Introduction to Security Hand-in 1 Report

Task1:

In task one, ElGamal encryption is asked, given the values

1. Public shared key (g)
2. Public shared prime (p)
3. System’s public key (h)
4. Ciphertexts (c1, c2)

The first thing we must recover, is our private key x, using system’s public key h. Here is the formula for this:

h = g^x (mod p) (1)

In the code, using the information 1 <= x <= p-2, a brute force functions is used, with optimized custom modulo exponential function which takes the third argument “modulo”.

To recover the message following formula is used:

M = c2 * (c1^x)^-1 (mod p) (2)

To be able to compute (c1^x)^-1 (inverse modulo) , Fermat’s formula is used, since it is computationaly faster considering p is a prime and not a huge number.

Task2:

In this task, modulo inverse and modulo exponential functions are reused. Modifying ciphertexts are based on formula (2) and also the fact that we want our target to satisfy:

M = T (mod p) (3) which implies there is a multiplier such that:

m = T * M^-1 (mod p)

Based on this idea c1 and c2 can be found through:

c2’ = c2 * m (mod p), and

c1’ = c1 * m (mod p)

Task3:

In task 3, proto and grpc plugins are used to create a server & client connection. As asked in the instructions, TLS protocol and a self-signed certificate is used in the connection configuration. The server and client only has send method and connection is started through flags in command line:

```
go run . // in server folder
```

```
go run . -addr localhost:7007 -msg "hello from client" -ca ..\server\server.crt -servername localhost // in client folder (with example message)
```

Certificate is created with openssl.