

pycrypto download fails windows 10



pycrypto 2.6.1.

This is a collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.). The package is structured to make adding new modules easy. This section is essentially complete, and the software interface will almost certainly not change in an incompatible way in the future; all that remains to be done is to fix any bugs that show up. If you encounter a bug, please report it in the Launchpad bug tracker at:

An example usage of the SHA256 module is:

An example usage of an encryption algorithm (AES, in this case) is:

One possible application of the modules is writing secure administration tools. Another application is in writing daemons and servers. Clients and servers can encrypt the data being exchanged and mutually authenticate themselves; daemons can encrypt private data for added security. Python also provides a pleasant framework for prototyping and experimentation with cryptographic algorithms; thanks to its arbitrary-length integers, public key algorithms are easily implemented.

As of PyCrypto 2.1.0, PyCrypto provides an easy-to-use random number generator:

A stronger version of Python's standard "random" module is also provided:

Caveat: For the random number generator to work correctly, you must call `Random.atfork()` in both the parent and child processes after using `os.fork()`

Installation.

PyCrypto is written and tested using Python version 2.1 through 3.3. Python 1.5.2 is not supported.

The modules are packaged using the Distutils, so you can simply run "python setup.py build" to build the package, and "python setup.py install" to install it.

If the setup.py script crashes with a DistutilsPlatformError complaining that the file /usr/lib/python2.2/config/Makefile doesn't exist, this means that the files needed for compiling new Python modules aren't installed on your system. Red Hat users often run into this because they don't have the python2-devel RPM installed. The fix is to simply install the requisite RPM. On Debian/Ubuntu, you need the python-dev package.

To verify that everything is in order, run "python setup.py test". It will test all the cryptographic modules, skipping ones that aren't available. If the test script reports an error on your machine, please report the bug using the bug tracker (URL given above). If possible, track down the bug and include a patch that fixes it, provided that you are able to meet the eligibility requirements at <http://www.pycrypto.org/submission-requirements/>.

It is possible to test a single sub-package or a single module only, for instance when you investigate why certain tests fail and don't want to run the whole suite each time. Use "python setup.py test --module=name", where 'name' is either a sub-package (Cipher, PublicKey, etc) or a module (Cipher.DES, PublicKey.RSA, etc). To further cut test coverage, pass also the option "--skip-slow-tests".

To install the package under the site-packages directory of your Python installation, run "python setup.py install".

If you have any comments, corrections, or improvements for this package, please report them to our mailing list, accessible via the PyCrypto website:

Microsoft Windows Python-3.6 PyCrypto installation error.

pip install pycrypto works fine with python3.5.2 but fails with python3.6 with the following error:

```
inttypes.h(26): error C2061: syntax error: identifier 'intmax_t'
```

The file include\pyport.h in Python installation directory does not have #include <stdint.h> anymore. This leaves intmax_t undefined.

A workaround for Microsoft VC compiler is to force include stdint.h via OS environment variable CL :

Open command prompt with admin privileges Run vsvars32.bat from your version of VC set CL=-FI"%VCINSTALLDIR%\INCLUDE\stdint.h" %CL% Run setup command python setup.py install
pip install pycrypto.

Thanks to user1960422's answer. PowerShell steps for pycrypto 2.6.1 (via simple-crypt) / Python 3.6 / Windows 10:

I also needed to follow the answer in: <https://stackoverflow.com/a/24822876/8751739> to fix a winrandom module error.

I've succeeded install pycrypto 2.6.1 on python 3.6 and windows 10.

Set an environment variable like below. Open cmd.exe Activate virtualenv Download pycrypto 2.6.1 release from github and unzip it. set an environment variable for pycrypto set CL=-FI"%VCINSTALLDIR%\INCLUDE\stdint.h" %CL% Run setup command python setup.py install
my result is here.

I would be glad if this could help someone.

I've succeeded install pycrypto 2.6.1 on python 3.6, Windows 10, Visual Studio 2017.

open "x86_x64 Cross-Tools Command Prompt for VS 2017" with administrator privilege in start menu. set CL=-FI"%VCINSTALLDIR%\Tools\MSVC\14.11.25503\include\stdint.h" pip install pycrypto.

Use PyCryptodome instead of pycrypto . Pycrypto is discontinued and is no longer actively supported.

Related Posts.

