eth-hash Documentation

Release 0.2.0

Jason Carver

Sep 13, 2018

Contents

The Ethereum hashing function, keccak256, sometimes (erroneously) called sha3

CHAPTER 1

Contents

1.1 Quickstart

1.1.1 Choose a hashing backend

If you're not sure, choose "pycryptodome" because it supports pypy3.

You can find a full list of each currently supported backend in `eth_hash.backends`.

1.1.2 Install

Put the backend you would like to use in brackets during install, like:

```
pip install eth-hash[pycryptodome]
```

1.1.3 Compute a Keccak256 Hash

```
>>> from eth_hash.auto import keccak
>>> keccak(b'')
b"\xc5\xd2F\x01\x86\xf7#<\x92~}\xb2\xdc\xc7\x03\xc0\xe5\x00\xb6S\xca\x82';
 ↪{\xfa\xd8\x04]\x85\xa4p"
```

You may also compute hashes incrementally

```
>>> from eth_hash.auto import keccak
>>> preimage = keccak.new(b'part-a')
>>> preimage.update(b'part-b')
>>> preimage.digest()
b
 ↪'6\x911\xdd50\xd6[\x7f\xf9B\xff\xc9SW\x98\xc3\xaal\xd9\xde\xdd6I\xb7\x91\x9e\xf4`pl\x08
 ↪'
```

The preimage object returned may be copied as well.

```
>>> from eth_hash.auto import keccak
>>> preimage = keccak.new(b'part-a')
>>> preimage_copy = preimage.copy()
>>> preimage.update(b'part-b')
>>> preimage.digest()
b
↪'6\x911\xdd50\xd6[\x7f\xf9B\xff\xc9SW\x98\xc3\xaal\xd9\xde\xdd6I\xb7\x91\x9e\xf4`pl\x08
↪'
>>> preimage_copy.update(b'part-c')
>>> preimage_copy.digest()
b'\xffcy45\xea\xdd\xdf\x8e(\x1c\xfcF\xf3\xd4\xa1S\x0f\xdf\xd8\x01!
↪\xb2(\xe1\xc7\xc6\xa3\x08\xc3\n\x0b'
```

1.1.4 Select one of many installed backends

If you have several backends installed, you may want to explicitly specify which one to load. You can specify in an environment variable, or at runtime.

Specify backend by environment variable

```
$ ETH_HASH_BACKEND="pysha3" python
>>> from eth_hash.auto import keccak
# This runs with the pysha3 backend
>>> keccak(b'')
b"\xc5\xd2F\x01\x86\xf7#<\x92~}\xb2\xdc\xc7\x03\xc0\xe5\x00\xb6S\xca\x82';
↪{\xfa\xd8\x04]\x85\xa4p"
```

Specify backend at runtime

```
>>> from eth_hash.backends import pysha3
>>> from eth_hash import Keccak256
>>> keccak = Keccak256(pysha3)
>>> keccak(b'')
b"\xc5\xd2F\x01\x86\xf7#<\x92~}\xb2\xdc\xc7\x03\xc0\xe5\x00\xb6S\xca\x82';
↪{\xfa\xd8\x04]\x85\xa4p"
```

1.2 Release Notes

1.2.1 v0.2.0

Released September 5, 2018

- set *pycryptodome* version to *>=3.6.6,<4* to fix a recently discovered vulnerability

1.2.2 v0.1.4

Released May 28, 2018

- Ensure the auto backend is pickleable (#19)

1.2.3 v0.1.3

Released May 14, 2018

- The pycryptodome backend now allows `update()`, then `digest()`, then `update()`.

1.2.4 v0.1.2

Released Apr 2, 2018

- You can now import eth-hash without a backend, it won't fail until trying to generate a hash

1.2.5 v0.1.1

Released Mar 15, 2018

- upgrade pycryptodome to v3.5.1+
- performance improvements with preimage
- Better docs and tests

1.2.6 v0.1.0

Released Feb 28, 2018

- Add support for `bytearray` input to keccak
- Add support for incrementally building hash results

1.2.7 v0.1.0-alpha.3

Released Feb 7, 2018

- Add pycryptodome backend support
- Add pysha3 backend support
- Can specify backend in environment variable `ETH_HASH_BACKEND`
- New *Quickstart* docs

1.2.8 v0.1.0-alpha.2

Released Feb 6, 2018

- Bugfix pypy3 reference in pypi

1.2.9 v0.1.0-alpha.1

- Launched repository, claimed names for pip, RTD, github, etc

1.3 eth_hash.backends package

1.3.1 Submodules

1.3.2 eth_hash.backends.auto module

1.3.3 eth_hash.backends.pycryptodome module

1.3.4 eth_hash.backends.pysha3 module

1.3.5 Module contents

1.4 eth_hash package

1.4.1 eth_hash.auto module

1.4.2 eth_hash.main module

CHAPTER 2

Indices and tables

- genindex
- modindex