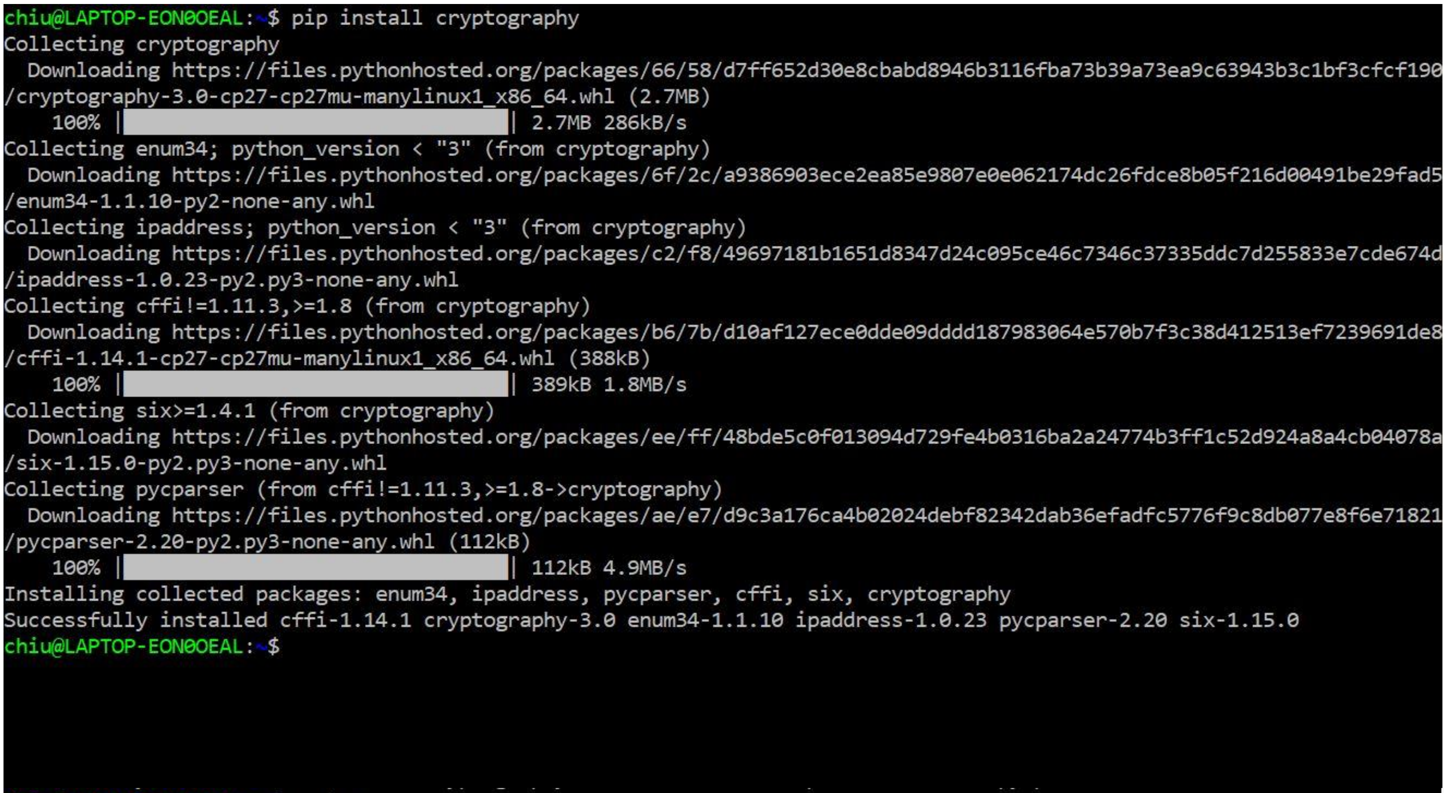
Q4.

