SOMAIYA VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering, Mumbai-400077

Department of Computer Engineering

Roll No.: 16010122313 Group No: -
Name of the student: Aashutosh Panda
Div: B
Branch: Computer Engineering
IA No: IA1
Date: 17/2/25

Subject: Information Security

TITLE: Implementation of - Basic Steganography Tools

AIM: To implement and analyze basic steganography tools for securely hiding and extracting secret messages within digital media using different techniques.

Literature survey/Theory:

Steganography is the technique of hiding secret information within digital media such as images, audio, and video files. Unlike encryption, which scrambles data to make it unreadable, steganography ensures the hidden message remains undetectable. Various steganographic tools have been developed to facilitate this process, each using different encoding techniques. Some of the commonly used tools include Stegosuite, Steghide, Xiao Steganography, SSuite Picsel, OpenPuff, and Camouflage.

Concept/Algorithms:

Least Significant Bit (LSB) Encoding:

LSB encoding is a simple and widely used steganography technique where the least significant bit of each pixel in an image (or sample in an audio file) is modified to hide secret data.

How It Works:

1. Convert the secret message into binary form.
2. Take the cover image and extract its pixel values.
3. Replace the least significant bit of selected pixels with bits from the secret message.
4. Save the modified image as a stego image, which looks visually identical to the original.

Example:

- Original pixel (Red component in RGB): 10110101
- Secret bit to hide: 0
- Modified pixel: 10110100

Since only the last bit is changed, the difference is imperceptible to the human eye.

Extraction Process:

1. Load the stego image.
2. Read the LSB of each modified pixel.
3. Convert extracted bits back into readable text.

Pseudocode/Flowchart/Implementations/Screenshots with steps:

Pseudocode for Image-Based Steganography (LSB Method)

Algorithm HideMessage

Input: Cover Image, Secret Message, Output Image

Output: Stego Image with Hidden Message

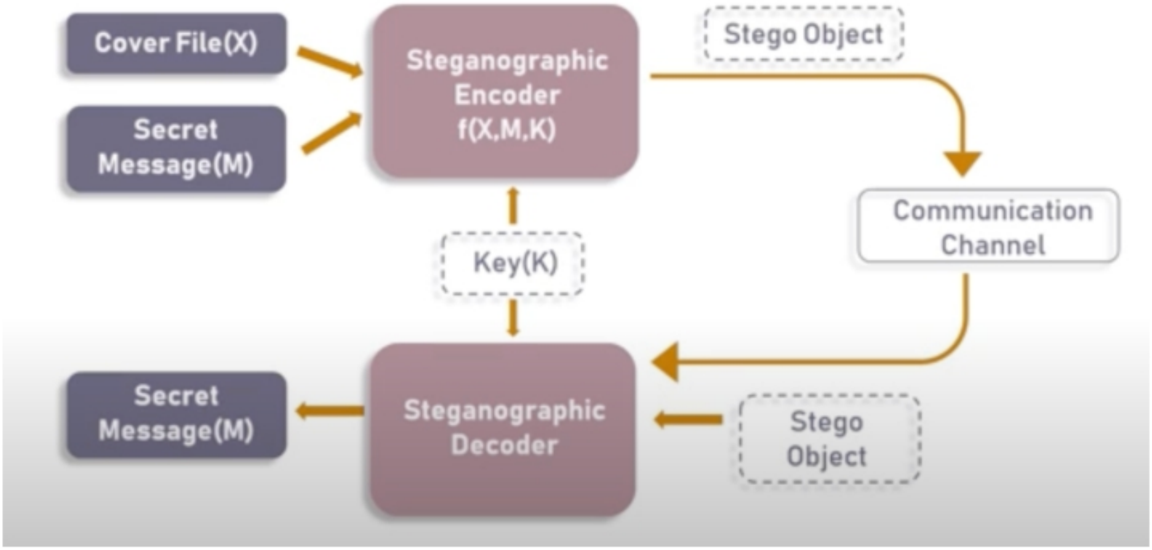
1. Convert the Secret Message into Binary.
2. Load the Cover Image and extract pixel values.
3. Replace the Least Significant Bit (LSB) of each pixel with bits from the Secret Message.
4. Save the modified image as Stego Image.
5. Return Stego Image.

