HW4

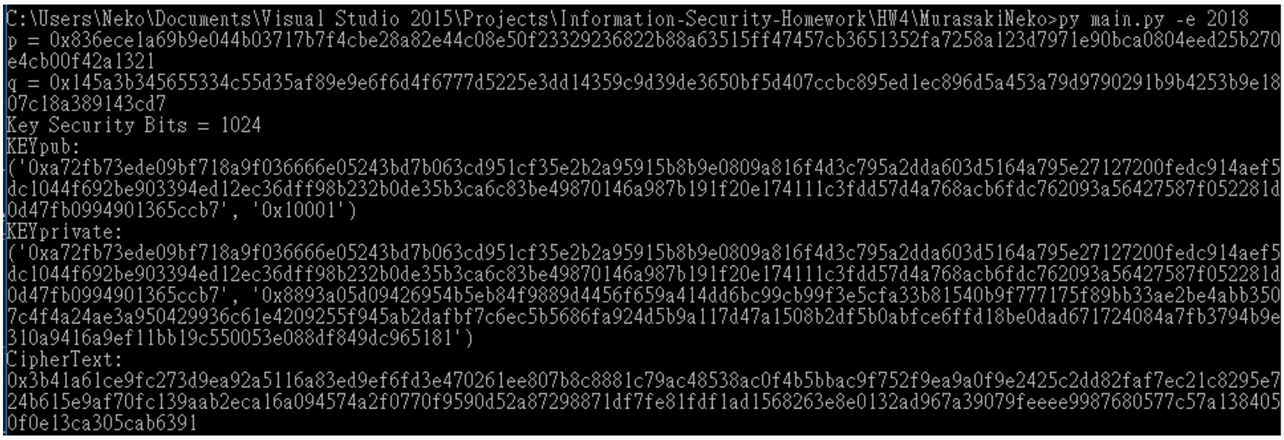
建置環境

- Python 3.6.5 :: Anaconda, Inc. on windows 10

使用方式

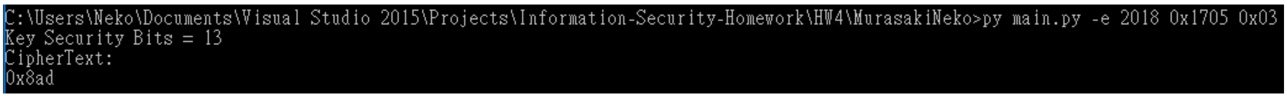
- 輸入格式說明:
  僅有 plaintext、ciphertext 可以輸入 10、16 進制的數值，ex `2018` or `0x7e2`
  其餘如 p、q、key 等等皆須輸入 16 進制的數值，長度不限
  輸出皆為 16 進制
- 加密:

1. 不指定 Key，命令列輸入`python main.py -e <plaintext>`
   (Key 為自動隨機產生共 1024bits p=512bits q=513bits，質數驗證進行 20 次 Miller Rabin Test)



