
Stegano Documentation

Release 0.9.4

Cédric Bonhomme

May 23, 2022

Contents

1	Requirements	2
2	Tutorial	2
2.1	Installation	2
2.2	Using Stegano as a Python module	2
2.3	Using Stegano in command line	4
2.4	Steganalysis	8
3	License	8
4	Donation	9
5	Contact	9

(<https://builds.sr.ht/~cedric/stegano>) **Stegano** (<https://sr.ht/~cedric/stegano>) is a pure Python **steganography** (<http://en.wikipedia.org/wiki/Steganography>) module.

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. Consequently, functions provided by Stegano only hide messages, without encryption. Steganography is often used with cryptography.

Stegano implements these methods of hiding:

- using the red portion of a pixel to hide ASCII messages;
- using the **Least Significant Bit** (http://en.wikipedia.org/wiki/Least_significant_bit) (LSB) technique;
- using the LSB technique with sets based on generators (Sieve for Eratosthenes, Fermat, Mersenne numbers, etc.);
- using the description field of the image (JPEG and TIFF).

Moreover some methods of **steganalysis** (<http://en.wikipedia.org/wiki/Steganalysis>) are provided:

- steganalysis of LSB encoding in color images;
- statistical steganalysis.

You can also use Stegano through this [Web service](https://stegano-web.herokuapp.com) (<https://stegano-web.herokuapp.com>). Not all functionalities of Stegano are covered.

1 Requirements

- Python (<https://www.python.org>) 3;
- Pillow (<https://pypi.python.org/pypi/Pillow>);
- piexif (<https://pypi.python.org/pypi/piexif>).

2 Tutorial

2.1 Installation

```
$ poetry install Stegano
```

You will be able to use Stegano in your Python programs or as a command line tool.

If you want to retrieve the source code (with the unit tests):

```
$ git clone https://git.sr.ht/~cedric/stegano
```

(<https://builds.sr.ht/~cedric/stegano>)

2.2 Using Stegano as a Python module

You can find more examples in the [unit tests directory](https://git.sr.ht/~cedric/stegano/tree/master/tests) (<https://git.sr.ht/~cedric/stegano/tree/master/tests>).

LSB method

```
Python 3.10.0 (default, Oct 17 2021, 09:02:57) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from stegano import lsb
>>> secret = lsb.hide("./tests/sample-files/Lenna.png", "Hello world!")
>>> secret.save("./Lenna-secret.png")
>>> print(lsb.reveal("./Lenna-secret.png"))
Hello world!
```

LSB method with sets

Sets are used in order to select the pixels where the message will be hidden.

```
Python 3.10.0 (default, Oct 17 2021, 09:02:57) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from stegano import lsbset
>>> from stegano.lsbset import generators

# Hide a secret with the Sieve of Eratosthenes
```

(continues on next page)

```

>>> secret_message = "Hello World!"
>>> secret_image = lsbset.hide("./tests/sample-files/Lenna.png",
                               secret_message,
                               generators.eratosthenes())
>>> secret_image.save("./image.png")

# Try to decode with another generator
>>> message = lsbset.reveal("./image.png", generators.fibonacci())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/cedric/projects/Stegano/stegano/lsbset/lsbset.py", line 111, in reveal
    for color in img_list[generated_number]:
IndexError: list index out of range

# Decode with Eratosthenes
>>> message = lsbset.reveal("./image.png", generators.eratosthenes())
>>> message
'Hello World!'

>>> # Generators available
>>> import inspect
>>> all_generators = inspect.getmembers(generators, inspect.isfunction)
>>> for generator in all_generators:
...     print(generator[0], generator[1].__doc__)
...
Dead_Man_Walking None
OEIS_A000217
    http://oeis.org/A000217
    Triangular numbers:  $a(n) = C(n+1,2) = n(n+1)/2 = 0+1+2+\dots+n$ .

ackermann
    Ackermann number.

carmichael None
eratosthenes
    Generate the prime numbers with the sieve of Eratosthenes.

eratosthenes_composite
    Generate the composite numbers with the sieve of Eratosthenes.

fermat
    Generate the n-th Fermat Number.

fibonacci
    A generator for Fibonacci numbers, goes to next number in series on each call.
    This generator start at 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610,
    ↪987, 1597, 2584, 4181, 6765, 10946, ...
    See: http://oeis.org/A000045

identity
     $f(x) = x$ 

log_gen
    Logarithmic generator.

mersenne
    Generate  $2^n - 1$ .

