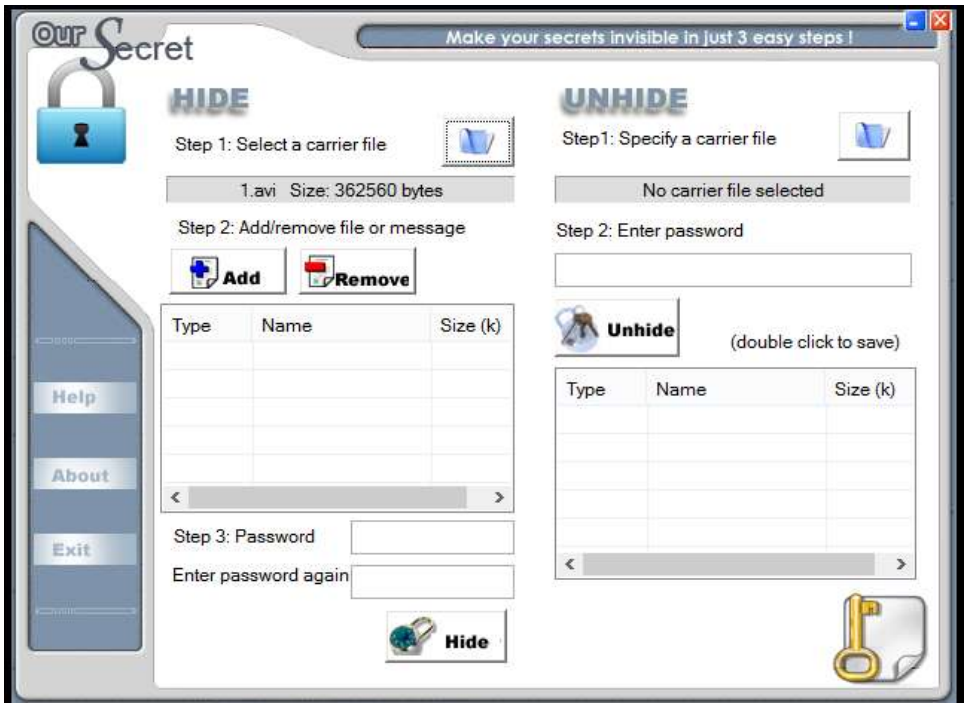


Рисунок 10.1 - Головне вікно програми

Файл-носій є повністю функціональним і ідентичним оригінальному файлу, тому, якщо повідомлення було приховано у зображення, то ніяких змін не буде. Програма дає змогу приховати текстові повідомлення, або які-небудь інші файли, які вам потрібно. Для збільшення секретності, можна ввести пароль, щоб витягти приховані файли.

Плюси

Програма дозволяє приховати будь-який файл в інший файл. Крім того, файл-носій може бути розшифрований тільки за допомогою пароля, інтерфейс інтуїтивно зрозумілий і дуже простий у використанні.

Мінуси

Немає опції відновлення у випадку, якщо користувач втрачає пароль для файлу-носія.

Приклад приховування інформації за допомогою програми OurSecret:

Щоб приховати інформації за допомогою програми OurSecret необхідно обрати файл-контейнер. Програма дає змогу працювати з різними типами файлів, такими як: аудіо, відео, графічні, програмні та текстові файли. Щоб обрати файл-контейнер необхідно в розділі «Hide» натиснути кнопку-зображення у вигляді теки.

Обравши файл-контейнер необхідно обрати інформацію, яку буде приховано в контейнері. Для цього необхідно обрати кнопку «Add» у розділі «Hide» (див. рис. 1). Після цього з’явиться вікно, у якому необхідно обрати тип інформації, яку буде приховано: файл або текстове повідомлення. Вікно для вибору інформації зображено на рис. 10.2.

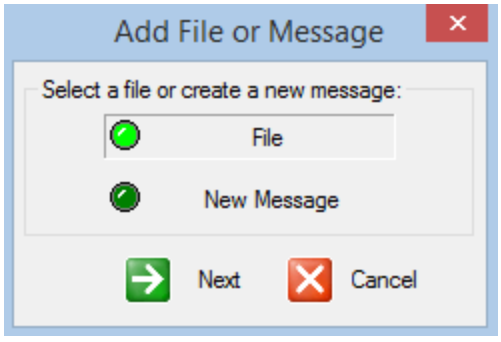


Рисунок 10.2 – Вікно вибору інформації для приховування

Для підтвердження приховування обраної інформації у файлі-контейнері необхідно ввести пароль. Щоб підтвердити введення пароля та приховати інформацію необхідно натиснути на кнопку «Hide». Вікно, у якому необхідно ввести пароль для приховання інформації зображено на рис. 10.3.

