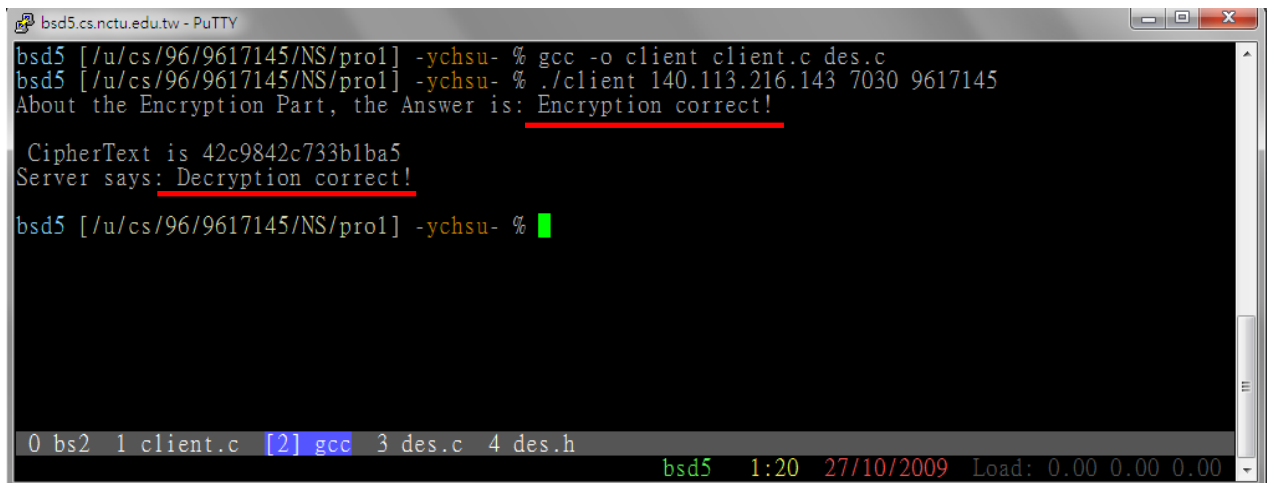


網路安全

Project 1 Report

9617145 資工 3C 許晏峻

1. 執行截圖：



```
bsd5 [/u/cs/96/9617145/NS/pro1] -ychsu- % gcc -o client client.c des.c
bsd5 [/u/cs/96/9617145/NS/pro1] -ychsu- % ./client 140.113.216.143 7030 9617145
About the Encryption Part, the Answer is: Encryption correct!
CipherText is 42c9842c733b1ba5
Server says: Decryption correct!
bsd5 [/u/cs/96/9617145/NS/pro1] -ychsu- % █

0 bs2 1 client.c [2] gcc 3 des.c 4 des.h
bsd5 1:20 27/10/2009 Load: 0.00 0.00 0.00
```

2. Code：

修改的部分為程式中註解 step 的地方：

首先加密部份：

1. 依照 function 使用方法 setkey
2. 接著將 plaintext 加密成 ciphertext
3. 然後開始處理 ciphertext 字串，用 for 迴圈依序讀出每個字元，然後用 `sprintf(my_tmp, "%.2x", tmp)` 來作 Hex 的 ASCII 轉換，.2x 是為了處理有 0 的時候預設會被忽略掉的情況
4. 最後再把 my_tmp 的 2Byte 字元存入 my_buf。

以上即完成加密部份。

接著解密部份：

1. 先將收到的 buf 將 Hex ASCII 轉回類似加密部份的 ciphertext 的字串。作法是用 for 迴圈一次讀出兩個字元(2 Byte)，接著用 `tmp_ascii=strtol(my_tmp2, NULL, 16);`，把 my_tmp2 中的字串轉成 long int type，再用 `sprintf(&tmp2, "%c", tmp_ascii);`，將這個 long int 轉回字元(1 Byte)存進 tmp2，最後將 tmp2 存進 ciphertext 中。
2. 接著將 ciphertext 解密成 recoverd。
`gl_des_ecb_decrypt(&context, ciphertext, recoverd);`。

以上即完成解密部份。

3. 遇到的問題：

一開始加密部份會有時候過，有時候不過，問題點就在當將密文轉成 Hex ASCII 時，若為例如 0x05，第一個字為 0 的會被忽略掉，造成錯誤，因此在格式中加上%.2x，將空格自動補 0 就可以解決。

4. 心得：

因為加解密的 `source code` 都給了，不用自己寫出加解密裡的東西，只要寫個 `client 端 code` 讓他可以 `call source code` 裡的 `function`，所以只要知道步驟，還算滿快的。最大的問題大概就像上面所說的字串處理部份，也可以當作字串處理的複習？忘記怎麼用的 `function` 去查一下很快就知道怎麼使用，所以在這邊花點時間就可以。這個 `project` 還算滿有意思的，讓我更了解加解密的過程。