對稱式金鑰加密與訊息認證實作

612410070 劉品彣

首先，先讀取被加密檔案（secretFile.txt），並印出其檔案的大小（134830082 bytes）

```
68  if __name__ == "__main__":
69
70      ##### 1. 請使用加密相關的library，實作加密檔案的程式(檔案需至少100MB)。#####
71
72      # read file
73      with open('secretFile.txt', 'rb') as f:
74          data = f.read()
75          dataSize = os.path.getsize('secretFile.txt')
76          print('the size of the file:', dataSize, ' bytes')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  5752802833.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

(1) 使用 AES-CCM mode 加密和解密

a. 定義 AES-CCM mode 加密函式

其中有三個參數：key 為加密用的金鑰、data 為要被加密的資料內容、nonce 為在串流加密上只使用一次的隨機數，用於確保安全。

- Line 14: 使用金鑰去建立一個 AESCCM 的物件
- Line 15: 使用剛建立的 AESCCM 物件去對資料 data 進行加密，其中使用到一次性的數值 nonce，而 None 表示沒有附加資料
- Line 13,16,17: 記錄加密的起始和結束時間，並計算整個加密所花費的時間
- Line 19: 回傳加密後的資料（encryptData）和加密所花費的時間（total_time）

```
12  def AESCCM_encrypt(key, data, nonce): #使用AES-CCM mode加密
13      start = time.time()
14      aesccm = AESCCM(key)
15      encryptData = aesccm.encrypt(nonce, data, None)
16      end = time.time()
17      total_time = end - start
18
19      return encryptData,total_time
```

b. 呼叫 AES-CCM mode 加密函式

傳入三個引數： key 為加密用的金鑰、data 為要被加密的資料內容、

AESCCM_nonce 為在 AES-CCM mode 加密方法中只使用一次的隨機數，最後得到加密資料和加密花費時間。

再以二進制的方式開啟一個名為 AESCCM_CipherFile.txt 的檔案並寫入以 AES-CCM mode 所加密的資料。

最後印出使用 AES-CCM mode 加密檔案的速度（每秒加密多少 data bytes）。

```
83      #AES-CCM encrypt
84      AESCCM_encryptData,AESCCM_time = AESCCM_encrypt(key, data, AESCCM_nonce)
85      AESCCM_CipherFile = open("AESCCM_CipherFile.txt","wb")
86      AESCCM_CipherFile.write(AESCCM_encryptData)
87      print('speed of AES-CCM encryption: ', str(dataSize/AESCCM_time), 'bytes/s')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

c. 定義 AES-CCM mode 解密函式

其中有三個參數：key 為解密用的金鑰、data 為要被解密的資料內容、nonce 為在串流解密上只使用一次的隨機數，用於確保安全。

- Line 42: 使用金鑰去建立一個 AESCCM 的物件
- Line 43: 使用剛建立的 AESCCM 物件去對資料 data 進行解密，其中使用到一次性的數值 nonce，而 None 表示沒有附加資料
- Line 46: 回傳解密後的資料（decryptData）

```
42  def AESCCM_decrypt(key, data, nonce):    #使用AES-CCM mode解密
43      aesccm = AESCCM(key)
44      decryptData = aesccm.decrypt(nonce, data, None)
45
46      return decryptData
```

d. 呼叫 AES-CCM mode 解密函式

傳入三個引數： key 為解密用的金鑰、AESCCM_encryptData 為要被解密的已加密資料內容、AESCCM_nonce 為在 AES-CCM mode 解密方法中只使用一次的隨機數，最後得到解密資料。

