NAME

steghide − a steganography program

SYNOPSIS

steghide *command* [*arguments*]

DESCRIPTION

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.

Features include the compression of the embedded data, encryption of the embedded data and automatic integrity checking using a checksum. The JPEG, BMP, WAV and AU file formats are supported for use as cover file. There are no restrictions on the format of the secret data.

Steghide uses a graph-theoretic approach to steganography. You do not need to know anything about graph theory to use steghide and you can safely skip the rest of this paragraph if you are not interested in the technical details. The embedding algorithm roughly works as follows: At first, the secret data is compressed and encrypted. Then a sequence of postions of pixels in the cover file is created based on a pseudo-random number generator initialized with the passphrase (the secret data will be embedded in the pixels at these positions). Of these positions those that do not need to be changed (because they already contain the correct value by chance) are sorted out. Then a graph-theoretic matching algorithm finds pairs of positions such that exchanging their values has the effect of embedding the corresponding part of the secret data. If the algorithm cannot find any more such pairs all exchanges are actually performed. The pixels at the remaining positions (the positions that are not part of such a pair) are also modified to contain the embedded data (but this is done by overwriting them, not by exchanging them with other pixels). The fact that (most of) the embedding is done by exchanging pixel values implies that the first-order statistics (i.e. the number of times a color occurs in the picture) is not changed. For audio files the algorithm is the same, except that audio samples are used instead of pixels.

The default encryption algorithm is Rijndael with a key size of 128 bits (which is AES − the advanced encryption standard) in the cipher block chaining mode. If you do not trust this combination for whatever reason feel free to choose another algorithm/mode combination (information about all possible algorithms and modes is displayed by the **encinfo** command). The checksum is calculated using the CRC32 algorithm.

COMMANDS

In this section the commands for steghide are listed. The first argument must always be one of these commands. You can supply additional arguments to the **embed**, **extract** and **info** commands. The other commands to not take any arguments.

embed, --embed
Embed secret data in a cover file thereby creating a stego file.

extract, --extract
Extract secret data from a stego file.

info, --info
Display information about a cover or stego file.

encinfo, --encinfo
Display a list of encryption algorithms and modes that can be used. No arguments required.

version, --version
Display short version information. No arguments required.

license, --license
Display steghide's license. No arguments required.

help, --help
Display a help screen. No arguments required.

EMBEDDING

You should use the **embed** command if you want to embed secret data in a cover file. The following arguments can be used with the **embed** command:

-ef, --embedfile *filename*
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. When extracting data (see below) the default behaviour is to save the embedded file into the current directory under its original name. If this argument is omitted or *filename* is **-**, steghide will read the secret data from standard input.

-cf, --coverfile *filename*
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 (the extension is not relevant). If this argument is omitted or *filename* is **-**, steghide will read the cover file from standard input.

-sf, --stegofile *filename*
Specify the name for the stego file that will be created. If this argument is omitted when calling steghide with the **embed** command, then the modifications to embed the secret data will be made directly to the cover file without saving it under a new name.

-e, --encryption *algo* [*mode*] | *mode* [*algo*]
Specify encryption parameters. This option must be followed by one or two strings that identify an encryption algorithm and/or mode. You can get the names of all available algorithms and supported modes with the **encinfo** command. The default encryption is **rijndael-128** (AES) in the **cbc** mode. If you do not want to use any encryption, use **-e none**.

-z, --compress *level*
Specify the compression level. The compression level can be any number in 1...9 where 1 means best speed and 9 means best compression.

-Z, --dontcompress
Do not compress the secret data before embedding it.

-K, --nochecksum
Do not embed a CRC32 checksum. You can use this if the secret data already contains some type of checksum or if you do not want to embed those extra 32 bits needed for the checksum.

-N, --dontembedname
Do not embed the file name of the secret file. If this option is used, the extractor needs to specify a filename to tell steghide where to write the embedded data.

EXTRACTING

If you have received a file that contains a message that has been embedded with steghide, use the **extract** command to extract it. The following arguments can be used with this command.

-sf, --stegofile *filename*

Specify the stego file (the file that contains embedded data). If this argument is omitted or *filename* is **-**, steghide will read a stego file from standard input.

-xf, --extractfile *filename*

Create a file with the name *filename* and write the data that is embedded in the stego file to it. This option overrides the filename that is embedded int the stego file. If this argument is omitted, the embedded data will be saved to the current directory under its original name.

GETTING INFORMATION ABOUT A COVER/STEGO FILE

You can use the **info** command to get some information about a cover or stego file (for example the capacity). You might want to use this if you have received a file and you are not sure if it contains an embedded message or if you consider using a certain file as cover file and want to find out its capacity.

The command line **steghide info** <*filename*> will print information about <*filename*> and then ask you if you would like to get information about data that is embedded in that file. If you answer with yes you have to supply the passphrase that was used to embed the data in that file.

You can also supply the **-p, --passphrase** argument (see below) to the **info** command which has the effect that steghide will automatically try to get information about the data that has been embedded using the given passphrase.

COMMON OPTIONS

The following options can be used with all commands (where it makes sense).

-p, --passphrase

Use the string following this argument as the passphrase. If your passphrase contains whitespace, you have to enclose it in quotes, for example: **-p "a very long passphrase"**.

-v, --verbose

Display detailed information about the status of the embedding or extracting process.

-q, --quiet

Supress information messages.

-f, --force

Always overwrite existing files.

FILE NAME OPTIONS

All file name arguments (**-cf**, **-ef**, **-sf**, **-xf**) also accept **-** as a filename which makes steghide use standard input or standard output (whichever makes sense). Omitting the corresponding file name argument will have the same effect as using **-** with two exceptions: If **-sf** is omitted for the embed command, then the modifications will be done directly in the cover file. If **-xf** is omitted for extraction, then the embedded data will be saved under the file name that is embedded in the stego file. So when you want to be sure that standard input/output is used, use **-** as filename.

EXAMPLES

The basic usage is as follows:

```
$ steghide embed -cf picture.jpg -ef secret.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret.txt" in "picture.jpg"... done
```

This command will embed the file secret.txt in the cover file picture.jpg.

After you have embedded your secret data as shown above you can send the file picture.jpg to the person who should receive the secret message. The receiver has to use steghide in the following way:

```
$ steghide extract -sf picture.jpg
Enter passphrase:
wrote extracted data to "secret.txt".
```

If the supplied passphrase is correct, the contents of the original file secret.txt will be extracted from the stego file picture.jpg and saved in the current directory.

If you have received a file that contains embedded data and you want to get some information about it before extracting it, use the info command:

```
$ steghide info received_file.wav
"received_file.wav":
  format: wave audio, PCM encoding
  capacity: 3.5 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "secret.txt":
    size: 1.6 KB
    encrypted: rijndael-128, cbc
    compressed: yes
```

After printing some general information about the stego file (format, capacity) you will be asked if steghide should try to get information about the embedded data. If you 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.

RETURN VALUE

Steghide returns 0 on success and 1 if a failure occured and it had to terminate before completion of the requested operation. Warnings do not have an effect on the return value.

AUTHOR

Stefan Hetzl <shetzl@chello.at>