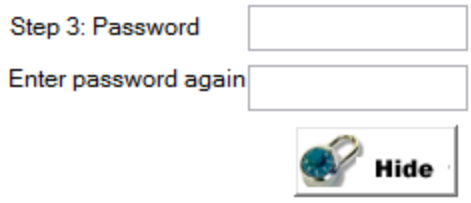


Рисунок 10.3 – Вікно підтвердження пароля

Після підтвердження пароля, з'являється вікно, в якому необхідно ввести назву, тип файлу, а також теку у яку буде збережено файл з прихованою інформацією.

Приклад діставання прихованої інформації з файлу за допомогою програми OurSecret:

Щоб отримати приховану інформацію з файлу-контейнера необхідно обрати файл з прихованою інформацію. Для цього необхідно в розділі «Unhide» обрати кнопку у вигляді теки. Після цього з'являється вікно, у якому необхідно обрати файл з прихованою інформацією. На рис. 10.4 зображено розділ «Unhide» та кнопку у вигляді теки.

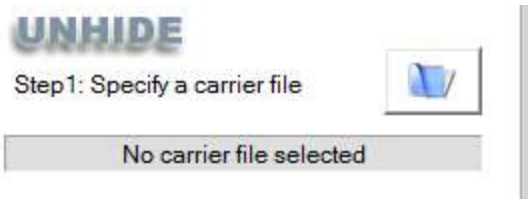


Рисунок 10.4 – Вигляд розділу «Unhide»

Обравши файл з прихованою інформацією, необхідно ввести пароль для діставання прихованої інформації з файлу-контейнера. Підтвердивши пароль, необхідно натиснути кнопку «Unhide». Поле для введення пароля зображено на рис. 10.5.

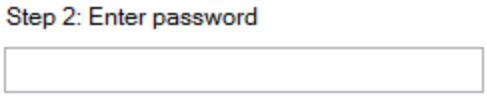


Рисунок 10.5 – Вигляд поля для введення пароля

Після підтвердження пароля у розділі «Unhide» з'явиться список з прихованими файлами. Список прихованих файлів, які можна дістати з файлу контейнера, зображено на рис. 10.6.

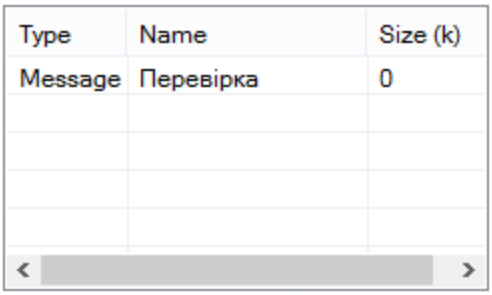


Рисунок 10.6 – Список прихованих файлів

Натиснувши на файл зі списку, з'являється вікно, у якому зображена прихована інформація. Щоб зберегти отриману інформацію з файлу-контейнера необхідно натиснути на кнопку «Save as». Вікно, у якому зображено приховану інформацію з файлу-контейнера, зображено на рис. .