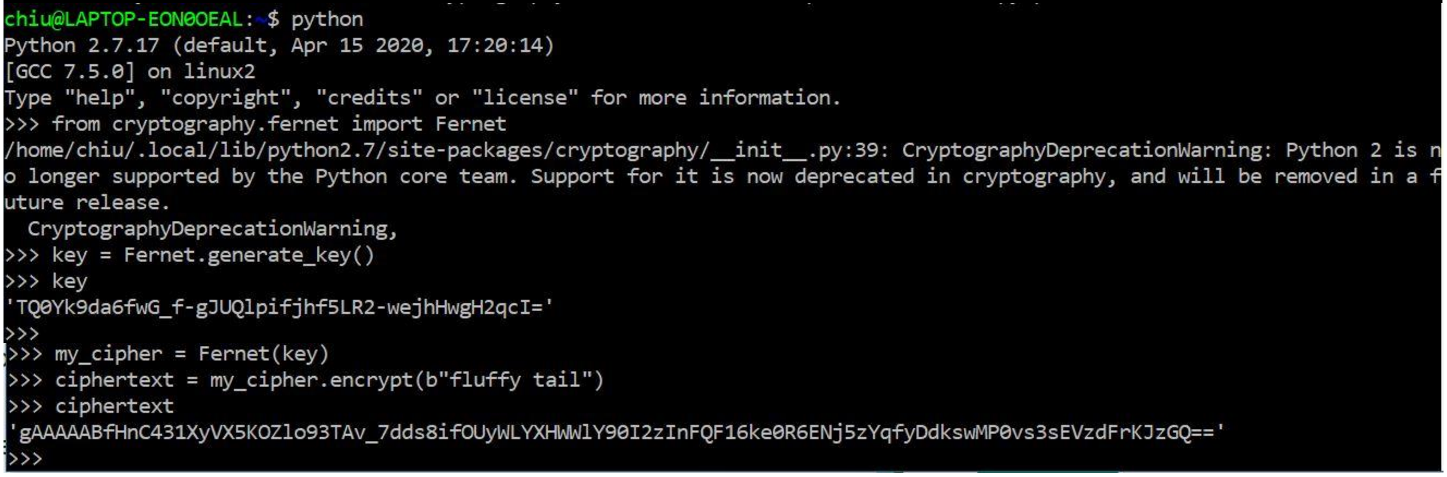
Python library that’s aptly named cryptography. It’s available on PyPI, so you can install it with pip:

$ pip install cryptography

Create a key for Fernet to work correctly, encrypt a message:



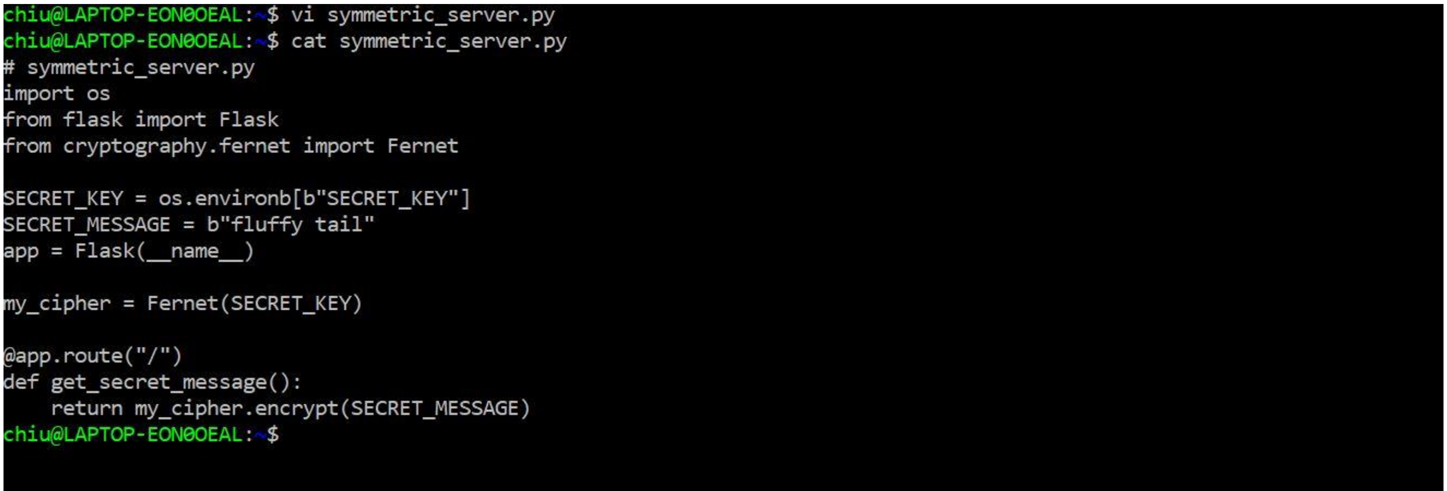
```
chiu@LAPTOP-EON0OEAL:~$ pip install cryptography
Collecting cryptography
  Downloading https://files.pythonhosted.org/packages/66/58/d7ff652d30e8cbabd8946b3116fba73b39a73ea9c63943b3c1bf3cfcf190
/cryptography-3.0-cp27-cp27mu-manylinux1_x86_64.whl (2.7MB)
    100% |████████████████████████████████| 2.7MB 286kB/s
Collecting enum34; python_version < "3" (from cryptography)
  Downloading https://files.pythonhosted.org/packages/6f/2c/a9386903ece2ea85e9807e0e062174dc26fdce8b05f216d00491be29fad5
/enum34-1.1.10-py2-none-any.whl
Collecting ipaddress; python_version < "3" (from cryptography)
  Downloading https://files.pythonhosted.org/packages/c2/f8/49697181b1651d8347d24c095ce46c7346c37335ddc7d255833e7cde674d
/ipaddress-1.0.23-py2.py3-none-any.whl
Collecting cffi!=1.11.3,>=1.8 (from cryptography)
  Downloading https://files.pythonhosted.org/packages/b6/7b/d10af127ece0dde09dddd187983064e570b7f3c38d412513ef7239691de8
/cffi-1.14.1-cp27-cp27mu-manylinux1_x86_64.whl (388kB)
    100% |████████████████████████████████| 389kB 1.8MB/s
Collecting six>=1.4.1 (from cryptography)
  Downloading https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a
/six-1.15.0-py2.py3-none-any.whl
Collecting pycparser (from cffi!=1.11.3,>=1.8->cryptography)
  Downloading https://files.pythonhosted.org/packages/ae/e7/d9c3a176ca4b02024debf82342dab36efadfc5776f9c8db077e8f6e71821
/pycparser-2.20-py2.py3-none-any.whl (112kB)
    100% |████████████████████████████████| 112kB 4.9MB/s
Installing collected packages: enum34, ipaddress, pycparser, cffi, six, cryptography
Successfully installed cffi-1.14.1 cryptography-3.0 enum34-1.1.10 ipaddress-1.0.23 pycparser-2.20 six-1.15.0
chiu@LAPTOP-EON0OEAL:~$
```



```
chiu@LAPTOP-EON0OEAL:~$ python
Python 2.7.17 (default, Apr 15 2020, 17:20:14)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from cryptography.fernet import Fernet
/home/chiu/.local/lib/python2.7/site-packages/cryptography/__init__.py:39: CryptographyDeprecationWarning: Python 2 is n
o longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in a f
uture release.
  CryptographyDeprecationWarning,
>>> key = Fernet.generate_key()
>>> key
'TQ0Yk9da6fwG_f-gJUQlpifjhf5LR2-wejhHwgH2qcI='
>>>
>>> my_cipher = Fernet(key)
>>> ciphertext = my_cipher.encrypt(b"fluffy tail")
>>> ciphertext
'gAAAAABfHnC431XyVX5KOZlo93TAv_7dds8ifOUyWLYXHWWlY90I2zInFQF16ke0R6ENj5zYqfyDdkswMP0vs3sEVzdFrKJzGQ=='
>>>
```

