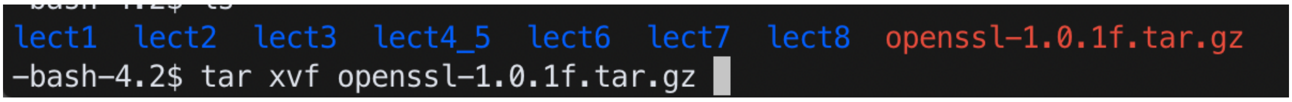
ssl-second

12141163 이욱진

Homework

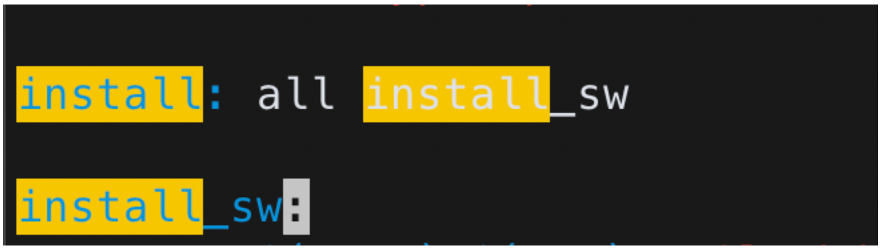
1) Do the steps in Section 2.



```
-bash-4.2$ ls
lect1  lect2  lect3  lect4_5  lect6  lect7  lect8  openssl-1.0.1f.tar.gz
-bash-4.2$ tar xvf openssl-1.0.1f.tar.gz
```

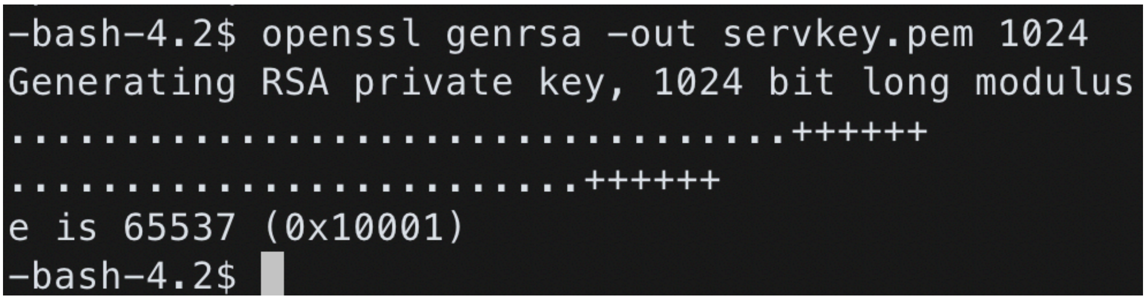
tar 명령어를 통해 압축을 풀었습니다.

config를 설정하고 make명령어를 실행한 후에 Makefile의 install부분을 다음과같이 수정하였습니다.



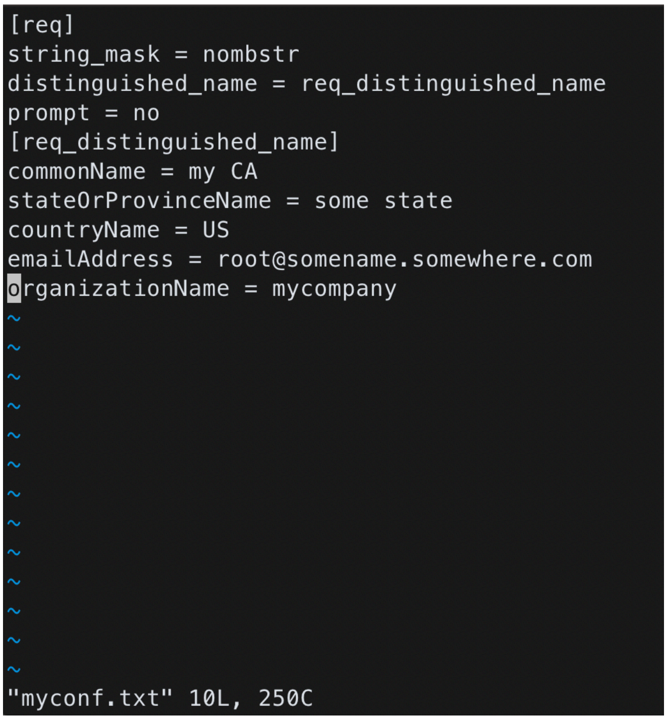
```
install: all install_sw

install_sw:
```

