Digital Assignment-5

Cryptography and Network Security Lab

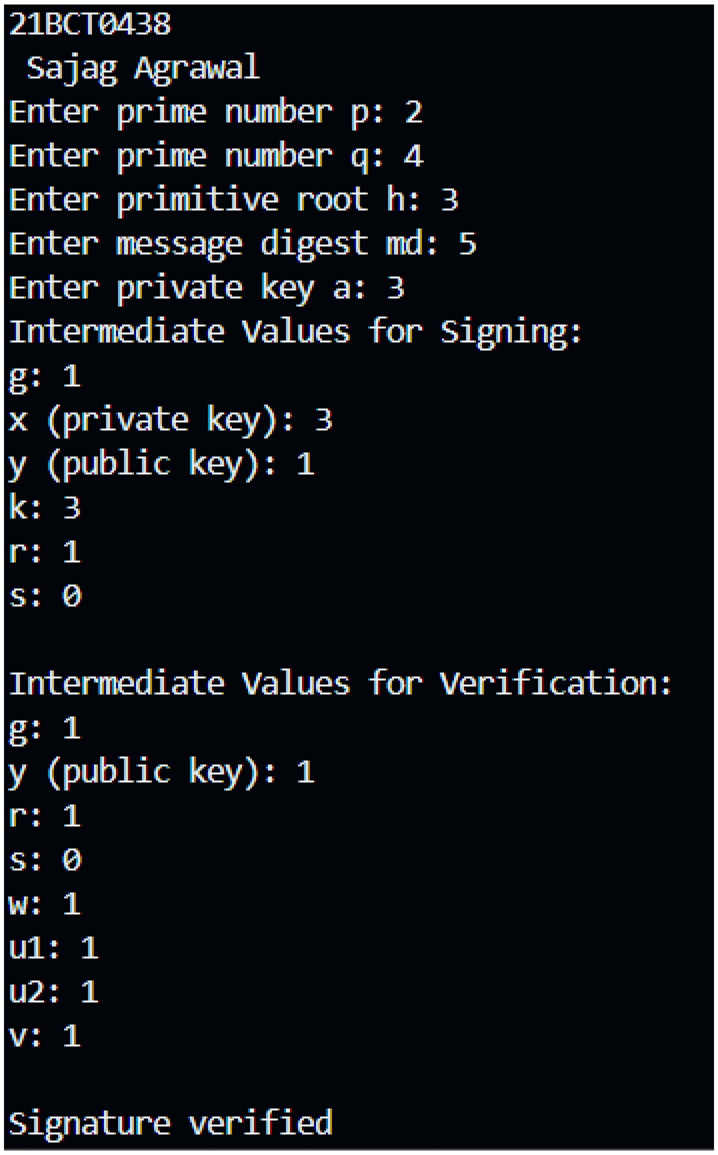
Sajag Agrawal

21BCT0438

Q1 DSS Code

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;
int powerModulo(int a, int b, int m) {
 int result = 1;
 a = a % m;
 while (b > 0) {
 if (b % 2 == 1) {
 result = (result * a) % m;
 }
 b = b / 2;
 a = (a * a) % m;
 }
 return result;
}
int multiplicativeInverse(int a, int m) {
 a = a % m;
 for (int x = 1; x < m; x++) {
 if ((a * x) % m == 1) {
 return x;
  }
 }
 return 1;
}
int generateRandomNumber(int q) {
 return rand() % (q - 1) + 1;
}
struct Signature {
 int r;
 int s;
 int y;
 Signature(int r, int s, int y) : r(r), s(s), y(y) {}
};
Signature sign(int p, int q, int h, int md, int a) {
 srand(time(0));
 int k = generateRandomNumber(q);
 int g = powerModulo(h, (p - 1) / q, p);
 int x = a;
 int y = powerModulo(g, x, p);
 int r = powerModulo(g, k, p) % q;
 int s = (multiplicativeInverse(k, q) * (md + x * r)) % q;
cout << "Intermediate Values for Signing:\n";
 cout << "g: " << g << "\n";
 cout << "x (private key): " << x << "\n";
 cout << "y (public key): " << y << "\n";
 cout << "k: " << k << "\n";
 cout << "r: " << r << "\n";
 cout << "s: " << s << "\n";
 return Signature(r, s, y);
}
bool verify(int p, int q, int h, int md, int y, int r, int s) {
 int g = powerModulo(h, (p - 1) / q, p);
 int w = multiplicativeInverse(s, q);
 int u1 = (md * w) % q;
 int u2 = (r * w) % q;
 int v = ((powerModulo(g, u1, p) * powerModulo(y, u2, p)) % p) % q;
 cout << "\nIntermediate Values for Verification:\n";
 cout << "g: " << g << "\n";
 cout << "y (public key): " << y << "\n";
 cout << "r: " << r << "\n";
 cout << "s: " << s << "\n";
 cout << "w: " << w << "\n";
 cout << "u1: " << u1 << "\n";
 cout << "u2: " << u2 << "\n";
cout << "v: " << v << "\n";
 return v == r;
}
int main() {
 int p, q, h, md, a;
 cout<<"21BCT0438\n Sajag Agrawal"<<endl;
 cout << "Enter prime number p: ";
 cin >> p;
 cout << "Enter prime number q: ";
 cin >> q;
 cout << "Enter primitive root h: ";
 cin >> h;
 cout << "Enter message digest md: ";
 cin >> md;