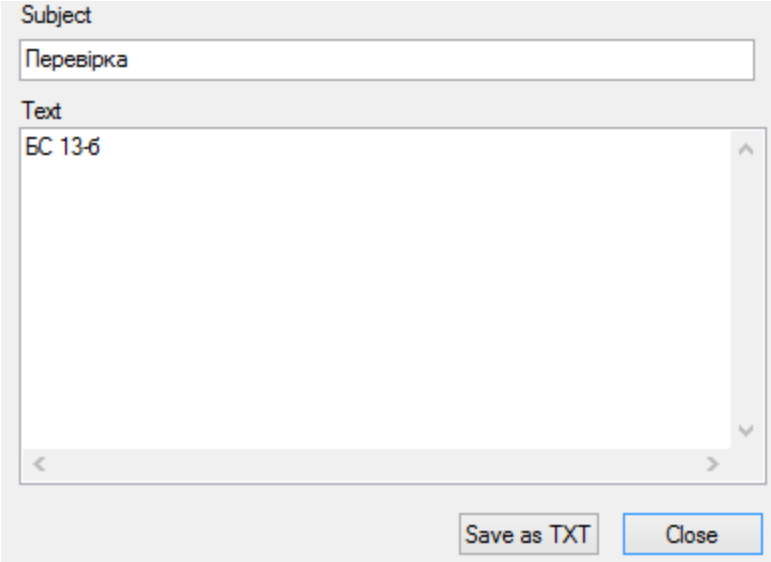


Рисунок 10.7 – Вікно зі збереженою інформацією

Щоб відмінити збереження прихованої інформації необхідно, натисну на кнопку «Close» внизу вікна.

Порядок виконання роботи

- **Завдання 1:**

1. Виберіть файл контейнер, відповідно варіанту, заданого типу і файл для схову (розмір контейнера має бути достатній для вміщення приховуваних даних).
2. Виберіть текст, який буде вбудовано в контейнер. Текстом має бути: ім'я, прізвище студента, група та варіант.

3. Виконайте вбудовування даних.
4. Порівняйте контейнери до і після вбудовування.
5. Отримайте сховані в контейнері дані за різними паролями (спробувати отримати інформацію, ввівши невірний пароль).

- **Завдання 2:**

1. Виберіть файл контейнер, відповідно варіанту, заданого типу і файл для схову (розмір контейнера має бути достатній для вміщення приховуваних даних).
2. Виберіть файл, який буде приховано в контейнерів, відповідно варіанту.
3. Виконайте вбудовування даних.
4. Порівняйте контейнери до і після вбудовування.
5. Отримайте сховані в контейнері дані.

Варіанти студентів:

Варіант	Тип контейнера	Тип прихованої інформації
1	Кольорове зображнння *.gif	Текстовий файл *.txt
2	Чорно-біле зображнння *.gif	Текстовий файл *.doc
3	Кольорове зображнння *.png	Текстовий файл *.html
4	Чорно-біле зображнння *.png	Аудіо файл *.wav
5	Кольорове зображнння *.jpeg	Аудіо файл *.mp3
6	Чорно-біле зображнння *.jpeg	Аудіо файл *.wav
7	Відео-файл *.mp4	Кольорове зображнння *.gif
8	Відео-файл *.3gp	Чорно-біле зображнння *.gif
9	Відео-файл *.avi	Кольорове зображнння *.png
10	Відео-файл *.mkv	Текстовий файл *.txt
11	Аудіо файл *.mp3	Текстовий файл *.doc
12	Аудіо файл *.wav	Текстовий файл *.html
13	Текстовий файл *.txt	Чорно-біле зображнння *.png
14	Текстовий файл *.doc	Кольорове зображнння *.jpeg
15	Виконуваний файл *.exe	Аудіо файл *.mp3

Зміст звіту

- короткі відомості про використані методи приховування інформації;
- короткі відомості про програму Our Secret;
- висновки про результати експериментів.

Контрольні питання

1. З якими видами файлів можлива робота програми?
2. Які режими роботи з паролем використовуються в програмі і чи обов'язковий він?
3. Які особливості має програма Our Secret, плюси і мінуси ?
4. Яким чином відбувається вбудовування інформації в контейнер?
5. Як отримати зашифровану інформацію з контейнера?
6. В якому форматі зберігається заповнений контейнер?

Лабораторна робота №11

РОЗРОБКА СТЕГАНОГРАФІЧНОЇ ПРОГРАМИ

Мета роботи: Дослідити роботу основних стеганографічних алгоритмів для приховування інформації у картинках, аудіо та відео файлах. Практично реалізувати один із алгоритмів.

Порядок виконання роботи

1. Напишіть програму вказану у вашому варіанті (Таблиця 1).
2. Протестуйте правильність роботи вашої програми.
3. Вбудуйте інформацію допустимих розмірів у контейнер.
4. Порівняйте контейнер до і після вбудовування.
5. Вилучіть інформацію із контейнера.
6. Вбудуйте в контейнер файл більше допустимого розміру.
7. Опишіть результат.