그 후 1024비트의 rsa key pair 를 생성하였습니다

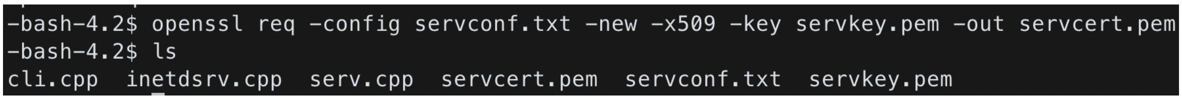


```
-bash-4.2$ openssl genrsa -out servkey.pem 1024
Generating RSA private key, 1024 bit long modulus
.................................++++++
..........................++++++
e is 65537 (0x10001)
-bash-4.2$
```

lect8의 myconf.txt 를 그대로 가져왔습니다.

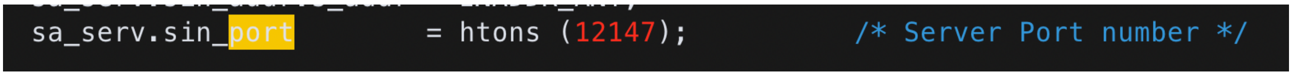


```
[req]
string_mask = nombstr
distinguished_name = req_distinguished_name
prompt = no
[req_distinguished_name]
commonName = my CA
stateOrProvinceName = some state
countryName = US
emailAddress = root@somename.somewhere.com
organizationName = mycompany
~
~
~
~
~
~
~
~
~
~
~
~
~
"myconf.txt" 10L, 250C
```



```
-bash-4.2$ openssl req -config servconf.txt -new -x509 -key servkey.pem -out servcert.pem
-bash-4.2$ ls
cli.cpp  inetdsrv.cpp  serv.cpp  servcert.pem  servconf.txt  servkey.pem
```

다음명령어를 통하여 servcert.pem을 생성하였습니다.

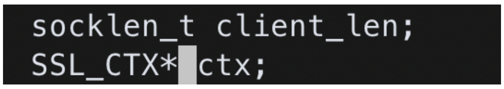


```
  sa_serv.sin_port         = htons (12147);          /* Server Port number */
```

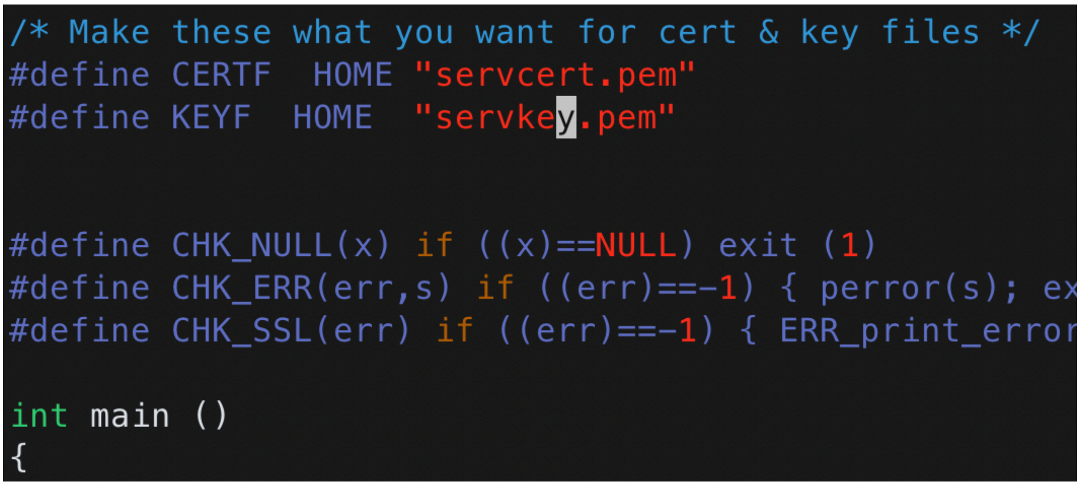
serv.cpp, cli.cpp 의 일부내용을 수정하였습니다.
certf,keyf 의 파일이름과 main문의 반환형 그리고 portnumber는 제가 생각한 12147로 설정해주었습니다.

