Mandatory Assignment I

ElGamal Encryption:
This program was created as an exercise for the Security-1 course at the IT-University of Copenhagen in March 2021. The ElGamal encryption scheme is based on the Diffie-Hellman key-exchange/key-agreement algorithm, where “the purpose of the algorithm is to enable two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages.” (Stallings & Brown, 2018).

(a):
Alice wants to send 2000 kr to Bob through a confidential message. She decides to use the ElGamal public key method. The keying material used to send the message to Bob is as follows:

- The shared base *g* = 666
- The shared prime *p* = 6661
- Bob’s public key *PK* = g^x mod p = 2227

Build an encrypted message containing ‘2000’.

(b):
Eve intercepts Alice’s encrypted message. Find Bob’s private key and reconstruct Alice’s message.

(c):
Assume that Eve run on a constrained device and is unable to find Bob’s private key. Modify Alice’s encrypted message so that when Bob decrypts it, he will get the double amount originally sent from Alice (so, if Alice sends ‘2000’, then Bob would decrypt ‘4000’). Note that you don’t have to encrypt a message containing ‘4000’.

Solution:

(**a**):
In order for Alice to send an encrypted message to Bob and vice versa, they need to agree on a shared secret-key. Alice computes the shared secret-key as:

$(g^x)^y \, mod \, p = K$

which is identical to:

$PK^y \, mod \, p = \mathrm{K}$

PK is Bob's public-key which is computed by:

$g^x \, mod \, p = PK$

We can input the values of *g, p* and PK since they are publicly known. The exponent *x* is Bob's private-key, and only Bob is suppose to know the value of *x*. Alice is able to get Bob's public-key, since it is public to everyone. This is the computation for Bob's public-key when inserting the publicly known values:

$666^x \text{ mod } 6661 = 2227$

Now that Alice has Bob's public-key, which she is able to receive from the internet, she can compute the shared secret-key:

$2227^y \, mod \, 6661 = K$

The exponent *y* is Alice's secret-key, which only Alice knows. The value of *y* have not been given to us in the assignment description, so we will have to choose a value between. As long as $y \in \mathbb{Z}^*_p$ is true, then Alice will be able to compute the shared secret-key. So if $y = 5$, then:

$2227^5 \, mod \, 6661 = 1930$

This is what makes asymmetric encryption powerful – the ability for two users to exchange a secret-key without the requirement of meeting in person or communicating over a secure channel, which in contrast is required by symmetric encryption.
Now that Alice has obtained the shared secret-key, she can use it to encrypt her message. She does this with the ElGamal encryption-scheme:

$c = g^{xy} * m \, mod \, p$

We know that $m \in \mathbb{Z}^*_p$, since the value is 2000. We can then input all the values:

$2227^5 * 2000 \text{ mod } 6661 = 3281$

Alice has now encrypted her plaintext message with the ElGamal encryption-scheme.

(**b**):
The Diffie-Hellman algorithm was designed in a way, so that when given the value of PK:

$g^x \, mod \, p = PK$

666^x mod 6661 = 2227

The reverse procedure of finding the value of the exponent *x*, is hard. This is called the discrete logarithm problem, which is possible to solve if the shared prime *p* is of small size. A secure size would typically be 128-bit long, which is 2^{128}. In this assignment *p* is of size 6661, so Eve knows that Bob's secret-key *x* is between 1-6660. This means that Eve is able to find the value *x* in a reasonable amount of time by executing a brute-force attack. A brute-force attack is a trial and error, attack, where every single key combination is attempted.

(**c**):
In order to modify the encrypted message from Alice, Eve will change the original encrypted message, and thereby compromises the integrity of the message without Alice knowing.

$g^{xy} * 2m = 2g^{xy * m}$