[수업]

#define CERTF HOME "foo-cert.pem" ==> #define CERTF HOME "servcert.pem"

#define KEYF HOME "foo-key.pem" ==> #define KEYF HOME "servey.pem"

servcert.pem is your certificate file and servkey.pem is your rsa key file

Do not copy and paste the commands; it may include hidden characters. Type the command by hand. And when to retype it, you can use the arrow key to replay previous commands.

: 명령어 칠 때 복사/붙이기 하면 숨은 글자가 포함되어 에러를 유발할 수 있습니다. 직접 타이핑하기 바랍니다. 이전 명령어를 다시 재플레이하려면 화살 키를 이용할 수 있습니다.

[homework]

1) Do the steps in Section 2.

=> 1) Copy openssl-1.0.1f.tar.gz into your home directory(assume it is /home/sec/12345), and un-tar it to create openssl-1.0.1f directory.

$ pwd

/home/sec/12345

$ cp ../../openssl-1.0.1f.tar.gz .

$ tar xvf openssl-1.0.1f.tar.gz

텍스트이(가) 표시된 사진

자동 생성된 설명

…

2) config

$ mkdir openssl

$ cd openssl-1.0.1f

$ pwd

/home/sec/12345/openssl-1.0.1f

Let config know it should use the **openssl** direcotry in the current user's home direcotry(/home/sec/12345).

$ ./config no-dso --prefix=/home/sec/12345/openssl --openssldir=/home/sec/12345/openssl

텍스트이(가) 표시된 사진

자동 생성된 설명

…..

3) compile openssl-1.0.1f library

$ make

will compile openssl library code. libssl.a, libcrypto.a will be created in the top directory (/home/sec/12345/openssl-1.0.1f).

텍스트이(가) 표시된 사진

자동 생성된 설명

….

4) Install

4.1) Remove "install\_docs" from Makefile (to reduce compile time) as below.

$ vi Makefile

and change

install: all install\_docs install\_sw

to

install: all install\_sw



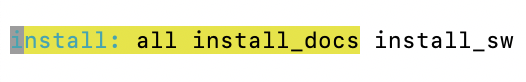
openssl-10… 폴더내의 Makefile 파일 내의 마지막 부분에 있는 install\_docs를 주석처리해줍니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이와같이 ..

&&



를

텍스트이(가) 표시된 사진

자동 생성된 설명

이로 변경해 줍니다!

4.2) Install

$ make install

will install openssl in /home/sec/12345/openssl.

Now you should have all necessary files in /home/sec/openssl directory.

All files will be installed in /home/sec/12345/openssl (make sure you have them all).

lib will have libssl.a, libcrypto.a

include/openssl will have header files

bin will have openssl binary files

certs will have certificates

private will have private keys

텍스트이(가) 표시된 사진

자동 생성된 설명

makefile 내용을 수정해 준후, make install 을 해줍니다.

5) Compile application: Compile cli.cpp and serv.cpp in demos directory.

5.1) Go to demos/ssl directory.

$ cd openssl-1.0.1f/demos/ssl

5.2) Generate server rsa key pair: servkey.pem

openssl genrsa –out servkey.pem 1024

5.3) Make a config file servconf.txt and generate server public key certificate(the server public key will be self-signed): servcert.pem

vi servconf.txt

........type the contents (refer to myconf.txt in lect8) ...........

openssl req –config servconf.txt –new –x509 –key servkey.pem –out servcert.pem

5.4) Change followings.

serv.cpp:

- change the port number

- change the file name for the certificate (CERTF) and key file (KEYF)

- change the return data type of main() to "int"

- change "size\_t client\_len" to "socklen\_t client\_len"

cli.cpp:

- change the server port number and IP address

- change the return data type of main() to "int"

- include <unistd.h>

- Change ssl version to TLSv1: use "TLSv1\_client\_method()" instead of "SSLv2\_client\_method()" in cli.cpp.

