Experiment 11

SECURE MAIL USING S/MIME

11.1 Aim

Secure mail using S/MIME.

11.2 Theory

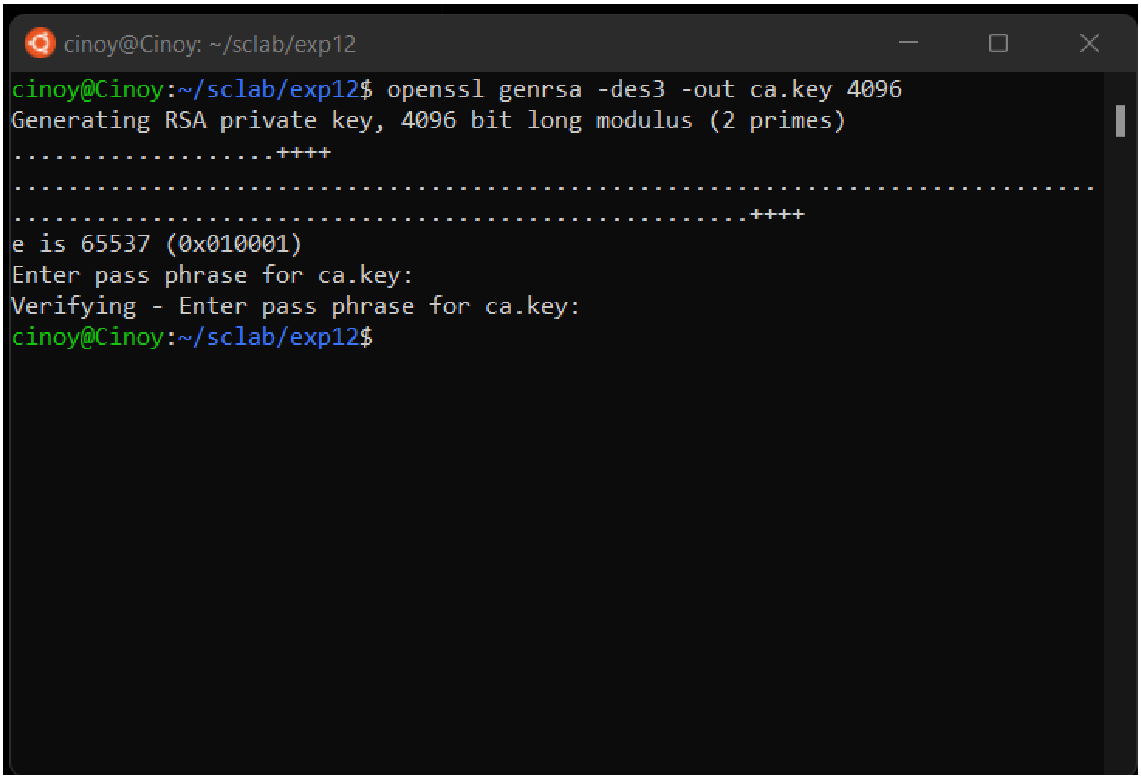
S/MIME (Secure/Multipurpose Internet Mail Extensions) is a standard for public key encryption and signing of MIME data.
S/MIME provides the following cryptographic security services for electronic messaging applications:

- Authentication.
- Message integrity.
- Non-repudiation of origin (using digital signatures)
- Privacy.
- Message integrity.
- Data security (using encryption).

Before S/MIME can be used in any of the above applications, an individual key/certificate must be obtained and installed, either from one's own certificate authority (CA) or from a public CA. The established best practice is to employ different private keys (and accompanying certificates) for signing and encryption, since this allows the encryption key to be escrow without jeopardizing the signature key's non-repudiation characteristic. Encryption necessitates the presence of the destination party's certificate in the database (which is typically automatic upon receiving a message from the party with a valid signing certificate). While it is technically feasible to transmit an encrypted message without having one's own certificate to digitally sign it, in practice, S/MIME clients will need the user to install their own certificate before allowing encrypting to others. This is required in order for the message to be encrypted for both the receiver and the sender, as well as for a copy of the message to be preserved (in the sent folder) and readable by the sender.

