CEI_423 Assignment - Sotiris Gypsiotis (Student ID: 22983)

All .txt files must be saved as **UTF-8** so it doesn't include BOM at the beginning of the text. I used PyCharm 2024.2.4 for this project since it's the best IDE for python.

Preparation Installations

Use the following command to install the Python library **pycryptodome**.

```
pip install pycryptodome
```

Usage and Examples

Caesar

Write a random plaintext in the file.txt file for encryption and a key (integer) in the my_key.txt for the shift value.

file.txt	caesar_key.txt
file.txt - Notepad File Edit Format View Help caesar secret message	caesar_key.txt - Notepad File Edit Format View Help 65

Run the following command in a terminal for encryption and decryption respectively:

- Encryption:

```
python encrypt-decrypt.py --encrypt --algorithm caesar --key caesar_key.txt --input
file.txt --output caesar.enc
```

- Decryption:

```
python encrypt-decrypt.py --decrypt --algorithm caesar --key caesar_key.txt --input
caesar.enc --output caesar-decrypted.txt
```

Results:

encrypted file	decrypted file
caesar.enc - Notepad File Edit Format View Help *(,:(9ç:,*9,;ç4,::(.,	caesar-decrypted.txt - Notepad File Edit Format View Help caesar secret message

One Time Pad (OTP)

Same as Caesar, write a plain text in the file.txt file and a key in the key.txt file.

Make sure the plaintext is the same length as the key ("Secret Message" is 14 characters with the space included, so is my_key)

file.txt	otp_key.txt
file.txt - Notepad File Edit Format View Help otp secure message	otp_key.txt - Notepad File Edit Format View Help 7 7Dç0„ŒP\ýé¹à

Run the following command in a terminal for encryption and decryption respectively:

- Encryption:

```
python encrypt-decrypt.py --encrypt --algorithm otp --key otp_key.txt --input file.txt --output otp.enc
```

- Decryption:

```
python encrypt-decrypt.py --decrypt --algorithm otp --key otp_key.txt --input otp.enc --output otp-decrypted.txt
```

Results:

encrypted file	decrypted file
otp.enc - Notepad File Edit Format View Help ?_‘ŒtÅºêM5Gø-1–Q	otp-decrypted.txt - Notepad File Edit Format View Help otp secure message

AES

Same as before, write a plaintext for the file.txt and type your 16-byte key for AES-128 or 32-byte key for AES-256.

For this example I'm using a 16-byte hexadecimal key.

file.txt	aes_key.txt
file.txt - Notepad File Edit Format View Help AES confidential data	aes_key.txt - Notepad File Edit Format View Help a13237187df767a069cf3120f23d792a

Run the following command in a terminal for encryption and decryption respectively:

- Encryption:

```
python encrypt-decrypt.py --encrypt --algorithm aes --key aes_key.txt --input file.txt --output aes.enc
```

- Decryption:

```
python encrypt-decrypt.py --decrypt --algorithm aes --key aes_key.txt --input aes.enc --output aes-decrypted.txt
```

Results:

encrypted file	decrypted file
aes.enc - Notepad File Edit Format View Help Gêð +Q r0ZÍ±'€û'4çÚÚR µóúaº*5KCBUn-ÎËR¬â›a«Q	aes-decrypted.txt - Notepad File Edit Format View Help AES confidential data

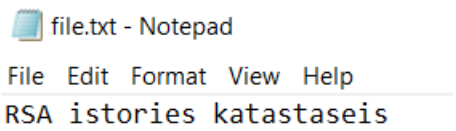
RSA

In RSA you need to generate your private and public key using *openssl*. But for this example I have implemented a function to create both keys automatically because it's an annoying process that requires powershell commands in Windows.

Here is the private and public key it generated:

public_key.pem	private_key.pem
public_key.pem - Notepad File Edit Format View Help -----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkZ6m0eiVv35V1AW6Q4Zt mVPEBTE/68FmMHtJN7nUl6KsC1s6RG7aqzBGS+q9COLCpyjHCdxhKg6/zy6qAnRi Up2dZAYiBgJwrGzY/92N6jJsm9oFMChNtb1DYxzHs4WeRgGDKz5a19ylbqM/kaqW RkZ4cH3xqw8SE1fgMY2DS3XoJZSOmbocmrxc4FEXbK0GHAKsWVGizms58x+l4Y6h rCvN6dQ13xq9a+pZvkzr5fiARclsUJTpBn5YShuweDR9QrsOsodXoLjODv5hZlhS srt9mF8w0XnxQkGOw0U+OlwR4IAC4nLPd+oNkDc5Tw2GEbhEs8rw3lDo6Pf4/AxE dQIDAQAB -----END PUBLIC KEY-----	private_key.pem - Notepad File Edit Format View Help -----BEGIN RSA PRIVATE KEY----- MIIEowIBAAKCAQEAkZ6m0eiVv35V1AW6Q4ZtmVPEBTE/68FmMHtJN7nUl6KsC1s6 RG7aqzBGS+q9COLCpyjHCdxhKg6/zy6qAnRiUp2dZAYiBgJwrGzY/92N6jJsm9oF MChNtb1DYxzHs4WeRgGDKz5a19ylbqM/kaqWRkZ4cH3xqw8SE1fgMY2DS3XoJZSO mbocmrxc4FEXbK0GHAKsWVGizms58x+l4Y6hrCvN6dQ13xq9a+pZvkzr5fiARcls UJTpBn5YShuweDR9QrsOsodXoLjODv5hZlhSsrt9mF8w0XnxQkGOw0U+OlwR4IAC 4nLPd+oNkDc5Tw2GEbhEs8rw3lDo6Pf4/AxEdQIDAQABAoIBACMzXQOWa+aNsU0u skNC8mjzTwnrSRaiGmQDVWRnYzzhiaG4JIpo2qqxoXiWDqqgeCnkyVJ4p7M9PAAm +VldDWyHpcWJVv2pZtfDeyREyBqyJD97cTAeOgTB8tF5Ykvi86rcEDtqmBagw4vO Yh5CBJAJ1nDo/PubCcVLw4nu+atKgHf1SZ9GvwJ08rrOO4ugtkZDKH1ZU2i0f4Df F+biCcNYme+N2yKV3f/GV2UcdDSKtwWa6ogx4p9cUwNw6LpSI/X3IXuai5/oWoOR TjVAOZPV6jMqYimK2Xs8XbpcTx5c/+22rsi2WQxUqW/2sMiIb2AOR4/zljUuz1jo +xNSZMkCgYEAte4L9+HCgfrq2Gpbw6QXYe1YyDOFPZC4dxDDKhqnJwaunh/4W9vv E/QzOZ8ucGpULrjzCPixBu1O3vAle1anNqmxwVdDg3/Kt9tPoVzde/tyEorDrydr ctfFa+UbNS0PGVuzX/Jx7psLfaS+FjHzLNlbl95Bx5GuzOLrnCV/cx0CgYEAzOge U+o6lnz+6wYf0hQ2zcn1uAIjRZrHCKd5onFMEYk1VniuiKWpGv0VtyGU7uKEdqWT B53/rA7BZhXQqq2muN7qdHI2+3kqEDVC1XBremBoBPH1HuyIJhyqAUj8G1uRAT/Z izSIcVo88tAVyipF9cBptKnjFdr22TSPx0+dvzkCgYAwd/8QR0E+x86RYaSY9rqU CFvC2pcA0/GRJxIXwYoOx6T106ajMnD7a+/JxjLlGC7dU8DL/xWKtuZwVqJlvaxG z+bfj5QZEI69dNw+LTOEuHSno/iGTYY+zmV2XU9mV17Sbq/XPAxGeWASVTWfTCdx J2dTio8T0Cg1xa7npsXRsQKBgFd6hXAX4Hlc1cvqqm0auibzINc0+m/92xkjvrcO HYajCEVpckHeT6q4/6tWR+oQhLLOj03CM2P2TLTTfWlQh4wHhxq9pCGKId4Xjvfm bcwqGl56Gcbg/mMe+6leylq3pdlds/dNLy8NclXLZnIPyBxZ0AcUPGGZQKeMXBGg ON1pAoGBAIrm+Kgi9PDpXX8HcUMaXa3Bjnq+W0JbVeTHzQsaHeOvXpWgiEKJVwOM cZmnJjWJ7mnt9Vzwzeeqcyu QzOk3qsMPnZCwc1dKS7u9Z+u5o8uAgbubmv1OvLM6 9matVGB6v5rHIPOxk0v0ZnQxD3rfQ86+qT/3YRIt75I0Puttxvhv -----END RSA PRIVATE KEY-----

And the file.txt:



Run the following command in a terminal for encryption and decryption respectively:

- Encryption:

```
python encrypt-decrypt.py --encrypt --algorithm rsa --key public_key.pem --input
file.txt --output rsa.enc
```

- Decryption:

```
python encrypt-decrypt.py --decrypt --algorithm rsa --key private_key.pem --input
rsa.enc --output rsa-decrypted.txt
```

Results:

ecrypted file	decrypted file