Student Name: RITI	USN: 1SI19CS144	Batch No:B4	Date:

Evaluation:

Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)

Sl.No	Name of the Faculty In-Charge	Signature
1.	Dr. H K Vedamurthy	
2.	Dr. A H Shanthakumara	

Question No: 13. Compute common secret key between client and serverusing Diffie-Hellman key exchange technique. Perform encryption and decryption of message using the shared secret key (Use simple XOR operation to encrypt and decrypt the message.)

Algorithm:

Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{XA} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{XB} \bmod q$

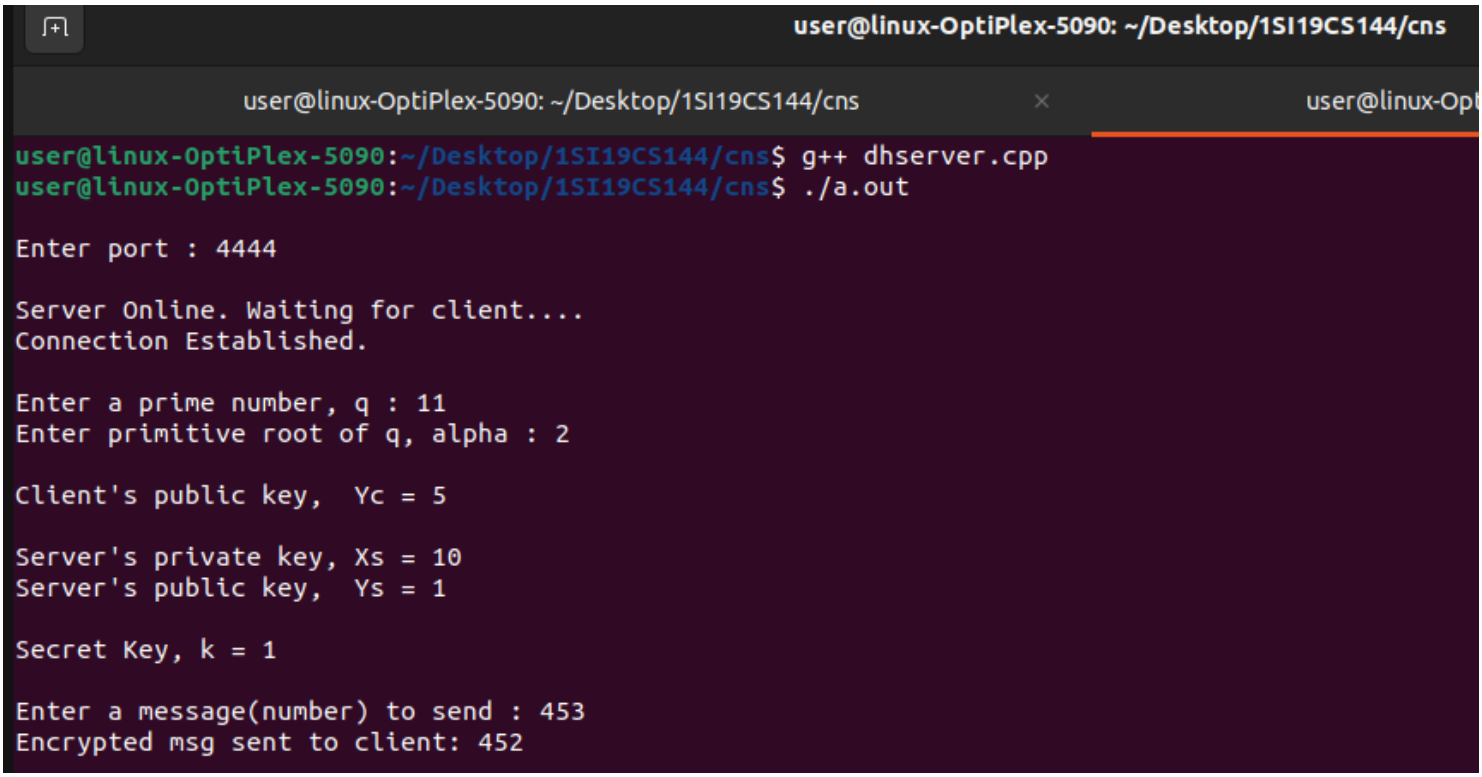
CODE:-

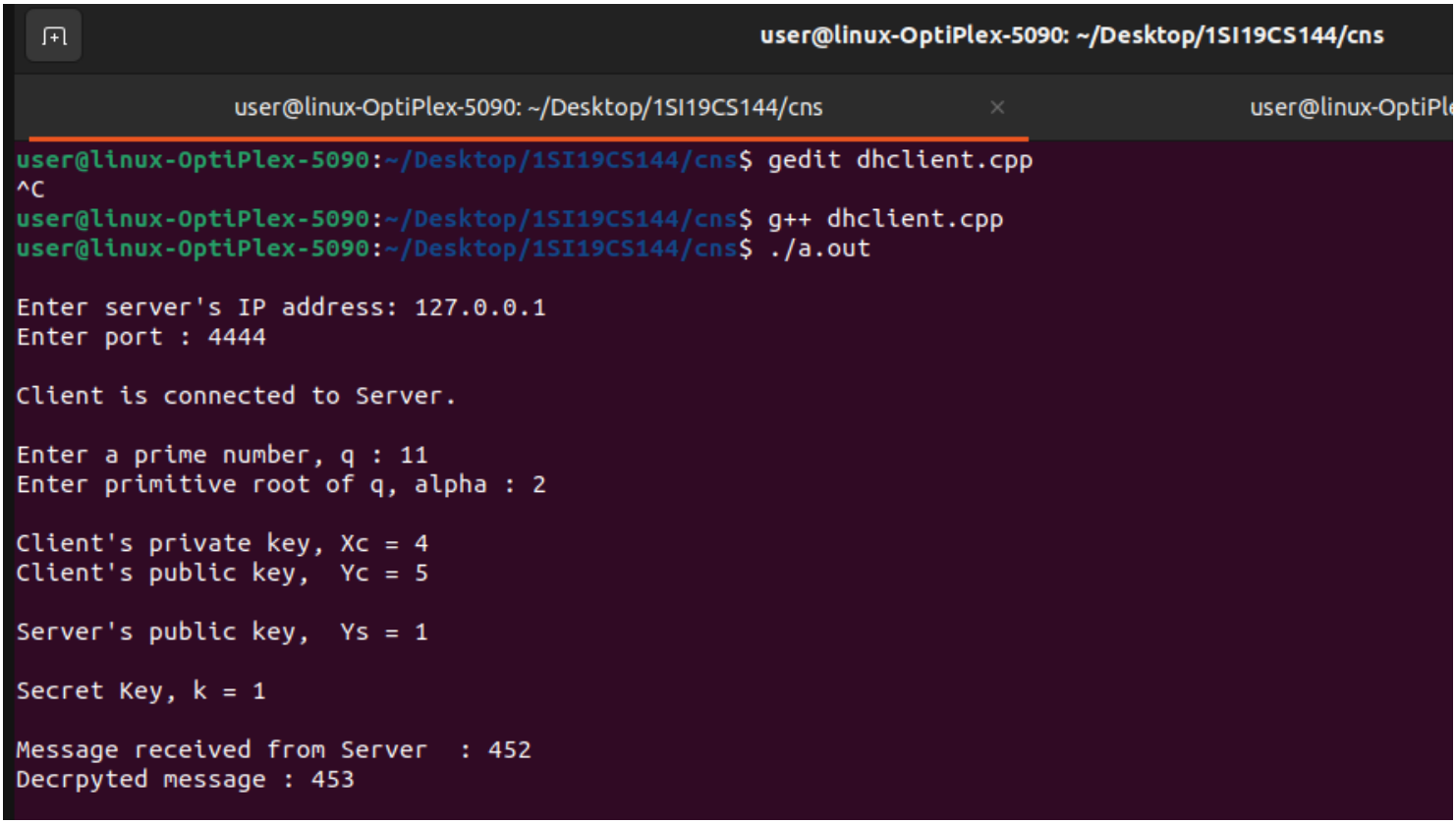
```
//server's code
#include <bits/stdc++.h>
#include <arpa/inet.h>

using namespace std;

int createServer(int port)
{
	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 powermod(long a, long b, long q)
{
	long res=1;
	for(long i=0;i<b;i++)
	{
		res=(res*a)%q;
	}
	return res;
}

int main()
{
	int port;
	cout<<"\n Enter port:
	";cin>>port;
	int sock=createServer(port);

	long q,alpha;
	cout<<"\nEnter a prime number, q:
	";cin>>q;
	cout<<"\nEnter primitve root of q, alpha:
	";cin>>alpha;

	long Yc; recv(sock,&Yc,sizeof(Yc),0);
	cout<<"\nClient's public key, Yc= "<<Yc<<endl;

	srand(time(NULL));
	long Xs=rand()%(q-2)+2;
	cout<<"\nServer's private key, Xs= "<<Xs<<endl;

	long Ys=powermod(alpha,Xs,q);
```

```
        send(sock,&Ys,sizeof(Ys),0);