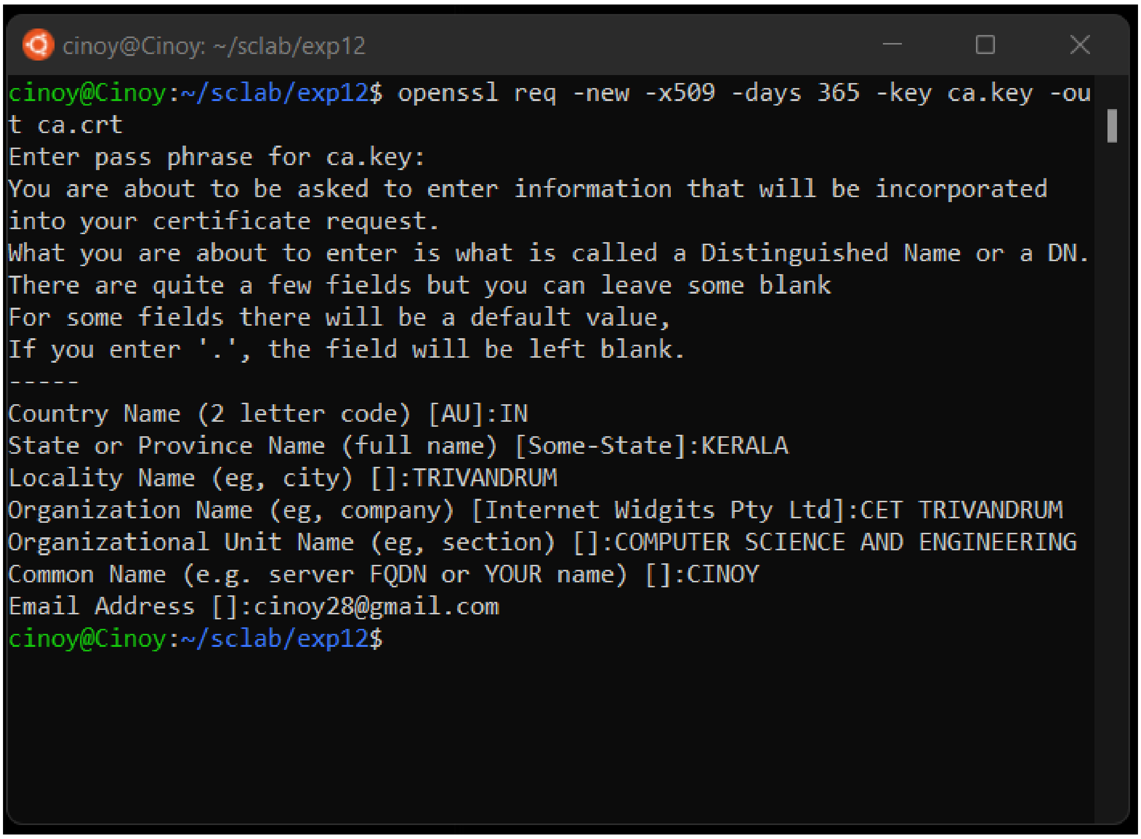
If the private key for the corresponding key pair is missing or otherwise useless (e.g., the certificate has been deleted or lost, or the private key's password has been forgotten), any encrypted message stored by a S/MIME email client cannot be decoded. An expired, revoked, or untrusted certificate, on the other hand, can still be used for cryptographic reasons. Some email clients may not be able to index the clear text of encrypted messages. These possible stumbling blocks aren't exclusive to S/MIME, but rather to cipher text in general, and they don't apply to S/MIME communications that are merely signed and not encrypted.