Create a new file called symmetric_server.py

This code combines your original server code with the Fernet object

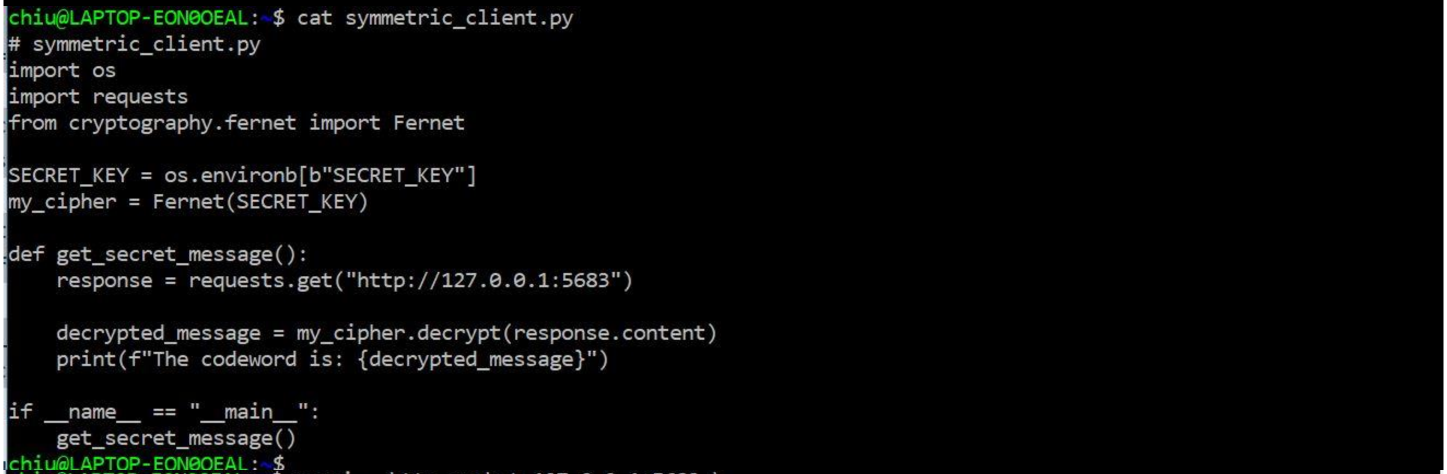


```
chiu@LAPTOP-EON0OEAL:~$ vi symmetric_server.py
chiu@LAPTOP-EON0OEAL:~$ cat symmetric_server.py
# symmetric_server.py
import os
from flask import Flask
from cryptography.fernet import Fernet

SECRET_KEY = os.environb[b"SECRET_KEY"]
SECRET_MESSAGE = b"fluffy tail"
app = Flask(__name__)

my_cipher = Fernet(SECRET_KEY)

@app.route("/")
def get_secret_message():
    return my_cipher.encrypt(SECRET_MESSAGE)
chiu@LAPTOP-EON0OEAL:~$
```



