Encryptr: Easily Encrypt and Decrypt Sensitive Data with R

Cameron Fairfield

2019-Sep-13

Why we are waking up to the value of our own data

Scotsman, Aug-30, 2019

Data Governance

- Data is precious and sensitive

Data Governance

- Data is precious and sensitive
- Confidential data (when possible) should be anonymised or pseudonomysed

Data Governance

- Data is precious and sensitive
- Confidential data (when possible) should be anonymised or pseudonomysed
- Patients expect us to safeguard their data

Data Governance

- Data is precious and sensitive
- Confidential data (when possible) should be anonymised or pseudonomysed
- Patients expect us to safeguard their data
- Data can be minimised, deleted or encrypted

Research and Data

- Data governance approvals are key components of research approvals

Research and Data

- Data governance approvals are key components of research approvals
- GDPR (Europe) and HIPAA etc. (USA)

Research and Data

- Data governance approvals are key components of research approvals
- GDPR (Europe) and HIPAA etc. (USA)
- Data breaches are financially and reputationally costly

Research and Data

- Data governance approvals are key components of research approvals
- GDPR (Europe) and HIPAA etc. (USA)
- Data breaches are financially and reputationally costly
- Not all data can be removed from records

Encryptr

- Easy pseudonymisation by encryption

Encryptr

- Easy pseudonymisation by encryption
- RSA encryption with private / public key pair (asymmetric)

Encryptr

- Easy pseudonymisation by encryption
- RSA encryption with private / public key pair (asymmetric)
- Encryption of vectors, variables and files

Encryptr

- Easy pseudonymisation by encryption
- RSA encryption with private / public key pair (asymmetric)
- Encryption of vectors, variables and files
- Secure storage of confidential data (and allocation concealment / blinding)

Encryptr on CRAN / Github

```r
install.packages("encryptr") # CRAN
remotes::install_github("SurgicalInformatics/encryptr")
```

https://github.com/surgicalinformatics

```r
### <b>
library(encryptr)
### </b>
library(dplyr) # Used in presentation examples

# Encryptr comes with an example data set of GPs (Family Pl
### <b>
gp
```

```
## # A tibble: 1,212 x 12
##   organisation_co~ name  address1 address2 address3 city
##   <chr>            <chr> <chr>    <chr>    <chr>    <chr
## 1 S10002           MUIR~ LIFF RO~ MUIRHEAD <NA>     DUNI
## 2 S10017           THE ~ CRIEFF ~ KING ST~ <NA>     CRIE
## 3 S10036           ABER~ TAYBRID~ <NA>     <NA>     ABER
## 4 S10060           ABER~ TAYBRID~ <NA>     <NA>     ABER
## 5 S10106           GROV~ 129 DUN~ BROUGHT~ <NA>     DUNI
## 6 S10125           ALYT~ NEW ALY~ ALYTH    <NA>     BLAI
## # ... with 1,206 more rows, and 5 more variables: postco
## #   opendate <date>, closedate <date>, telephone <chr>,
```

Public and Private Keys

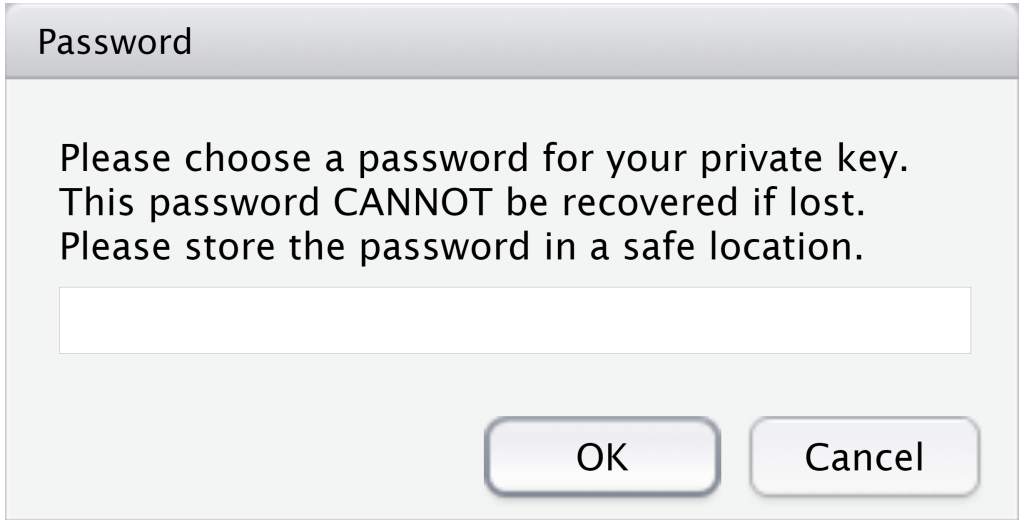
```r
### <b>
genkeys()
```

```
## Private key written with name 'id_rsa'
## Public key written with name 'id_rsa.pub'
```

```r
### </b>
```

