

Forum:[Hack Forums](#)

Thread: [SFEDXOR 0.0.2 - Single File Encryption & Decryption - Random Keys or Manual Keys](#)

Hackers - February 20, 2024, 6:58 am

SFEDXOR

Version 0.0.2

Single File Encryption & Decryption: XOR

Encrypt Any Filetype

Decrypt Your Encrypted File

Single String Key / "Password"

Use Random Encryption Key (automatically matches byte size of file you are encrypting)

or Supply Your Own Key

Output Key to File (for use later)

Use File for Key Input (instead of string)

Like what you see? Check out the more secure version SFED-AES-256 - same functionality, same usage - More security!

<https://hackforums.net/showthread.php?tid=6264320>

Free Source Code - Just download & compile!

Code <https://github.com/LNodesL/SFEDXOR2>

ENCRYPT EXAMPLES

Code `./SFEDXOR -i <inputfile> -o <outputfile> -m encrypt [-k <keyfile>] [-ko <keyoutputfile>] [-ks <keystring>]`- <inputfile>: Path to the input file to be encrypted.

- <outputfile>: Path to save the encrypted output file.

- -m encrypt: Mode for encryption.

- -k <keyfile>: Optional. Path to a file containing the encryption key. If not provided, a key will be auto-generated.

- -ko <keyoutputfile>: Optional. Path to save the encryption key to a file.

- -ks <keystring>: Optional. Inline key string to use for encryption.

Encrypt with auto-generated key:

Code `./SFEDXOR -i input.txt -o encrypted.txt -m encrypt`

Encrypt with specified key file and save key to another file:

Code `./SFEDXOR -i input.txt -o encrypted.txt -m encrypt -k key.txt -ko key_saved.txt`

Encrypt with inline key string:

Code `./SFEDXOR -i input.txt -o encrypted.txt -m encrypt -ks mysecretpassword`

DECRYPT EXAMPLES

Code `./SFEDXOR -i <inputfile> -o <outputfile> -m decrypt [-k <keyfile>] [-ks <keystring>]`-

<inputfile>: Path to the encrypted input file.
- <outputfile>: Path to save the decrypted output file.
- -m decrypt: Mode for decryption.
- -k <keyfile>: Optional. Path to a file containing the decryption key.
- -ks <keystring>: Optional. Inline key string to use for decryption.

Decrypt with specified key file:

Code `./SFEDXOR -i encrypted.txt -o decrypted.txt -m decrypt -k key.txt`

Decrypt with inline key string:

Code `./SFEDXOR -i encrypted.txt -o decrypted.txt -m decrypt -ks mysecretpassword`

The original SFEDXOR 0.0.1 does not include input/output key files, only inline string keys, and did not have random encryption key generation:

<https://hackforums.net/showthread.php?tid=6263383>

Disclaimer:

This is not a crypter, it is a single file encryption & decryption tool. You can use it to lock down your own files for security, to decrypt later. You can also use the final program in scripts that access command line for bulk encryption/decryption. The program does not have built in folder support, only single files. This works with any type of file, but you must make sure to keep the same file extension on the output when you decrypt. The file extension used for the encrypted file doesn't matter as long as it is decrypted into the final, correct format.

XOR is two-way symmetrical encryption. If you use a key that is shorter than the file you are encrypting/decrypting, you can expect partially exposed data.

Klonopin - February 20, 2024, 7:52 am

consider using `std::random_device` along with `std::mt19937` for better random number generation instead of `srand(time(nullptr))`

```
~ std::mt19937
~ srand(time(nullptr))
```

In the XorEncryptDecrypt function, replace direct file reading/writing with buffered input/output streams to handle files efficiently in chunks.
(For dealing with massive files)

Cool project though, hoping to see more updates in the future.

Nightmare - February 20, 2024, 7:54 am

Very nice [Hackers](#) !!! <3 ?

My bro [Klonopin](#) master of the art!We posted together by the way hahaha !

Hackers - February 20, 2024, 7:56 am

(Feb 20, 2024 08:52 AM)Klonopin Wrote: [CITE](#) consider using `std::random_device` along with `std::mt19937` for better random number generation instead of `srand(time(nullptr))`

```
~ std::mt19937
~ srand(time(nullptr))
```

In the XorEncryptDecrypt function, replace direct file reading/writing with buffered input/output streams to handle files efficiently in chunks.
(For dealing with massive files)

Cool project though, hoping to see more updates in the future. This is smart for thinking about file buffering / chunks.

I will look into that if I release a 0.0.3 as well as making sure rand is more reliably random.

(Feb 20, 2024 08:54 AM)-Nightingale- Wrote: [\[CITE\]](#) Very nice [Hackers](#) !!! <3 ?Thank you! Happy to share it

Grim - March 21, 2024, 2:38 pm

very nice thanks for the share!

Hackers - March 21, 2024, 7:14 pm

(Mar 21, 2024 02:38 PM)Grim Wrote: [\[CITE\]](#) very nice thanks for the share!Glad you like it & no problem!

Let me know if you have any questions