Cryptography and Network Security
Lab Assignment - IV

Program 1: RSA Key Generation

Code:

```python
def gcd(a,b):
        if(b==0):
            return a
        else:
            return gcd(b,a%b)
def totient(n):
    res=1
    for i in range(2,n):
        if(gcd(n,i)==1):
            res+=1
    return res

print("Siddhanth Monnappa\t22BCE3061\n")
p=int(input("Enter p: "))
q=int(input("Enter q: "))
e=int(input('Enter e: '))
n=p*q
phi=(p-1)*(q-1)
if gcd(e,phi)!=1:
    print("Multiplicative Inverse of e does not exist")
    exit()
d=e**(totient(phi)-1)%phi

print(f"Private Keys: ({d},{n})")
```

Input/Output:

```
Siddhanth Monnappa      22BCE3061

Enter p: 17
Enter q: 11
Enter e: 7
Private Keys: (23,187)
```

```
Siddhanth Monnappa      22BCE3061

Enter p: 3
Enter q: 11
Enter e: 7
Private Keys: (3,33)
```

Program 2: RSA Key Generation & Encryption/Decryption

Code:

```
def gcd(a,b):
        if(b==0):
            return a
        else:
            return gcd(b,a%b)
def totient(n):
    res=1
    for i in range(2,n):
        if(gcd(n,i)==1):
            res+=1
    return res

print("Siddhanth Monnappa\t22BCE3061\n")
p=int(input("Enter p: "))
q=int(input("Enter q: "))
e=int(input('Enter e: '))
n=p*q
phi=(p-1)*(q-1)
if gcd(e,phi)!=1:
    print("Multiplicative Inverse of e does not exist")
    exit()
d=e**(totient(phi)-1)%phi

num=int(input("Enter number of plaintexts: "))
pt=[]
for i in range(num):
    x=int(input("Enter plaintext: "))
    pt.append(x)
cipher_text=[]
for plain_text in pt:
    cipher_text.append(pow(plain_text,e,n))
deciphered_text=[]
for cipher in cipher_text:
    deciphered_text.append(pow(cipher,d,n))
print(f"\nPublic Keys: ({e},{n})")
print(f"Private Keys: ({d},{n})")
print(f"Cipher Text: {cipher_text}")
print(f"Deciphered Text: {deciphered_text}")
```

Input/Output:

```
Siddhanth Monnappa      22BCE3061

Enter p: 17
Enter q: 11
Enter e: 7
Enter number of plaintexts: 1
Enter plaintext: 88

Public Keys: (7,187)
Private Keys: (23,187)
Cipher Text: [11]
Deciphered Text: [88]
```

```
Siddhanth Monnappa      22BCE3061

Enter p: 5953
Enter q: 83
Enter e: 4097
Enter number of plaintexts: 1
Enter plaintext: 1234

Public Keys: (4097,494099)
Private Keys: (20609,494099)
Cipher Text: [208053]
Deciphered Text: [1234]
```

```
Siddhanth Monnappa      22BCE3061

Enter p: 5953
Enter q: 83
Enter e: 4097
Enter number of plaintexts: 3
Enter plaintext: 123
Enter plaintext: 456
Enter plaintext: 789

Public Keys: (4097,494099)
Private Keys: (20609,494099)
Cipher Text: [282710, 376148, 304784]
Deciphered Text: [123, 456, 789]
```

Program 3: Elgamal Key Generation & Encryption/Decryption

Code:

```python
def fast_expo(a, x, n):
    x_bin=bin(x)[2:]
    y=1
    for bit in x_bin[::-1]:
        if bit=='1':
            y=(y*a)%n
        a=(a*a)%n
    return y

def elgamal_encrypt(q, alpha, Xa, k, M):
    Ya=fast_expo(alpha, Xa, q)
    K=fast_expo(Ya, k, q)
    C1=fast_expo(alpha, k, q)
    C2=(K*M)%q
    return Ya, C1, C2

def elgamal_decrypt(q, Xa, C1, C2):
    K=fast_expo(C1, Xa, q)
    K_inv=pow(K,-1,q)
    plaintext=(C2*K_inv)%q
    return plaintext

print("Siddhanth Monnappa\t22BCE3061\n")
q=int(input("Enter q: "))
alpha=int(input("Enter alpha: "))
Xa=int(input("Enter Xa: "))
k=int(input("Enter k: "))
M=int(input("Enter plaintext: "))

Ya, C1, C2=elgamal_encrypt(q, alpha, Xa, k, M)
decrypted_text=elgamal_decrypt(q, Xa, C1, C2)
print(f"\nYa: {Ya}\nCi: ({C1},{C2})")
print(f"Decrypted Text: {decrypted_text}")
```