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"



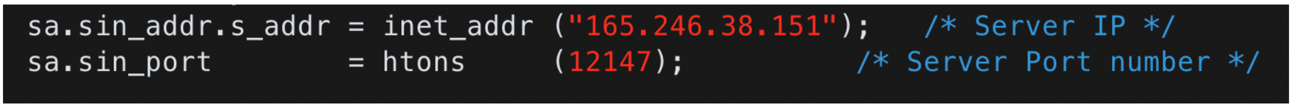
```
  socklen_t client_len;
  SSL_CTX* ctx;
```



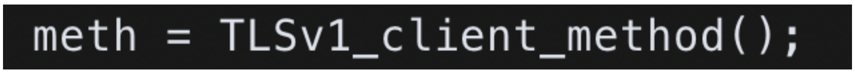
```
/* Make these what you want for cert & key files */
#define CERTF  HOME "servcert.pem"
#define KEYF  HOME  "servkey.pem"


#define CHK_NULL(x) if ((x)==NULL) exit (1)
#define CHK_ERR(err,s) if ((err)==-1) { perror(s); ex
#define CHK_SSL(err) if ((err)==-1) { ERR_print_error

int main ()
{
```

cli.cpp

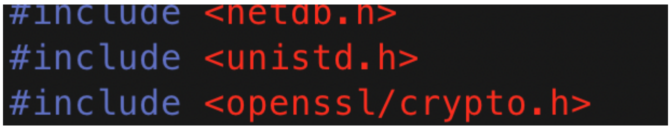
- change the server port number and IP address
- change the return data type of main() to "int"
- include <unistd.h>
- Change ssl version to TLSv1: use "TLSv1_client_method()" instead of "SSLv2_client_method()" in cli.cpp.



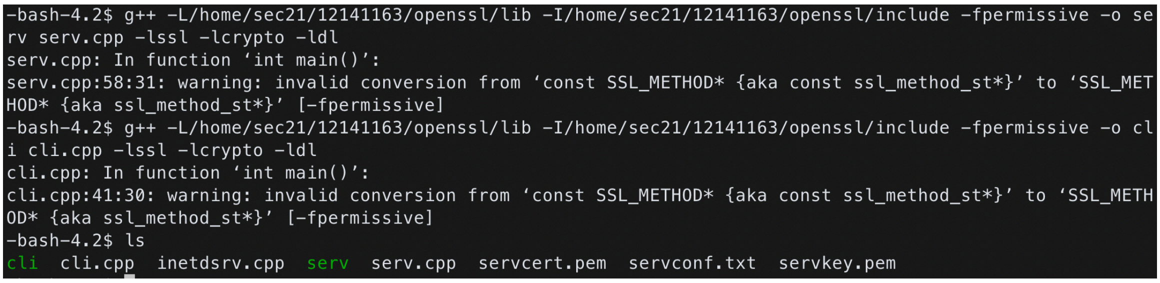
```
  sa.sin_addr.s_addr = inet_addr ("165.246.38.151");   /* Server IP */
  sa.sin_port        = htons     (12147);          /* Server Port number */
```



```
  meth = TLSv1_client_method();
```



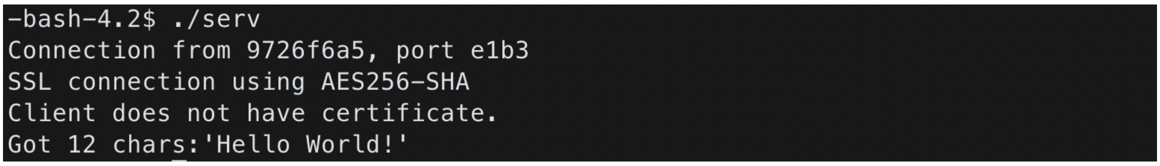
```
#include <netdb.h>
#include <unistd.h>
#include <openssl/crypto.h>
```