```

(continued from previous page)

```
syracuse
    Generate the sequence of Syracuse.

shi_tomashi Shi-Tomachi corner generator of the given points
    https://docs.opencv.org/4.x/d4/d8c/tutorial_py_shi_tomasi.html

triangular_numbers Triangular numbers:  $a(n) = C(n+1,2) = n(n+1)/2 = 0+1+2+\dots+n$ .
    http://oeis.org/A000217
```

Description field of the image

For JPEG and TIFF images.

```
Python 3.10.0 (default, Oct 17 2021, 09:02:57) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from stegano import exifHeader
>>> secret = exifHeader.hide("./tests/sample-files/20160505T130442.jpg",
                             "./image.jpg", secret_message="Hello world!")
>>> print(exifHeader.reveal("./image.jpg"))
```

2.3 Using Stegano in command line

The command `stegano-lsb`

Hide and reveal a message with the LSB method.

Display help

```
$ stegano-lsb --help
usage: stegano-lsb [-h] {hide,reveal} ...

positional arguments:
  {hide,reveal}  sub-command help
  hide           hide help
  reveal         reveal help

optional arguments:
  -h, --help      show this help message and exit
```

```
$ stegano-lsb hide --help
usage: stegano-lsb hide [-h] -i INPUT_IMAGE_FILE [-e {UTF-8,UTF-32LE}]
                        (-m SECRET_MESSAGE | -f SECRET_FILE) -o
                        OUTPUT_IMAGE_FILE

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT_IMAGE_FILE, --input INPUT_IMAGE_FILE
                        Input image file.
  -e {UTF-8,UTF-32LE}, --encoding {UTF-8,UTF-32LE}
                        Specify the encoding of the message to hide. UTF-8
```

(continues on next page)

(continued from previous page)

```
(default) or UTF-32LE.
-m SECRET_MESSAGE Your secret message to hide (non binary).
-f SECRET_FILE    Your secret to hide (Text or any binary file).
-o OUTPUT_IMAGE_FILE, --output OUTPUT_IMAGE_FILE
                  Output image containing the secret.
```

```
$ stegano-lsb reveal --help
usage: stegano-lsb reveal [-h] -i INPUT_IMAGE_FILE [-e {UTF-8,UTF-32LE}]
                        [-o SECRET_BINARY]

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT_IMAGE_FILE, --input INPUT_IMAGE_FILE
                        Input image file.
  -e {UTF-8,UTF-32LE}, --encoding {UTF-8,UTF-32LE}
                        Specify the encoding of the message to reveal. UTF-8
                        (default) or UTF-32LE.
  -o SECRET_BINARY      Output for the binary secret (Text or any binary
                        file).
```

Hide and reveal a text message

```
$ stegano-lsb hide -i ./tests/sample-files/Lenna.png -m 'Hello World!' -o ./Lenna_enc.
↪png
$ stegano-lsb reveal -i ./Lenna_enc.png
Hello World!
```

Specify an encoding

```
$ stegano-lsb hide -i ./tests/sample-files/Lenna.png -m 'I love and .' -e UTF-32LE -
↪o ./Lenna_enc.png
$ stegano-lsb reveal -i ./Lenna_enc.png
I love and .
```

The default encoding is UTF-8.

Hide and reveal a binary file

```
$ wget http://www.gnu.org/music/free-software-song.ogg
$ stegano-lsb hide -i ./tests/sample-files/Montenach.png -f ./free-software-song.ogg -
↪o ./Montenach_enc.png
$ rm free-software-song.ogg
$ stegano-lsb reveal -i ./Montenach_enc.png -o ./song.ogg
```

The command `stegano-lsb-set`

Sets are used in order to select the pixels where the message will be hidden.

Hide and reveal a text message

```
# Hide the message with the Sieve of Eratosthenes
$ stegano-lsb-set hide -i ./tests/sample-files/Montenach.png --generator eratosthenes
↳ -m 'Joyeux Noël!' -o ./surprise.png

# Try to reveal with Mersenne numbers
$ stegano-lsb-set reveal --generator mersenne -i ./surprise.png

# Try to reveal with fermat numbers
$ stegano-lsb-set reveal --generator fermat -i ./surprise.png

# Try to reveal with carmichael numbers
$ stegano-lsb-set reveal --generator carmichael -i ./surprise.png

# Try to reveal with Sieve of Eratosthenes
$ stegano-lsb-set reveal --generator eratosthenes -i ./surprise.png
```