Prevent taps from passing through buttons in XAML/WPF.

Questions: In my project, I have a large container with a handler for taps. Inside this container, I also have a button. My goal is to handle all taps on the background container UNLESS the user click.

Send some keys to inactive window with python.

Questions: I'm tryin to send some keys to inactive window/process/programm (win32/64) using python. Already read about pywinauto and SendKeys, but both of them activate window before sendin keys.

Performance Counter shows different values when ReadOnly=false.

Questions: I was trying to figure out why certain performance counters aren't updated in our production server, when I've hit this weird issue – the counter seems to return different.

Python 3 - Install pycrypto on Windows.

I'm using Python 3.7 and Windows 10. ' pip install pycrypto ' doesn't work with the following error:

The file include\pyport.h in Python installation directory does not have #include <stdint.h> anymore. This leaves intrmax_t undefined.

A workaround for Microsoft VC compiler is to force include stdint.h via OS environment variable CL :

Open command prompt, and Setup VC environment by runing vcvars*.bat (choose file name depending on VC version and architecture)

set CL=-FI"Full-Path\stdint.h" (use real value for Full-Path for the environment)

then, try to install pycrypto again:

Another option that I'll try in future is to use PyCryptodome instead of pycrypto . pycrypto is discontinued and is no longer actively supported. PyCryptodome exposes almost the same API as pycrypto (source).

Installation¶

The installation procedure depends on the package you want the library to be in. PyCryptodome can be used as:

an almost drop-in replacement for the old PyCrypto library . You install it with:

In this case, all modules are installed under the Crypto package. You can test everything is right with:

You can test everything is right with:

The procedures below go a bit more in detail, by explaining how to setup the environment for compiling the C extensions for each OS, and how to install the GMP library.

Compiling in Linux Ubuntu¶

If you want to install under the Crypto package, replace below pycryptodomex with pycryptodome .

Compiling in Linux Fedora¶

If you want to install under the Crypto package, replace below pycryptodomex with pycryptodome .

Windows (from sources, Python 2.x, Python <=3.2)¶

If you want to install under the Crypto package, replace below pycryptodomex with pycryptodome .

Windows does not come with a C compiler like most Unix systems. The simplest way to compile the PyCryptodome extensions from source code is to install the minimum set of Visual Studio components freely made available by Microsoft.

Run Python from the command line and note down its version and whether it is a 32 bit or a 64 bit application.

For instance, if you see:

you clearly have Python 2.7 and it is a 32 bit application.

[Only once] Install Virtual Clone Drive.

[Only once] Download the ISO image of the MS SDK for Windows 7 and . NET Framework 3.5 SP1. It contains the Visual C++ 2008 compiler.

There are three ISO images available: you will need GRMSDK_EN_DVD.iso if your Windows OS is 32 bits or GRMSDKX_EN_DVD.iso if 64 bits.

Mount the ISO with Virtual Clone Drive and install the C/C++ compilers and the redistributable only.

If your Python is a 64 bit application, open a command prompt and perform the following steps:

Replace /x64 with /x86 if your Python is a 32 bit application.

Compile and install PyCryptodome:

To make sure everything work fine, run the test suite:

Windows (from sources, Python 3.3 and 3.4)¶

If you want to install under the Crypto package, replace below pycryptodomex with pycryptodome .

Windows does not come with a C compiler like most Unix systems. The simplest way to compile the Pycryptodome extensions from source code is to install the minimum set of Visual Studio components freely made available by Microsoft.

Run Python from the command line and note down its version and whether it is a 32 bit or a 64 bit application.

For instance, if you see:

you clearly have Python 2.7 and it is a 32 bit application.

[Only once] Install Virtual Clone Drive.

[Only once] Download the ISO image of the MS SDK for Windows 7 and . NET Framework 4. It contains the Visual C++ 2010 compiler.

There are three ISO images available: you will need GRMSDK_EN_DVD.iso if your Windows OS is 32 bits or GRMSDKX_EN_DVD.iso if 64 bits.

Mount the ISO with Virtual Clone Drive and install the C/C++ compilers and the redistributable only.

If your Python is a 64 bit application, open a command prompt and perform the following steps:

Replace /x64 with /x86 if your Python is a 32 bit application.

