



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیووتر

پروژه دوم

درس مبانی امنیت اطلاعات

هلياسادات هاشمي پور - ۹۸۳۱۱۰۶

استاد

دکتر شهریاری

آذر ۰۱

کد زده شده در محیط کولب میباشد.

بخش اول: رمزنگاری

در این بخش در اصل یک مقدار را به عنوان کلید در نظر گرفته و سپس اقدام به رمزنگاری میکنیم. (اگر کاربر E را وارد کند)

- ابتدا با گرفتن کلید اولیه الگوریتم از فایل مورد نظر و سپس اضافه کردن salt

```
#  
#  
print('-----')  
enter = input("Choose E for Encryption and D for Decryption: ")  
print('-----')  
if enter == "E":  
    print("Encryption")  
    file = open("Key.txt", "r") # open the file in read mode  
    encryptionKey = file.read() # read the file  
    salt = os.urandom(16) # generate a random 16 byte salt
```

- پس خواندن کلید از فایل مربوطه اش، طول آن را با استفاده از کتابخانه pbkdf2 به ۲۵۶ بیت رسانده و سپس با استفاده از کتابخانه binascii هگز آن را به اپراتور استفاده کننده از ابزار نشان دادم. و در فایلی ذخیره کردم.

```
print('Algorithm key:', binascii.hexlify(key)) # print the key  
  
file = open("SaltedKey.txt", "w") # open the file in write mode  
TEXT = str(binascii.hexlify(key)).maketrans(" ", " ", "'b") # remove the b and ' from the decrypted text  
file.writelines(str(binascii.hexlify(key)).translate(TEXT)) # write the reduced key to the file  
file.close() # close the file
```

- پس از آن با استفاده از کتابخانه secrets برای استفاده در مدل CTR initial vector یک secrets تولید کردم و اقدام به رمزنگاری با استفاده از کتابخانه pyaes و متد AESModeOfOperationCTR کردم.

```
iv = secrets.randbits(256) # generate a random 256 bit number  
aes = pyaes.AESModeOfOperationCTR(key, pyaes.Counter(iv)) # create the AES object  
  
file = open("PlainText.txt", "r") # open the file in read mode  
plaintext = file.read() # read the file  
ciphertext = aes.encrypt(plaintext) # encrypt the plaintext  
print('Encrypted Text:', binascii.hexlify(ciphertext)) # print the ciphertext
```

- در آخر هم ciphertext را در یک فایلی کردم.

```

file = open("Encrypted.txt", "w") # open the file in write mode
TEXT = str(binascii.hexlify(ciphertext)).maketrans(" ", " ", "'b") # remove the b and ' from the ciphertext
file.writelines(str(binascii.hexlify(ciphertext)).translate(TEXT)) # write the ciphertext to the file
file.close() # close the file

```

بخش دوم: رمزگشایی

با استفاده از initial vector و کلید به دست آمده از مرحله قبلی بخش رمزگشایی را پیاده سازی کردم به این شکل که پس از تبدیل بایت های خروجی به رشتہ مناسب خود آن را در فایلی ذخیره کردم. (کاربر D را وارد کند)

```

elif enter == "D":
    print("Decryption")
    print('Algorithm key:', binascii.hexlify(key)) # print the key
    aes2 = pyaes.AESModeOfOperationCTR(key, pyaes.Counter(iv)) # create the AES object
    decrypted = aes2.decrypt(ciphertext) # decrypt the ciphertext
    print('Decrypted Text:', decrypted) # print the decrypted text

    plainText = str(decrypted).maketrans(" ", " ", "'b") # remove the b and ' from the decrypted text
    plainText = str(decrypted).translate(plainText) # remove the b and ' from the decrypted text

    file = open("Decrypted.txt", "w") # open the file in write mode
    file.writelines(plainText) # write the decrypted text to the file
    file.close() # close the file

```

خروجی با وارد کردن E و D به شکل زیر است:

```

-----
Choose E for Encryption and D for Decryption:   E
-----
Encryption
Algorithm key: b'9ea02d16dd9cc9c1185f54f0e02096896e75ea9ecf3111f807716dd091268fd5'
Encrypted Text: b'c41c1aa50f4a86'

-----
Choose E for Encryption and D for Decryption:   D
-----
Decryption
Algorithm key: b'9ea02d16dd9cc9c1185f54f0e02096896e75ea9ecf3111f807716dd091268fd5'
Decrypted Text: b'9831106'

-----
Choose E for Encryption and D for Decryption:   [REDACTED]
-----
```

فایل‌ها هم به صورت زیر هستند.

```
Key.txt ×  
1 AUT*ICTSec*2022
```

```
Encrypted.txt ×  
1 c41c1aa50f4a86
```

```
SaltedKey.txt ×  
1 9ea02d16dd9cc9c1185f54f0e02096896e75ea9ecf3111f807716dd091268fd5
```

```
PlainText.txt ×  
1 9831106
```



```
Decrypted.txt ×  
1 9831106
```