An other example:

```
# Hide the message - LSB with a set defined by the identity function (f(x) = x).
stegano-lsb-set hide -i ./tests/sample-files/Montenach.png --generator identity -m 'I
↳ like steganography.' -o ./enc-identity.png

# Hide the message - LSB only.
stegano-lsb hide -i ./tests/sample-files/Montenach.png -m 'I like steganography.' -o .
↳ /enc.png

# Check if the two generated files are the same.
shasum ./enc-identity.png ./enc.png

# The output of lsb is given to lsb-set.
stegano-lsb-set reveal -i ./enc.png --generator identity

# The output of lsb-set is given to lsb.
stegano-lsb reveal -i ./enc-identity.png
```

Sometimes it can be useful to skip the first values of a set. For example if you want to hide several messages or because due to the selected generator (Fibonacci starts with 0, 1, 1, etc.). Or maybe you just want to add more complexity. In this case, simply use the optional arguments `--shift`:

```
stegano-lsb-set reveal -i ./tests/sample-files/Lenna.png --generator fibonacci --
↳ shift 7
```

List all available generators

```
$ stegano-lsb-set list-generators
Generator id:
  ackermann
Description:
  Ackermann number.

Generator id:
  ackermann_naive
Description:
```

(continues on next page)

Ackermann number.

Generator id:
carmichael

Description:
Composite numbers n such that $a^{(n-1)} \equiv 1 \pmod{n}$ **for** every a coprime to n .
<https://oeis.org/A002997>

Generator id:
composite

Description:
Generate the composite numbers using the sieve of Eratosthenes.
<https://oeis.org/A002808>

Generator id:
eratosthenes

Description:
Generate the prime numbers with the sieve of Eratosthenes.
<https://oeis.org/A000040>

Generator id:
fermat

Description:
Generate the n -th Fermat Number.
<https://oeis.org/A000215>

Generator id:
fibonacci

Description:
Generate the sequence of Fibonacci.
<https://oeis.org/A000045>

Generator id:
identity

Description:
 $f(x) = x$

Generator id:
log_gen

Description:
Logarithmic generator.

Generator id:
mersenne

Description:
Generate $2^p - 1$, where p is prime.
<https://oeis.org/A001348>

Generator id:
triangular_numbers

Description:
Triangular numbers: $a(n) = C(n+1, 2) = n(n+1)/2 = 0+1+2+\dots+n$.
<http://oeis.org/A000217>

The command `stegano-red`

Hide and reveal a text message with the red portion of a pixel.

Display help

```
$ stegano-red hide --help
usage: stegano-red hide [-h] [-i INPUT_IMAGE_FILE] [-m SECRET_MESSAGE]
                        [-o OUTPUT_IMAGE_FILE]

optional arguments:
-h, --help            show this help message and exit
-i INPUT_IMAGE_FILE, --input INPUT_IMAGE_FILE
                        Image file
-m SECRET_MESSAGE     Your secret message to hide (non binary).
-o OUTPUT_IMAGE_FILE, --output OUTPUT_IMAGE_FILE
                        Image file
```

Hide and reveal a text message

```
$ stegano-red hide -i ./tests/sample-files/Lenna.png -m 'Basic steganography_
↳ technique.' -o ~/Lenna1.png
$ stegano-red reveal -i ~/Lenna1.png
Basic steganography technique.
```

2.4 Steganalysis

Parity

```
# Hide the message with Sieve of Eratosthenes
stegano-lsb-set hide -i ./tests/sample-files/20160505T130442.jpg -o ./surprise.png --
↳ generator eratosthenes -m 'Very important message.'

# Steganalysis of the original photo
stegano-steganalysis-parity -i ./tests/sample-files/20160505T130442.jpg -o ./surprise_
↳ st_original.png

# Steganalysis of the secret photo
stegano-steganalysis-parity -i ./surprise.png -o ./surprise_st_secret.png

# Reveal with Sieve of Eratosthenes
stegano-lsb-set reveal -i ./surprise.png --generator eratosthenes
```

You can have a look at the [unit tests](https://git.sr.ht/~cedric/stegano/tree/master/tests) (<https://git.sr.ht/~cedric/stegano/tree/master/tests>).

3 License

Stegano (<https://sr.ht/~cedric/stegano>) is under GPL v3 license.

4 Donation

If you wish and if you like Stegano, you can [donate](https://github.com/sponsors/cedricbonhomme) (https://github.com/sponsors/cedricbonhomme).

5 Contact

My home page (<https://www.cedricbonhomme.org>)