Compile and install PyCryptodome:

To make sure everything work fine, run the test suite:

Windows (from sources, Python 3.5 and newer)¶

If you want to install under the Crypto package, replace below pycryptodomex with pycryptodome .

Windows does not come with a C compiler like most Unix systems. The simplest way to compile the PyCryptodome extensions from source code is to install the minimum set of Visual Studio components freely made available by Microsoft.

[Once only] Download Build Tools for Visual Studio 2019. In the installer, select the C++ build tools , the Windows 10 SDK , and the latest version of MSVC v142 x64/x86 build tools .

Compile and install PyCryptodome:

To make sure everything work fine, run the test suite:

Documentation¶

Project documentation is written in reStructuredText and it is stored under Doc/src . To publish it as HTML files, you need to install sphinx and

use:

It will then be available under Doc/_build/html/.

PGP verification¶

All source packages and wheels on PyPI are cryptographically signed. They can be verified with the following PGP key:

cannot build PyCrypto in windows (64-bit)

I tried to build PyCrypto in my PC, but it failed. I use Python 2.6 in Windows Server 2008 R2 (64-bit).

the following messages are the whole error:

```
running build running build_py running build_ext warning: GMP library not found; Not building Crypto. PublicKey. _fastmath. building
'Crypto.Cipher.AES' extension C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\ BIN\amd64\ cl.exe /c /nologo /Ox /MD /W3 /GS-
/DNDEBUG -Isrc/ -Isrc /inc-msvc/ -IC:\Python26\ include -IC:\Python26\PC /Tsrc/AES.c /Fobuild\ temp.win- amd64-2. 6\Release\ src/AES.
obj AES.c c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(435) : error C2275: 'PyObject' : illegal use of this type as an expression
c:\python26\ include\ object. h(108) : see declaration of 'PyObject' c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(435) : error
C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(436) : error C2065: 'ctr' : undeclared identifier
c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(436) : warning C4047: '=' : 'int' differs in levels of indirection from 'void *' c:\users\
administrator\ pycrypto- 2.x\src\ block_template. c(440) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\
block_template. c(444) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(444) : error C2065:
'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(444) : error C2065: 'ctr' : undeclared identifier c:\users\
administrator\ pycrypto- 2.x\src\ block_template. c(448) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\
block_template. c(448) : warning C4047: 'function' : 'PyObject *' differs in levels of indirection from 'int' c:\users\ administrator\ pycrypto- 2.x\src\
block_template. c(448) : warning C4024: 'PyString_Size' : different types for formal and actual parameter 1 c:\users\ administrator\ pycrypto-
2.x\src\ block_template. c(453) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(453) : error
C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(453) : error C2065: 'ctr' : undeclared identifier
c:\users\ administrator\ pycrypto- 2.x\src\ block_template. c(458) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto-
2.x\src\ block_template. c(458) : warning C4047: 'function' : 'PyObject *' differs in levels of indirection from 'int' c:\users\ administrator\ pycrypto-
2.x\src\ block_template. c(458) : warning C4024: 'PyString_AsString' : different types for formal and actual parameter 1 c:\users\ administrator\
pycrypto- 2.x\src\ block_template. c(461) : error C2065: 'ctr' : undeclared identifier c:\users\ administrator\ pycrypto- 2.x\src\ block_template.
c(461) : error C2065: 'ctr' : undeclared identifier error: command "C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\ BIN\amd64\ cl.exe" failed with exit status 2 PS
C:\Users\ Administrator\ pycrypto- 2.x> PS C:\Users\ Administrator\ pycrypto- 2.x> PS C:\Users\ Administrator\ pycrypto- 2.x> PS C:\Users\
Administrator\ pycrypto- 2.x> python setup.py build > error warning: GMP library not found; Not building Crypto. PublicKey. _fastmath. error:
command "C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\ BIN\amd64\ cl.exe" failed with exit status 2.
```

Nevens (nevens-bartolomeo) wrote on 2009-11-20 : #1.

The fix for this is to modify line 435 of block_template.c change it from:

```
Py_BLOCK_THREADS; PyObject *ctr = PyObject_CallObject( self->counter, NULL); if(ctr == NULL) free(buffer); return NULL; >
```

```
PyObject *ctr; Py_BLOCK_THREADS; ctr = PyObject_CallObject( self->counter, NULL); if(ctr == NULL) free(buffer); return NULL; >
```