 cout << "Enter private key a: ";
 cin >> a;
 Signature signature = sign(p, q, h, md, a);
 if (verify(p, q, h, md, signature.y, signature.r, signature.s)) {
 cout << "\nSignature verified\n";
 } else {
 cout << "\nSignature not verified\n";
 }
 return 0;
}
```

Output



```
21BCT0438
 Sajag Agrawal
Enter prime number p: 2
Enter prime number q: 4
Enter primitive root h: 3
Enter message digest md: 5
Enter private key a: 3
Intermediate Values for Signing:
g: 1
x (private key): 3
y (public key): 1
k: 3
r: 1
s: 0

Intermediate Values for Verification:
g: 1
y (public key): 1
r: 1
s: 0
w: 1
u1: 1
u2: 1
v: 1

Signature verified
```

Q2 Source Code

Server Program

```
# include <bits/stdc++.h>

# include <arpa/inet.h>

using namespace std;

int createServer(int port) // TCP connection

{

 int sersock = socket(AF_INET, SOCK_STREAM, 0);
 struct sockaddr_in addr = {AF_INET, htons(port), INADDR_ANY};
 bind(sersock, (struct sockaddr *) &addr, sizeof(addr));
 cout << "\nServer Online. Waiting for client...." << endl;
 listen(sersock, 5);
 int sock = accept(sersock, NULL, NULL);
 cout << "Connection Established." << endl;
 return sock;
}
long randInRange(long low, long high) // excluding high and low
{
 return rand()%(high-(low+1)) + (low+1) ;
}
long mod(long a, long b)
{
return a >= 0 ? (a%b) : b-(abs(a)%b) ;
}
long powermod(long a, long b, long c)
{
 long res=1;
 for(int i=0; i<b; i++)
 {
 res = (res * a) % c;
 }
 return res;
}
long findInverse(long R , long D)
{
 int i = 0;
 long N = D; // copy D to N for taking mod
 long p[100] = {0,1};
 long q[100] = {0} ;
 while(R!=0)
 {
 q[i] = D/R ;
 long oldD = D ;
 D = R ;
 R = oldD%R ;
 if(i>1)
 {
 p[i] = mod(p[i-2] - p[i-1]*q[i-2], N) ;
 }
 i++ ;
 }
 if (i == 1) return 1;
 else return p[i] = mod(p[i-2] - p[i-1]*q[i-2], N) ;
}
long H(long M) // Hash Function
{
return (M ^ 1234); //hash key = 1234
}
int main()
{
 int port; cout << "\nEnter port : "; cin >> port;
 int sock = createServer(port);
 long p, q; // prime numbers
 long r, s; // signature
 long k, x, y, g; // keys
 long M, hashval; // Message and Hash
```

