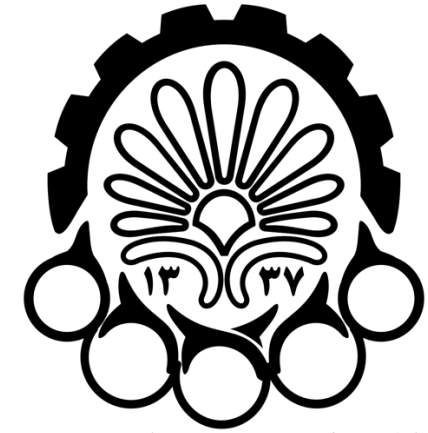
به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

تمرین عملی دوم درس مبانی امنیت اطلاعات: رمزنگاری

استاد درس: دکتر شهریاری

دانیال حمدی – ۹۷۳۱۱۱۱

پاییز ۱۴۰۱

۱. توضیح کد

همان‌طور که دستور کار گفته شده، از کتابخانه و ماژول‌های os, secrets, pbkdf2, binascii و pyaes استفاده شده است.

برای ساخت کلید، تابع initialize_key فراخوانده می‌شود. در بدنه‌ی این تابع، در تابع get_key کلید از فایل خوانده می‌شود، با تابع get_salt به کمک ماژول os پایتون، یک رشته بایت رندوم تولید می‌شود. سپس طی تابع hash_key به کمک کتابخانه‌ی pbkdf2 کلید و سالت تولید شده به یک رشته‌ی ۲۵۶ بیتی نگاشت می‌شوند. در نهایت طی تابع to_hex به کمک ماژول binascii کلید ساخته شده به شکل هگزادسیمال درآمده و به کاربر نمایش داده می‌شود.

برای ساخت عدد تصادفی اولیه‌ی شمارنده‌ی AES، طی تابع get_counter_initial_value، به کمک ماژول os، یک عدد رندوم بین 1 تا 10^12 تولید می‌شود.

در نهایت برای رمزنگاری و رمزگشایی، دو شئ از کلاس pyaes. AESModeOfOperationCTR نگه می‌داریم. این نکته قابل توجه است که به دلیل متقارن بودن رمزنگاری و رمزگشایی AES در مُد شمارنده، هر دو متد encrypt. و decrypt. این کلاس، مشابه‌اند.

```
class AESModeOfOperationCTR(AESStreamModeOfOperation):
    """AES Counter Mode of Operation...."""


    name = "Counter (CTR)"

    def __init__(self, key, counter = None):...

    def encrypt(self, plaintext):...

    def decrypt(self, crypttext):
        # AES-CTR is symetric
        return self.encrypt(crypttext)
```

با هر بار فراخوانده شدن یکی از متدهای encrypt و decrypt، مقدار شمارنده یک واحد زیاد می‌شود. پس اگر متن خامی را با یک شئ AES رمزنگاری کنیم، با فراخوانی تابع decrypt روی همان شئ و ورودی دادن پیام رمزنگاری شده، دوباره به متن خام نمی‌رسیم. این به این دلیل است که رمزنگاری با شمارنده‌ی x و رمزگشایی با شمارنده‌ی x+1 انجام می‌شود.

به همین دلیل، در کد برای رمزنگاری و رمزگشایی دو شئ مختلف از pyaes. AESModeOfOperationCTR می‌داریم.

```
encryption_counter = pyaes.Counter(initial_value=initial_counter_value)
aes_encryption = pyaes.AESModeOfOperationCTR(key, counter=encryption_counter)
decryption_counter = pyaes.Counter(initial_value=initial_counter_value)
aes_decryption = pyaes.AESModeOfOperationCTR(key, counter=decryption_counter)
```

در نهایت توابع encrypt و decrypt، پیاده‌سازی‌شان بدیهی‌ست؛ این توابع محتویات فایل ورودی را خوانده، این محتویات را با متدهای encrypt. یا decrypt. شئ aes، رمز کرده، و خروجی را در فایل مناسب ذخیره می‌کنند.

