

Simple Encryption in PHP.

🔖 531 commits

🌿 12 branches

📦 9 releases

👥 26 contributors

📄 MIT

Branch: master ▾
















New pull request

Create new file

Upload files

Find file

Clone or download ▾

 defuse	committed on Nov 18, 2017 Merge pull request #372 from defuse/use-libsodium-readme ...	Latest commit 1eabcfcd on Nov 18, 2017
 bin	Add command line script for generating new keys	10 months ago
 dist	Don't rebuild when signing the .phar	a year ago
 docs	Merge pull request #342 from chriseskow/generate-key-cli	9 months ago
 src	Update KeyProtectedByPassword.php	4 months ago
 test	Update CryptoTest.php	9 months ago
 .gitattributes	Don't ignore dist in gitarchive as it is used to create phar	2 years ago
 .gitignore	Add signed phar to .gitignore	2 years ago
 .php_cs	cleanup outdated dirs	2 years ago
 .travis.yml	Begin Psalm integration.	10 months ago
 LICENSE	Remove BSD license comments and fix copyright attribution.	2 years ago
 README.md	Add "Other Language Support" to README.md	3 months ago
 composer.json	Merge pull request #342 from chriseskow/generate-key-cli	9 months ago
 psalm.xml	Begin Psalm integration.	10 months ago
 test.sh	update test to check for null output	2 years ago

📖 README.md

php-encryption

build passing

This is a library for encrypting data with a key or password in PHP. **It requires PHP 5.4 or newer.** The current version is v2.0.0, which is expected to remain stable and supported by its authors with security and bugfixes until at least January 1st, 2019.

The library is a joint effort between [Taylor Hornby](#) and [Scott Arciszewski](#) as well as numerous open-source contributors.

What separates this library from other PHP encryption libraries is, firstly, that it is secure. The authors used to encounter insecure PHP encryption code on a daily basis, so they created this library to bring more security to the ecosystem. Secondly, this library is "difficult to misuse." Like [libsodium](#), its API is designed to be easy to use in a secure way and hard to use in an insecure way.

Dependencies

This library requires no special dependencies except for PHP 5.4 or newer with the OpenSSL extensions enabled (this is the default). It uses [random_compat](#), which is bundled in with this library so that your users will not need to follow any special installation steps.

Getting Started

Start with the [Tutorial](#). You can find instructions for obtaining this library's code securely in the [Installing and Verifying](#) documentation.

After you've read the tutorial and got the code, refer to the formal documentation for each of the classes this library provides:

- [Crypto](#)
- [File](#)
- [Key](#)
- [KeyProtectedByPassword](#)

If you encounter difficulties, see the [FAQ](#) answers. The fixes to the most commonly-reported problems are explained there.

If you're a cryptographer and want to understand the nitty-gritty details of how this library works, look at the [Cryptography Details](#) documentation.

If you're interested in contributing to this library, see the [Internal Developer Documentation](#).

Other Language Support

This library is intended for server-side PHP software that needs to encrypt data at rest. If you are building software that needs to encrypt client-side, or building a system that requires cross-platform encryption/decryption support, we strongly recommend using [libsodium](#) instead.

Examples

If the documentation is not enough for you to understand how to use this library, then you can look at an example project that uses this library:

- [encutil](#)
- [fileencrypt](#)

Security Audit Status

This code has not been subjected to a formal, paid, security audit. However, it has received lots of review from members of the PHP security community, and the authors are experienced with cryptography. In all likelihood, you are safer using this library than almost any other encryption library for PHP.

If you use this library as a part of your business and would like to help fund a formal audit, please [contact Taylor Hornby](#).

Public Keys

The GnuPG public key used to sign releases is available in [dist/signingkey.asc](#). Its fingerprint is:

```
2FA6 1D8D 99B9 2658 6BAC  3D53 385E E055 A129 1538
```

You can verify it against Taylor Hornby's [contact page](#) and [twitter](#).