Algorithm ExtractMessage

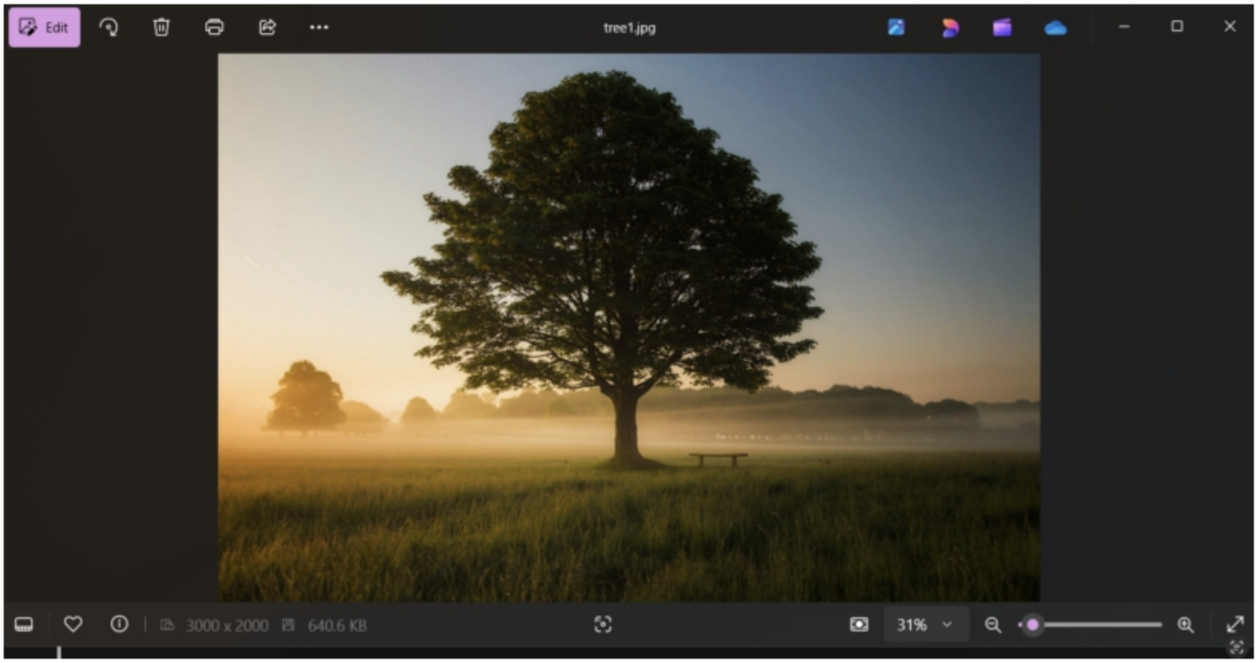
Input: Stego Image

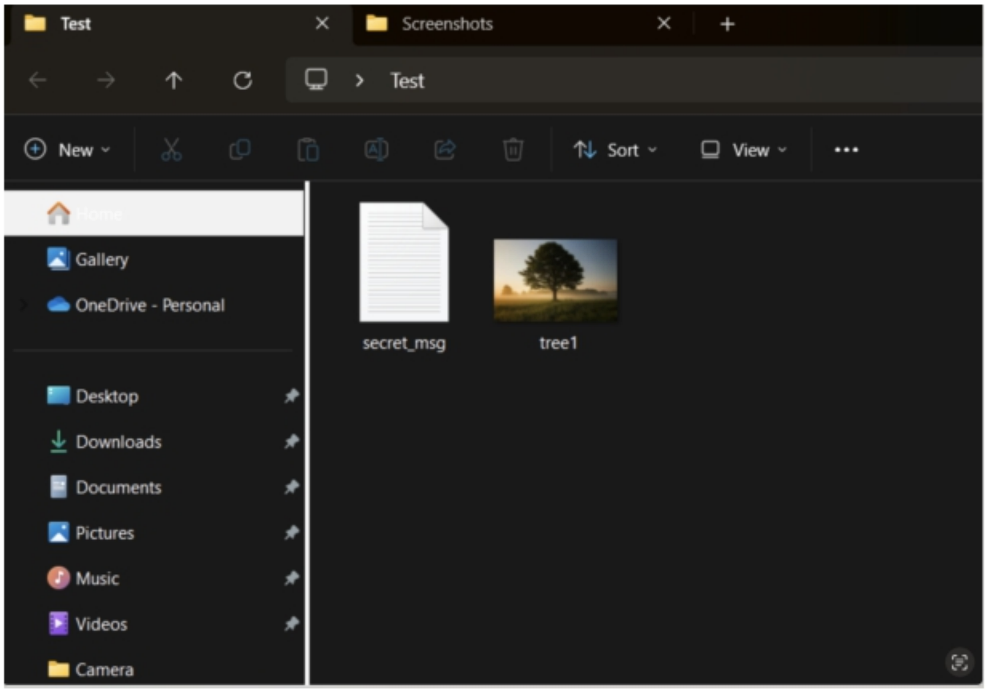
Output: Secret Message

