signethic
Release 0.2

Francois Dion

Jan 08, 2020

CONTENTS:

ONE

SIGNETHIC MODULE DOCUMENTATION

1.1 `signethic`

signethic.**gen_key_pair**(*path=None*, *key_len=1024*)
> Generates a private and public key pair. These will be generated based on the path of the private key (the public key will use the same file name plus the extension .pub)
>
> Alternatively, this can be generated using a tool like openssl:
>
> > openssl genpkey -out signing_key.pem 1024
> >
> > openssl rsa -in signing_key.pem -pubout > signing_key.pem.pub
>
> **Parameters**
>
> > * **path** – a string of the private key file name, with relative or full path
> >
> > * **key_len** – 1024 bits is default, other lengths can be used like 2048, 4096
>
> **Returns**

signethic.**get_private_key**(*private_key_path=None*)
> Loads private key. If no path is specified, loads signing_key.pem from the current directory.
>
> **Parameters** **private_key_path** – a string of the private key file name, with relative or full path
>
> **Returns** the private key

signethic.**get_public_key**(*public_key_path=None*, *private_key_path=None*)
> Loads public key. If no path is specified, loads signing_key.pem.pub from the current directory. If a private key path is provided, the public key path is ignored and the public key is loaded from the private key.
>
> **Parameters**
>
> > * **public_key_path** – a string of the public key file name, with relative or full path
> >
> > * **private_key_path** – a string of the private key file name, with relative or full path
>
> **Returns**

signethic.**main**()
> main entry point, called by command line script signethic.
>
> Calling signethic with a filename will generate a new file with the .signed extension that includes the signature and the file. It will be signed using the key *signing_key.pem* in the current directory, unless the SIGNING_KEY environment variable is set.
>
> Usage:

signethic

signethic filename

To specify signing key, set environment variable, i.e.:

export SIGNING_KEY=/path/to/key.pem

Returns

signethic.**sign**(*thing*, *private_key_path=None*)
Creates a digital signature of "thing" using *RSASSA-PKCS1-v1_5*

Parameters

- **thing** – binary data from any source, or a string

- **private_key_path** – a string of the private key file name, with relative or full path

Returns the PKCS1 v1.5 signature

signethic.**sign_and_persist**(*thing*, *path*, *private_key_path=None*)
Commit to disk the signature generated by *sign* and the "thing", in one file, specified by the path.

Parameters

- **thing** – binary data from any source, or a string

- **path** – a string of the resulting file name, with relative or full path (ie. something.zip.signed)

- **private_key_path** – a string of the private key file name, with relative or full path

Returns the PKCS1 v1.5 signature

signethic.**verify**(*thing*, *signature*, *public_key_path=None*)
Verify the signature corresponds to the "thing", based on the public key provided. That public key has to match the private key that was used to sign. Returns 0 if verification failed.

Parameters

- **thing** – binary data from any source, or a string

- **signature** – the PKCS1 v1.5 signature previously generated by sign, or loaded from signed file

- **public_key_path** – a string of the public key file name, with relative or full path

Returns test for True / False on this return value

signethic.**verify_file**(*path*, *public_key_path=None*)
Loads the signed file and verify the signature and the "thing" match, given a public key.

Parameters

- **path** – a string of the signed file name, with relative or full path (ie. something.zip.signed)

- **public_key_path** – a string of the public key file name, with relative or full path

Returns the "thing" in the signed file, or None if signature failed

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

G

M

S

V