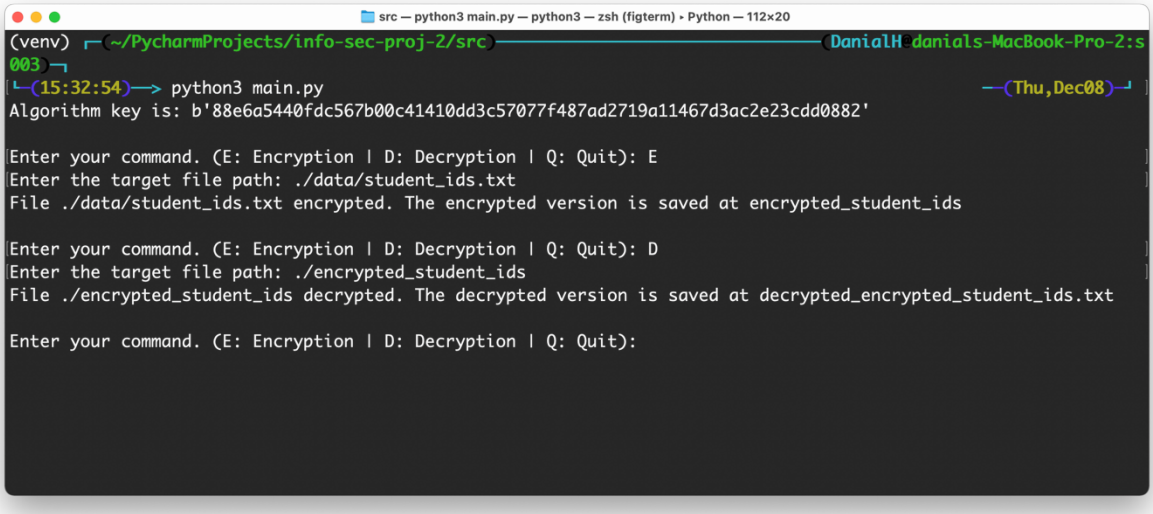
```python
def encryption(aes: pyaes.AESModeOfOperationCTR):
    """
    Note that encryption and decryption in AES:CounterMode is symmetric, and
     everytime the .encrypt method is called, the counter is incremented by 1.
     Therefor, the aes object created for encryption, cannot be used directly for decryption.
     (Because the decryption would be computed with the wrong counter value)
    """
    target_file_path = input('Enter the target file path: ')
    target_file_name = os.path.basename(target_file_path)
    target_file_name_without_extension = os.path.splitext(target_file_name)[0]

    with open(target_file_path, 'r') as f:
        input_text = f.read()

    result = aes.encrypt(input_text)

    result_file_name = f'encrypted_{target_file_name_without_extension}'
    with open(result_file_name, 'wb') as f:
        f.write(result)

    print(f'File {target_file_path} encrypted. The encrypted version is saved at {result_file_name}')
```

۲. توضیح خروجی

رمزنگاری: در تصویر ۱، پس از اجرای برنامه، ابتدا با وارد کردن دستور E و دادن آدرس ./data/student_ids.txt فایل شماره‌های دانشجویی را رمزنگاری می کنیم. همان‌طور که در خروجی این دستور چاپ شده، نتیجه در فایلی به آدرس ./encrypted_student_ids ذخیره می‌شود. به طور کلی اگر فایل ورودی برنامه به صورت /parent_1/parent_2/file_name.extension باشد، خروجی در فایلی به آدرس ./encrypted_file_name ذخیره می‌شود.

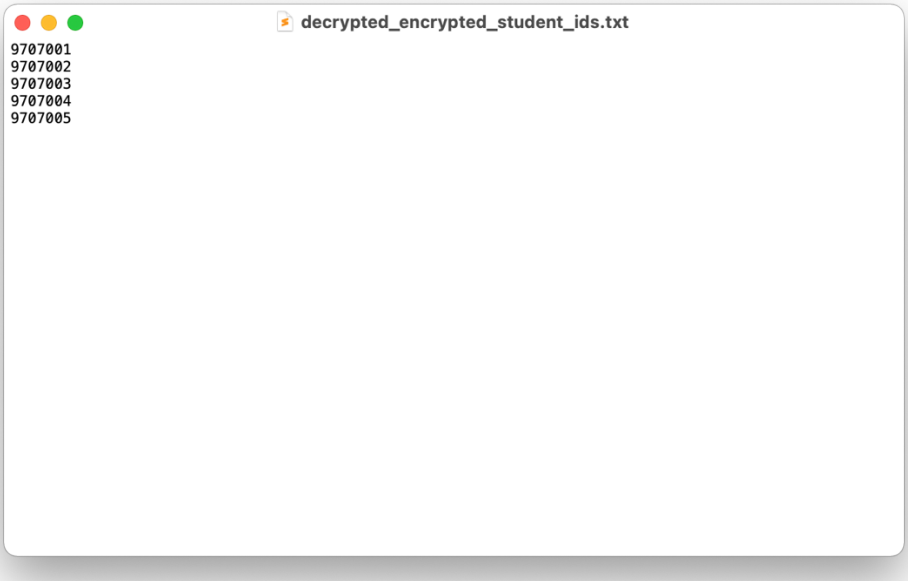
نتیجه‌ی این رمزنگاری، رشته‌ای باینری‌ست و نمایش آن به صورت رشته‌های utf-8 یا ascii معنی‌دار نخواهد بود. در نتیجه فایل بدون پسوند ذخیره می‌شود. محتویات این فایل در تصویر ۲ قابل مشاهده است.

رمزگشایی: در تصویر ۱، پس از دستورات رمزنگاری، دستورات D و آدرس ./encrypted_student_ids برای رمزگشایی وارد شده است. خروجی رمزگشایی به صورت فایل متنی با پسوند txt ذخیره می‌شود. همچنین برای مشخص کردن عمل رمزنگاری، به نام فایل پیشوند _decrypted اضافه می‌شود. نتیجه‌ی این دستور (محتویات فایل) در تصویر ۳ قابل مشاهده است.



تصویر ۱

تصویر ۲



تصویر ۳