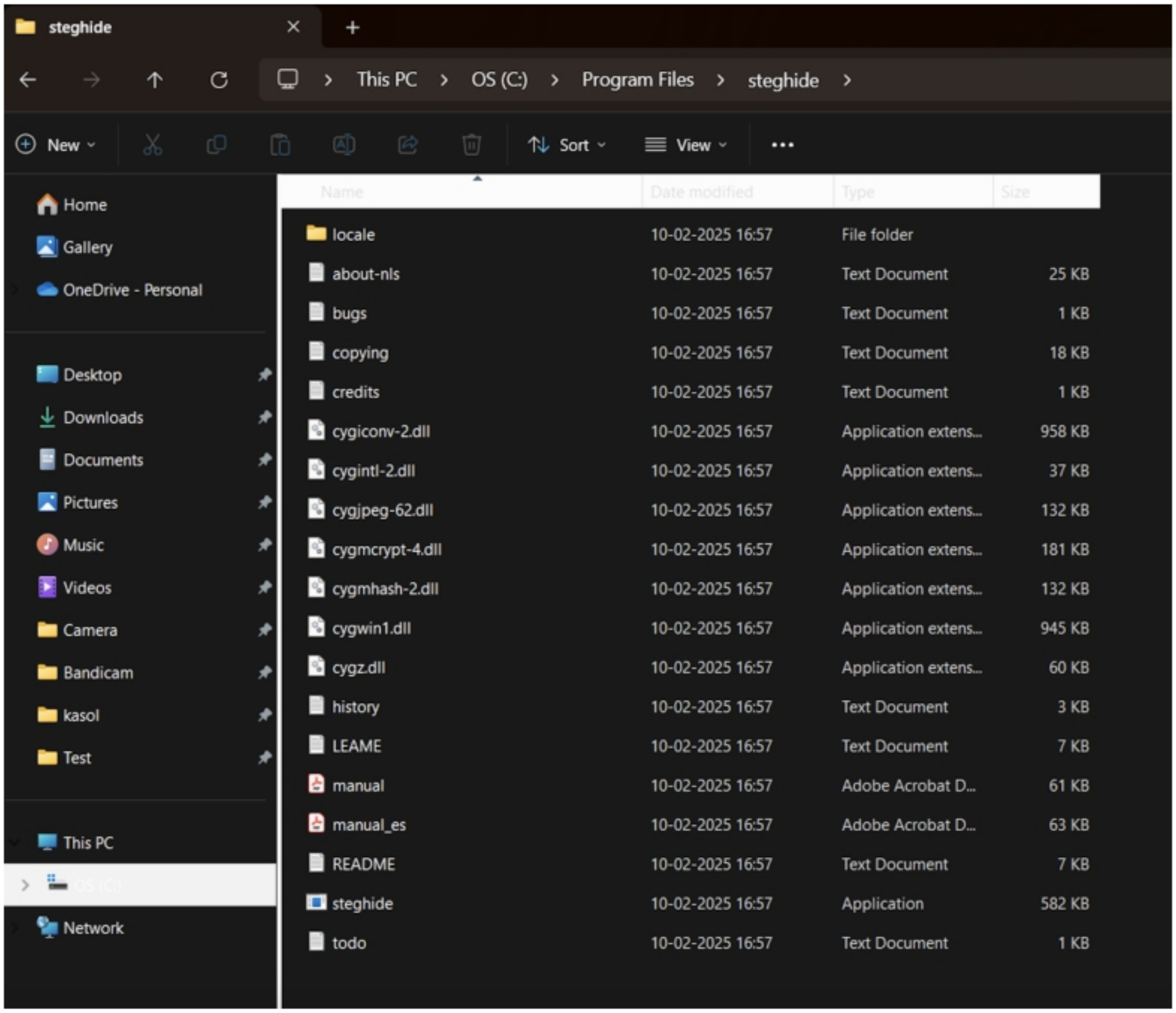
1. Load the Stego Image.
2. Extract the LSB of each pixel.
3. Convert the extracted binary values to text.
4. Display the Secret Message.