Input/Output:

```
Siddhanth Monnappa      22BCE3061

Enter q: 19
Enter alpha: 10
Enter Xa: 5
Enter k: 6
Enter plaintext: 17

Ya: 3
Ci: (11,5)
Decrypted Text: 17
```

```
Siddhanth Monnappa      22BCE3061

Enter q: 17
Enter alpha: 3
Enter Xa: 15
Enter k: 13
Enter plaintext: 10

Ya: 6
Ci: (12,15)
Decrypted Text: 10
```

Program 4: Diffie Hellman Key Exchange

Code:

```python
print("Siddhanth Monnappa\t22BCE3061\n")

q=int(input('Enter q: '))
alpha=int(input('Enter alpha: '))
alpha=alpha%q
Xa=int(input('Enter Xa: '))
Xb=int(input('Enter Xb: '))
Ya=pow(alpha, Xa, q)
Yb=pow(alpha, Xb, q)

print(f"\nPublic Keys - \tYa: {Ya}\tYb: {Yb}")

Ka=pow(Yb, Xa, q)
Kb=pow(Ya, Xb, q)

print(f'\nSecret Session Keys - \tKa: {Ka}\tKb: {Kb}')

```

Input/Output:

```
Siddhanth Monnappa      22BCE3061

Enter q: 353
Enter alpha: 3
Enter Xa: 97
Enter Xb: 233

Public Keys -   Ya: 40  Yb: 248

Secret Session Keys -   Ka: 160 Kb: 160
```

```
Siddhanth Monnappa      22BCE3061

Enter q: 71
Enter alpha: 7
Enter Xa: 5
Enter Xb: 12

Public Keys -   Ya: 51  Yb: 4

Secret Session Keys -   Ka: 30  Kb: 30
```

Program 5: Diffie Hellman Man-In-The-Middle Attack with Key Exchange

Code:

```python
print("Siddhanth Monnappa\t22BCE3061\n")
q=int(input('Enter q: '))
alpha=int(input('Enter alpha: '))
alpha=alpha%q
Xa=int(input('Enter Xa: '))
Xb=int(input('Enter Xb: '))
Ya=pow(alpha, Xa, q)
Yb=pow(alpha, Xb, q)

print(f"\nPublic Keys - \tYa: {Ya}\tYb: {Yb}\n")

Xda=int(input("Enter Xda: "))
Xdb=int(input("Enter Xdb: "))
Yda=pow(alpha, Xda, q)
Ydb=pow(alpha, Xdb, q)

print(f'\nAttackers Public Keys - \tYda: {Yda}\tYdb: {Ydb}\n')

Ka=pow(Yda, Xa, q)
Kb=pow(Ydb, Xb, q)

print(f'Attackers Shared Secret Keys - \tKa: {Ka}\tKb: {Kb}\n')

Kda=pow(Ya, Xda, q)
Kdb=pow(Yb, Xdb, q)

print(f'Shared Session Keys - \tKda: {Kda}\tKdb: {Kdb}')

```

Input/Output:

```
Siddhanth Monnappa      22BCE3061

Enter q: 19
Enter alpha: 10
Enter Xa: 7
Enter Xb: 8

Public Keys -   Ya: 15  Yb: 17

Enter Xda: 4
Enter Xdb: 12

Attackers Public Keys -         Yda: 6  Ydb: 7

Attackers Shared Secret Keys -  Ka: 9   Kb: 11

Shared Session Keys -   Kda: 9  Kdb: 11
```

```
Siddhanth Monnappa      22BCE3061

Enter q: 17
Enter alpha: 7
Enter Xa: 5
Enter Xb: 4

Public Keys -   Ya: 11  Yb: 4

Enter Xda: 4
Enter Xdb: 8

Attackers Public Keys -         Yda: 4  Ydb: 16

Attackers Shared Secret Keys -  Ka: 4   Kb: 1

Shared Session Keys -   Kda: 4  Kdb: 1
```