Таблиця 1 – Варіанти завдань

№ варіанту	Завдання	
	Контейнер:	Файл для схову
1	BMP	*.doc
2		*.txt
3		*.avi
4		*.jpeg
5		*.bmp
6		*.wav
7		*.gif
8		*.exe
9		*.com
10		*.rar
11	WAV	*.doc
12		*.txt
13		*.avi
14		*.jpeg
15		*.bmp

№ варіанту	Завдання	
	Контейнер:	Файл для схову
16	WAV	*.wav
17		*.gif
18		*.exe
19		*.com
20		*.rar
21	AVI	*.doc
22		*.txt
23		*.avi
24		*.jpeg
25		*.bmp
26		*.wav
27		*.gif
28		*.exe
29		*.com
30		*.rar

Зміст звіту

- короткі відомості про використані методи приховування інформації;
- короткі відомості про вашу програму;
- висновки про результати експериментів.

Контрольні питання

1. Які вимоги вашої програми до контейнера.
2. Який алгоритм приховування використовує ваша програма.
3. Яким чином обчислюється максимальна допустима довжина приховуваного файлу.
4. Які налаштування має ваша програма.
5. Які вам відомі способи приховування інформації у картинках.
6. Які вам відомі способи приховування інформації у аудіо файлах.
7. Які вам відомі способи приховування інформації у відео файлах.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Аграновский А. В. Основы компьютерной стеганографии / А. В. Аграновский, П. Н. Девянин, Р. А. Хади– М.: Радио и связь, 2003. – 151 с.
2. Грибунин В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. – Санкт-Петербург: Солон-Пресс, 2002. – 272 с.
3. Конахович Г. Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – К.:МК-Пресс, 2006. – 288 с.
4. Конахович Г. Ф. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних [підручник] / Г.Ф. Конахович, Д. О. Прогонов, О.Ю. Пузиренко.— К. : «Центр навчальної літератури», 2018. — 558 с.
5. Хорошко В. О. Основи комп'ютерної стеганографії: [навч. пос.] / В.О.Хорошко, О. Д. Азаров, М. Є. Шелест– Вінниця: ВДТУ, 2003. – 143 с.
6. Юдін О. К. Захист інформації в мережах передачі даних: підручник МОН України [для студ. вищ. навч. закл.] / О. К. Юдін, О.Г.Корченко, Г. Ф. Конахович– К.: Дельфін, 2009. – 714 с.
7. Задірака В. К. Аналіз стійкості стеганографічних систем в моделі пасивного противника / В. К. Задірака, Н. В. Кошкіна, О. С. Олексюк // Искусственный интеллект. – 2004. – № 3. – С. 801-805.
8. Яковлев В.А. Защита информации на основе кодового зашумления. Часть 1. Теория кодового зашумления. / Под ред. В.И. Коржика.– СПб.: ВАС, 1993. –245 с.
9. Лукічов В. В. Методи та засоби стеганографічного захисту інформації на основі вейвлет-перетворень / В. В. Лукічов, В. А. Лужецький, А. С. Васюра. – Вінниця: ВНТУ, 2014. – 160 с.
10. Шелухин О.И. Стеганография. Алгоритмы и программная реализация / О.И. Шелухин, С.Д. Канаев. – М.: Горячая линия-Телеком, 2017. — 592 с.

ГЛОССАРІЙ

Контейнер – будь-яка інформація (потік даних, файл та ін.), призначена для приховування інформації стеганографічним перетворенням.

Стеганоконтейнер (стеганограма) – контейнер (стеганотекст, стеганозвук, стеганозображення і т. п.), отриманий у результаті стеганографічного перетворення і такий, що містить приховану інформацію.

Порожній контейнер – контейнер без вбудованого повідомлення.

Приховане повідомлення – повідомлення, вбудоване в контейнер.

Стеганограф – програмний засіб, що забезпечує процес приховування і витягання приховуваної інформації.

Стеганографічний алгоритм – набір математичних і логічних правил і процедур, за допомогою яких проводиться стеганографічне перетворення.

Стеганографічний ключ (***стегоключ)*** – додаткові секретні дані, які необхідні для керування стеганографічним перетворенням. Якщо для приховування і витягання захищуваних даних потрібен той самий ключ, то він називається симетричним, у іншому випадку – асиметричним.

Стеганографічний метод – принцип реалізації стеганографічного перетворення.

Стеганографічне перетворення – сукупність операцій, пов’язаних із приховуванням і витяганням приховуваної інформації.