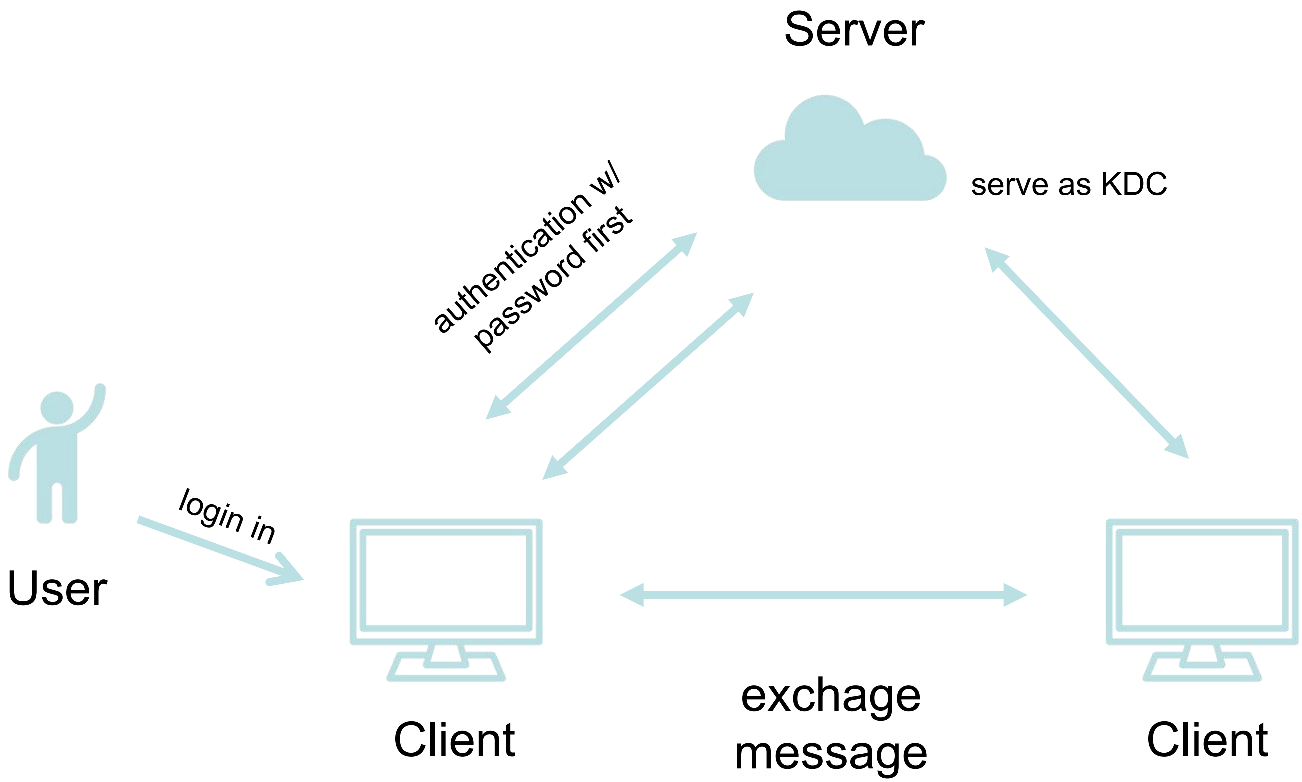
Setup

- Architecture



Setup

- Assumptions

  KDC stores all the user's keys

  Each user will be assigned unique keys

- Services

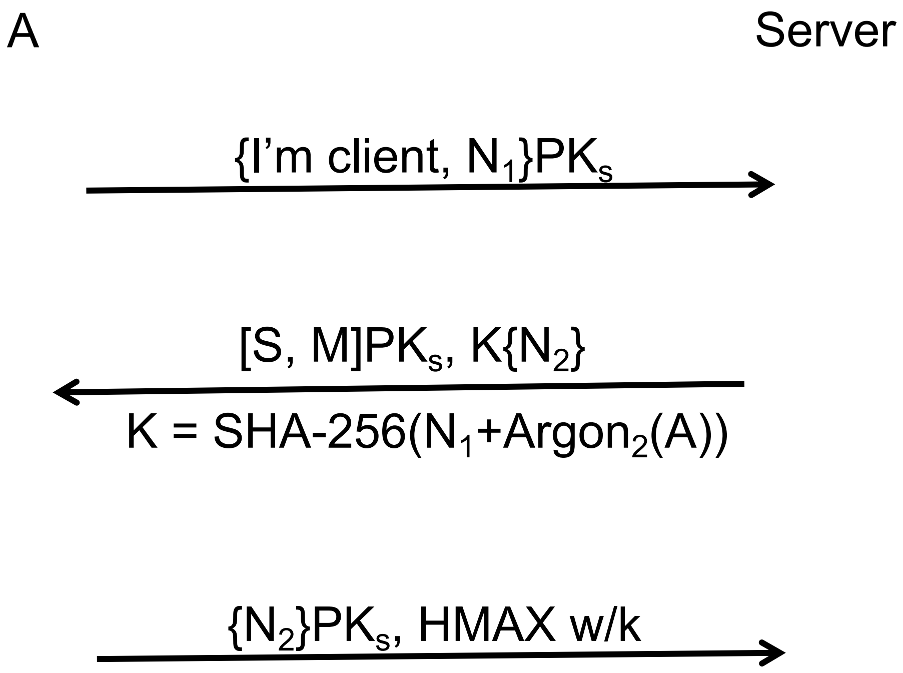
  Clients identificate each other via Server.

