Tutorial #6
Security in Computing COSC2356/2357

(RSA Signature System)

Q1. Suppose Bob (the sender) wants to send a signed message **m=500** to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **RSA encryption algorithm** and sends the public key to Alice. Bob uses parameter **p=113** and **q=89**, and chooses a suitable public key parameter **e=29**. How would Bob sign message **m=500**? How would Alice verify the signed message from Bob?

Q2. **(Do It Yourself)** Suppose Bob (the sender) wants to send a signed message **m=1234** to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **RSA encryption algorithm** and sends the public key to Alice. Bob uses parameter **p=131** and **q=97**, and chooses a suitable public key parameter **e=11**. How would Bob sign message **m=1234**? How would Alice verify the signed message from Bob?

(ElGamal Digital Signature System)

Q3. Suppose Bob (the sender) wants to send a signed message $m = 37$ to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **ElGamal encryption algorithm** and sends the public key to Alice. Bob chooses $p = 8081$, $g = 2849$, $x = 53$. How would Bob sign message $m = 37$? How would Alice verify the signed message from Bob?

Q4. **(Do It Yourself)** Suppose Bob (the sender) wants to send a signed message $m = 23$ to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **ElGamal encryption algorithm** and sends the public key to Alice. Bob chooses $p = 83$, $g = 79$, $x = 29$. How would Bob sign message $m = 23$? How would Alice verify the signed message from Bob?

TASK-1 (RSA Digital Signature using OpenSSL).

Say, you have a plain-text file, called "**plain-text.txt",** with your name and student ID in that file. Apply **openssl's RSA algorithm** to generate **2048** bit keys. Show that you can generate public and private keys and apply them to sign/verify. Make sure you apply a hash algorithm before signing the document.

Step-1: Run **openssl** command

```
openssl
```

Step-2: Generate a private key and store that private key on computer

```
genrsa -out private-key.pem 2048
```

Step-3: Generate the public key from the private key

```
rsa -in private-key.pem -pubout -out public-key.pem
```

Step-4: Hash the file "**plain-text.txt"** using **SHA256** and sign that file using the private key. The signed file **sign.sha256** is a binary file.

```
dgst -sha256 -sign private-key.pem –out sign.sha256 plain-text.txt
```

Step-5: Verify themessage:

```
dgst -sha256 -verify public-key.pem -signature sign.sha256 plain-text.txt
```

If the message is verified successfully, the following message will appear:

Verified OK