

---

# AES-Toolbox

**Emídio Neto**

**Dec 27, 2022**



## INSTALLATION USAGE:

<b>1</b>	<b>Getting started</b>	<b>1</b>
1.1	AES-Toolbox . . . . .	1
1.2	Features . . . . .	1
1.3	Installation . . . . .	1
1.4	Using AES-Toolbox . . . . .	2
<b>2</b>	<b>Core Modules</b>	<b>5</b>
2.1	aestoolbox . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



## GETTING STARTED

### 1.1 AES-Toolbox

An AES Toolbox for computing Rijndael key schedule given a 128, 192, or 256-bit key.

- Documentation: <https://aestoolbox.readthedocs.io>.

### 1.2 Features

- AES Key Schedule
- AES Encrypt/Decrypt (work in progress)

### 1.3 Installation

#### 1.3.1 Stable release via pip

To install AES-Toolbox, run this command in your terminal:

```
$ pip install aestoolbox
```

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### 1.3.2 Get from source

The sources for AES-Toolbox can be downloaded from the [Github repo](#). This is the preferred method to install AES-Toolbox, as it will always install the most recent stable release.

You can either clone the public repository:

```
$ git clone git://github.com/emdneto/aestoolbox
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/emdneto/aestoolbox/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

## 1.4 Using AES-Toolbox

### 1.4.1 Usage (via CLI)

```
$ aes-schedule [-h] [-v] [-i] key  
$ aes-schedule 0x01010101020202020303030304040404 -i -v
```

The above command should output:

```
{'xk':  
{0: '0x01010101020202020303030304040404',  
1: '0xf2f3f3f3f0f1f1f1f3f2f2f7f6f6f6',  
2: '0xb2b1b19b4240406ab1b2b2984644446e',  
3: '0xadaa2ec1efea6eab5e58dc33181c985d',  
4: '0x39ec626cd6060cc7885ed0f4904248a9',  
5: '0x05beb10cd3b8bdc5be66d3fcba42596',  
6: '0x6c812113bf399cd8e4dff1e72f7bd471',  
7: '0x0dc98206b2f01ede562fef3979543b48',  
8: '0xad2bd0b01fdbce6e49f4215730a01a1f',  
9: '0x568910b44952deda00a6ff8d3006e592',  
10: '0x0f505fb04602816a46a47ee776a29b75'},  
'xki':  
{0: '0x01010101020202020303030304040404',  
1: '0xfdfafef8fff8fcfafcfbfff9f8fffbfd',  
2: '0xc263931b3d9b6fe1c1609018399f6be5',  
3: '0x70e738474d7c57a68c1cc7beb583ac5b',  
4: '0xa68450a9ebf8070f67e4c0b1d2676cea',  
5: '0xb86800d6539007d93474c768e613ab82',  
6: '0xffd917eeac491037983dd75f7e2e7cdd',  
7: '0xe238ed774e71fd40d64c2a1fa86256c2',  
8: '0xc20b68478c7a95075a36bf18f254e9da',  
9: '0x7edace11f2a05b16a896e40e5ac20dd4',  
10: '0x0f505fb04602816a46a47ee776a29b75'}}}
```

### 1.4.2 Usage as Python Library

Soon

#### Disclaimer

AES-Toolbox implementations should not be used in security software or production environments. The AES-Toolbox is for research purposes.





## CORE MODULES

### 2.1 aestoolbox

#### 2.1.1 aestoolbox package

##### Subpackages

aestoolbox.core package

##### Subpackages

aestoolbox.core.base package

##### Subpackages

aestoolbox.core.base.logs package

##### Submodules

aestoolbox.core.base.logs.logger module

AES-Toolbox Basic Logger Module

```
class aestoolbox.core.base.logs.logger.ToolboxLogger
```

```
    Bases: object
```

```
    static loadConfig(config_file, debug=False)
```

##### Module contents

##### Module contents

##### Submodules

**aestoolbox.core.aes\_schedule module**

This implementation is derived in part from the reference Golang AES implementation, which carries the following notice:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright

**notice, this list of conditions and the following disclaimer.**

- Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Google Inc. nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**class** aestoolbox.core.aes\_schedule.**KeySchedule**(key, dec=False)

Bases: object

**expand\_key()**

Computes the expanded AES key given a 128, 192 or 256 bit key. :return: Expanded AES Key. If *dec* is True, also returns the

decryption expanded AES Key in a tuple.

**static format\_hkeys**(xkb, Nr)

Formats the array of expanded key (xkb) or inverse expanded key (xki) in hexadecimal values and returns a dictionary with all round keys.

**Param**

xkb: Array of the expanded AES key.

**Param**

Nr: The numbers of rounds base on key length.

**Returns**

dkeys: Dictionary of formatted round keys.

**hexdump()**

**static rotw**(w)

Simple rotate transformation to a 4-byte word.

**Param**

w: 4-byte word.

**Returns**

4-byte transformed word.

**static subw(*w*)**

Apply Sbox match to each byte in word w.

**Param**

4-byte word

**Returns**

4-byte transformed word.

**validate\_key()****aestoolbox.core.const module****Module contents****Submodules****aestoolbox.release module**

This file states the AES-Toolbox metadata release information.

**Module contents**

Top-level package for aestoolbox.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### a

- `aestoolbox`, [7](#)
- `aestoolbox.core`, [7](#)
- `aestoolbox.core.aes_schedule`, [6](#)
- `aestoolbox.core.base`, [5](#)
- `aestoolbox.core.base.logs`, [5](#)
- `aestoolbox.core.base.logs.logger`, [5](#)
- `aestoolbox.core.const`, [7](#)
- `aestoolbox.release`, [7](#)





## INDEX

### A

aestoolbox  
    module, 7  
aestoolbox.core  
    module, 7  
aestoolbox.core.aes\_schedule  
    module, 6  
aestoolbox.core.base  
    module, 5  
aestoolbox.core.base.logs  
    module, 5  
aestoolbox.core.base.logs.logger  
    module, 5  
aestoolbox.core.const  
    module, 7  
aestoolbox.release  
    module, 7

### E

expand\_key() (aestoolbox.core.aes\_schedule.KeySchedule method), 6

### F

format\_hkeys() (aestoolbox.core.aes\_schedule.KeySchedule static method), 6

### H

hexdump() (aestoolbox.core.aes\_schedule.KeySchedule method), 6

### K

KeySchedule (class in aestoolbox.core.aes\_schedule), 6

### L

loadConfig() (aestoolbox.core.base.logs.logger.ToolboxLogger static method), 5

### M

module

aestoolbox, 7  
aestoolbox.core, 7  
aestoolbox.core.aes\_schedule, 6  
aestoolbox.core.base, 5  
aestoolbox.core.base.logs, 5  
aestoolbox.core.base.logs.logger, 5  
aestoolbox.core.const, 7  
aestoolbox.release, 7

### R

rotw() (aestoolbox.core.aes\_schedule.KeySchedule static method), 6

### S

subw() (aestoolbox.core.aes\_schedule.KeySchedule static method), 7

### T

ToolboxLogger (class in aestoolbox.core.base.logs.logger), 5

### V

validate\_key() (aestoolbox.core.aes\_schedule.KeySchedule method), 7