5.5) Compile the client and server

g++ -L/home/sec/12345/openssl/lib -I/home/sec/12345/openssl/include

-fpermissive -o serv serv.cpp -lssl -lcrypto -ldl

g++ -L/home/sec/12345/openssl/lib -I/home/sec/12345/openssl/include

-fpermissive -o cli cli.cpp -lssl -lcrypto -ldl

-L is option for additional library directory and -I for additional header file directory.

-lssl means "link with libssl.a" (for static library) or "link with libssl.so" (for dynamically linked library, or shared object) and -lcrypto "link with libcrypto.a" or "link with libcrypto.so".

5.6). Run the server first and run the client to see if they can talk.

2) Modify cli.cpp such that it displays "Start SSL protocol in client" before it calls SSL\_connect(ssl). Also modify serv.cpp such that it displays "Start SSL protocol in server" before it calls SSL\_accept(ssl). Recompile cli, serv, and rerun them to see the effect.



serv.cpp



cli.cpp



텍스트이(가) 표시된 사진

자동 생성된 설명

각 파일을 수정한대로 문자열이 출력되는 것을 확인할 수 있습니다.

3) cli.cpp calls SSL\_connect() which in turn calls ssl3\_connect() (defined in openssl-1.0.1f/ssl/s3\_clnt.c). Add

printf("ssl3\_connect begins\n");

in the beginning of ssl3\_connect(). Go to the SSL top directory (openssl-1.0.1f) and recompile ssl library with "make". Re-install ssl library with "make install". Now go to demos/ssl and recompile cli.cpp and serv.cpp and rerun them to see if the client prints "ssl3\_connect begins". If the output does not reflect your change, check the lib directory location in g++ command.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

정상적으로 실행되는 것을 확인할 수 있습니다.

4) serv.cpp calls SSL\_accept() which in turn calls ssl3\_accept() (defined in openssl-1.0.1f/ssl/s3\_srvr.c). Add

printf("ssl3\_accept begins\n");

in the beginning of ssl3\_accept(). Recompile and re-install ssl library. Recompile cli.cpp and serv.cpp and see if the server displays the above message.



텍스트이(가) 표시된 사진

자동 생성된 설명

코드를 수정한 후, 결과를 확인했습니다.

이또한 정상적으로 실행됩니다.

5) Modify ssl3\_connect(), ssl3\_accept() such that they print some message at each ssl protocol stage. Recompile ssl libraries, cli, serv, and rerun. Match the state changes in the client and the server with the state changes explained in Section 1.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

결롸를 보면, ssl 사이사이 추가적인 항목들이 있는 것을 확인할 수 있습니다.

client가 server로 hello를 보내고 server는 이를 받습니다.

이후 서버는 클라이언트로 hello를 보내면서 서버 인증서를 전송합니다.

이때 일반적으로 서버 인증서와 함께 서버의 공용키가 클라이언트에 전달됩니다.   
인증서가 없거나 인증서가 서명용으로 사용될 때 server key exchange 메세지를 보냅니다.

만약 클라이언트 인증서가 필요하다면, 인증서 요청도 함께 보내게 됩니다.

서버가 클라이언트에게 보낼 메세지를 모두 보냈음으로 hello done을 보냅니다.

클라이언트가 서버로 인증서를 보낼 때 클라이언트의 인증서 개인키로 암호화하고, 서버는 클라이언트 인증서 공개키를 이용해서 확인하게 됩니다.

client key를 서버에 보내고 change cipher spec으로 이후에 전송되는 모든 메세지는 서버와 협상된 알고리즘과 키를 이용해 암호화 시키겠다고 서버에 알립니다.

그리고 바로 finished 메세지를 서버에 전송합니다.

그렇게 핸드쉐이크가 종료됩니다.

5-1) Modify openssl library so that your ssl client program displays the premaster secret byte sequence.

..................

premaster secret size:48

premaster secret is:3 1 bd ee 28 .............61 c

위와같은 결과를 내기 위해서는 , openssl 라이브러리를 수정해주어야 합니다.

그과정은 다음과 같습니다.