再開啟一個名為 AESCCM_PlaintFile.txt 的檔案並寫入以 AES-CCM mode 所解密的資料。

```
101     #AES-CCM decrypt
102     AESCCM_decryptData = AESCCM_decrypt(key, AESCCM_encryptData, AESCCM_nonce)
103     AESCCM_PlaintFile = open("AESCCM_PlaintFile.txt","w")
104     AESCCM_PlaintFile.write(str(AESCCM_decryptData))
```

e. 驗證 AES-CCM mode 加解密實作是否正確

若解密後的資料內容和原本的檔案內容相同，則表示加解密實作正確；反之則表示加解密實作有誤。

```
116     # Vertify the result of implemetations
117     print('Is the implementation of AES-CCM encryption and decryption correct?')
118     if (AESCCM_decryptData == data):
119         print('Correct!')
120     else:
121         print('Incorrect!')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

(2) 使用 AES-CTR mode (counter mode)加密和解密

a. 定義 AES-CTR mode (counter mode) 加密函式

其中有三個參數：key 為加密用的金鑰、data 為要被加密的資料內容、iv 為初始化向量，用於起始計數器的值。

- Line 23: 建立了一個 Cipher 對象，使用 AES algorithm 並且以 CTR（counter）mode 進行加密。
- Line 24: 建立了一個加密器，用於執行加密操作
- Line 25: 進行加密， update()函數將資料 data 傳遞給加密器，然後 finalize()函數完成加密過程，在 AES-CTR mode 下，這個方法沒有任何輸入，僅用於告訴加密器當前沒有更多的資料要加密了，可以返回最終的結果。最後將兩函數的結果串接起來，形成最終的加密資料。
- Line 22,26,27: 記錄加密的起始和結束時間，並計算整個加密所花費的時間
- Line 29: 回傳加密後的資料（encryptData）和加密所花費的時間（total_time）

```
21  def AESCTR_encrypt(key, data, iv): #使用AES-CTR mode (counter mode)加密
22      start = time.time()
23      cipher = Cipher(algorithms.AES(key), modes.CTR(iv))
24      encryptor = cipher.encryptor()
25      encryptData = encryptor.update(data) + encryptor.finalize()
26      end = time.time()
27      total_time = end - start
28
29      return encryptData,total_time
```

b. 呼叫 AES-CTR mode (counter mode) 加密函式

傳入三個引數： key 為加密用的金鑰、data 為要被加密的資料內容、 iv 為初始

化向量，用於起始計數器的值。最後得到加密資料和加密花費時間。
再以二進制的方式開啟一個名為 AESCTR_CipherFile.txt 的檔案並寫入以 AES- CTR mode 所加密的資料。
最後印出使用 AES-CTR mode 加密檔案的速度（每秒加密多少 data bytes）。

```
89      #AES-CTR encrypt
90      AESCTR_encryptData,AESCTR_time = AESCTR_encrypt(key, data, iv)
91      AESCTR_CipherFile = open("AESCTR_CipherFile.txt","wb")
92      AESCTR_CipherFile.write(AESCTR_encryptData)
93      print('speed of AES-CTR encryption: ', str(dataSize/AESCTR_time), 'bytes/s')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

c. 定義 AES-CTR mode (counter mode) 解密函式
其中有三個參數：key 為解密用的金鑰、data 為要被解密的資料內容、 iv 為初始化向量，用於起始計數器的值。最後得到加密資料和加密花費時間。

- Line 49: 建立了一個 Cipher 對象，使用 AES algorithm 並且以 CTR（counter）mode 進行解密。
- Line 50: 建立了一個解密器，用於執行解密操作
- Line 51: 進行解密， update()函數將資料 data 傳遞給解密器，然後 finalize()函數完成解密過程，在 AES-CTR mode 下，這個方法沒有任何輸入，僅用於告訴解密器當前沒有更多的資料要解密了，可以返回最終的結果。最後將兩函數的結果串接起來，形成最終的解密資料。
- Line 53: 回傳解密後的資料（decryptData）

```
48  def AESCTR_decrypt(key, data, iv): #使用AES-CTR mode (counter mode)解密
49      cipher = Cipher(algorithms.AES(key), modes.CTR(iv))
50      decryptor = cipher.decryptor()
51      decryptData = decryptor.update(data) + decryptor.finalize()
52
53      return decryptData
```

