

# Package: keyringr (via r-universe)

August 26, 2024

**Type** Package

**Title** Decrypt Passwords from Gnome Keyring, Windows Data Protection API and macOS Keychain

**Version** 0.4.0

**Author** Josh Gilfillan

**Maintainer** Josh Gilfillan <joshua.i.gilfillan+keyringr@gmail.com>

**Description** Decrypts passwords stored in the Gnome Keyring, macOS Keychain and strings encrypted with the Windows Data Protection API.

**License** MIT + file LICENSE

**LazyData** TRUE

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown

**Imports** stringr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-02-17 23:55:46

## Contents

decrypt_dpapi_pw . . . . .	2
decrypt_gk_pw . . . . .	3
decrypt_kc_pw . . . . .	3
get_kc_account . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

decrypt_dpapi_pw	<i>Decrypt passwords encrypted with the Microsoft Data Protection API</i>
------------------	---

---

## Description

Decrypt passwords encrypted with the Microsoft Data Protection API

## Usage

```
decrypt_dpapi_pw(file)
```

## Arguments

file	File that holds a password encrypted using DPAPI
------	--

## Details

Requires Powershell to be installed and execution policy set to RemoteSigned. This can be achieved by running `Set-ExecutionPolicy RemoteSigned` from Powershell.

## Value

An decrypted password as an invisible string. Invisible means that the password won't be displayed in the console, but can be assigned to a variable or used inline.

## Examples

```
## Not run:
# First run the command below from Powershell:
# Read-Host "PW?" -AsSecureString | ConvertFrom-SecureString | Out-File "C:\Temp>Password.txt"
# Now execute the following R commands to decrypt the password and save it in
# variable "x". Note that if run without assignment, the password will not
# be displayed in the console. Passwords must be saved to a variable or used
# inline within a connection string.
library("keyringr")
x <- decrypt_dpapi_pw("C:\\Temp\\Password.txt")

# function is best used in a connection string command:
ch <- odbcConnect("some dsn", uid = "user1", pwd = decrypt_dpapi_pw("C:\\Temp\\Password.txt"))

## End(Not run)
```

---

decrypt_gk_pw	<i>Get a password from Gnome Keyring using secret-tool</i>
---------------	--

---

**Description**

Get a password from Gnome Keyring using secret-tool

**Usage**

```
decrypt_gk_pw(key_value_pairs)
```

**Arguments**

key\_value\_pairs  
A string of key value pairs as expected by secret-tool

**Details**

Requires the Gnome Keyring and secret-tool to be installed.

**Value**

An decrypted password as an invisible string. Invisible means that the password won't be displayed in the console, but can be assigned to a variable or used inline.

**Examples**

```
## Not run:  
# First encrypt a password using secret-tool as follows:  
# secret-tool store --label=mylabel db mydb user user1  
# now return the password above to the R environment  
x <- decrypt_gk_pw("db mydb user user1")  
  
# function is best used in a connection string command:  
ch <- odbcConnect("some dsn", uid = "user1", pwd = decrypt_gk_pw("db mydb user user1"))  
  
## End(Not run)
```

---

decrypt_kc_pw	<i>Get a generic password from macOS Keychain using the 'security' cli</i>
---------------	--

---

**Description**

Get a generic password from macOS Keychain using the 'security' cli

**Usage**

```
decrypt_kc_pw(label, type = "generic")
```

**Arguments**

label	Keychain password label
type	Keychain password type. Either "generic" or "internet".

**Details**

Passwords must be saved in Keychain prior to using the function.

macOS may require the user to grant access to "security" the first time the function is run for each password. It is important to select "Always allow", which will prevent similar dialogs in the future.

**Value**

Returns a decrypted password as an invisible string. Invisible means that the password won't be displayed in the console, but can be assigned to a variable or used inline.

**Examples**

```
## Not run:
# First store a password in Keychain
# now return the password above to the R environment
x <- decrypt_kc_pw("label")

# function is best used in a connection string command:
ch <- odbcConnect("some dsn", uid = "user1", pwd = decrypt_kc_pw("mydb_myuser"))

## End(Not run)
```

---

get_kc_account	<i>Get a account name from macOS Keychain using the 'security' cli</i>
----------------	--

---

**Description**

Get a account name from macOS Keychain using the 'security' cli

**Usage**

```
get_kc_account(label, type = "generic")
```

**Arguments**

label	Keychain password label
type	Keychain password type. Either "generic" or "internet".

**Value**

Returns the account value attached to the label.

macOS may require the user to grant access to "security" the first time the function is run for each stored credential. It is important to select "Always allow", which will prevent similar dialogs in the future.

**Examples**

```
## Not run:  
# First store a set of credentials in Keychain  
# now return the account name to the R environment  
x <- get_kc_account("label")  
  
## End(Not run)
```

# Index

decrypt\_dpapi\_pw, 2

decrypt\_gk\_pw, 3

decrypt\_kc\_pw, 3

get\_kc\_account, 4