The Setup

- Architecture

There are two types of entities: Server and Clients.

- Assumptions

The clients have the Public Key of the Server (PKserver).
The server has the usernames and SHA256 hash of the Password of all the users.

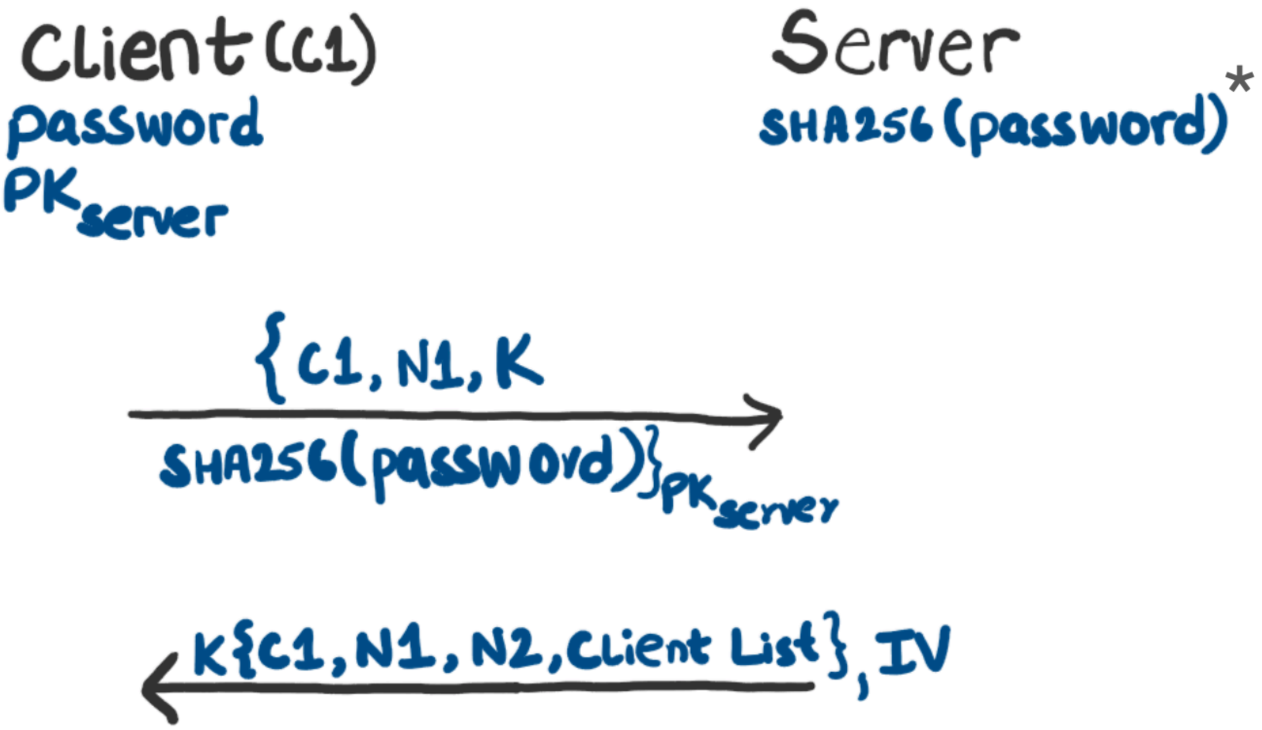
- Services

Perfect Forward Secrecy, Confidentiality, Authentication, Integrity, Non-Repudiation, and Endpoint Hiding.

The application runs on UDP protocol to avoid connection-based attacks.

