Overview: Needham-Schroeder Key Establishment Protocol

- A trusted server $S$
- Every node shares a secret with the trusted server
- $K_A$ is key shared between $A$ and server
- To establish a session secret $K$ with $B$

$$
A \rightarrow S : \ A \ || \ B \ || \ E_{K_A}(A \ || \ B)
$$

$$
S \rightarrow A : \ E_{K_A}(K \ || \ T)
$$

$$
T = \ E_{K_B}(K \ || \ A \ || \ B)
$$

$$
A \rightarrow B : \ T
$$

- Inter-nodal secrets are established by using the trusted server for mediation
- Requires *on-line* server
Kerberos Components

- Authentication server (AS)
- Ticket granting server (TGS)
- Servers (V)
- Clients (C)
- TGS and all clients share a secret (individually) with the AS
- All servers share a secret with the TGS.
Kerberos Operation

- Once a day: Clients login and obtain a ticket that will be honored by the TGS
- Once for every service (for example Email server, print server, file server etc.): Clients approach TGS to obtain a ticket (for each server)
- For every service session: Tickets provided by TGS (for that service) provided to the server (email / print / file)
- Tickets are authenticated with an “Authenticator” to prevent replay attacks.
Kerberos Overview

1. User logs on to workstation and requests service on host.
2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
5. Workstation sends ticket and authenticator to server.
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Figure 14.1 Overview of Kerberos

Authentication Server (AS)
Ticket-granting Server (TGS)
Kerberos (Version 4) Messages

$C \rightarrow AS : \quad ID_c \parallel ID_{tgs} \parallel TS_1$

$AS \rightarrow C : \quad E_C (K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel LT_2 \parallel T_{tgs})$

$T_{tgs} = \quad E_{TGS} (K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel LT_2)$

$C \rightarrow TGS : \quad ID_v \parallel T_{tgs} \parallel \{A_{c,tgs} = E_{c,tgs} (ID_c \parallel AD_c \parallel TS_3)\}$

$TGS \rightarrow C : \quad E_{c,tgs} (K_{c,v} \parallel ID_v \parallel TS_4 \parallel T_v)$

$T_v = \quad E_{tgs,v} (K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel LT_4)$

$C \rightarrow V : \quad T_v \parallel \{A_{c,v} = E_{c,v} (ID_c \parallel AD_c \parallel TS_5)\}$

$V \rightarrow C : \quad E_{c,v} (TS_5 + 1)$
Multiple Kerberi

- The Kerberos servers in different realms share a key.
- Typically the TGSs.
- Usually AS and TGS run on the same server.
Kerberos vs PKI

- Symmetric / asymmetric
- Online Server / Offline Server (ad hoc authentication)
- Scalability?
- Versatility? No equivalent for signatures with Kerberos.
- Revocation?
Key Predistribution

1. Key distribution center (KDC)’
2. Unlimited number of entities with unique IDs
3. KDC chooses a set of $P$ secrets $\Phi$.
4. Entity with identity $A$ receives secrets $\Phi_A = F(\Phi, A)$.
5. Entity with identity $B$ receives secrets $\Phi_B = F(\Phi, B)$.
6. Shared secret between $A$ and $B$ is $K_{AB} = G(\Phi_A, B) = G(\Phi_B, A)$.
7. No need for involvement of the KDC after distribution of secrets
Blom’s Polynomial Scheme

1. KDC chooses a $k - 1$-degree symmetric polynomial

$$F(x, y) \equiv \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} a_{ij} x^i y^j \mod p$$

2. The coefficients $a_{ij}$ (a total of $\binom{k}{2}$ unique coefficients) are the $P$ secrets chosen by the KDC ($P = \binom{k}{2}$).

3. $\Phi_A = G_A(X) \equiv F(X, A) \mod p$

4. $\Phi_B = G_B(X) \equiv F(X, B) \mod p$

5. $G_A(X), G_B(X)$ are $k - 1$ degree polynomials.

6. The $k$ coefficients of $G_A(X)$ are the $k$ secrets provided to $A$.

7. The $k$ coefficients of $G_B(X)$ are the $k$ secrets provided to $B$.

8. Both $A$ and $B$ can independently compute

$$G_A(B) = G_B(A) = K_{AB}$$

9. $G_A(B) = G_B(A) = F(A, B) = F(B, A)$
Whats the catch?

1. $n = \frac{k}{2} - 1$ entities pooling their secrets together can compute any secret - even all $P$ secrets chosen by the KDC.

2. Using $nk$ secrets ($k$ secrets each from $n$ entities) an attacker can construct $nk$ simultaneous equations.

3. If $nk \geq P = \binom{k}{2}$ the attacker can solve for all values of $P$.

4. $n \geq \frac{k}{2} - 1$ nodes need to collude.

5. If we need to resist collusions of 99 entities we need to choose $k = 200$.

6. Each node stores only 200 secrets.

7. Unlimited network size (limited only by the size of $p$ - as each node needs a unique ID in $\mathbb{Z}_p$).