  Clients can establish connection with others to exchange message instantly. One connection at a time.

  User can login client using username and a single password.

Protocols

- Login



A

Server

{I’m client, N_1}PK_s

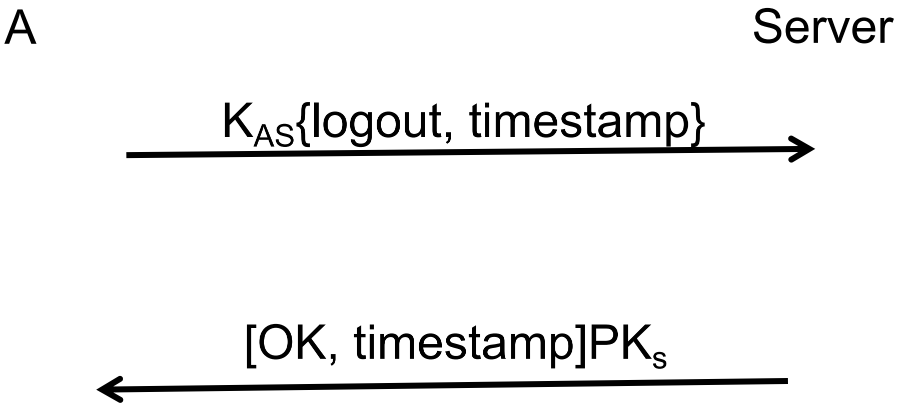
[S, M]PK_s, K{N_2}

K = SHA-256(N_1+Argon_2(A))

{N_2}PK_s, HMAX w/k

Protocols

- Logout



A

Server

K_{AS}\{logout, timestamp\}

[OK, timestamp]PK_s

Discussion

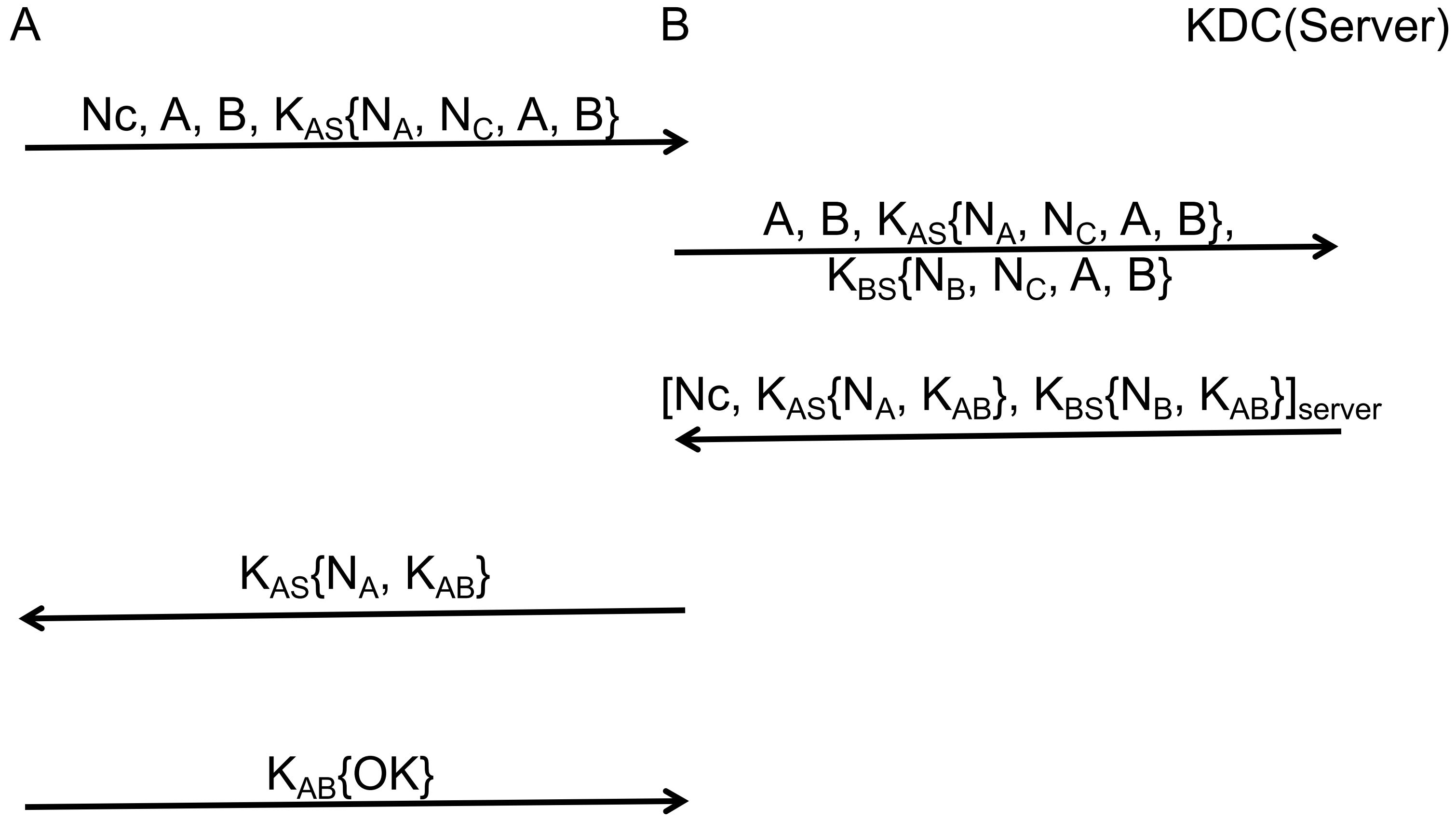
The login authentication and key establishment protocol provides both DoS protection and PFS