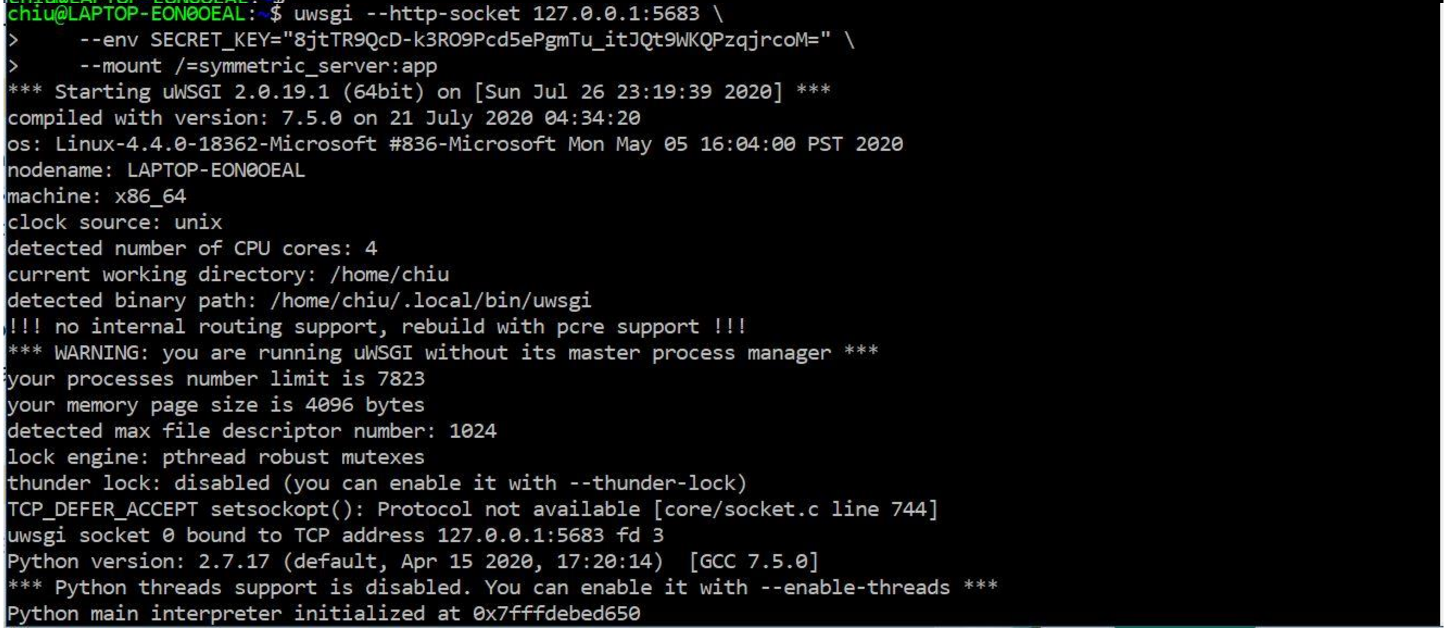
```
chiu@LAPTOP-EON0OEAL:~$ cat symmetric_client.py
# symmetric_client.py
import os
import requests
from cryptography.fernet import Fernet

SECRET_KEY = os.environb[b"SECRET_KEY"]
my_cipher = Fernet(SECRET_KEY)

def get_secret_message():
    response = requests.get("http://127.0.0.1:5683")

    decrypted_message = my_cipher.decrypt(response.content)
    print(f"The codeword is: {decrypted_message}")

if __name__ == "__main__":
    get_secret_message()
chiu@LAPTOP-EON0OEAL:~$
```

