

# Package ‘anonymizer’

August 29, 2016

**Title** Anonymize Data Containing Personally Identifiable Information

**Version** 0.2.0

**Description** Allows users to quickly and easily anonymize data containing Personally Identifiable Information (PII) through convenience functions.

**URL** <https://github.com/paulhendricks/anonymizer>

**BugReports** <https://github.com/paulhendricks/anonymizer/issues>

**Depends** R (>= 3.1.2)

**License** MIT + file LICENSE

**LazyData** true

**Suggests** digest, testthat

**NeedsCompilation** no

**Author** Paul Hendricks [aut, cre]

**Maintainer** Paul Hendricks <paul.hendricks.2013@owu.edu>

**Repository** CRAN

**Date/Publication** 2015-09-08 19:51:21

## R topics documented:

anonymize . . . . .	2
anonymizer . . . . .	2
hash . . . . .	3
salt . . . . .	3
unsalt . . . . .	4
<b>Index</b>	<b>6</b>

---

anonymize	<i>Anonymize a vector.</i>
-----------	----------------------------

---

### Description

anonymize anonymizes a vector `.x` by first salting it with `salt` and then hashing it with `hash`. See both functions for additional documentation.

### Usage

```
anonymize(.x, .algo = "sha256", .seed = 0, .chars = letters,
         .n_chars = 5L, ...)
```

### Arguments

<code>.x</code>	a vector.
<code>.algo</code>	the name of the algorithm.
<code>.seed</code>	an integer to seed the random number generator.
<code>.chars</code>	set of characters to salt with.
<code>.n_chars</code>	an integer; number of characters to salt with.
<code>...</code>	additional arguments to be passed to hash.

### Details

The user is advised to check out [Wikipedia](#) for more information.

### Value

An anonymized version of the vector.

### Examples

```
set.seed(1)
anonymize(letters)
```

---

anonymizer	<i>anonymizer: Anonymize Data Containing Personally Identifiable Information</i>
------------	--

---

### Description

anonymizer: Anonymize Data Containing Personally Identifiable Information

---

hash	<i>Hash a vector.</i>
------	-----------------------

---

### Description

See [digest](#) for additional documentation.

### Usage

```
hash(.x, .algo = "sha256", .seed = 0, ...)
```

### Arguments

.x	a vector.
.algo	the name of the algorithm.
.seed	an integer to seed the random number generator.
...	additional arguments to be passed to <a href="#">digest</a> .

### Details

The user is advised to check out [Wikipedia](#) for more information.

### Value

A hashed version of the vector.

### Examples

```
# All algorithms available to digest::digest are available here
set.seed(1)
hash(letters, .algo = "sha256")
hash(letters, .algo = "crc32")
```

---

salt	<i>Salt a vector.</i>
------	-----------------------

---

### Description

`salt` takes a vector `.x` and returns a salted version of it. The algorithm for salting a vector is:

1. Select a random sample of characters of length `.n_chars` from `.chars`. Call this random sample `.y`.
2. Concatenate `.y`, the vector `.x`, and `.y` again in a vectorized fashion.

**Usage**

```
salt(.x, .seed = NULL, .chars = letters, .n_chars = 5L)
```

**Arguments**

`.x` a vector.  
`.seed` an integer to seed the random number generator.  
`.chars` set of characters to salt with.  
`.n_chars` an integer; number of characters to salt with.

**Details**

The user is advised to check out [Wikipedia](#) for more information.

**Value**

A salted version of the vector.

**Examples**

```
# Use various number of characters
salt(letters, .n_chars = 0L)
salt(letters, .n_chars = 1L)
salt(letters, .n_chars = 5L)
salt(letters)

# Use other sets of characters to salt with
salt(letters, .chars = letters[1:2])
```

---

unsalt	<i>Unsalt a vector.</i>
--------	-------------------------

---

**Description**

`salt` takes a vector `.x` and returns an unsalted version of it. The algorithm for unsalting a vector is:

1. Select a random sample of characters of length `.n_chars` from `.chars`. Call this random sample `.y`.
2. Substitute `.y` out of the vector `.x` wherever it occurs, in a vectorized fashion.

**Usage**

```
unsalt(.x, .seed = NULL, .chars = letters, .n_chars = 5L)
```

**Arguments**

<code>.x</code>	a vector.
<code>.seed</code>	an integer to seed the random number generator.
<code>.chars</code>	set of characters to unsalt with.
<code>.n_chars</code>	an integer; number of characters to unsalt with.

**Details**

The user is advised to check out [Wikipedia](#) for more information.

**Value**

An unsalted version of the vector.

**Examples**

```
# Use various number of characters
unsalt(salt(letters, .n_chars = 0L))
unsalt(salt(letters, .n_chars = 1L))
unsalt(salt(letters, .n_chars = 5L))
unsalt(salt(letters))

# Use other sets of characters to salt with
unsalt(salt(letters, .chars = letters[1:2]), .chars = letters[1:2])
```

# Index

[anonymize](#), [2](#)  
[anonymizer](#), [2](#)  
[anonymizer-package \(anonymizer\)](#), [2](#)  
  
[digest](#), [3](#)  
  
[hash](#), [2](#), [3](#)  
  
[salt](#), [2](#), [3](#)  
  
[unsalt](#), [4](#)