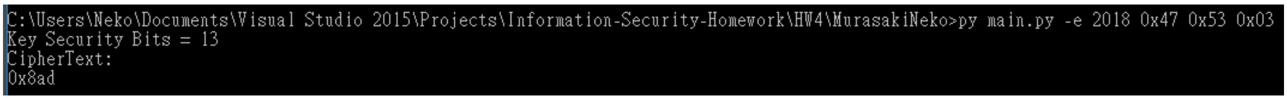
```
C:\Users\Neko\Documents\Visual Studio 2015\Projects\Information-Security-Homework\HW4\MurasakiNeko>py main.py -e 2018
p = 0x836ece1a69b9e044b03717b7f4cbe28a82e44c08e50f23329236822b88a63515ff47457cb3651352fa7258a123d7971e90bca0804eed25b270
e4cb00f42a1321
q = 0x145a3b345655334c55d35af89e9e6f6d4f6777d5225e3dd14359c9d39de3650bf5d407ccbc895ed1ec896d5a453a79d9790291b9b4253b9e18
07c18a389143cd7
Key Security Bits = 1024
KEYpub:
('0xa72fb73ede09bf718a9f036666e05243bd7b063cd951cf35e2b2a95915b8b9e0809a816f4d3c795a2dda603d5164a795e27127200fedc914aef5
dc1044f692be903394ed12ec36dff98b232b0de35b3ca6c83be49870146a987b191f20e174111c3fdd57d4a768acb6fdc762093a56427587f052281d
0d47fb0994901365ccb7', '0x10001')
KEYprivate:
('0xa72fb73ede09bf718a9f036666e05243bd7b063cd951cf35e2b2a95915b8b9e0809a816f4d3c795a2dda603d5164a795e27127200fedc914aef5
dc1044f692be903394ed12ec36dff98b232b0de35b3ca6c83be49870146a987b191f20e174111c3fdd57d4a768acb6fdc762093a56427587f052281d
0d47fb0994901365ccb7', '0x8893a05d09426954b5eb84f9889d4456f659a414dd6bc99cb99f3e5cfa33b81540b9f777175f89bb33ae2be4abb350
7c4f4a24ae3a950429936c61e4209255f945ab2dafbf7c6ec5b5686fa924d5b9a117d47a1508b2df5b0abfce6ffd18be0dad671724084a7fb3794b9e
310a9416a9ef11bb19c550053e088df849dc965181')
CipherText:
0x3b41a61ce9fc273d9ea92a5116a83ed9ef6fd3e470261ee807b8c8881c79ac48538ac0f4b5bbac9f752f9ea9a0f9e2425c2dd82faf7ec21c8295e7
24b615e9af70fc139aab2eca16a094574a2f0770f9590d52a87298871df7fe81fdf1ad1568263e8e0132ad967a39079feeee9987680577c57a138405
0f0e13ca305cab6391
```

2. 使用 public key 進行加密，命令列輸入`python main.py -e <plaintext> <n> <public key>`



```
C:\Users\Neko\Documents\Visual Studio 2015\Projects\Information-Security-Homework\HW4\MurasakiNeko>py main.py -e 2018 0x1705 0x03
Key Security Bits = 13
CipherText:
0x8ad
```

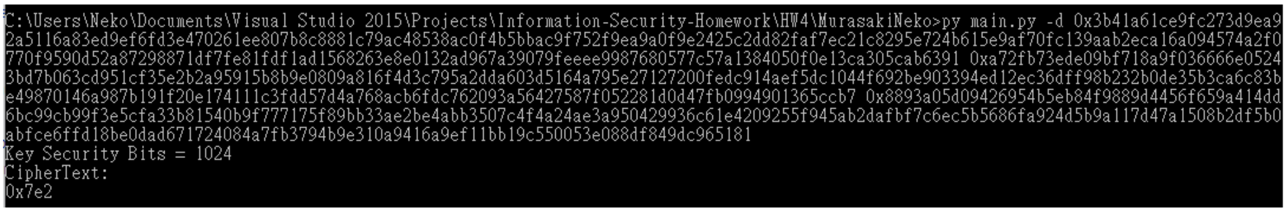
3. 使用兩個質數與 public key ,
   命令列輸入`python main.py -e <plaintext> <prime p> <prime q> <public key>`



```
C:\Users\Neko\Documents\Visual Studio 2015\Projects\Information-Security-Homework\HW4\MurasakiNeko>py main.py -e 2018 0x47 0x53 0x03
Key Security Bits = 13
CipherText:
0x8ad
```

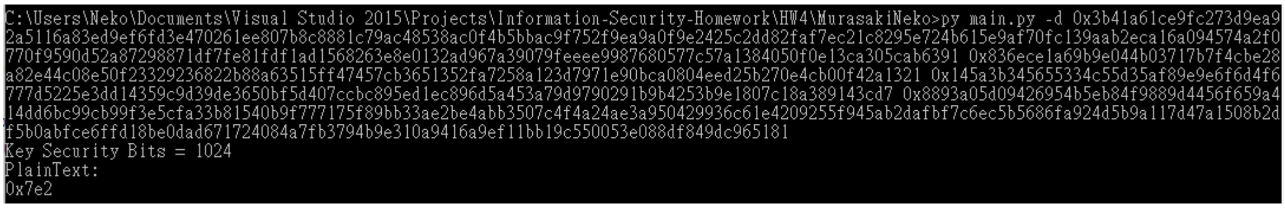
- 解密:

1. 使用 private key 進行解密(僅有 Square & multiply),
   命令列輸入`python main.py -d <ciphertext> <n> <private key>`
   (這裡用上面自動產生的 key 做示範)



```
C:\Users\Neko\Documents\Visual Studio 2015\Projects\Information-Security-Homework\HW4\MurasakiNeko>py main.py -d 0x3b41a61ce9fc273d9ea9
2a5116a83ed9ef6fd3e470261ee807b8c8881c79ac48538ac0f4b5bbac9f752f9ea9a0f9e2425c2dd82faf7ec21c8295e724b615e9af70fc139aab2eca16a094574a2f0
770f9590d52a87298871df7fe81fdf1ad1568263e8e0132ad967a39079feeee9987680577c57a1384050f0e13ca305cab6391 0xa72fb73ede09bf718a9f036666e0524
3bd7b063cd951cf35e2b2a95915b8b9e0809a816f4d3c795a2dda603d5164a795e27127200fedc914aef5dc1044f692be903394ed12ec36dff98b232b0de35b3ca6c83b
e49870146a987b191f20e174111c3fdd57d4a768acb6fdc762093a56427587f052281d0d47fb0994901365ccb7 0x8893a05d09426954b5eb84f9889d4456f659a414dd
6bc99cb99f3e5cfa33b81540b9f777175f89bb33ae2be4abb3507c4f4a24ae3a950429936c61e4209255f945ab2dafbf7c6ec5b5686fa924d5b9a117d47a1508b2df5b0
abfce6ffd18be0dad671724084a7fb3794b9e310a9416a9ef11bb19c550053e088df849dc965181
Key Security Bits = 1024
CipherText:
0x7e2
```

2. 使用兩個質數與 private key 進行解密(中國餘式定理),
   命令列輸入`python main.py -d <ciphertext> <prime p> <prime q> <private key>`



```
C:\Users\Neko\Documents\Visual Studio 2015\Projects\Information-Security-Homework\HW4\MurasakiNeko>py main.py -d 0x3b41a61ce9fc273d9ea9
2a5116a83ed9ef6fd3e470261ee807b8c8881c79ac48538ac0f4b5bbac9f752f9ea9a0f9e2425c2dd82faf7ec21c8295e724b615e9af70fc139aab2eca16a094574a2f0
770f9590d52a87298871df7fe81fdf1ad1568263e8e0132ad967a39079feeee9987680577c57a1384050f0e13ca305cab6391 0x836ece1a69b9e044b03717b7f4cbe28
a82e44c08e50f23329236822b88a63515ff47457cb3651352fa7258a123d7971e90bca0804eed25b270e4cb00f42a1321 0x145a3b345655334c55d35af89e9e6f6d4f6
777d5225e3dd14359c9d39de3650bf5d407ccbc895ed1ec896d5a453a79d9790291b9b4253b9e1807c18a389143cd7 0x8893a05d09426954b5eb84f9889d4456f659a4
14dd6bc99cb99f3e5cfa33b81540b9f777175f89bb33ae2be4abb3507c4f4a24ae3a950429936c61e4209255f945ab2dafbf7c6ec5b5686fa924d5b9a117d47a1508b2d
f5b0abfce6ffd18be0dad671724084a7fb3794b9e310a9416a9ef11bb19c550053e088df849dc965181
Key Security Bits = 1024
PlainText:
0x7e2
```

實作過程困難與心得

- 在撰寫的過程中，在做平方時最後忘記除以模數，以至於算到後面數字很大，嚴重影響效能，還以為是加速不夠，幸好最後有發現!
- 在設計輸入輸出時也遇到一些問題，例如要輸入哪些資訊來進行加解密比較合理、或者是要輸入 16 進制的值還是 10 進制也煩惱了一陣子。