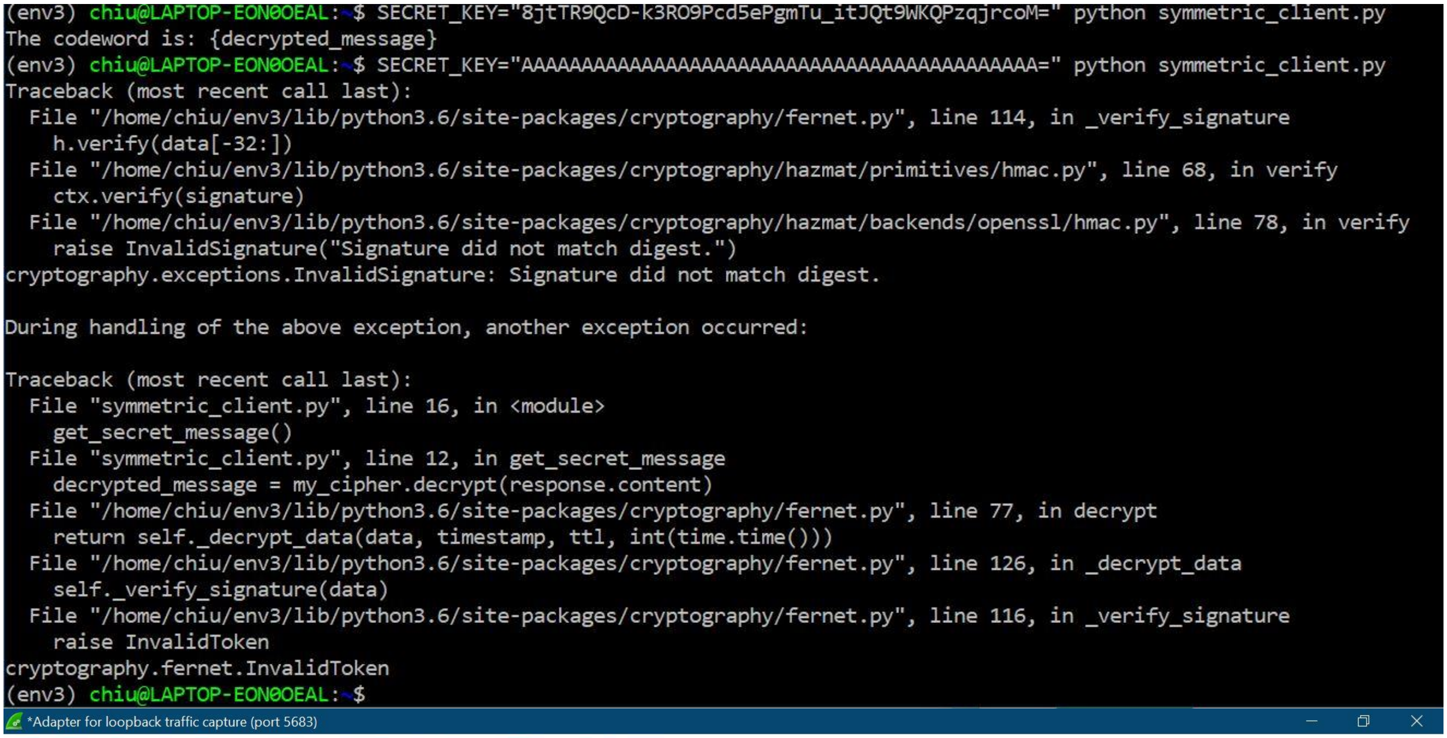


```
chiu@LAPTOP-EON0OEAL:~$ uwsgi --http-socket 127.0.0.1:5683 \
>     --env SECRET_KEY="8jtTR9QcD-k3RO9Pcd5ePgmTu_itJQt9WKQPzqjrcoM=" \
>     --mount /=symmetric_server:app
*** Starting uWSGI 2.0.19.1 (64bit) on [Sun Jul 26 23:19:39 2020] ***
compiled with version: 7.5.0 on 21 July 2020 04:34:20
os: Linux-4.4.0-18362-Microsoft #836-Microsoft Mon May 05 16:04:00 PST 2020
nodename: LAPTOP-EON0OEAL
machine: x86_64
clock source: unix
detected number of CPU cores: 4
current working directory: /home/chiu
detected binary path: /home/chiu/.local/bin/uwsgi
!!! no internal routing support, rebuild with pcre support !!!
*** WARNING: you are running uWSGI without its master process manager ***
your processes number limit is 7823
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
TCP_DEFER_ACCEPT setsockopt(): Protocol not available [core/socket.c line 744]
uwsgi socket 0 bound to TCP address 127.0.0.1:5683 fd 3
Python version: 2.7.17 (default, Apr 15 2020, 17:20:14)  [GCC 7.5.0]
*** Python threads support is disabled. You can enable it with --enable-threads ***
Python main interpreter initialized at 0x7fffdebed650
```

Run both the server and the client

Start the server on port 5683 again.