```
srand(time(NULL));

cout << "\nEnter a large prime number, p : "; cin >> p;

cout << "Enter a prime number, q (p-1 divisible by q & q>2) : "; cin >> q;

if( (p-1)%q != 0 || q <3) { cout << "\nInvalid input\n"; exit(-1); }

cout<<"Enter message, M = "; cin >> M;

hashval = H(M);

cout << "\nH(M) = " << hashval << endl;

long h;

do{

h = randInRange(1, p-1); // 1 < h < p-1

g = powermod(h,(p-1)/q, p); //g > 1

} while(g<=1);

cout << "g = " << g;

x = randInRange(1, q); cout << "\nServer's Private key, x = " << x;

y = powermod(g, x, p); cout << "\nServer's Public key, y = " << y;

k = randInRange(1, q); cout << "\nSecret key, k = " << k << endl;

//Signing

r = powermod(g, k, p) % q;

s = (findInverse(k,q) * (hashval + x*r )) % q;

cout<<"Sajag Agrawal \n 21BCT0438"<<endl;

cout << "\nServer's Signature {r,s} = {" << r << ", " << s << "}" << endl;

send(sock, &p, sizeof(p), 0);

send(sock, &q, sizeof(q), 0);

send(sock, &g, sizeof(g), 0);

send(sock, &y, sizeof(y), 0);

send(sock, &M , sizeof(M), 0);

send(sock, &r, sizeof(r), 0);

send(sock, &s, sizeof(s), 0);

cout << "\nSent p, q, g, and public key to client.";
```

```
 cout <<"\nSent message along with signature to client." << endl << endl;
}
```

Client Program

```
# include <bits/stdc++.h>
# include <arpa/inet.h>
using namespace std;
int connectToServer(const char* ip, int port)
{
 int sock = socket(AF_INET, SOCK_STREAM, 0);
 struct sockaddr_in addr = {AF_INET, htons(port),inet_addr(ip)};
 if(connect(sock, (struct sockaddr *) &addr, sizeof(addr)) < 0 ){
 cout << "\nRun server program first." << endl; exit(0);
 }else{
 cout << "\nClient is connected to Server." << endl;
 }
 return sock;
}
long mod(long a, long b)
{
 return a >= 0 ? (a%b) : b-(abs(a)%b) ;
}
long powermod(long a, long b, long c)
{
 long res=1;
 for(int i=0; i<b; i++)
 {
 res = (res * a) % c;
 }
 return res;
```

```
}
long findInverse(long R , long D)
{
 int i = 0;
 long N = D; // copy D to N for taking mod
 long p[100] = {0,1};
 long q[100] = {0} ;

