Lab-07

DATE
PAGE No.

Aim:- Write a program to implement Elgamal cryptosystem.
→ fn for primitive roots and Mul: Group.
→ Key Generation
→ Encryption
→ Decryption

Code:-

```
#include "function.h"

struct key{
 int e1, e2, p, d;
};

vector<int> findGroupZp(int p)
{
    vector<int> zp;
    for(int i=0; i< p; ++i)
        if(multiplicativeInverse(i,p)!=-1)
            zp.push_back(i);

    return zp;
}
```

```
int findPrimitiveRoot(int prime)
{
    int a;
    vector<int> groupZp = findGroupZp(prime);
    for(int j=0; j<groupZp.size(); ++j)
    {
        a = groupZp[j];
        for(int i=2; i<=(prime-1); ++i)
        {
            int m = squareAndMultiply(a,i,prime);
            if( i==(prime-1) && m==1)
                return a;
            else if(m==1)
                break;
        }
    }
    return 0;
}

Key generateKey()
{
    int prime;
    Key keys;
    while(1)
    {   cout<< "Enter a large prime number";
```

```
        cin>>prime;
        if(isPrime(prime))
            break;
    }
    vector<int> groupZp = findGroupZp(prime);
    keys.e1 = findPrimitiveRoot(prime);
    int k=0;
    while(groupZp[k]<=1 || group[k]>(prime-2))
        k++;
    keys.d = groupZp[k];
    keys.e2 = squareAndMultiply(keys.e1, keys.d, prime);
    keys.p = prime;
    return keys;
}

Pair<int,int> encryption(int msg, int e1, int e2, int p)
{
    vector<int> groupZp = findGroupZp(p);
    int r = groupZp[2];
    cout<<"\nr:"<<r<<endl;
    int c1 = squareAndMultiply(e1, r, p);
    int c2 = ((msg*r.p) * squareAndMultiply(e2,r,p))%p;
    return make_pair(c1, c2);
}
```

```
int decryption(int c1, int c2, int d, int p)
{
    int c1_d = squareAndMultiply(c1, d, p);
    int c1_inverse = multiplicativeInverse(c1_d, p);

    return (c1_inverse * c2) % p;
}

int main()
{
    Key keys;
    int msg;
    cout<<"Enter your msg: ";
    cin>>msg;
    keys = generateKeys();
    pair<int, int> encryptedMsg =
                encryption(msg, keys.e1, keys.e2, keys.p);
    cout<<"e1: "<<keys.e1<<" e2 "<<keys.e2
        <<" d: "<<keys.d<<" p: "<<keys.p<<endl;
    cout<<"C1: "<<encryptedMsg.first<<endl;
    cout<<"C2: "<<encryptedMsg.second<<endl;
    cout<<decryption(encryptedMsg.first, encryptedMsg.second, keys.d, keys.p);
    return 0;
}
```

Input

Enter your msg : 1234
Enter prime number: 3499

output

r: 3
e1: 2, e2 = 2, d = 1, p: 3499
C1 = 8, C2 = 2874
Decrypted msg: 1234