        cout<<"\nServer's public key, Ys= "<<Ys<<endl;
        long k=powermod(Yc,Xs,q);
        cout<<"\nSecret Key, k="<<k<<endl;

        long msg;
        cout<<"\nEnter a message(number) to send:
        ";cin>>msg;

        long cipher=msg^k;
        send(sock,&cipher,sizeof(cipher),0);
        cout<<"Encrypted msg sent to client: "<<cipher<<endl<<endl;
}

//client's code

#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);
        }
        els
        e
        {       cout<<"\nCLient is connected to Server."<<endl;

        }

        return sock;
}

long powermod(long a,long b, long q)
{
        long res=1;
        for(long i=0;i<b;i++)
        {
                res=(res*a)%q;
        }
        return res;
}
```

```
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 q,alpha;
	cout<<"\nEnter a prime number, q:
	";cin>>q;
	cout<<"Enter primitive root of q, alpha: ";
	cin>>alpha;

	srand(time(NULL));
	long Xc=rand()%(q-2)+2;
	cout<<"\nClient's private key, Xc= "<<Xc<<endl;

	long Yc=powermod(alpha,Xc,q);
	send(sock,&Yc,sizeof(Yc),0);
	cout<<"Client's public key, Yc= "<<Yc<<endl;

	long Ys;
	recv(sock,&Ys,sizeof(Ys),0);
	cout<<"\nServer's public key, Ys= "<<Ys<<endl;

	long k=powermod(Ys,Xc,q);
	cout<<"\nSecret Key, k="<<k<<endl;

	long cipher;
	recv(sock,&cipher,sizeof(cipher),0);
	cout<<"\nMessage received from Server: "<<cipher<<endl;

	long decipher=cipher^k;
	cout<<"Decrypted Message: "<<decipher<<endl<<endl;
```

