readme

倪浚桐

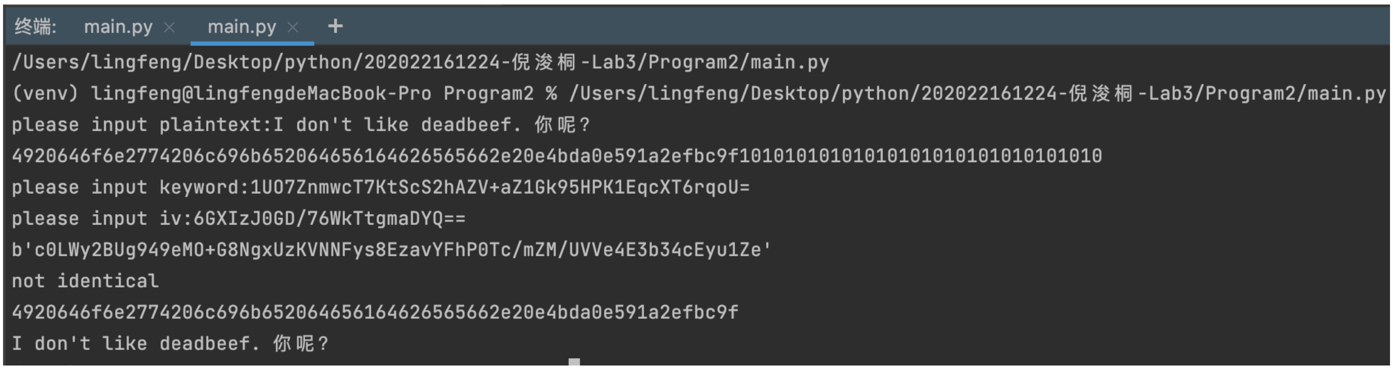
202022161224

Lab3-progarm2

macOS Big Sur 11.5.2(20G95)

Pycharm 11.0.11 x86-64

Python 3.9.5



```
/Users/lingfeng/Desktop/python/202022161224-倪浚桐-Lab3/Program2/main.py
(venv) lingfeng@lingfengdeMacBook-Pro Program2 % /Users/lingfeng/Desktop/python/202022161224-倪浚桐-Lab3/Program2/main.py
please input plaintext:I don't like deadbeef. 你呢?
4920646f6e2774206c696b6520646561646265662e20e4bda0e591a2efbc9f10101010101010101010101010101010
please input keyword:1UO7ZnmwcT7KtScS2hAZV+aZ1Gk95HPK1EqcXT6rqoU=
please input iv:6GXIzJ0GD/76WkTtgmaDYQ==
b'c0LWy2BUg949eMO+G8NgxUzKVNNFys8EzavYFhP0Tc/mZM/UVVe4E3b34cEyu1Ze'
not identical
4920646f6e2774206c696b6520646561646265662e20e4bda0e591a2efbc9f
I don't like deadbeef. 你呢?
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import base64
from Crypto.Cipher import AES


def Pad(text: bytes, byteAlignLen: int) -> bytes:
    count: int = len(text)
    mod_num: int = count % byteAlignLen
    add_num: int = byteAlignLen - mod_num
    add_text: str = chr(add_num) * add_num
    add_text: bytes = add_text.encode('utf-8')
    return text + add_text


def Unpad(text: bytes) -> bytes:
    text: str = text.decode('utf-8')
    remainder: str = text[-1]
    padding_text: str = ord(remainder) * remainder
    return text.rstrip(padding_text).encode('utf-8')

```

```

def encrypt(text: bytes, key: bytes, iv: bytes) -> bytes:
    mode: int = AES.MODE_CBC
    cryptos = AES.new(key, mode, iv)
    cipher_text: bytes = cryptos.encrypt(text)
    return cipher_text


def decrypt(text: bytes, key: bytes, iv: bytes) -> bytes:
    mode = AES.MODE_CBC
    cryptos = AES.new(key, mode, iv)
    plain_text = cryptos.decrypt(text)
    return plain_text


plaintext: str = input("please input plaintext:")
plaintext: bytes = plaintext.encode('utf-8')
plaintext_copy: bytes = plaintext
plaintext: bytes = Pad(plaintext, 16)
print(plaintext.hex())

keyword: str = input("please input keyword:")
keyword: bytes = base64.b64decode(keyword)

if len(keyword) != 32:
    raise Exception('key length mismatch')

iv_text: str = input("please input iv:")
iv_text: bytes = base64.b64decode(iv_text)

if len(iv_text) != 16:
    raise Exception('IV length mismatch')

ciphertext: bytes = encrypt(plaintext, keyword, iv_text)
print(base64.b64encode(ciphertext))

de_plaintext: bytes = decrypt(ciphertext, keyword, iv_text)

if de_plaintext == plaintext_copy:
    print("identical")
else:
    print("not identical")

de_plaintext: bytes = Unpad(de_plaintext)
print(de_plaintext.hex())

print(de_plaintext.decode('utf-8'))

```