Network and Systems Security (SIL-765)

(Assignment-3)

Made By :-
Ruchir Dhiman (2016MCS2685)
Shantanu Agarwal (2016MCS2661)

Problem Statement and Assumptions

- 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.

Assumptions:

1. Alice and Bob are already know the parameters q and α (the prime number and the primitive, 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.

Design

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

The HMAC Secret

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

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

- Alice selects a private key Xa, calculates Ya using the formula given above.
- She, then, proceeds to send Ya, to Bob, and this message is accompanied with the MAC calculated using the HMAC algorithm.

Encryption by Bob

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)

- Bob performs the operations given in this figure, to encrypt and send the messages.
- The contents of the message include the ciphertext given above, as well as the MAC of both C1 and C2.

Encoding of Message to Integer

- Conversion to an integer is done by utilising the unicode used to represent characters.
- An inbuilt python function is used to obtain the unicode for each character and the way the message is translated is illustrated by this example.

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

Converted Message: **10110000101100010** (1||a||b)

Decryption by Alice

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2K^{-1}) \bmod q$

- Alice decrypts the message by performing the operations given above. She also uses the MACs received to ensure that the message hasn't been altered in between.
- The plaintext she obtains is a number, and she obtains the message by performing the reverse of the operations given in this diagram.

Thank You