**Openssl**

Confidentiality

1. Alice creates a random key of size 128 bits and stores it in file symm.key.

This key will be used for the purpose of encrypting and decrypting data

using symmetric ciphers.

openssl rand -out symm.key 16

2. Alice creates a file plain.txt, adds some dummy data to the file.

vi plain.txt

3. Alice encrypts the contents of plain.txt to cipher.txt using AES-128

algorithm in CBC mode. Use symm.key for the purpose of encryption.

openssl enc -aes-128-cbc -in plain.txt -out cipher.txt -kfile symm.key -e

4. Alice creates a 2048 bit RSA private key. Store in file alicepriv.key.

openssl genrsa -out alicepriv.key -aes256 2048

5. Alice extracts the public key from alicepriv.key and store in file

alicepub.key.

openssl rsa -in alicepriv.key -out alicepub.key -pubout

6. Repeat step 4 and 5 to create private and public key of Bob. bobpriv.key

and bobpub.key. Alice and Bob exchange their public keys.

(repeat)

7. Alice sends cipher.txt to Bob.

cp cipher.txt ../bob/

8. Alice encrypts symm.key using the public key of Bob. Store in

symm.enc.key.

openssl rsautl -in symm.key -out symm.enc.key -inkey bobpub.key -pubin -encrypt

9. Bob decrypts symm.enc.key using his private key and stores the output

in symm.dec.key.

openssl rsautl -in symm.enc.key -out symm.dec.key -inkey bobpriv.key -decrypt

10.Bob decrypts cipher.txt using symm.dec.key and stores the output in

cipher.dec.txt. The cipher.dec.txt and plain.txt should have same

contents.

openssl enc -in cipher.txt -out cipher.dec.txt -d -aes-128-cbc -kfile symm.dec.key

Integrity Check

11.Alice computes sha-512 hash on plain.txt and store in hash.txt.

openssl dgst -sha256 -out hash.txt plain.txt

12. Alice verifies the hash.

cat hash.txt

13.Make minor changes to plain.txt and check that verification of hash now

fails.

diff hash.txt hash2.txt

Authentication check

14.Alice computes MAC on plain.txt using sha-512 and store in plain.mac.

openssl dgst -sha256 -out hash2.mac -hmac key1 plain.txt

15.Alice verifies the MAC.

16.Make minor changes to plain.txt and check that verification of MAC now

fails.

openssl dgst -sha256 -out hmac\_dec.mac -hmac key1 plain.txt

Digital Signature

17.Alice creates sha-512 hash on plain.txt and signs it using her private key.

Store signed hash in file hash.sign.

openssl dgst -sha256 -out hash.sign -sign alicepriv.key plain.txt

18. Alice sends plain.txt and hash.sign to Bob.

19.Bob verifies the signature using the public key of Alice.

openssl dgst -sha256 -signature hash.sign -verify alicepub.key plain.txt

20.Check that the verification fails if the file plain.txt is modified.