11.3 Procedure



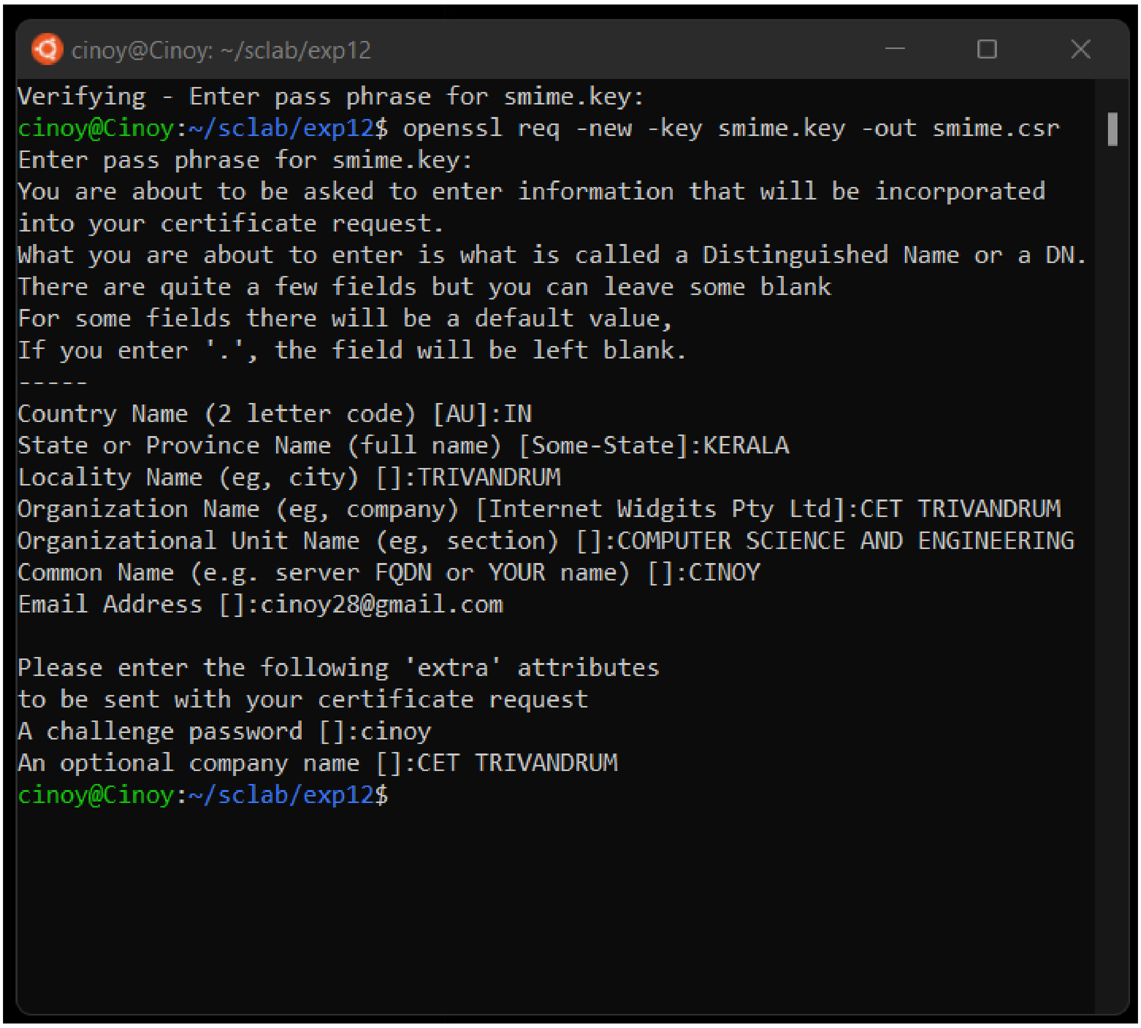
```
cinoy@Cinoy:~/sclab/exp12$ openssl genrsa -des3 -out ca.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
...................++++
..............................................................................
.....................................................++++
e is 65537 (0x010001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
cinoy@Cinoy:~/sclab/exp12$
```