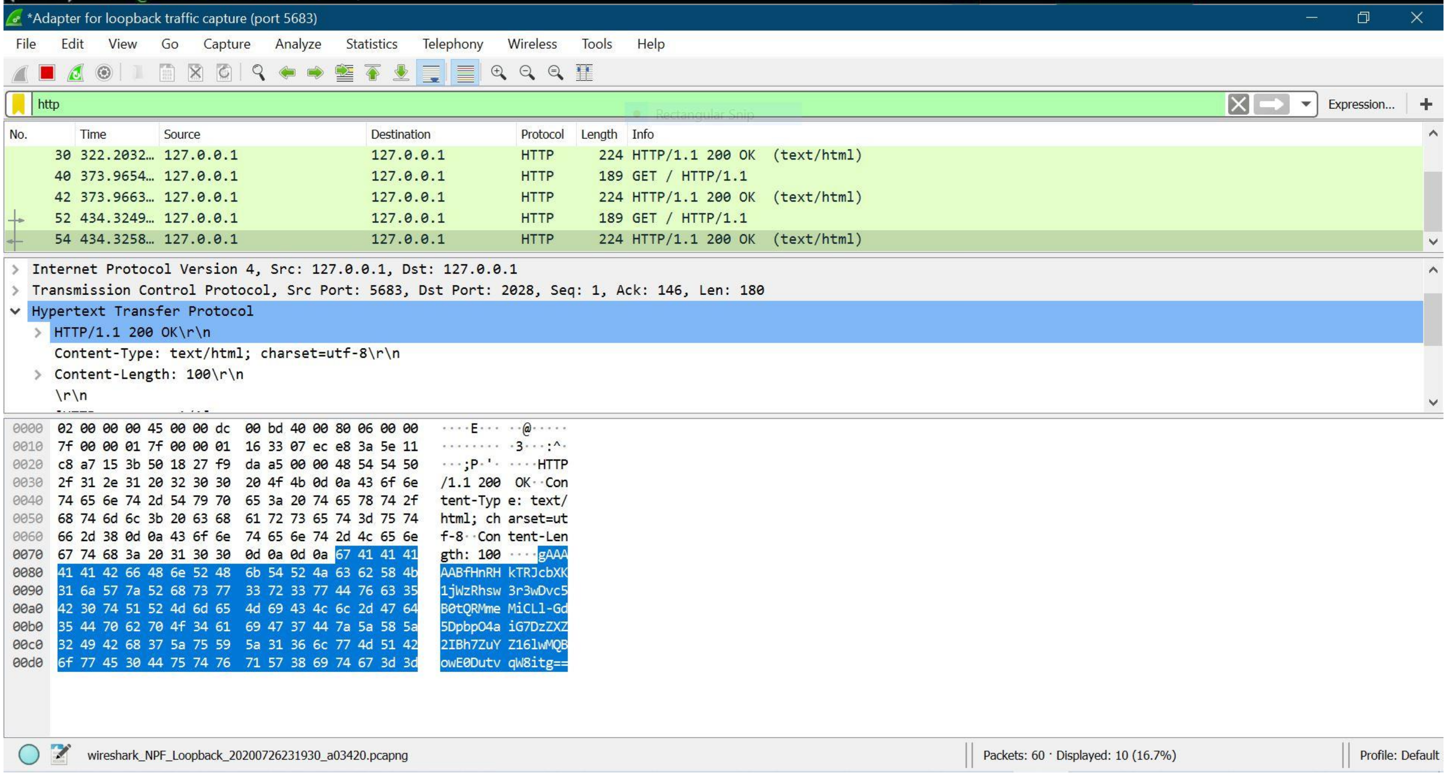
This time, you pass in a SECRET_KEY which must be at least a 32-length base64 encoded string.



```
(env3) chiu@LAPTOP-EON0OEAL:~$ SECRET_KEY="8jtTR9QcD-k3RO9Pcd5ePgmTu_itJQt9WKQPzqjrcoM=" python symmetric_client.py
The codeword is: {decrypted_message}
(env3) chiu@LAPTOP-EON0OEAL:~$ SECRET_KEY="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=" python symmetric_client.py
Traceback (most recent call last):
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/fernet.py", line 114, in _verify_signature
    h.verify(data[-32:])
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/hazmat/primitives/hmac.py", line 68, in verify
    ctx.verify(signature)
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/hazmat/backends/openssl/hmac.py", line 78, in verify
    raise InvalidSignature("Signature did not match digest.")
cryptography.exceptions.InvalidSignature: Signature did not match digest.

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "symmetric_client.py", line 16, in <module>
    get_secret_message()
  File "symmetric_client.py", line 12, in get_secret_message
    decrypted_message = my_cipher.decrypt(response.content)
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/fernet.py", line 77, in decrypt
    return self._decrypt_data(data, timestamp, ttl, int(time.time()))
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/fernet.py", line 126, in _decrypt_data
    self._verify_signature(data)
  File "/home/chiu/env3/lib/python3.6/site-packages/cryptography/fernet.py", line 116, in _verify_signature
    raise InvalidToken
cryptography.fernet.InvalidToken
(env3) chiu@LAPTOP-EON0OEAL:~$
```