d. 呼叫 AES-CTR mode (counter mode) 解密函式
傳入三個引數： key 為加密用的金鑰、AESCTR_encryptData 為要被解密的已加密資料內容、 iv 為初始化向量，用於起始計數器的值。最後得到解密資料。
再開啟一個名為 AESCTR_PlaintFile.txt 的檔案並寫入以 AES-CTR mode 所解密的資料。

```
106     #AES-CTR decrypt
107     AESCTR_decryptData = AESCTR_decrypt(key, AESCTR_encryptData, iv)
108     AESCTR_PlaintFile = open("AESCTR_PlaintFile.txt","w")
109     AESCTR_PlaintFile.write(str(AESCTR_decryptData))
```

e. 驗證 AES-CTR mode (counter mode) 加解密實作是否正確

若解密後的資料內容和原本的檔案內容相同，則表示加解密實作正確；反之則表示加解密實作有誤。

```
123     print('\nIs the implementation of AES-CTR encryption and decryption correct?')
124     if (AESCTR_decryptData == data):
125         print('Correct!')
126     else:
127         print('Incorrect!')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

(3) 使用 ChaCha20 加密和解密

a. 定義 ChaCha20 加密函式

其中有三個參數：key 為加密用的金鑰、data 為要被加密的資料內容、nonce 為在串流加密上只使用一次的隨機數，用於確保安全。

- Line 33: 使用金鑰和 nonce 建立一個 ChaCha20 加密 algorithm 的物件，。
- Line 34: 建立了一個 Cipher 物件，使用 ChaCha20 加密 algorithm ，因為 ChaCha20 不需要額外的模式參數，所以未指定加密模式。
- Line 35: 建立了一個加密器，用於執行加密操作
- Line 36: 進行加密， update()函數將資料 data 傳遞給加密器
- Line 32,37,38: 記錄加密的起始和結束時間，並計算整個加密所花費的時間
- Line 39: 回傳加密後的資料（encryptData）和加密所花費的時間（total_time）

```
31  def ChaCha20_encrypt(key, data, nonce):    #使用ChaCha20加密
32      start = time.time()
33      algorithm = algorithms.ChaCha20(key, nonce)
34      cipher = Cipher(algorithm, mode=None)
35      encryptor = cipher.encryptor()
36      encryptData = encryptor.update(data)
37      end = time.time()
38      total_time = end - start
39      return encryptData,total_time
```

b. 呼叫 ChaCha20 加密函式

傳入三個引數： key 為加密用的金鑰、data 為要被加密的資料內容、ChaCha20_nonce 為在 ChaCha20 加密方法中只使用一次的隨機數，最後得到加密資料和加密花費時間。

再以二進制的方式開啟一個名為 ChaCha20_CipherFile.txt 的檔案並寫入以 ChaCha20 所加密的資料。

最後印出使用 ChaCha20 加密檔案的速度（每秒加密多少 data bytes）。

```
87      #ChaCha20 encrypt
88      ChaCha20_encryptData,ChaCha20_time = ChaCha20_encrypt(key, data, ChaCha20_nonce)
89      ChaCha20_CipherFile = open("ChaCha20_CipherFile.txt","wb")
90      ChaCha20_CipherFile.write(ChaCha20_encryptData)
91      print('speed of ChaCha20 encryption: ', str(dataSize/ChaCha20_time), 'bytes/s')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

c. 定義 ChaCha20 解密函式

其中有三個參數：key 為解密用的金鑰、data 為要被解密的資料內容、nonce 為在串流解密上只使用一次的隨機數，用於確保安全。

- Line 48: 使用金鑰和 nonce 建立一個 ChaCha20 解密 algorithm 的物件，。
- Line 49: 建立了一個 Cipher 物件，使用 ChaCha20 解密 algorithm ，因為 ChaCha20 不需要額外的模式參數，所以未指定解密模式。
- Line 50: 建立了一個解密器，用於執行解密操作
- Line 51: 進行解密， update()函數將資料 data 傳遞給解密器
- Line 53: 回傳解密後的資料（decryptData）

```
47  def ChaCha20_decrypt(key, data, nonce): #使用ChaCha20解密
48      algorithm = algorithms.ChaCha20(key, nonce)
49      cipher = Cipher(algorithm, mode=None)
50      decryptor = cipher.decryptor()
51      decryptData = decryptor.update(data)
52
53      return decryptData
```

