Pradyumna Bhosale Roll NO:-20141224

root@ip-172-31-46-197:~

```
[root@ip-172-31-46-197 ~]# vim KeyGen.sh
[root@ip-172-31-46-197 ~]# cat KeyGen.sh
#!/bin/sh



# Key generation

# Create Alice's key pair
openssl genrsa > alice.private

# Obtain Alice's public key
openssl rsa -in alice.private -pubout -out alice.public

# Create Bob's key pair
openssl genrsa > bob.private

# Obtain Bob's public key
openssl rsa -in bob.private -pubout -out bob.public
[root@ip-172-31-46-197 ~]# source KeyGen.sh
Generating RSA private key, 2048 bit long modulus (2 primes)
....................+++++
..........................................................................................+++++
e is 65537 (0x010001)
writing RSA key
Generating RSA private key, 2048 bit long modulus (2 primes)
..........................+++++
...............................................................+++++
e is 65537 (0x010001)
writing RSA key
[root@ip-172-31-46-197 ~]# ls
alice.private  alice.public  bob.private  bob.public  KeyGen.sh
[root@ip-172-31-46-197 ~]#
```

```
[root@ip-172-31-46-197 ~]# vim AliceMsg.sh
[root@ip-172-31-46-197 ~]# cat AliceMsg.sh
#!/bin/sh



# Alice sends a short confidential message

# Secret message Alice wants to send to Bob
echo "Alice Loves you" > message.plain

# Alice encrypts the message using Bob's public key
openssl rsautl -encrypt -in message.plain -out message.encrypted
               -pubin -inkey bob.public

# Bob decrypts Alice's message using his private key
openssl rsautl -decrypt -in message.encrypted -out message.decrpyted
               -inkey bob.private
[root@ip-172-31-46-197 ~]# ls
AliceMsg.sh  alice.private  alice.public  bob.private  bob.public  KeyGen.sh
[root@ip-172-31-46-197 ~]# source AliceMsg.sh
[root@ip-172-31-46-197 ~]# ls
AliceMsg.sh    alice.public  bob.public  message.decrpyted  message.plain
alice.private  bob.private   KeyGen.sh   message.encrypted
[root@ip-172-31-46-197 ~]# cat message.plain
Alice Loves you
[root@ip-172-31-46-197 ~]# cat message.encrypted
▒▒▒▒z
     l▒d▒7▒h▒z▒E▒hNn▒W▒l▒▒c⊐h
B▒:▒on▒·;Ĩ*▒▒▒$▒E▒g▒q▒"[\%@ ▒:.▒H▒Q▒1*▒K▒8w¤TU▒_▒Œ%▒"▒{}_▒d[▒gy▒▒
▒]▒t▒n7▒'▒d(N▒(<▒@KW(▒PI4▒~▒W▒y▒\▒dc▒Y▒A▒-▒Khd▒▒P▒5▒+K▒=`▒t▒z▒W▒
□<i6▒[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]# cat message.decrpyted
Alice Loves you
[root@ip-172-31-46-197 ~]#
```

```
root@ip-172-31-46-197:~
[root@ip-172-31-46-197 ~]# cat BobSign.sh
#!/bin/sh



# Bob sends a short signed message

# Message Bob wants to sign
echo "Will you marry me ?" > message.plain

# Bob signs the message using his private key
openssl rsautl -sign -in message.plain -out message.signed -inkey bob.private

# Alice verifies Bob's message using his public key
openssl rsautl -verify -in message.signed -out message.verified -pubin -inkey
 bob.public

[root@ip-172-31-46-197 ~]# source BobSign.sh
[root@ip-172-31-46-197 ~]# ls
AliceMsg.sh    alice.public  bob.public  KeyGen.sh     message.signed
alice.private  bob.private   BobSign.sh  message.plain message.verified
[root@ip-172-31-46-197 ~]# cat message.plain
Will you marry me ?
[root@ip-172-31-46-197 ~]# cat message.signed
```


```
2▒}Y▒▒'▒:y%▒^▒▒)R"&▒
▒▒D▒P擴 ▒#▒n▒▒I}Jql%▒▒N▒▒X▒▒□▒B
                E▒X▒▒▒▒;▒▒▒&h▒pb>▒鸪 ▒▒▒1?k▒▒
                                              ▒S▒3▒▒*
▒!▒Y▒7▒▒Gu9▒▒▒▒▒▒Q▒▒▒v%[la▒▒A▒▒
                                  ;▒▒▒zh▒▒▒▒p*S▒▒=6}c悟{▒▒▒N▒zOS▒â▒C▒A▒▒;dU▒j4~ـ▒
S
 ▒▒▒I▒y▒IlE▒▒▒
              ▒t▒▒▒-▒▒▒▒[root@ip-172-31-46-197 ~]#
```


```
[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]# cat message.verified
Will you marry me ?
[root@ip-172-31-46-197 ~]#
```


```
[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]#
[root@ip-172-31-46-197 ~]# cat message.verified
Will you marry me ?
[root@ip-172-31-46-197 ~]#
```


```
root@ip-172-31-46-197:~
[root@ip-172-31-46-197 ~]# cat AliceLargeFile.sh
#!/bin/sh



# Alice sends a large signed and confidential message

# Secret message Alice wants to send to Bob
cat > message.plain << EOF
                        Marital AGREEMENT

THIS AGREEMENT, made this thirteen day of June, 2004 is between Bob
and Alice

1. PURPOSE. The parties expect to be married to death do them part,
   and hear by enter into this agrement vouluntarily.

2. EFFECT OF AGREEMENT. The parties agree that if one or the other
   commits infidelity during the duration of the marriage, that the person
   guilty of said act shall in effect and wholey forsake all material
   property, assets and rights to act as a parent of any children.

3. DEFINITON OF INFEDELITY. Infedelity is defined as follows: Any
   socializing with the intent to establish a realtionship, and/or
   physical contact with other person.

4. JOINT PROPERTY, ETC. This Agreement does not restrict, prohibit
   or condition any conveyance or transfer by the parties, or either of
   them alone, of the Separate Property of either party into tenancy in
   common, joint tenancy, tenency by the entireties or any other form of
   concurrent and/or undivided estate or ownership between the parties,
   or the acquisition of any property in any such form of ownership by the
   parties. The incidents and attributes of ownership and other rights
   of the parties with respect to any property so conveyed, transferred
   or acquired shall be determined under State law and shall not be
   governed by or otherwise determined with reference to this Agreement.

5. SEPARATE PROPERTY. The parties agree that there is no seperate
   property.

6. WAIVER OF RIGHTS. Except as otherwise provided in this Agreement,
```