Figure 1: Generating key for CA certificate



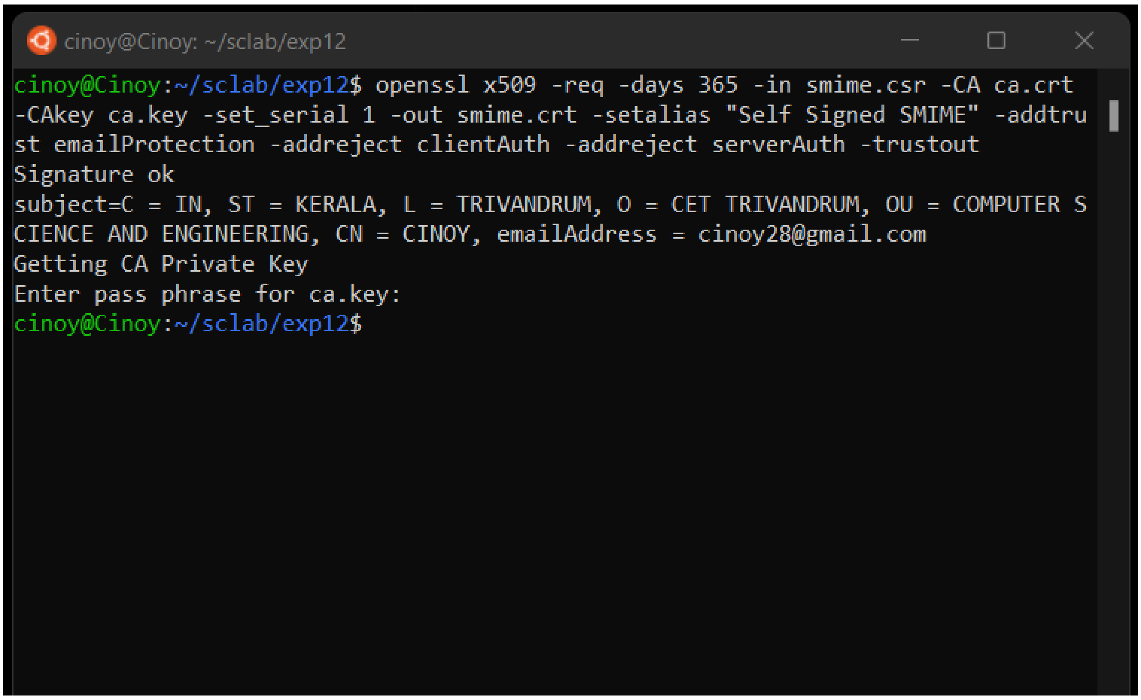
```
cinoy@Cinoy:~/sclab/exp12$ openssl req -new -x509 -days 365 -key ca.key -ou
t ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:KERALA
Locality Name (eg, city) []:TRIVANDRUM
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CET TRIVANDRUM
Organizational Unit Name (eg, section) []:COMPUTER SCIENCE AND ENGINEERING
Common Name (e.g. server FQDN or YOUR name) []:CINOY
Email Address []:cinoy28@gmail.com
cinoy@Cinoy:~/sclab/exp12$
```

Figure 2: Generating CA certificate



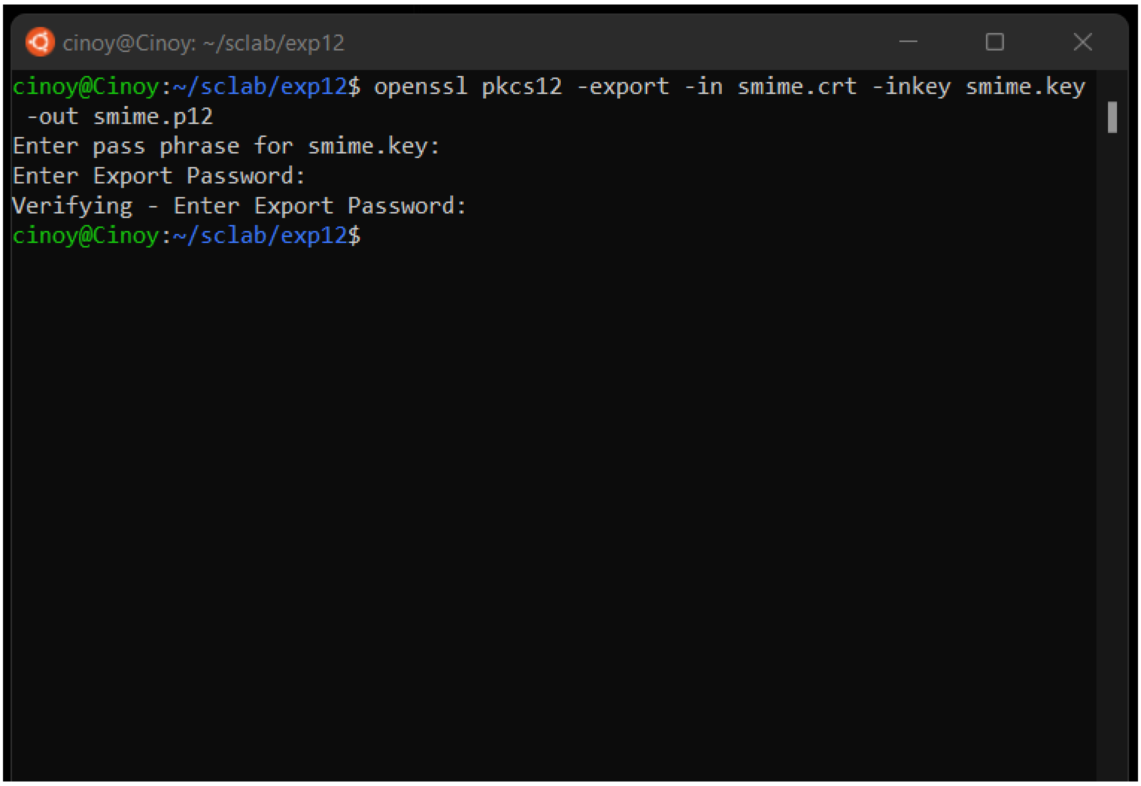
```
Verifying - Enter pass phrase for smime.key:
cinoy@Cinoy:~/sclab/exp12$ openssl req -new -key smime.key -out smime.csr
Enter pass phrase for smime.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:KERALA
Locality Name (eg, city) []:TRIVANDRUM
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CET TRIVANDRUM
Organizational Unit Name (eg, section) []:COMPUTER SCIENCE AND ENGINEERING
Common Name (e.g. server FQDN or YOUR name) []:CINOY
Email Address []:cinoy28@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cinoy
An optional company name []:CET TRIVANDRUM
cinoy@Cinoy:~/sclab/exp12$
```