Login Protocol

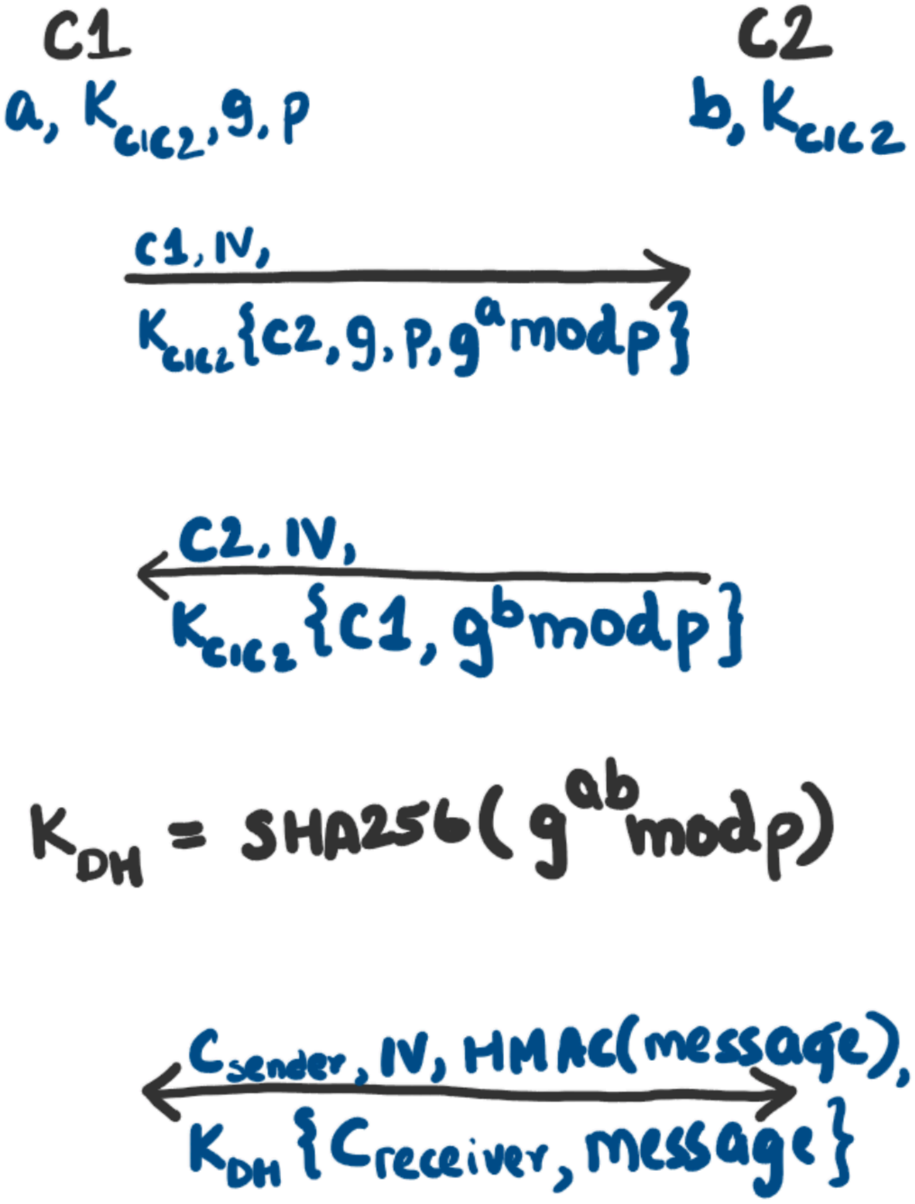
- The SHA256 hash of the password authenticates the client to the server
- N1 is a nonce that authenticates the server to the client
- K is a 256-bit shared key generated by client and used to encrypt communication between server and client (used in “login”, “list”, and “logout”)
- N2 is a nonce that will be later used in the logout protocol
- The server maintains a table with shared keys for each client pair
- A shared key is a randomly generated 256-bit value assigned to each client pair
- Each client receives a custom client list which includes the corresponding shared key
- Whenever a new client logs in, the updated shared keys table is sent to all the logged in clients