d. 呼叫 ChaCha20 解密函式

傳入三個引數： key 為解密用的金鑰、 ChaCha20_encryptData 為要被解密的已加密資料內容、 ChaCha20_nonce 為在 ChaCha20 解密方法中只使用一次的隨機數，最後得到解密資料。

再開啟一個名為 ChaCha20_PlaintFile.txt 的檔案並寫入以 ChaCha20 所解密的資

料。

```
103     #ChaCha20 decrypt
104     ChaCha20_decryptData = ChaCha20_decrypt(key, ChaCha20_encryptData, ChaCha20_nonce)
105     ChaCha20_PlaintFile = open("ChaCha20_PlaintFile.txt","w")
106     ChaCha20_PlaintFile.write(str(ChaCha20_decryptData))
```

e. 驗證 ChaCha20 加解密實作是否正確

若解密後的資料內容和原本的檔案內容相同，則表示加解密實作正確；反之則表示加解密實作有誤。

```
121     print('\nIs the implementation of ChaCha20 encryption and decryption correct?')
122     if (ChaCha20_decryptData == data):
123         print('Correct!')
124     else:
125         print('Incorrect!')
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```

(4) 計算出"I love cryptography." 這個字串（不含雙引號）的 SHA-3-512 message digest，並說明你使用的計算工具或程式。

說明：SHA-3-512 是 SHA-3 家族中的一種雜湊算法，產生 512-bit 的雜湊值。這個雜湊值是對輸入訊息進行高度安全的雜湊後的結果，通常用於確保數據的完整性和安全性， SHA-3-512 的雜湊值通常以十六進制形式呈現。

定義了一個函數 hash，接收一個參數：string 為將進行雜湊的字串。

- Line 56: 建立了一個 SHA-3-512 的雜湊物件。其中 SHA-3-512 是 SHA-3 家族中的一種雜湊算法，產生 512-bit 的雜湊值。
- Line 57: 將輸入的字串轉換為 byte string，並更新到 SHA-3-512 的雜湊物件中。update 函數用於將字串資料添加到雜湊運算中。
- Line 58: 回傳最終為十六進制字串形式的雜湊值

印出"I love cryptography." 這個字串的 SHA-3-512 message digest。

```
55  def hash(string):   #cryptographic hashes
56      h = SHA3_512.new()
57      h.update(string.encode())
58      return h.hexdigest()
```

```
128     #####2. 請計算出"I love cryptography." 這個字串（不含雙引號）的SHA-3-512 message digest#####
129     #cryptographic hashes
130     string = "I love cryptography."
131     print('The digest of \"', string, '\" : ', hash(string))
```

```
(ENV1) C:\Users\Wen\OneDrive - 國立中正大學\桌面\CCU\密碼學\612410070_HW1>python HW1.py
the size of the file: 134830082  bytes
speed of AES-CCM encryption:  575280283.3816309 bytes/s
speed of AES-CTR encryption:  958804698.843246 bytes/s
speed of ChaCha20 encryption:  855639018.6765759 bytes/s
Is the implementation of AES-CCM encryption and decryption correct?
Correct!

Is the implementation of AES-CTR encryption and decryption correct?
Correct!

Is the implementation of ChaCha20 encryption and decryption correct?
Correct!

The digest of " I love cryptography. " :  d140015da1ff581b8f981790de927a3b00ff03df37d201c59899e9d143be8c736a8307
cece8f125b4413e19e63f1db2fe562aec6453833ff7eb80e411e520733
```