Problem Set 4

Group Members: Janhavi Mahajan
Shruthi Kulkarni

Setup:

1. Architecture:

This design of a secure instant messaging system contains a server which is used for mutual authentication and login. The shared key is used by the user to give commands to the server and get valid responses from the server. The users can communicate with each other if they are logged into the server and at least one of the user has obtained a ticket to talk to other user or users. Once the users have finished communicating with each other they need to logout from the messaging system. If they don't log out then the session will be terminated after a fixed time.

The symmetric encryption algorithm is based on the 256-bit key variant of AES (Advanced Encryption Standard), using CBC (Cipher Block Chaining) mode of encryption. The asymmetric encryption algorithm is RSA-2048, while the cryptographic hash function employed is SHA-256.

1. Assumptions

The Diffie-Hellman parameters p and g are already known by both parties.
The Clients know the server's public key.
The public/private key pair used in any communication is generated for that session and is not stored.
The Server reads username, salt value and hash(salt||password) for each user from a file

2. Services

1. Protection against Denial of Service attack
2. Perfect Forward Secrecy
3. Endpoint Identifier Hiding
4. Protection against weak passwords
5. Protection against Man-in-the-middle/Replay attack
6. Protection against offline dictionary attack and fairly resistant to online password guesses
7. Protection against Reflection attack

1. PROTOCOLS

1. Login

1. Client -> Server "Connect me"
2. Server -> Client SHA256 (IP_{Client}, R), R[127:20]
3. Client -> Server R[19:0] $\{g^{\wedge client\_secret} \text{modp}, \text{client\_public\_key}, N1\}_{server\_public\_key}$
4. Server -> Client $\{g^{\wedge server\_secret} \text{modp}, N1+1, N2\}_{client\_public\_key}$
   At this stage they both have calculated the Shared Session Key from the DH component received. Shared Key $K = g^{\wedge client\_key * server\_key} modp$
5. Client -> Server K{username , password, N2+1,N3}

The server verifies the password by calculating the SHA256(salt||password) and verifying it with the stored SHA256(salt||password)

6. Server -> Client K{N3+1, N4}

2. List

1.Client -> Server K{LIST , N5 , N4+1}

2.Server -> Client K{List of users , N5+1 , N6}

3. Communication between two clients

Step 1: Get Permit from the server.

The initiating server gets permit from the server to talk to another client. A and B are clients

1. A -> Server K {"Connect me to B", N4+1, N5}
2. Server -> A K {TicketToB, K_{AB}, $B_{IPAddress}$, $B_{PortNumber}$, "B" , N5+1, N6 }
3. A -> Server K {"Received", N6+1}

   TicketToB = {K_{AB}, N5+1, $A_{IPAddress}$, $A_{PortNumber}$, "A"} $_{B\_public\_key}$
4. A -> B TicketToB, K_{AB} {"I wan to talk to you", N6+1, N7}
5. B -> A K_{AB} {$g^{\wedge B_secret}$modp ,N7+1,N8}
6. A ->B K_{AB} {$g^{\wedge A_secret}$modp , N8+1,N9}

At this stage they both have calculated the Shared Session Key from the DH component received. Shared Key $K_{ABNew} = g^{\wedge B\_secret * A\_secret} modp$

Step 2: Message Exchange.

A -> B K_{ABNew} {"Hi", N9+1, N10}

B -> A K_{ABNew} {"Hi, how are you", N10 + 1, N11}

4. Logout

1. Client -> Server K {LOGOUT, N11+1, N12}
2. Server -> Client K {"You have been logged out", N12+1}

If the clients have communication going on then the disconnecting client logs out from that connection as well. Server and Client delete the shared session key and also the sequences of the Nonces

4. Discussion

1. AES-256 has been used to ensure security and make the symmetric encryption
2. CBC mode of encryption makes the protocol robust with exchange of multiple messages.
3. RSA-2048 is used for asymmetric encryption, where the 2048-bit keys are in accordance with the current specification and considered secure enough until the end of year 2030.RSA 4096 is used as server key.
4. SHA-256 is used as a secure hashing function, with a 256-bit message digest, which may then be used as a key for AES-256 during Step 1.
5. Mutual Authentication is provided between every user and server, and also between the communicating users.
6. Establishment of shared session keys ensures Perfect Forward Secrecy. These session keys are discarded after session termination and new keys are established at the start of every session. This is valid between any user and the server, and also between any pair of communicating users.

7. The endpoints reveal their identity only after a session key establishment, and encrypted with the same. This ensures Endpoint Identifier Hiding.
8. Protection against weak passwords is provided by passing password encrypted in the shared session key. This is also effective against offline dictionary attacks.
9. The server stores hashes of all users' passwords and salt and not the passwords themselves making it difficult for the intruder to attack.
11. DoS attacks are prevented by messages 2 and 3 in LOGIN. The attacker needs to send a response.
13. Confidentiality is maintained as the messages exchanged between the users are encrypted with their shared session key.
14. Integrity is maintained by using HMAC and SHA256.