Secure Messenger Design

By Yifei Sun, Yiting Wu

Architecture: KDC + Clients

Assumptions: KDCs are not trusted

Workflow: Password → Argon2id → Ed25519 → CS X[1] → CC X[2]

Services: Login (w/ username or anonymous), List, Encrypted Messaging, Logout

- Servers generate new keys for every new "session".
- Keys generated with server are will only be used once for only one purpose

Login/Message/Logout Protocols

- Login: send identity (host + port), hash
- Message: encrypted text, hash
- Logout: remove all keys

---

[1] Client-Server Key Exchange

[2] Client-Client Key Exchange

Password $\rightarrow$ Encryption Key

P: a password of arbitrary length provided by client

c_t: time cost factor for Argon2id KDF (int)

c_m: memory cost factor for Argon2id KDF (int)

r: salt

$K = \text{Argon2id}(\text{SHA-2}(P), c_t, c_m, r)$

Client-Server Ephemeral Session Key Generation

Assumption:

- KDC (server) generates a long-lived public/private key pair
- The key pair will stay the same for entire lifetime of the server (a new one will be generated if the server dies)
- A has an Ed25519 public/private key pair generated based on K
- KDC (S) has an randomly generated Ed25519 public/private key pair

Client-Server Ephemeral Session Key Generation (Modified TLS Key Exchange)

Step 1: A $\longrightarrow$ S: A, K_A, T_1

Step 2: S $\longrightarrow$ A: $K_S, \{T_1, T_2\}_{K_A}$

Step 3: A $\longrightarrow$ S: $\{K_{\text{AS}}, T_2, T_3\}_{K_S}$

Step 4: S $\longrightarrow$ A: $\{\text{Op}(T_3)\}_{K_{\text{AS}}}$

Client-Client Ephemeral Session Key Generation (Modified Kerberos)

Step 1: A $\longrightarrow$ B: A

Step 2: B $\longrightarrow$ A: $\{A, N_B\}_{K_{\text{BS}}}$

Step 3: A $\longrightarrow$ S: A, B, N_A, $\{A, N_B\}_{K_{\text{BS}}}$

Step 4: S $\longrightarrow$ A: $\left\{N_A, K_{\text{AB}}, B, \{K_{\text{AB}}, A, N_B\}_{K_{\text{BS}}}\right\}_{K_{\text{AS}}}$

Step 5: A $\longrightarrow$ B: $\{K_{\text{AB}}, A, N_B\}_{K_{\text{BS}}}$

Step 6: B $\longrightarrow$ A: $\{N_B\}_{K_{\text{AB}}}$

Step 7: A $\longrightarrow$ B: $\{N_B - 1\}_{K_{\text{AB}}}$

Step 8: B $\longrightarrow$ A: $\{B', g, p\}_{K_{\text{AB}}}$

Step 9: A $\longrightarrow$ B: $\{A'\}_{K_{\text{AB}}}$

Session key between A and B:

For A: $K = {B'}^{a} \bmod p$

For B: $K = {A'}^{b} \bmod p$

Summary

Argon2id KDF
- Memory hard / Long execution time
- Prevents on-/off- line dictionary attacks

Modified Kerberos
- Server does not know the session keys between two clients

Perfect Forward Secrecy: Ephemeral session keys

Denial of Service Attacks
- Spawn more KDCs
- KDCs trustworthiness won't affect communication security

End-points Hiding
- Address by usernames / Address by host + port