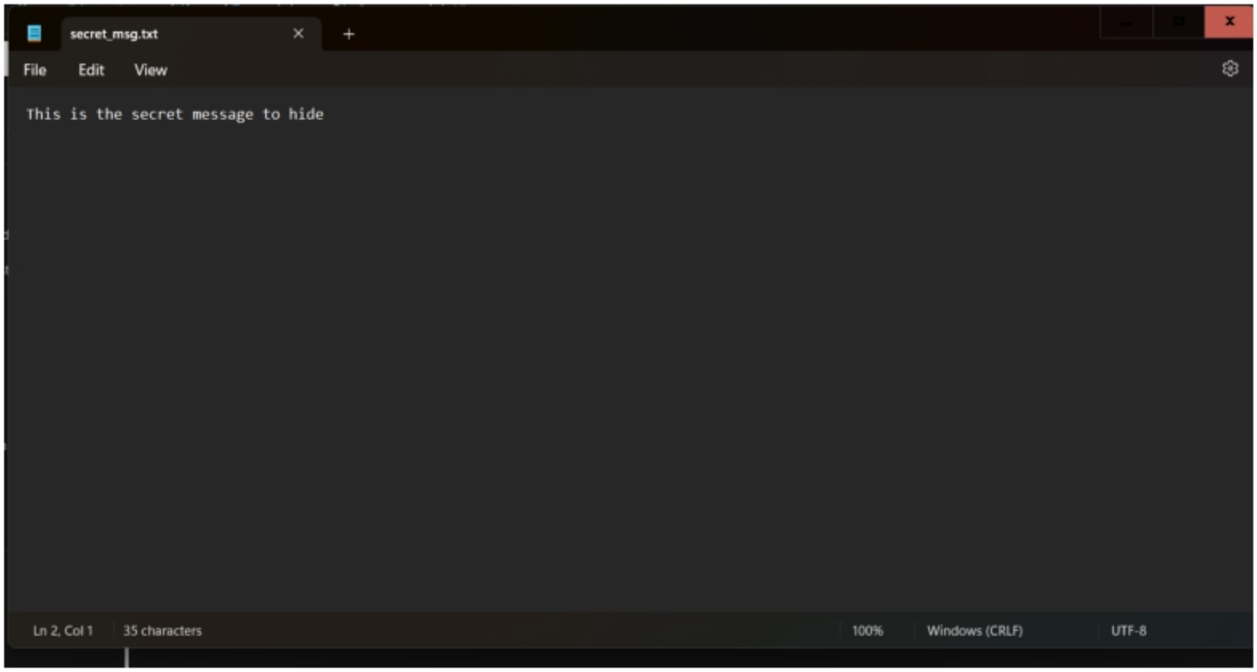
Output:


Edit
tree1.jpg
3000 x 2000
640.6 KB
31%

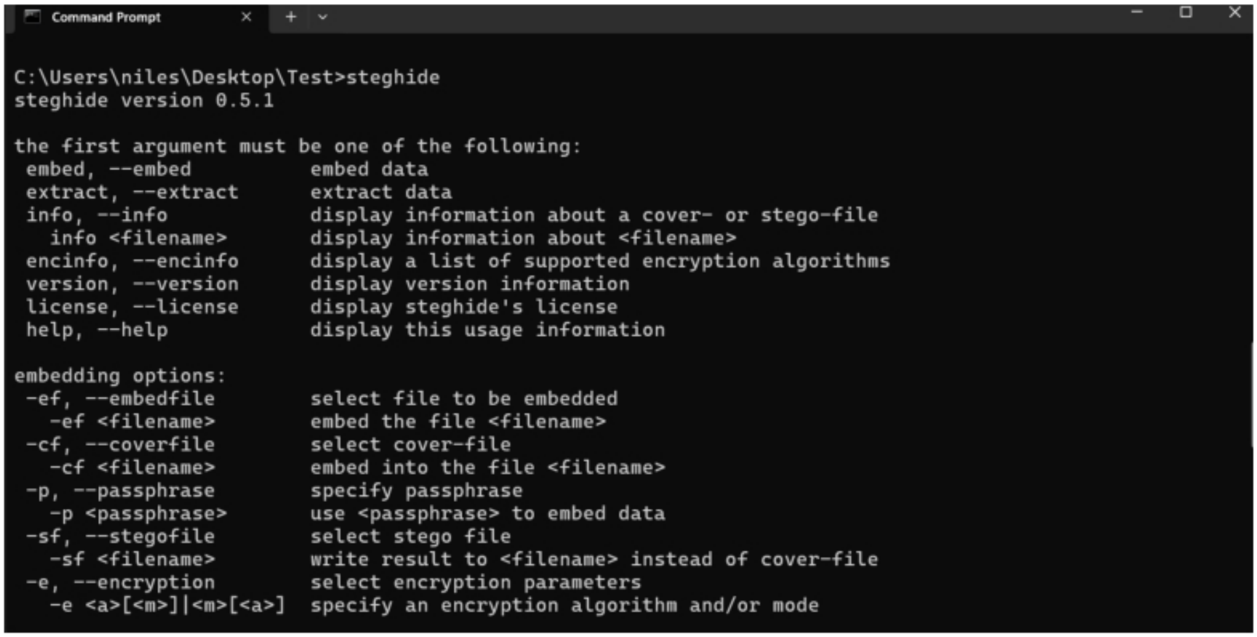

Test
Screenshots
Test
New
Sort
View
Home
Gallery
OneDrive - Personal
Desktop
Downloads
Documents
Pictures
Music
Videos
Camera
secret_msg
tree1