 while(R!=0)
 {
 q[i] = D/R ;
 long oldD = D ;
 D = R ;
 R = oldD%R ;
 if(i>1)
 {
 p[i] = mod(p[i-2] - p[i-1]*q[i-2], N) ;
 }
 i++ ;
 }
 if (i == 1) return 1;
 else return p[i] = mod(p[i-2] - p[i-1]*q[i-2], N) ;
}
long H(long M)
{
 return (M ^ 1234); //hash key = 1234
}
int main()
{
```

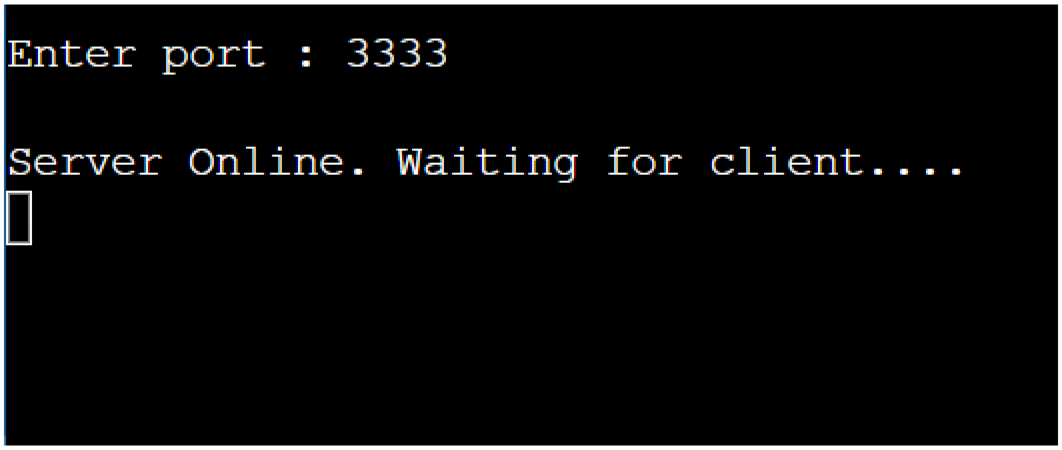
```
char ip[50]; cout << "\nEnter server's IP address: "; cin >> ip;
int port; cout << "Enter port : "; cin >> port;
int sock = connectToServer(ip, port);
long p, q; // prime numbers
long r, s; // signature
long g, y; // keys
long M, hashval; // Message and Hash
long w, v; // verify
srand(time(NULL));
recv(sock, &p, sizeof(p), 0);
recv(sock, &q, sizeof(q), 0);
recv(sock, &g, sizeof(g), 0);
recv(sock, &y, sizeof(y), 0);
recv(sock, &M , sizeof(M), 0);
recv(sock, &r, sizeof(r), 0);
recv(sock, &s, sizeof(s), 0);
cout << "Received p = " << p << endl;
cout << "Received q = " << q << endl;
cout << "Received g = " << g << endl;
cout << "Received y = " << y << endl;
cout << "Received M'= " << M << endl;
cout << "Received r' = " << r << endl;
cout << "Received s' = " << s << endl;
hashval = H(M) ;
cout << "\nH(M') = " << hashval << endl;
//Verifying
w = findInverse(s,q) % q; cout << "w = " << w << endl;
long u1 = (hashval * w) % q;
long u2 = (r * w) % q;
```

```
 v = ((powermod(g,u1,p)*powermod(y,u2,p)) %p) %q; cout<<"v = "<<v<<endl;
 if(v == r) cout<<"\nDigital Signature Verified. " << endl << endl;
 else cout<<"\nDigital Signature is invalid !!!" << endl << endl;
}
```

In local system issue of socket header file faced so need to use online compiler

Output

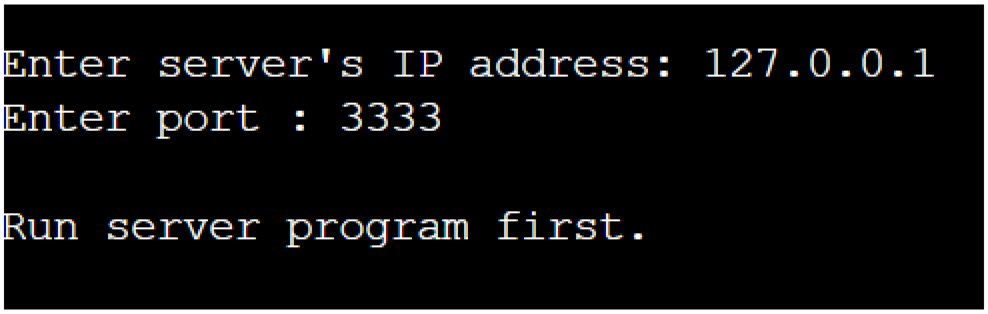
Server program



```
Enter port : 3333

Server Online. Waiting for client....
```

Client Program



```
Enter server's IP address: 127.0.0.1
Enter port : 3333

Run server program first.
```