# Package 'encryptedRmd'

May 8, 2020

**Title** Encrypt Html Reports Using 'Libsodium'

**Version** 0.2.0

**Description** Create encrypted html files that are fully self contained and do not require any additional software. Using the package you can encrypt arbitrary html files and also directly create encrypted 'rmarkdown' html reports.

**License** MIT + file LICENSE

**Copyright** file inst/COPYRIGHTS

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Imports** sodium, readr, rmarkdown

**Suggests** testthat (>= 2.1.0)

**Collate** 'encrypt_html_file.R' 'encrypted_html_document.R'

**NeedsCompilation** no

**Author** Dirk Schumacher [aut, cre], Jannis R. [aut]

**Maintainer** Dirk Schumacher <mail@dirk-schumacher.net>

**Repository** CRAN

**Date/Publication** 2020-05-08 17:00:05 UTC

## R topics documented:

`encrypted_html_document` *Create an encrypted HTML document*

## Description

In addition to a standard html file this `rmarkdown` format also creates an encrypted version together with the key as two separate files.

## Usage

```
encrypted_html_document(...)
```

## Arguments

| | |
|---|---|
| `...` | all parameters are passed to `rmarkdown::html_document` |

## Details

Two files are created:

**filename.enc.html** This is the password protected file.

**filename.enc.html.key** This file contains the key with which the report was encrypted.

Warning: You are using this at your own risk. Make sure your encryption key is strong enough. For serious use cases, please also review the code of the functions. Any feedback is appreciated. This is an early package version. Please only share the key file with trusted parties

## Value

R Markdown output format to pass to `render`

## See Also

`encrypt_html_file` for more information on the encryption.

encrypt_html_file *Encrypt an html file*

## Description

This function takes an html file, encrypts the complete file using `sodium::data_encrypt` and a given key. It then injects the encrypted content into an html template that contains the `sodium` decryption code compiled to javascript. The resulting file is fully self contained as it can decrypt itself. When the user enters the correct key, the DOM of the html files gets replaced with the initially encrypted html file.

## Usage

```
encrypt_html_file(
  path,
  output_path = paste0(path, ".enc.html"),
  key = sodium::random(32L),
  message_key = TRUE,
  write_key_file = FALSE,
 output_template_path = system.file("html-template.html", package = "encryptedRmd")
)
```

## Arguments

| | |
|---|---|
| `path` | the file you want to encrypt |
| `output_path` | optional, the output path |
| `key` | optional, the encryption key |
| `message_key` | optional, print the encryption key to the console |
| `write_key_file` | optional, write a key file in the same directory |
| `output_template_path` | a path to the output template. The output template needs have the same html form elements (same ids) and the same placeholders as the default template. Everything else can be customized. |

## Details

Warning: You are using this at your own risk. Make sure your encryption key is strong enough. For serious use cases, please also review the code of the functions. Any feedback is appreciated. This is an early package version.

## Value

The key used to encrypt the file as an invisible raw vector.

## References

The package follows the same approach as the node module self-decrypting-html-page. The decryption code is based on a number of great node modules. All licenses are also bundled with each encrypted html file.

# Index