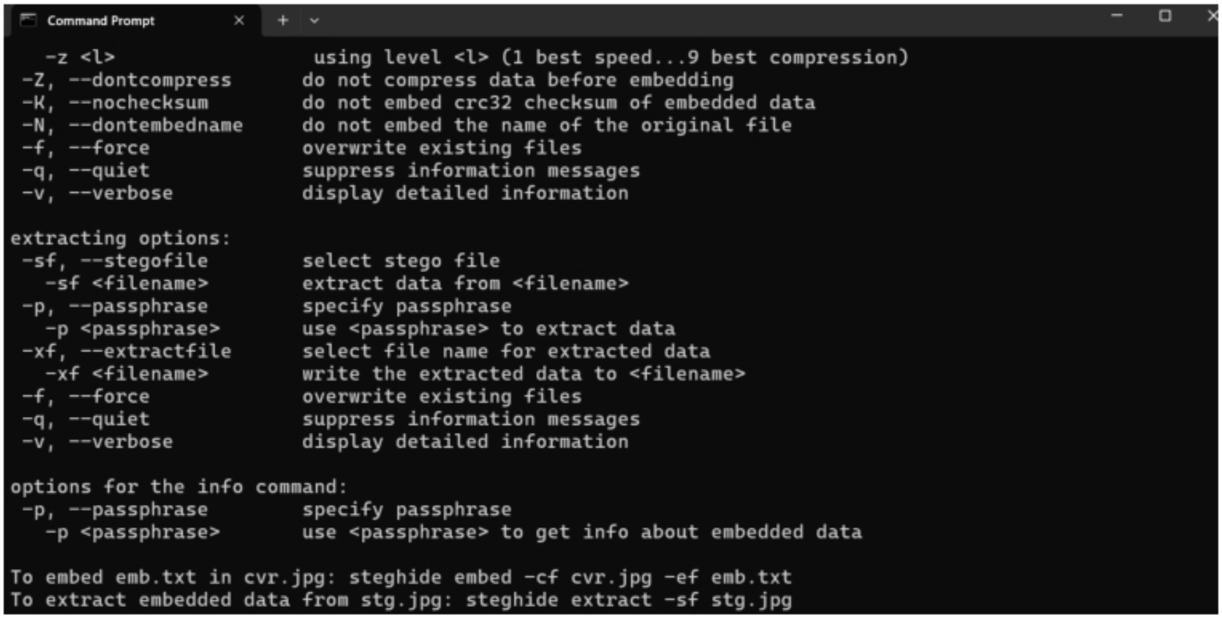
Name	Date modified	Type	Size
locale	10-02-2025 16:57	File folder	
about-nls	10-02-2025 16:57	Text Document	25 KB
bugs	10-02-2025 16:57	Text Document	1 KB
copying	10-02-2025 16:57	Text Document	18 KB
credits	10-02-2025 16:57	Text Document	1 KB
cygiconv-2.dll	10-02-2025 16:57	Application extens...	958 KB
cygintl-2.dll	10-02-2025 16:57	Application extens...	37 KB
cygjpeg-62.dll	10-02-2025 16:57	Application extens...	132 KB
cygmcrypt-4.dll	10-02-2025 16:57	Application extens...	181 KB
cygmhash-2.dll	10-02-2025 16:57	Application extens...	132 KB
cygwin1.dll	10-02-2025 16:57	Application extens...	945 KB
cygz.dll	10-02-2025 16:57	Application extens...	60 KB
history	10-02-2025 16:57	Text Document	3 KB
LEAME	10-02-2025 16:57	Text Document	7 KB
manual	10-02-2025 16:57	Adobe Acrobat D...	61 KB
manual_es	10-02-2025 16:57	Adobe Acrobat D...	63 KB
README	10-02-2025 16:57	Text Document	7 KB
steghide	10-02-2025 16:57	Application	582 KB
todo	10-02-2025 16:57	Text Document	1 KB



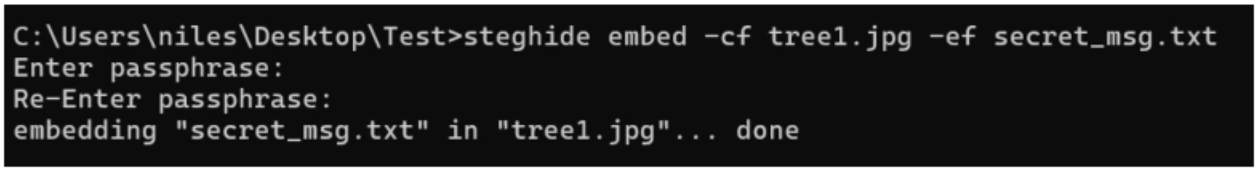
```
This is the secret message to hide
```