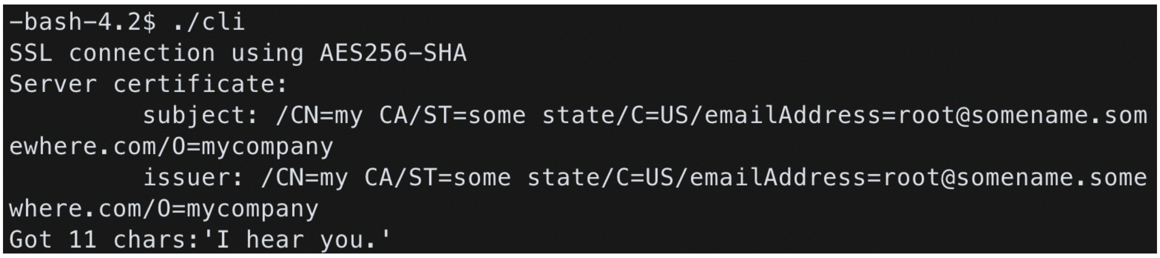
```
-bash-4.2$ g++ -L/home/sec21/12141163/openssl/lib -I/home/sec21/12141163/openssl/include -fpermissive -o se
rv serv.cpp -lssl -lcrypto -ldl
serv.cpp: In function ‘int main()’:
serv.cpp:58:31: warning: invalid conversion from ‘const SSL_METHOD* {aka const ssl_method_st*}’ to ‘SSL_MET
HOD* {aka ssl_method_st*}’ [-fpermissive]
-bash-4.2$ g++ -L/home/sec21/12141163/openssl/lib -I/home/sec21/12141163/openssl/include -fpermissive -o cl
i cli.cpp -lssl -lcrypto -ldl
cli.cpp: In function ‘int main()’:
cli.cpp:41:30: warning: invalid conversion from ‘const SSL_METHOD* {aka const ssl_method_st*}’ to ‘SSL_METH
OD* {aka ssl_method_st*}’ [-fpermissive]
-bash-4.2$ ls
cli  cli.cpp  inetdsrv.cpp  serv  serv.cpp  servcert.pem  servconf.txt  servkey.pem
```

serv.cpp 과 cli.cpp를 컴파일하였고 정상적으로 오브젝트파일이 생성되는것을 확인할 수 있었습니다.

서버를 처음 실행시키고 다른하나의 터미널에서는 클라이언트를 실행하니 다음과같이 연결됨을 알 수 있었습니다.



```
-bash-4.2$ ./serv
Connection from 9726f6a5, port e1b3
SSL connection using AES256-SHA
Client does not have certificate.
Got 12 chars:'Hello World!'
```

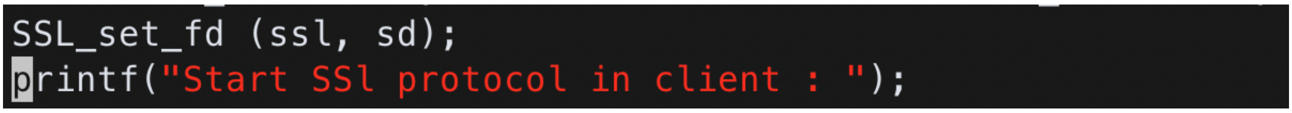


```
-bash-4.2$ ./cli
SSL connection using AES256-SHA
Server certificate:
         subject: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.som
ewhere.com/O=mycompany
         issuer: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.some
where.com/O=mycompany
Got 11 chars:'I hear you.'
```

클라이언트의 내용을보니 이전의 myconf.txt 의작성했던 인증서의 내용이 담겨있었습니다.

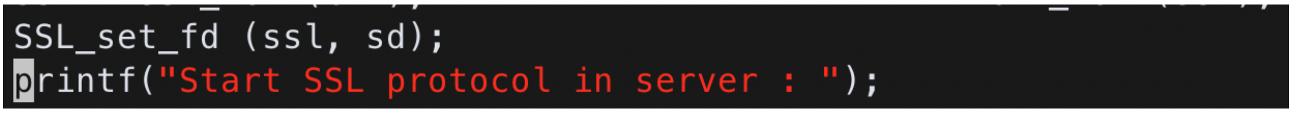
2) Modify cli.cpp such that it displays "Start SSL protocol in client" before it calls SSL_connect(ssl). Also modify serv.cpp such that it displays "Start SSL protocol in server" before it calls SSL_accept(ssl). Recompile cli, serv, and rerun them to see the effect.

cli.cpp



```
SSL_set_fd (ssl, sd);
printf("Start SSl protocol in client : ");
```

