First create Two folder(directories)
mkdir alice
mkdir bob

Then go to alice folder (cd alice)

1. Alice creates a random key of size 128 bits and stores it in file symm.key. This key will be used for the use 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
```

Inside plain.txt(type anything):- baro mare bob!!

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
```

Explanation: -aes-128-cbc - tells us that we are using advanced encrytion standard of 128 bit and the mode is CBC(cipher block chaining mode).
We are creating plain text into a cipher text using a 128 bit key(symm.key)
-kfile : which key we are using and it is present in which file
-e : encryption

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

Which command we use?
We use genpkey , but that's not the only thing there is something called genrsa too. Sir taught using genrsa

```
openssl genrsa -out alicepriv.key -aes256 2048
```

It will ask you pass phrase for alicepriv.key
Just enter anything which u will remember (for ex: 1234)

5. Alice extracts the public key from alicepriv.key and store in file alicepub.key
To just output the public part of a private key
Now we use rsa rather then genrsa

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

Why pubout?
By default a private key is output: with this option a public key will be output instead. This option is automatically set if the input is a public key.

After the above command he will ask for password again , just type what u typed in step 4 (1234)

Then,

```
openssl rsa -in alicepub.key -pubin
```

If u check in alice folder now (ls alice) you will find 5 files.
Alice public key , Alice private key, Alice symmetric key which is used for encryption , plain text, and cipher text .

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

First we have to share the public key of Alice with Bob,

```
cp alicepub.key ../bob/
```

Then get out of Alice folder(cd ..)

And go inside bob folder(cd bob)

To create private key of Bob:

```
openssl genrsa -out bobpriv.key -aes256 2048
```

Again it will ask for pass phrase
Enter anything (for ex: 4321)

To extract public key of bob from his private key

```
openssl rsa -in bobpriv.key -out bobpub.key -pubout
```

The pass phrase which u entered above (4321)
Then,

```
openssl rsa -in bobpub.key -pubin
```

Also bob will send his public key to Alice

```
cp bobpub.key ../alice/
```

7. Alice sends cipher.txt to Bob
Come out of bob folder (cd ..) and go to alice folder(cd alice)

```
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
```

Alice then sends “symm.enc.key” to bob

```
cp symm.enc.key ../bob/
```

Now get out of alice folder (cd ..) and go to Bob’s folder(cd bob)

9. Bob decrypts symm.enc.key using his private key and stores the ouput in symm.dec.key

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

It will ask for pass phrase (which u entered for bob’s private key) (4321)

10. Bob decrypts cipher.txt using symm.dec.key and stores the ouput 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
```

Now check cipher.dec.txt and u will see text which was send by alice that was “baro mare bob!!”

```
cat cipher.dec.txt
```

Integrity Check

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

Sha-512 is a hash function . we are getting a hash of a message(that is plaint text(baro mare bob!!) by alice)

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

12. Alice verifies the hash
    Just do : cat hash.txt and u will see that hash is generated
13. Make minor changes to plain.txt and check that verification of hash now fails.

    Open plain text file : vi plain.txt
    Inside the text edit the text , we had text as : “baro mare bob!!”
    U can change it is as : “baro mare bob!!, kide chala?”

    And then create the hash of the new message .

```
openssl dgst -sha256 -out hash2.txt plain.txt
```

Then compare two hashes and u will see that there are two different hashes for same file bcs we changed the content.

```
diff hash.txt hash2.txt
```

Send the new hash(hash2.txt) to bob along with plain.txt , bcs u updated that too

```
cp hash2.txt plain.txt ../bob/
```

Then bob will calculate the hash for that plain text and if he gets the same hash as hash2.txt that means that bob recieved from alice and no one else.

Authentication Check

14. Alice computes MAC on plain.txt using sha-512 and store in plain.mac
    Mac C=(k,m)
    Mac is generated combining secret key and message

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

    Then send the mac as well as plain text again to bob (no need to send plain text again)

```
cp hash2.txt ../bob/
```

15. Alice verifies the mac
16. Make minor changes to plain.txt and check that verification of MAC now fails.
    Just change the plain text again and u will see that MAc changes too . and then send the updated the mac and plain text to bob.

Now go to bob's folder (cd bob) and check the mac of the plain text , if it is same then it sent by alice

```
openssl dgst -sha256 -out hmac_dec.txt -hmac key1 plain.txt
```

and then u compare hash2.txt with hmac_dec.txt

```
cat hash2.txt

cat hmac_dec.txt
```

You will see that they are same…

Digital Signature

Creating a hash using message and private key is termed as 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
```

It will ask for pass phase again , type the pass which we entered for alice private key . (1234)

18. Alice sends plain.txt and hash.sign to Bob
    No need to send plain.txt again , if u had made changes to plain.txt then only resend it to bob

```
cp hash.sign ../bob/
```

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

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