*The hash of the password is stored salted to prevent offline password attacks

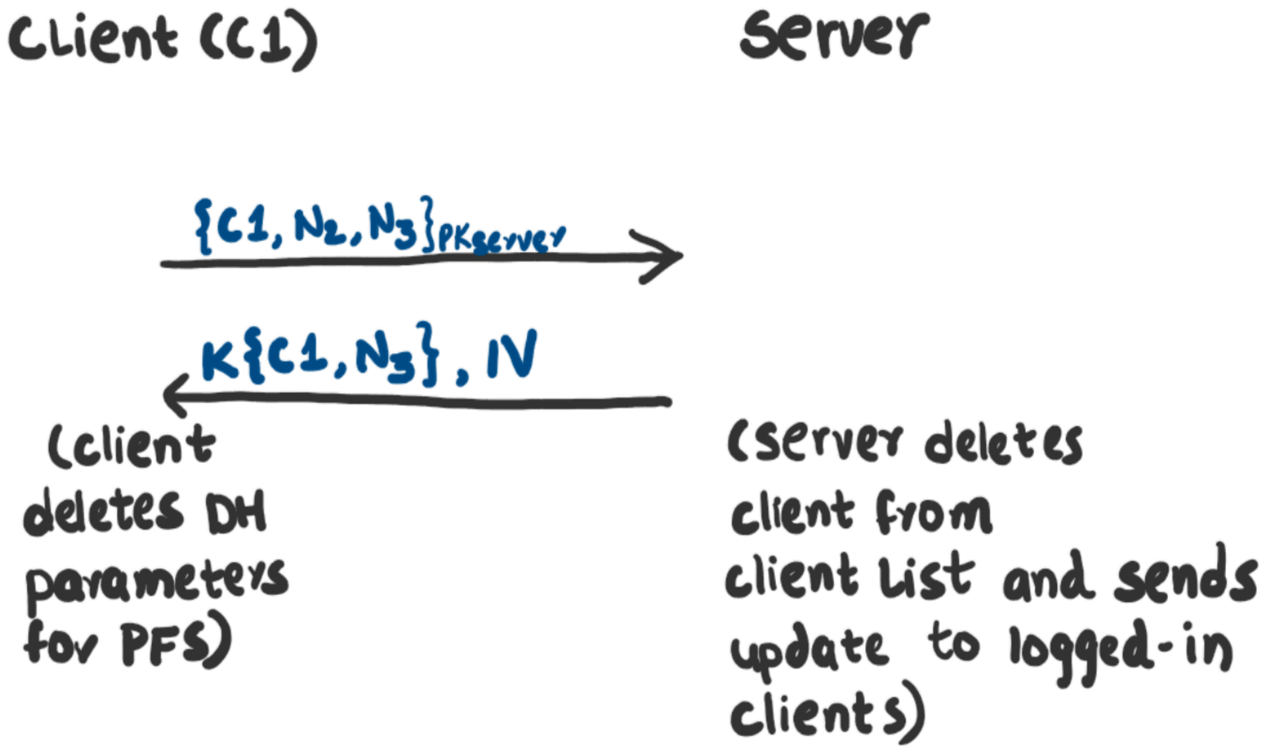
Key Establishment and messaging protocol

- The shared key, K_{c1c2}, is used to encrypt the Diffie-Hellman parameters used to setup the final shared key, K_{DH}
- The protocol supports partial endpoint hiding
- This authenticates the clients mutually and prevents MITM attacks while setting up K_{DH}
- HMAC of the messages are included to provide integrity and authentication


C1
C2
a, K_{c1c2}, g, p
b, K_{c1c2}
$C1, IV,$
$K_{c1c2}\{C2, g, p, g^a \mathrm{mod} p\}$
$C2, IV,$
$K_{c1c2}\{C1, g^b \mathrm{mod} p\}$
$K_{DH} = SHA256(g^{ab} \mathrm{mod} p)$
$C_{sender}, IV, HMAC(message),$
$K_{DH}\{C_{receiver}, message\}$

Logout Protocol

- Encrypting the logout request from the client with server's public key allows full endpoint hiding
- N_2 (from Login protocol) is used to prevent de-authentication attacks by legitimizing the logout request
- N_3 is a nonce used to prevent replay attack where a previous logout response from the server can be used to trick the client to think it has logged out
- Perfect Forward Secrecy is provided by forgetting a, b, and K_{DH} at the end of the logout process


Client (C1)
Server
$\{C1, N_2, N_3\}_{PKserver}$
$K\{C1, N_3\}$, IV
(client deletes DH parameters for PFS)
(server deletes client from client list and sends update to logged-in clients)