# INFA723 Cryptography and Network Security

# Lab4 Use OpenSSL Crypto Library to Encrypt/Decrypt a File and Send/Receive through a Socket Connection

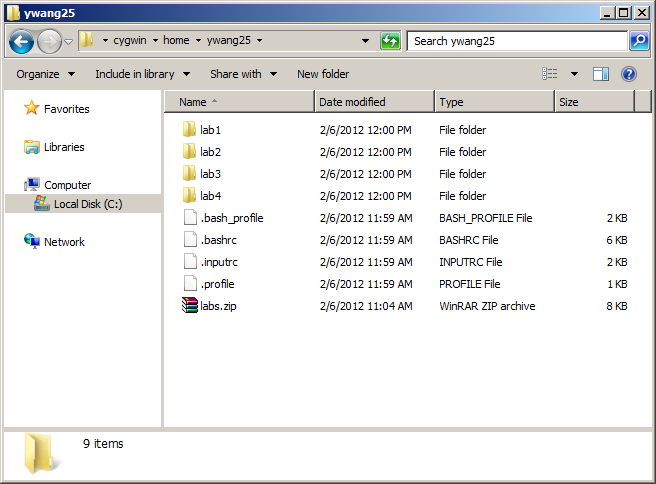
# 1 Introduction

OpenSSL toolkit provides a crypto library used for a wide range of crypto operations. In this exercise, we are going to learn how to use the crypto library to encrypt a file and send the file through a socket. We are also going to demo how to receive the file in the server side and decrypt the message. The cipher we are going to use is DES and the mode is ECB. A lab package has been posted on the class’ website. If you have not installed the package on your computer, follow Section 2 to install the package. If you already installed the package, go to Section 3.

Note that the sample program is for demo purpose only. The program use ‘\0’ as the end of a file. The program cannot work properly if the cipher text has a ‘\0’ in it. The lab procedure assumes the Cygwin is used. If you have a Linux OpenSSL environment, open a terminal and the lab procedure is similar.

# 2 Setup

A Lab package (labs1-4.zip) has been posted on the class’s website. Go to class’s website and download the package to your local computer. Assume you use the default Cygwin installation directory “C:\cygwin”, copy labs.zip to “C:\cygwin\home” directory. Unzip the file to your home directory and you will see four Labs listed here: lab1, lab2, lab3, and lab4.



# Two execution files will be created in the Lab, client.exe and server.exe. The formation of these two programs is listed below.

client server\_ip\_address port\_num plain\_text\_file

server\_ip\_address: server’s IP address.

port\_num: server’s port number.

plain\_text\_file: a text file you want to send to server

server host\_ip\_address port\_num

host\_ip\_address: host IP address

port\_num: server’s port number

Note that when the compiler (gcc) links the object file, you need to tell the compiler to link with openssl library (libcrypto.a). Otherwise, you will see some error meesage as below:

$ gcc client.exe

client.exe:client.c:(.text+0x0): multiple definition of `\_mainCRTStartup'

/usr/lib/gcc/i686-pc-cygwin/3.4.4/../../../crt0.o:(.text+0x0): first defined her e

client.exe:client.c:(.text+0x0): multiple definition of `\_WinMainCRTStartup'

/usr/lib/gcc/i686-pc-cygwin/3.4.4/../../../crt0.o:(.text+0x0): first defined her e

collect2: ld returned 1 exit status

Check the Makefile to see how to link with openssl library.

# 3 Use OpenSSL Encrypt/Decrypt a File and Send/Receive through a Socket Connection

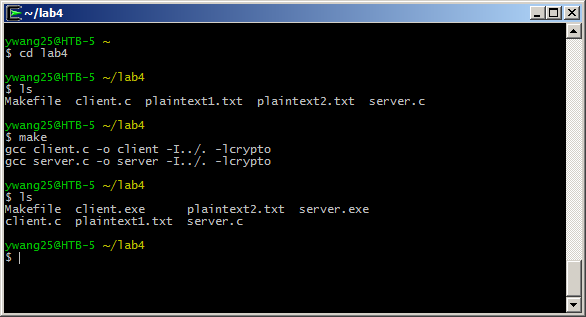
1. Open a cgywin window and go to “lab3” directory.

$ cd lab4

1. Check the files under the directory and make sure you have all the files as listed below:



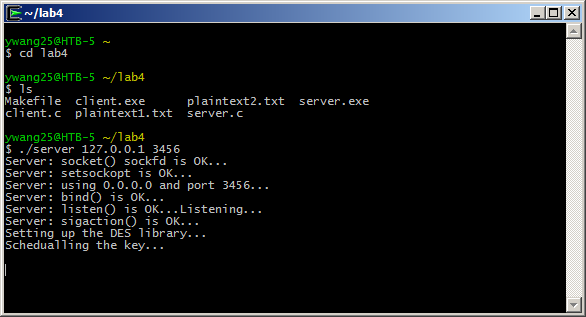
1. Enter and run “make” in the prompt. Two new files client.exe and server.exe will be created.