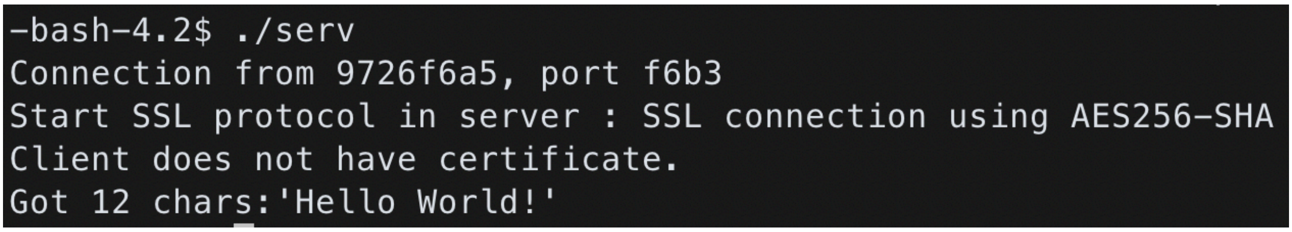
serv.cpp



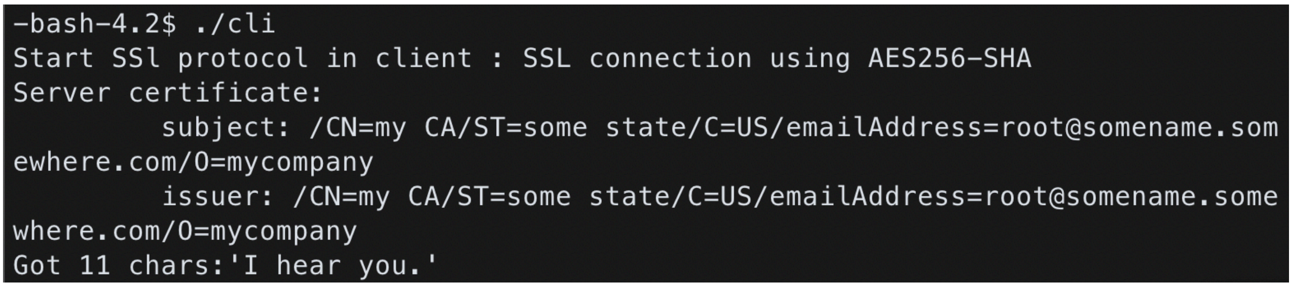
```
SSL_set_fd (ssl, sd);
printf("Start SSL protocol in server : ");
```

SSL_connect, SSl_accept 이전의 출력문을 추가하였습니다.

그 후 다음과같이 출력되었습니다.



```
-bash-4.2$ ./serv
Connection from 9726f6a5, port f6b3
Start SSL protocol in server : SSL connection using AES256-SHA
Client does not have certificate.
Got 12 chars:'Hello World!'
```



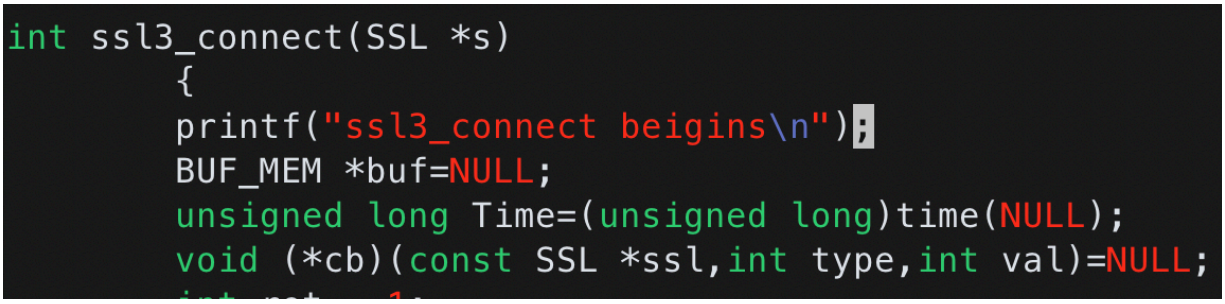
```
-bash-4.2$ ./cli
Start SSl protocol in client : SSL connection using AES256-SHA
Server certificate:
         subject: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.som
ewhere.com/O=mycompany
         issuer: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.some
where.com/O=mycompany
Got 11 chars:'I hear you.'
```

