

Obscuro

Obscuro is a collection of command-line stream encryption utilities based on libsodium. It is somewhat experimental ATM.

Building From Source (Linux)

If you have downloaded the source package, make sure you have libsodium installed. Then run make. You can also use “make install” to install it.

Usage

Obscuro

Encrypt data from stdin and output to stdout.

Usage: `obscuro [keyfile path]`

DeObscuro

Decrypt data from stdin and output to stdout.

Usage: `deobscuro [keyfile path]`

GenObKey

Generate a key for encryption.

Usage: `genobkey [keyfile path]`

oReveal

This utility decrypts a file in place. It works by reading a chunk of data, decrypting it, and writing it back to the file. This will leave a file in a corrupt state if it is interrupted. Only use this if you are really limited by storage space.

Usage: `oreveal [keyfile path] [encrypted file]`

Neat stuff you can do

Note: The Windows Command Prompt uses a boneheaded asinine pipe system. Many of these commands might work on windows, but work stupidly because of how Windows handles pipes. For instance, the command `foo | bar` will run `foo`, wait for it to exit, then pipe its output to `bar`. On a sensible system these programs would run in parallel. Additionally, the standard input and output won't work for binary data unless a program manually sets them to binary mode. Everything here assumes a Linux-based operating system running something like bash.

Quickly Compress and Encrypt a File/Directory

Note: This uses pigz, which is a fantastic utility that does gzip compression using multiple CPU cores. This is the same as using `gzip` or adding `-z` to `tar`, but much faster.

```
tar -c [file/directory path] | pigz | obscuro [key file] > archive.tar.gz.ob
```

Secure File/Directory Transfer Over a Network

Note: Assumes both machines already have the same keyfile. Also uses pigz, described above.

Send

```
tar -c [file/directory path] | pigz | obscuro [key file] | nc [ip] [port]
```

Receive

```
nc -lp [port] | deobscurso tk | unpigz | tar -x
```