root@ip-172-31-46-197:~

```
EOF

# Alice generates a short random key to be used for encrypting the message
openssl rand -out key.plain 16

# Alice encrypts the message with the short random key
openssl des3 -e -kfile key.plain -in message.plain -out message.encrypted

# Alice creates a message digest of the message to sign
openssl dgst -binary message.plain > message.digest

# Alice signs the digest using her private key
openssl rsautl -sign -in message.digest -out digest.signed -inkey alice.priva
te

# Alice encrypts the random key using Bob's public key
openssl rsautl -encrypt -in key.plain -out key.encrypted -pubin -inkey bob.pu
blic
# Alice sends Bob:
# - the encrypted message
# - the encrypted key
# - the signed message digest


# Bob decrypts Alice's encrypted key using his private key
openssl rsautl -decrypt -in key.encrypted -out key.decrypted -inkey bob.priva
te

# Bob decrypts the message using the decrypted key
openssl des3 -d -kfile key.decrypted -in message.encrypted -out message.decry
pted

# Bob verifies the digest Alice has signed using her public key
openssl rsautl -verify -in digest.signed -out message.digest1 -pubin -inkey
alice.public

# Bob calculates again a message digest of the message
openssl dgst -binary message.plain > message.digest2
# Bob decrypts the message using the decrypted key
openssl des3 -d -kfile key.decrypted -in message.encrypted -out message.decry
pted

# Bob verifies the digest Alice has signed using her public key
openssl rsautl -verify -in digest.signed -out message.digest1 -pubin -inkey
alice.public

# Bob calculates again a message digest of the message
openssl dgst -binary message.plain > message.digest2
# Bob compares the two message digests to verify Alice signed the agreement
# he has examined diff message.digest1 message.digest2
[root@ip-172-31-46-197 ~]#
```

root@ip-172-31-46-197:~

```
[root@ip-172-31-46-197 ~]# ls
AliceLargeFile.sh  alice.private  bob.private  BobSign.sh
AliceMsg.sh        alice.public   bob.public   KeyGen.sh
[root@ip-172-31-46-197 ~]# source AliceLargeFile.sh
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[root@ip-172-31-46-197 ~]# ls
AliceLargeFile.sh  bob.public     KeyGen.sh          message.digest2
AliceMsg.sh        BobSign.sh     key.plain          message.encrypted
alice.private      digest.signed  message.decrypted  message.plain
alice.public       key.decrypted  message.digest
bob.private        key.encrypted  message.digest1
[root@ip-172-31-46-197 ~]#
```