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");

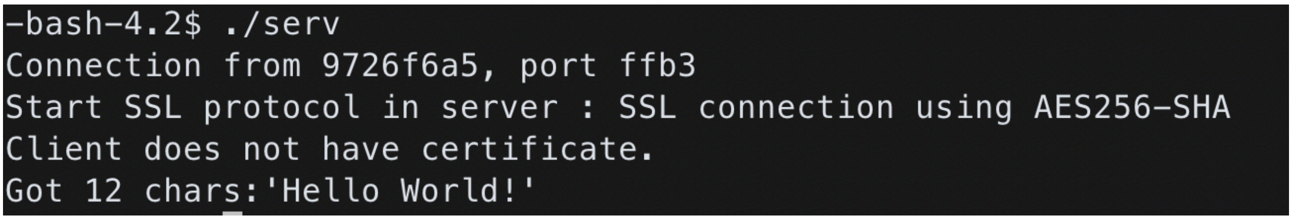
in the beginning of ssl3_connect(). Go to the SSL top directory (openssl-1.0.1f) and recompile ssl library with "make". Re-install ssl library with "make install". Now go to demos/ssl and recompile cli.cpp and serv.cpp and rerun them to see if the client prints "ssl3_connect begins". If the output does not reflect your change, check the lib directory location in g++ command.

openssl-1.0.1f/ssl/s3_clnt.c 파일내부 ssl3_connect함수의 첫줄에 출력문을 추가하였습니다.

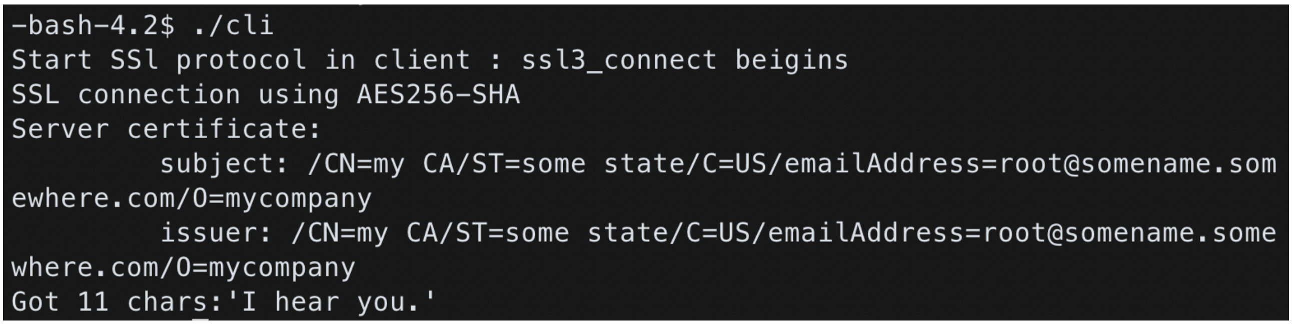


```
int ssl3_connect(SSL *s)
        {
        printf("ssl3_connect beigins\n");
        BUF_MEM *buf=NULL;
        unsigned long Time=(unsigned long)time(NULL);
        void (*cb)(const SSL *ssl,int type,int val)=NULL;
```

그 후 컴파일을 다시한 후에 실행시켜보았습니다.



```
-bash-4.2$ ./serv
Connection from 9726f6a5, port ffb3
Start SSL protocol in server : SSL connection using AES256-SHA
Client does not have certificate.
Got 12 chars:'Hello World!'
```



