ASSIGNMENT NO: 3

TITLE:
To implement the Diffie-Hellman Key Exchange algorithm.

ALGORITHM:

1.Both Alice and Bob share the same public keys g and p.
2.Alice selects a random public key a.
3.Alice computes his secret key A as ga mod p.
4.Then Alice sends A to Bob.
5.Similarly Bob also selects a public key b and computes his secret key as B and sends the same back to Alice.
6.Now both of them compute their common secret key as the other one's secret key power of a mod p.

ASSIGNMENT NO: 4

TITLE: Implementation of RSA algorithm.

ALGORITHM:
1. Start
2. Input two prime numbers p and q.
3. Calculate n = pq.
4. Calculate Ø(n) = (p-1)(q-1).
5. Input value of e.
6. Determine d.
7. Determine PU and PR.
8. Take input plaintext.
9. Encrypt the plaintext and show the output.
10. Stop.

ASSIGNMENT NO: 5

TITLE: Implementation of ECC algorithm.

Key Generation
Now, we have to select a number 'd' within the range of 'n'.
Using the following equation we can generate the public key
Q = d * P

d = The random number that we have selected within the range of (1 to n-1). P is the point on the curve.
‘Q’ is the public key and ‘d’ is the private key.

Encryption

Let ‘m’ be the message that we are sending. Consider ‘m’ has the point ‘M’ on the curve ‘E’.
Randomly select ‘k’ from [1 – (n-1)].
Two cipher texts will be generated, let it be C1 and C2.
C1 = k*P
C2 = M + k*Q

C1 and C2 will be sent.

Decryption

We have to get back the message ‘m’ that was send to us,
M = C2 – d * C1

M is the original message that we have sent.

Proof

How does we get back the message,
M = C2 – d * C1
‘M’ can be represented as ‘C2 – d * C1’
C2 – d * C1 = (M + k * Q) – d * (k * P) (C2 = M + k * Q and C1 = k * P)
= M + k * d * P – d * k *P
= M (Original Message)