Kashish Srivastava
500067405
R134218079

LAB EXPERIMENT 4

STEGOSUITE OR STEGHIDE TOOL USING KALI

Steghide is a steganography program that is able to hide data in various kinds of image- and audio-files. The color- respectivly sample-frequencies are not changed thus making the embedding resistant against first-order statistical tests.

Steps:

1. **Install Steghide tool in Kali Linux.**

   Command used : **apt-get install steghide**

```
root@kali:~/DF2# cd Exp4-Steghide/
root@kali:~/DF2/Exp4-Steghide# 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
   -e none                 do not encrypt data before embedding
 -z, --compress            compress data before embedding (default)
   -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
```

2. We create a text file named as *"secret_file.txt"* and write some secret message in it, which has to be hidden inside the image.

Command used: **nano secret_file.txt**

```
root@kali:~/DF2/Exp4-Steghide# nano secret_file.txt
root@kali:~/DF2/Exp4-Steghide# cat secret_file.txt
"This is my secret file"
-Kashish Srivastava 079
```

Also have an image file **"open_the_lock"** in which the message of our secret file has to be embedded.

image name: open_the_lock.jpeg

3. Now we will perform the embedding of the secret message in the image file.

Command used: **steghide embed -cf open_the_lock.jpeg -ef secret_file.txt**

embed : We use the embed command if we want to embed secret data in a cover file.
-cf : Specify the cover file that will be used to embed data. The cover file must be in one of the following formats: AU, BMP, JPEG or WAV. The file-format will be detected automatically based on header information
-ef : Specify the file that will be embedded (the file that contains the secret message). Note that steghide embeds the original file name in the stego file.

```
root@kali:~/DF2/Exp4-Steghide# ls
open_the_lock.jpeg  secret_file.txt
root@kali:~/DF2/Exp4-Steghide# steghide embed -cf open_the_lock.jpeg -ef secret_file.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret_file.txt" in "open_the_lock.jpeg"... done
root@kali:~/DF2/Exp4-Steghide#
```

Providing the passphrase as : "KEY"

We see that the embedding of the secret file has been done to the image.

4. We will check the info of the open_the_lock.jpeg image.

Command used: **steghide info open_the_lock.jpeg**

```
root@kali:~/DF2/Exp4-Steghide# steghide info open_the_lock.jpeg
"open_the_lock.jpeg":
  format: jpeg
  capacity: 323.0 Byte
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "secret_file.txt":
    size: 49.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

After printing some general information about the stego file (format, capacity) we will be asked if steghide should try to get information about the embedded data. If we answer with yes you have to supply a passphrase. Steghide will then try to extract the embedded data with that passphrase and - if it succeeds - print some information about it.

5. **We will perform extraction of the hidden message from the image file.**
<<changed the directory to prevent the overwriting in the same txt file>>

Command used: **steghide extract -sf open_the_lock.jpeg**

Specifying the passphrase as "KEY"

Command used: **cat secret_file.txt**

We get the hidden message extracted from the stego image file.
"This is my secret file"
-Kashish Srivastava 079

```
root@kali:~/DF2# ls
Exp4-Steghide  help.txt  message.txt  openstego_0.7.4-1_amd64.deb  openstego_0.7.4-1_i386.deb  open_the_lock.jpeg
root@kali:~/DF2# steghide extract -sf open_the_lock.jpeg
Enter passphrase:
wrote extracted data to "secret_file.txt".
root@kali:~/DF2# ls
Exp4-Steghide  help.txt  message.txt  openstego_0.7.4-1_amd64.deb  openstego_0.7.4-1_i386.deb  open_the_lock.jpeg  secret_file.txt
root@kali:~/DF2# cat secret_file.txt
"This is my secret file"
-Kashish Srivastava 079
root@kali:~/DF2#
```

Basic Commands Used -

```
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>
```