output:-


user@linux-OptiPlex-5090: ~/Desktop/1SI19CS144/cns
user@linux-OptiPlex-5090: ~/Desktop/1SI19CS144/cns
user@linux-Opt
user@linux-OptiPlex-5090:~/Desktop/1SI19CS144/cns$ g++ dhserver.cpp
user@linux-OptiPlex-5090:~/Desktop/1SI19CS144/cns$ ./a.out
Enter port : 4444
Server Online. Waiting for client....
Connection Established.
Enter a prime number, q : 11
Enter primitive root of q, alpha : 2
Client's public key, Yc = 5
Server's private key, Xs = 10
Server's public key, Ys = 1
Secret Key, k = 1
Enter a message(number) to send : 453
Encrypted msg sent to client: 452


user@linux-OptiPlex-5090: ~/Desktop/1SI19CS144/cns
user@linux-OptiPlex-5090: ~/Desktop/1SI19CS144/cns
user@linux-OptiPl
user@linux-OptiPlex-5090:~/Desktop/1SI19CS144/cns$ gedit dhclient.cpp
^C
user@linux-OptiPlex-5090:~/Desktop/1SI19CS144/cns$ g++ dhclient.cpp
user@linux-OptiPlex-5090:~/Desktop/1SI19CS144/cns$ ./a.out
Enter server's IP address: 127.0.0.1
Enter port : 4444
Client is connected to Server.
Enter a prime number, q : 11
Enter primitive root of q, alpha : 2
Client's private key, Xc = 4
Client's public key, Yc = 5
Server's public key, Ys = 1
Secret Key, k = 1
Message received from Server : 452
Decrpyted message : 453