Default values are “id_rsa” and “id_rsa.pub”

No Raw Text Password



```r
### <b>
gp_encrypt = gp %>%
  encrypt(name)
### </b>
gp_encrypt %>%
  select(organisation_code, name, address1)
```

```
## # A tibble: 1,212 x 3
##   organisation_code name
##   <chr>             <chr>
## 1 S10002            91238ee62e30d59b77a1df574f960ade46~
## 2 S10017            b901fec2d49e7464e2547bf39cfe617133~
## 3 S10036            2e61f70cc7b076505d78e3ef95b41a10cd~
## 4 S10060            8c892a195f127488047c8017fb79dc2ded~
## 5 S10106            390a88f935943d113cd10053e16a6db822~
## 6 S10125            08ce667040f79d6d97db9aa93af3ea6f9c~
## # ... with 1,206 more rows
```

```r
### <b>
gp_encrypt %>%
  slice(1:2) %>%
  decrypt(name) %>%
  ### </b>
  select(organisation_code, name, address1)
```

```
## # A tibble: 2 x 3
##   organisation_code name                    address1
##   <chr>             <chr>                   <chr>
## 1 S10002            MUIRHEAD MEDICAL CENTRE LIFF ROAD
## 2 S10017            THE BLUE PRACTICE       CRIEFF MEDIC
```

Encryptr Customisation

- Use look-up table - create object with encrypted output and ID variable on which to match

Encryptr Customisation

- Use look-up table - create object with encrypted output and ID variable on which to match
- Write a look-up file

Encryptr Customisation

- Use look-up table - create object with encrypted output and ID variable on which to match
- Write a look-up file
- Customise file names, key names, encrypt several variables

Encryptr Customisation

- Use look-up table - create object with encrypted output and ID variable on which to match
- Write a look-up file
- Customise file names, key names, encrypt several variables
- Use a publicly-available public key

Encryptr Customisation Examples

```
# Creating a lookup table with specified name and filename
### <b>
gp %>%
  encrypt(name, postcode,
          lookup = TRUE, write_lookup = TRUE,
          lookup_name = "new_lookup_name")


# Using a public key hosted at URL
gp %>%
  encrypt(name, public_key_path = "https://<some_url>/id_rs
### </b>
```

Encryptr File Encryption

```r
gp_encrypt %>% write_csv("gp_enc.csv")

### <b>
encrypt_file("gp.csv")




# Encrypted file will have suffix: `.encryptr.bin`
decrypt_file("gp.csv.encryptr.bin", file_name = "gp2.csv")
### </b>
```

Encryptr Ciphertexts Not Matchable

- Each repeat of encryption generates a unique number

Encryptr Ciphertexts Not Matchable

- Each repeat of encryption generates a unique number
- Prevents malicious, opportunistic use of public key

Encryptr Ciphertexts Not Matchable

- Each repeat of encryption generates a unique number
- Prevents malicious, opportunistic use of public key
- Alternative symmetric encryption outputs can be matched (and not always reversed)

Encryptr Ciphertexts Not Matchable

- Each repeat of encryption generates a unique number
- Prevents malicious, opportunistic use of public key
- Alternative symmetric encryption outputs can be matched (and not always reversed)
- Alternative methods need a “salt”

```r
### <b>
encrypt_vec(c("a name", "a name", "a name"))
```

```
## [1] "26d47f442588daaa3a8eeedfabe44a6937645e38b381512f25a
## [2] "170dcdd48b4f39a9ee439c8b6812108e7a3376f303a51d26826
## [3] "32f6eb6ea40f5dabfb540e5f6268c7864bf399a454074371a94
```

```r
### </b>
```

Technical Aspects of Encryptr

- Wrapper around OpenSSL (and purrr)

Technical Aspects of Encryptr

- Wrapper around OpenSSL (and purrr)
- RSA asymmetric encryption for vectors (each component in vector < 256 bytes)

Technical Aspects of Encryptr

- Wrapper around OpenSSL (and purrr)
- RSA asymmetric encryption for vectors (each component in vector < 256 bytes)
- File encryption uses AES technique with symmetric session key which is in turn encrypted by RSA public key

What Encryptr Is Useful For

- Storing confidential data in trials, cohorts, service evaluation, etc.

What Encryptr Is Useful For

- Storing confidential data in trials, cohorts, service evaluation, etc.
- Retrieving individual data if needed

What Encryptr Is Useful For

- Storing confidential data in trials, cohorts, service evaluation, etc.
- Retrieving individual data if needed
- Blinding in RCTs

What Encryptr Is Useful For

- Storing confidential data in trials, cohorts, service evaluation, etc.
- Retrieving individual data if needed
- Blinding in RCTs
- Data governance considerations

Questions