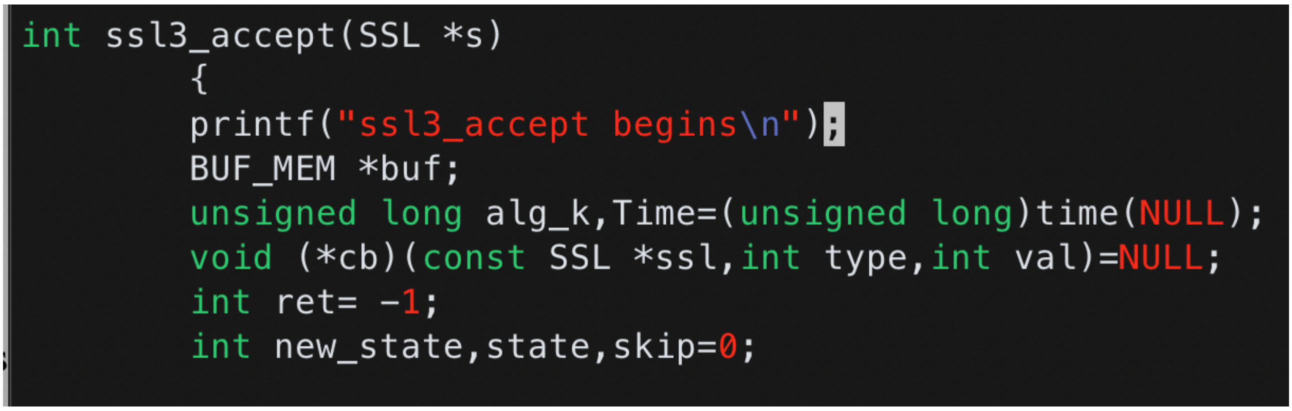
```
-bash-4.2$ ./cli
Start SSl protocol in client : ssl3_connect beigins
SSL connection using AES256-SHA
Server certificate:
         subject: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.som
ewhere.com/O=mycompany
         issuer: /CN=my CA/ST=some state/C=US/emailAddress=root@somename.some
where.com/O=mycompany
Got 11 chars:'I hear you.'
```

ssl3_connect begins문구를 확인할 수 있었습니다.

4) serv.cpp calls SSL_accept() which in turn calls ssl3_accept() (defined in openssl-1.0.1f/ssl/s3_srvr.c). Add

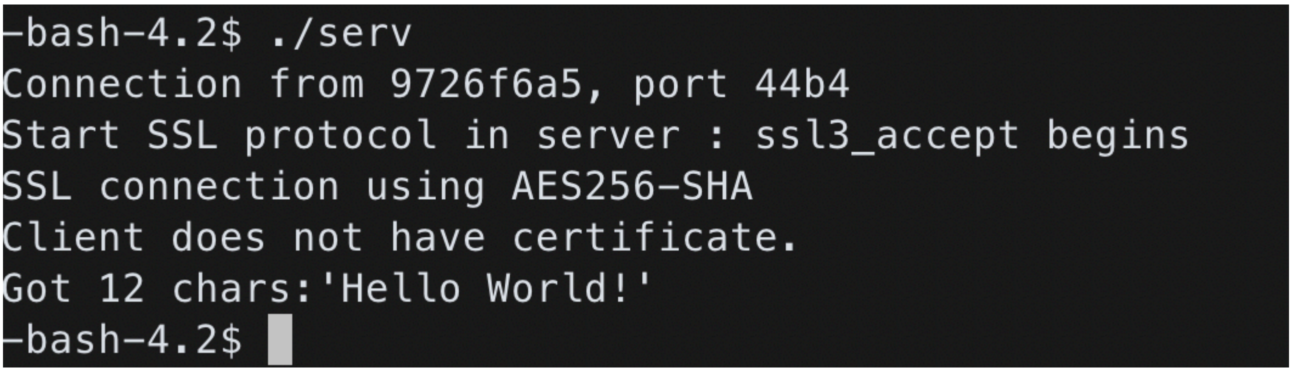
    printf("ssl3_accept begins\n");

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.



```
int ssl3_accept(SSL *s)
        {
        printf("ssl3_accept begins\n");
        BUF_MEM *buf;
        unsigned long alg_k,Time=(unsigned long)time(NULL);
        void (*cb)(const SSL *ssl,int type,int val)=NULL;
        int ret= -1;
        int new_state,state,skip=0;
```

s3_srvr.c 파일내부에서 ssl3_accept()를 찾을 수 있었고 출력문을 추가해주었습니다.
다시 컴파일을 진행하고 출력해보았을경우 serv에서 위 추가한 출력문이 나타나는것을 확인할 수 있었습니다.



```
-bash-4.2$ ./serv
Connection from 9726f6a5, port 44b4
Start SSL protocol in server : ssl3_accept begins
SSL connection using AES256-SHA
Client does not have certificate.
Got 12 chars:'Hello World!'
-bash-4.2$
```

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.

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
```