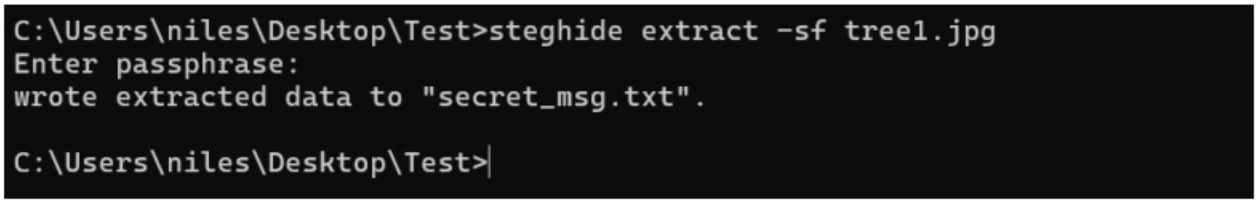
```
C:\Users\niles\Desktop\Test>steghide
steghide version 0.5.1

the first argument must be one of the following:
 embed, --embed          embed data
 extract, --extract      extract data
 info, --info            display information about a cover- or stego-file
   info <filename>       display information about <filename>
 encinfo, --encinfo      display a list of supported encryption algorithms
 version, --version      display version information
 license, --license      display steghide's license
 help, --help            display this usage information

embedding options:
 -ef, --embedfile        select file to be embedded
   -ef <filename>        embed the file <filename>
 -cf, --coverfile        select cover-file
   -cf <filename>        embed into the file <filename>
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to embed data
 -sf, --stegofile        select stego file
   -sf <filename>        write result to <filename> instead of cover-file
 -e, --encryption        select encryption parameters
   -e <a>[<m>]|<m>[<a>]  specify an encryption algorithm and/or mode
```



```
   -z <l>                 using level <l> (1 best speed...9 best compression)
 -Z, --dontcompress      do not compress data before embedding
 -K, --nochecksum        do not embed crc32 checksum of embedded data
 -N, --dontembedname     do not embed the name of the original file
 -f, --force             overwrite existing files
 -q, --quiet             suppress information messages
 -v, --verbose           display detailed information

extracting options:
 -sf, --stegofile        select stego file
   -sf <filename>        extract data from <filename>
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to extract data
 -xf, --extractfile      select file name for extracted data
   -xf <filename>        write the extracted data to <filename>
 -f, --force             overwrite existing files
 -q, --quiet             suppress information messages
 -v, --verbose           display detailed information

options for the info command:
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to get info about embedded data

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
```



```
C:\Users\niles\Desktop\Test>steghide embed -cf tree1.jpg -ef secret_msg.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret_msg.txt" in "tree1.jpg"... done
```



```
C:\Users\niles\Desktop\Test>steghide extract -sf tree1.jpg
Enter passphrase:
wrote extracted data to "secret_msg.txt".

C:\Users\niles\Desktop\Test>
```

Result/Discussion:

· The study and implementation of various steganographic tools demonstrate how information can be concealed in digital media effectively.

· Tools like Steghide and OpenPuff provide strong encryption options, while simpler tools like Stegosuite focus on basic text embedding.

· The efficiency of these tools depends on factors such as security level, ease of use, and file size limitations.

Limitations:

1. **Detection Risk:** Advanced forensic tools can sometimes detect hidden messages.
2. **File Distortion:** Some methods slightly alter the carrier file, which may be noticeable.
3. **Storage Limitations:** The amount of data that can be hidden depends on the carrier file’s size.
4. **Security Concerns:** Without encryption, hidden data may still be extracted if detected.

Applications:

1. **Secure Communication:** Used for sending confidential messages.
2. **Digital Watermarking:** Protects intellectual property by embedding ownership details.
3. **Forensic & Intelligence:** Used in cyber forensics and espionage.
4. **Stealth Malware & Cybersecurity:** Some malware hides itself using steganography.

References/Research Papers: (In IEEE format)

N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32-44, 2003.

A. Cheddad, J. Condell, K. Curran, and P. McKevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727-752, 2010.

J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," *Proc. 5th Information Hiding Workshop*, 2002, pp. 310-323.

https://www.youtube.com/watch?v=xepNoHgNj0w

Conclusion:

Steganography allows secure data hiding within digital files using tools like Steghide, OpenPuff, and Xiao Steganography. While useful for security and copyright protection, it also raises concerns about cybercrime. Ongoing research is essential to improve detection and security.