Figure 3: Generating CSR file for cert generation



```
cinoy@Cinoy:~/sclab/exp12$ openssl x509 -req -days 365 -in smime.csr -CA ca.crt
-CAkey ca.key -set_serial 1 -out smime.crt -setalias "Self Signed SMIME" -addtru
st emailProtection -addreject clientAuth -addreject serverAuth -trustout
Signature ok
subject=C = IN, ST = KERALA, L = TRIVANDRUM, O = CET TRIVANDRUM, OU = COMPUTER S
CIENCE AND ENGINEERING, CN = CINOY, emailAddress = cinoy28@gmail.com
Getting CA Private Key
Enter pass phrase for ca.key:
cinoy@Cinoy:~/sclab/exp12$
```

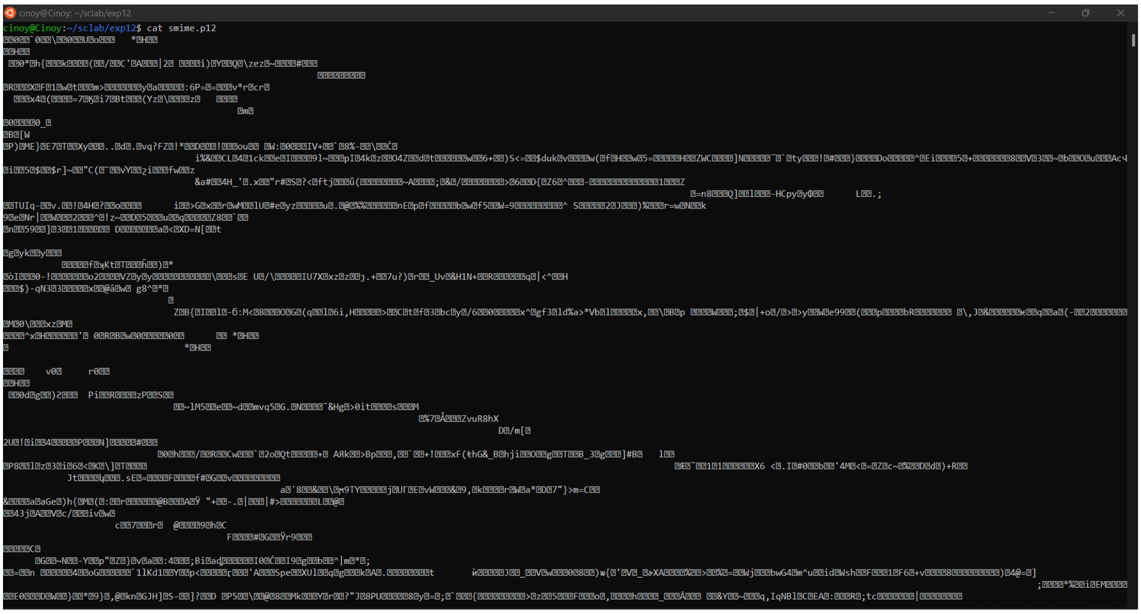
Figure 4: Generating S/MIME certificate



```
cinoy@Cinoy:~/sclab/exp12$ openssl pkcs12 -export -in smime.crt -inkey smime.key
 -out smime.p12
Enter pass phrase for smime.key:
Enter Export Password:
Verifying - Enter Export Password:
cinoy@Cinoy:~/sclab/exp12$
```

Figure 5: Generating a single file for key and cert

11.4 Output



```
cinoy@Cinoy:~/sclab/exp12$ cat smime.p12
```

Figure 6: S/MIME p12 certificate

11.5 Result

Generated S/MIME certificate for sending S/MIME Secured emails.