Prior was possible because we utilized challenge-response mechanisms which require the computing of Argon2 hash, this has minimal impact on normal users but can deter mass requests from automated scripts.

PFS was ensured since we opt for a session based scheme, where each session is initialized by login request and ended with logout, and a session key is established using the nonce generated during that session. Thus we can make sure no adversary who compromises client or server would be able to decrypt/replay past messages.

Protocols

- Comnication Between Clients



A B KDC(Server)

$Nc, A, B, K_{AS}\{N_A, N_C, A, B\}$

$A, B, K_{AS}\{N_A, N_C, A, B\}, K_{BS}\{N_B, N_C, A, B\}$

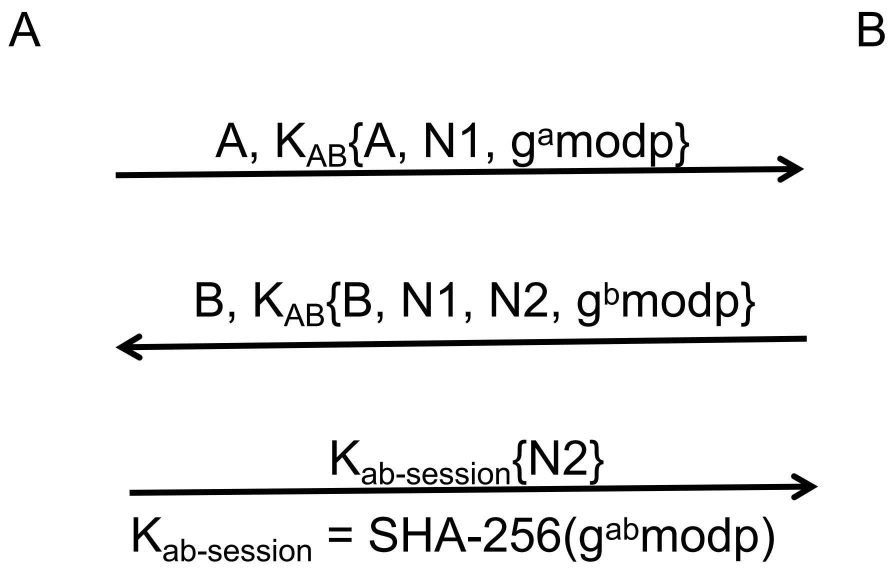
$[Nc, K_{AS}\{N_A, K_{AB}\}, K_{BS}\{N_B, K_{AB}\}]_{server}$

$K_{AS}\{N_A, K_{AB}\}$

$K_{AB}\{OK\}$

Protocols

- Comnication Between Clients



A

B

A, $K_{AB}\{A, N1, g^{a}modp\}$

B, $K_{AB}\{B, N1, N2, g^{b}modp\}$

$K_{ab\text{-}session}\{N2\}$

$K_{ab\text{-}session}$ = SHA-256($g^{ab}modp$)

Discussion

We choose a protocol in the style of Otway-Rees to communicate between clients. It provides a mechanism for clients to share a communication secret key. The original Otway-Rees protocol, however, has some flaws. And we have several changes. Firstly, to prevent key exposure, we generate secret key to communicate between client and KDC in each sessions. Further more, we sign the message sending by KDC to guarantee KDC is trustworthy and to guarantee message integrity and confidentiality.