Assignment 3

Network and Systems Security (SIL-765)

Submitted By:-

Ruchir Dhiman (2016MCS2685)
Shantanu Agarwal (2016MCS2661)

Problem Statement

Transferring encrypted messages between two clients *A and B,* using the ElGamal Cryptosystem. It is also required that man-in-the-middle attack is not possible.

To ensure that man-in-the-middle attack does not occur, all messages are sent with an added MAC, as an integrity check, which is based on the HMAC algorithm.

Introduction

ElGamal Cryptosystem

The ElGamal Encryption System is an asymmetric key encryption algorithm devised by Taher ElGamal, which uses public key cryptography for sending messages. ElGamal is based on the Diffie-Hellman key exchange. It asymmetrically encrypt keys previously used for symmetric message encryption (like the one generated by Diffie-Hellman).

The cryptosystem also defines a digital signature algorithm, but that is not discussed or used, here.

Technical Details

1. Both the clients A *(Alice)* and B *(Bob)* are implemented in Python 2.7.
2. The two communicate with each other using socket communication provided by Python.
3. The following functions are done using inbuilt functions: hashing (SHA256), MAC generation (HMAC), and RSA based encryption and decryption.

Implementation

Assumptions

1. Alice and Bob are already know the parameters **q** and **α** (the prime number and the primitive root of prime, respectively).
2. They also have RSA-based public and private key pairs, and both are aware of the other's public key.
3. Alice and Bob both already know the algorithms used for hashing, and signing etc.

Algorithm

The sharing of the HMAC secret is done as follows:

1. Alice generates a 128 bit random number, which is the *secret* used for HMAC calculation.
2. She encrypts the secret using Bob's public key.
3. She then signs the *encrypted secret* by first hashing it using SHA256 and then signing the hash.
4. She packs the *encrypted secret* and the *hash* into a message and sends it to Bob.
5. Bob can then decrypt the message to obtain the secret, as well as authenticate that the original sender was Alice using the signed hash.

Initial message passing and sharing of Alice's public key:

Key Generation by Alice

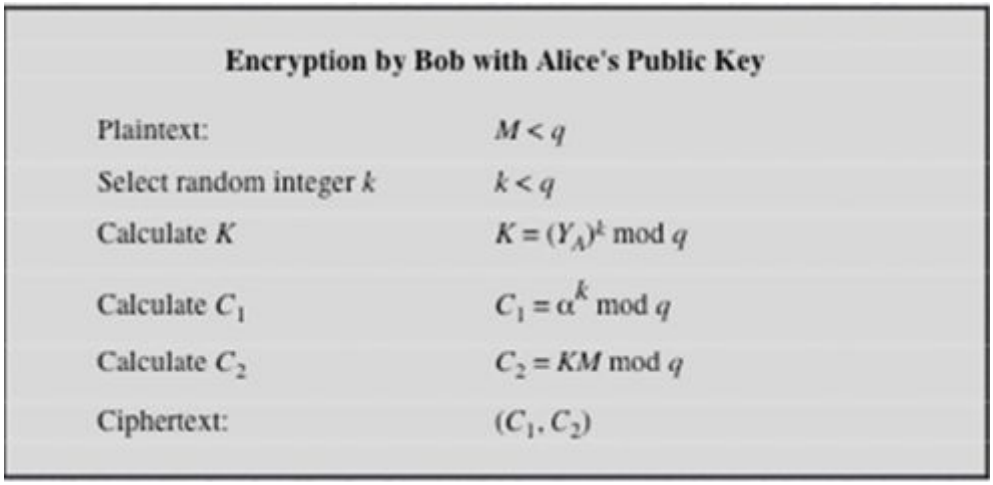
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

Figure 1.

1. Alice uses the pre-set number **Xa** (its private key) to calculate its public key **Ya** using the formula mentioned in Figure 1.

2. She, then, proceeds to send Ya, to Bob, and this message is accompanied with the MAC calculated using the HMAC algorithm.
3. Bob authenticates the message using the received MAC, and if authentication is successful, he now has the public key of Alice.

Transfer of messages:



Encryption by Bob with Alice's Public Key

Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Figure 2.

1. Bob then uses the mechanism mentioned in Figure 2 to send the given messages to Alice. This is done using the **elgamal** function which takes the message as an argument.
2. Since, ElGamal works with numbers, Bob first translates the message to be sent into an integer. This illustrated in the following example.

   String: **ab** (a: 01100001, b: 01100010)

   Converted to int: **10110000101100010** (1||a||b)

   (Note: The converted int is not a binary representation; it is an integer)
3. Once, Bob has the message as an integer, he randomly generates a random integer ***k*** (as given in Fig. 2), and calculates ***K, C1,*** and ***C2.*** The mac values associated with C1 and C2 are also calculated using the secret mentioned above.
4. Bob, then sends the messages to Alice, the messages have C1, C1's mac, C2, and C2's mac, in that order.

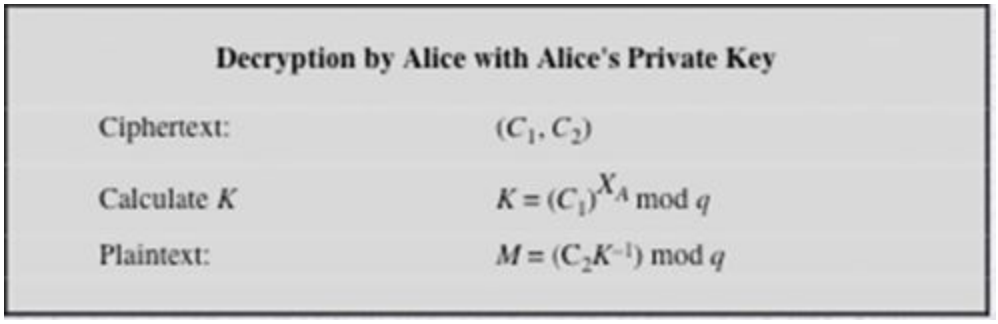


Figure 3.

5. Alice reads the messages as they arrive and decrypts them using the mechanism mentioned in Figure 3.
6. She also re-converts the received message into text, by performing the inverse of the operations performed by Bob.