mkdir openssl\_eg

cd openssl\_eg/

#Encryption

openssl version

openssl list -commands

openssl list -cipher-commands

nano message.txt / touch message.txt; echo "This is my secret" > message.txt

openssl <cmd> <cipher> -in <inputFile> -out <outputFile>

openssl enc -base64 -in message.txt -out enc-message-base64.bin

openssl enc -aes-256-cbc -in message.txt -out enc-message-aes.bin

openssl enc -aes-256-cbc -d -in enc-message-aes.bin -pass pass:<your password>

#Public Key Cryptography

\*\*Key Generation

openssl genpkey -algorithm RSA -out privkey-A.pem

cat privkey-A.pem | less

openssl pkey -in privkey-A.pem -text | less

openssl pkey -in privkey-A.pem -out pubkey-A.pem -pubout

ls

openssl genpkey -algorithm RSA -out privkey-B.pem

openssl pkey -in privkey-B.pem -out pubkey-B.pem -pubout

\*\*Encrypting Message using Hashing , hash algorithm sha1

openssl dgst -sha1 message.txt

\*\*Genrating Digital Signature and Encrypting Message using private key of A

openssl dgst -sha1 -sign privkey-A.pem -out signature.bin message.txt

\*\*Encrypting a message using Public Key of B (assuming you have publickey of B)

openssl pkeyutl -encrypt -in message.txt -pubin -inkey pubkey-B.pem -out ciphertext.bin

\*\*Things that will be availible other side B by your socket file transfer

1. Public Key of A ie pubkey-A.pem

2. Signature ie signature.bin

3. Encrypted Text Message ie ciphertext.bin

\*\*Side B

\*\*Decrypting Message using Private Key of B

openssl pkeyutl -decrypt -in ciphertext.bin -inkey privkey-B.pem -out received-message.txt

cat received-message.txt

\*\*To verify weather its different with original file or not

diff received-message.txt message.txt

\*\*Verify the Digital Signature

openssl dgst -sha1 -verify pubkey-A.pem -signature signature.bin received-message.txt