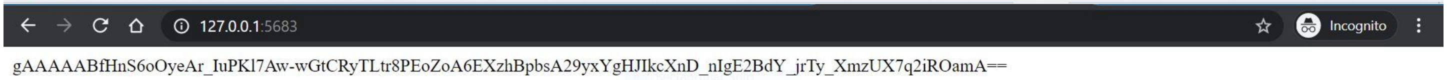
*Adapter for loopback traffic capture (port 5683)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
+

```
No.   Time        Source      Destination   Protocol  Length  Info
   30 322.2032…   127.0.0.1   127.0.0.1     HTTP         224  HTTP/1.1 200 OK  (text/html)
   40 373.9654…   127.0.0.1   127.0.0.1     HTTP         189  GET / HTTP/1.1
   42 373.9663…   127.0.0.1   127.0.0.1     HTTP         224  HTTP/1.1 200 OK  (text/html)
   52 434.3249…   127.0.0.1   127.0.0.1     HTTP         189  GET / HTTP/1.1
   54 434.3258…   127.0.0.1   127.0.0.1     HTTP         224  HTTP/1.1 200 OK  (text/html)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 5683, Dst Port: 2028, Seq: 1, Ack: 146, Len: 180
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Content-Type: text/html; charset=utf-8\r\n
  > Content-Length: 100\r\n
    \r\n

0000  02 00 00 00 45 00 00 dc  00 bd 40 00 80 06 00 00   ····E··· ··@·····
0010  7f 00 00 01 7f 00 00 01  16 33 07 ec e8 3a 5e 11   ········ ·3···:^·
0020  c8 a7 15 3b 50 18 27 f9  da a5 00 00 48 54 54 50   ···;P·'· ····HTTP
0030  2f 31 2e 31 20 32 30 30  20 4f 4b 0d 0a 43 6f 6e   /1.1 200  OK··Con
0040  74 65 6e 74 2d 54 79 70  65 3a 20 74 65 78 74 2f   tent-Typ e: text/
0050  68 74 6d 6c 3b 20 63 68  61 72 73 65 74 3d 75 74   html; ch arset=ut
0060  66 2d 38 0d 0a 43 6f 6e  74 65 6e 74 2d 4c 65 6e   f-8··Con tent-Len
0070  67 74 68 3a 20 31 30 30  0d 0a 0d 0a 67 41 41 41   gth: 100 ····gAAA
0080  41 41 42 66 48 6e 52 48  6b 54 52 4a 63 62 58 4b   AABfHnRH kTRJcbXK
0090  31 6a 57 7a 52 68 73 77  33 72 33 77 44 76 63 35   1jWzRhsw 3r3wDvc5
00a0  42 30 74 51 52 4d 6d 65  4d 69 43 4c 6c 2d 47 64   B0tQRMme MiCLl-Gd
00b0  35 44 70 62 70 4f 34 61  69 47 37 44 7a 5a 58 5a   5DpbpO4a iG7DzZXZ
00c0  32 49 42 68 37 5a 75 59  5a 31 36 6c 77 4d 51 42   2IBh7ZuY Z16lwMQB
00d0  6f 77 45 30 44 75 74 76  71 57 38 69 74 67 3d 3d   owE0Dutv qW8itg==

wireshark_NPF_Loopback_20200726231930_a03420.pcapng   Packets: 60 · Displayed: 10 (16.7%)   Profile: Default
```



gAAAAABfHnS6oOyeAr_IuPKl7Aw-wGtCRyTLtr8PEoZoA6EXzhBpbsA29yxYgHJIkcXnD_nIgE2BdY_jrTy_XmzUX7q2iROamA==

You can see the data was encrypted and that eavesdroppers have no clue what the message content actually is.