

python图片隐写_Python隐写工具，用来在图像中隐藏图像或文本

[weixin_39544101](#) 于 2020-12-09 19:46:36 发布 96 收藏

文章标签: [python图片隐写](#)

Матрёшка

About

Matroshka is a command-line steganography tool written in pure Python. You can use it to hide and encrypt images or text in the least significant bits of pixels in an image.

Encryption

The encryption uses HMAC-SHA256 to authenticate the hidden data. Therefore the supplied MAC password is hashed with SHA-256 digest to generate the HMAC-SHA256 key. The MAC and the message data is further encrypted using the XTEA algorithm in CFB mode running 32 iterations, before being embedded in the image data. The SHA-256 hash for the XTEA key is created using the 128 high-order bits of the supplied password. A random 8 byte seed is used in the CFB 64 bit block cipher.

Decryption

The random seed is appended to the hidden secret and is used with the user supplied password to decrypt the hidden message using XTEA block cipher according to the encryption process. Further the decrypted secret is authenticated by comparing the embeded hmac hash with the HMAC-SHA256 of the extracted hidden message and the user supplied mac password.

Note

This is a fun project. Do not use this for serious encryption purposes!

Installation

You may want to install Matroschka in a virtualenv that has Pillow installed. Clone the repo and navigate to the root dir. That's it.

```
pip install Pillow
```

```
git clone git@github.com:fgrimme/Matroschka.git
```

```
cd Matroschka
```

Usage

Only losless image formats are supported. Therefore It is recommended to use PNG or BMP images to hide your secret. The secret can be either a text file with the .txt extension or an image with the .png extension and format.

Note: If your image contains transparent pixels, most likely artifacts will be visible after embedding data. This is caused by the manipulation of the least-significant bit in every pixels color channels.

Hide Secret

```
python matroschka.py -hide -m -k
```

Reveal Secret

```
python matroschka.py -open -m -k
```

Example

matroschka_medium.png gets hidden and encrypted in matroschka_big.png

```
python matroschka.py -hide -m foo -k bar resources/matroschka_medium.png resources/matroschka_big.png
```

Decrypting the image will save the extracted image in resources/secret-image.png

```
python matroschka.py -open -m foo -k bar resources/matroschka_big.png
```

License

GNU General Public License 3.0