1. Open another Cygwin window (for server program, we will use previous one for client program) and go to “lab4” directory.
2. In the server Cygwin window, enter and run the command

$./server 127.0.0.1 3456

127.0.0.1 is your host IP address and 3456 is an unused port number.



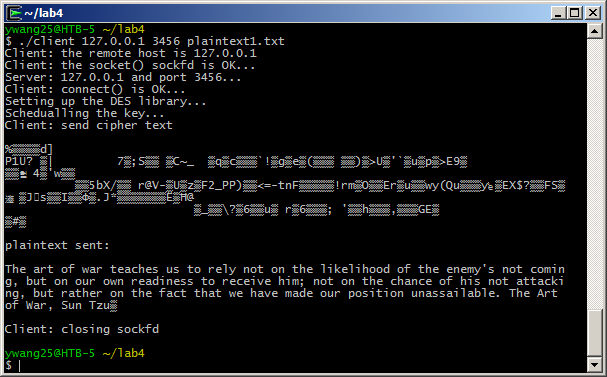
Windows will pop up a window to ask if you allow the access. Click the button “Allow access” to temporarily authorize the access.



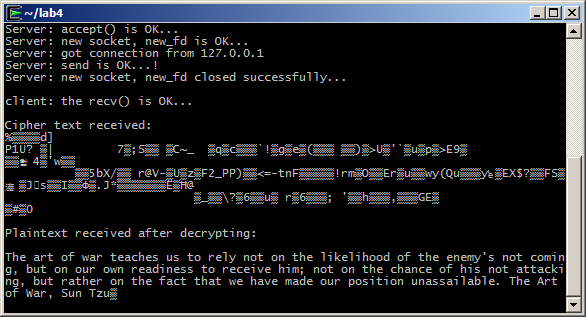
1. In the client Cygwin window, enter and run the command

$ ./client 127.0.0.1 3456 plaintext1.txt

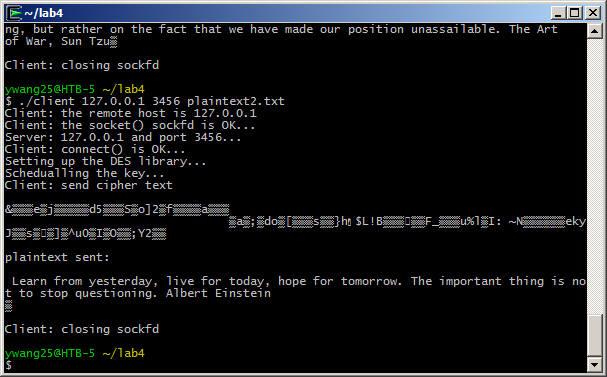
127.0.0.1 is you host IP address. 3456 is the port number and plaintext.txt is a clear text.



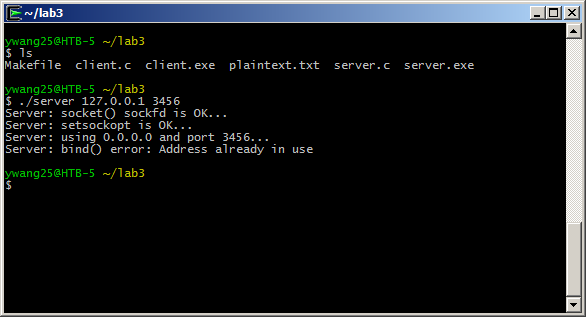
1. Check both client and server Cygwin windows and you will see a file is sent from the client to the server. Client Cygwin output is shown above and server Cygwin output is shown below.



1. You can enter and run the command “./client 127.0.0.1 3456 plaintext2.txt” and you will see the server will receive and decrypt another plaintext file.



1. Use a text editor to open Client.c and server.c to see how the source code is structured.
2. If you stop the server program (ctrl-d) and run the server program again. You may see a bind error message like below:



This is because the port number is not released after you terminated the server program. You can force kill the process (using kill) or simply use another port number to start the server program. Make sure the client and the server program use the same IP address and port number every time.

# 4 Questions

1. (5pts) Check the source code, client.c and server.c. Describe how the OpenSSL encryption and decryption functions are used in the source code.
2. (5pts) This lab demonstrates how to protect confidentiality via encryption for socket communications. As we discussed in the OSI security model, what security services are desired to protect data transmitted via socket communications? How to ensure/implement these security services as you identified?
3. (5pts) How to detect open ports in a computer?
4. (10pts) How to exploit an open pot in a computer? List at least two different approaches to exploit an open port.
5. (10pts) An initial port scanning doesn’t show any open ports in a network. Does it indicate the network is secure? What else you can do to exploit the network?
6. (5pts) What is a meet-in-middle attack?
7. (10pts) Is it possible to perform encryptions in parallel on multiple blocks of plaintext in CBC mode? How about decryption?
8. (10pts) What is SSL splitting? Explain a scenario how to use SSL splitting to monitor SSL/TLS traffic?