php-encryptor Documentation

Release 1.1.0

Mar 15, 2021

CONTENTS

1 Installation 3

2 Usage 5

3 File Format 7

4 License 9

A small and simple tool to encrypt small files with with libsodium.

CHAPTER

ONE

INSTALLATION

Install for local user with composer:

```
composer global require arokettu/encryptor
```

Install for all users by downloading prebuilt phar:

```
sudo wget https://github.com/arokettu/php-encryptor/releases/latest/download/
↪encryptor.phar -O /usr/local/bin/encryptor
sudo chmod +x /usr/local/bin/encryptor
```

CHAPTER

TWO

USAGE

```
encryptor encrypt|decrypt [-o|--output=OUTPUT_FILE] [--stdout] [-k|--key=KEY]
          [-p|--password=PASSWORD] [-s|--strength=STRENGTH] [<INPUT_FILE>]
```

-o, --output=OUTPUT_FILE Output file.

--stdout Force output to stdout.

-k, --key=KEY Encrypt/decrypt data with a binary key. The key must be 32 bytes long encoded in hexadecimal.

-p, --password=PASSWORD Encrypt/decrypt data with password.

-s, --strength=STRENGTH Encryption only: Key derivation strength for password encryption. (1-3, default 2)

If no input file is specified, the tool will read from stdin.

If neither `--output` nor `--stdout` are specified:

- If data is read from stdin, output will be stdout
- On encryption: INPUT_FILE.encrypted
- On decryption: if input file is FILENAME.enctypted, then FILENAME, otherwise INPUT_FILE.decrypted

If neither key nor password are given in parameters, a password will be requested interactively

Key derivation strength sets opslimit/memlimit for Argon2id key derivation. Default level is MODERATE

Strength	Limit constants
1	INTERACTIVE
2	MODERATE
3	SENSITIVE

CHAPTER

THREE

FILE FORMAT

Encrypted file is a bencoded dictionary with the following keys:

key	value	description
_a	“sfenc”	Header
_v	1 or 2	Container version
salt	16 random bytes	Password salt. Unset if encrypted with a key
ops	integer	Argon2id opslimit. Unset if encrypted with a key (v2 only)
mem	integer	Argon2id memlimit. Unset if encrypted with a key (v2 only)
nonce	24 random bytes	Xsalsa20 nonce
payload	long binary string	Xsalsa20 + Poly1305 encrypted payload

The file is guaranteed to start with `d2:_a5:sfenc2:_v`

V1 and V2 differences:

- V2 uses Argon2id, V1 uses Argon2i
- V2 uses ops and mem form the container, V1 always uses SENSITIVE (ops=4, mem=1_073_741_824, hard-coded since 1.1)
- V1 and V2 are equal when encrypting with a key except for the version header

V1 was used during early development. If you somehow used my dev version, you can still decode your files but it may break if libsodium changes the constants.

CHAPTER

FOUR

LICENSE

The library is available as open source under the terms of the MIT License.