Cryptography: Key Distribution

Luca Veltri
(mail.to: luca.veltri@unipr.it)

Corso di Sicurezza nelle reti di telecomunicazioni, a.a. 2009/2010
http://www.tlc.unipr.it/veltri

Key Distribution

- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
  - Often secure system failure due to a break in the key distribution scheme
- Given parties A and B have various key distribution alternatives:
  - A can select key and physically deliver to B
  - Third party can select & deliver key to A & B
  - If A & B have communicated previously can use previous key to encrypt a new key
  - If A & B have secure communications with a third party C, C can relay key between A & B

Trusted intermediaries

- With N nodes, each node must authenticate each other.. N-1 keys maintained by each node
- Possible solution: Key Distribution Center - KDC
  - Similar to CAs for public-key cryptography

![Key Distribution Diagram]

![Trusted Intermediaries Diagram]
Multiple trusted intermediaries

- Web of trusted KDCs
- Hierarchical trusted KDCs

Key Distribution in practice

- Key Distribution in theory (Client-Server)
- Key Distribution in practice (Client-Server)

Key Distribution

- Needham-Schroeder Protocol
  - original third-party key distribution protocol
  - for session between A B mediated by KDC

Protocol overview:
1. A → KDC : req || $N_f$ = $ID_A || ID_B || N_f$
2. KDC → A : $E_{Ks}[Ks || req || N_f || ticket]$ = $E_{Ks}[Ks || ID_A || N_f || E_{Ks}(Ks||ID_A)]$
3. A → B : ticket = $E_{Ks}[Ks||ID_A]$
4. B → A : challenge = $E_{Ks}[N_f]$
5. A → B : response = $E_{Ks}(f(N_f))$
**Key Distribution (cont.)**

- Lo scopo è quello di creare una nuova chiave di sessione $K_S$ tra A e B
- $K_A$ e $K_B$ sono chiavi condivise rispettivamente tra A e KDC, e tra B e KDC (master keys)
- $N_1$ e $N_2$ sono due valori di nonce utilizzati per evitare attacchi di tipo “replay” (replay attack)
- I passi da 1 a 3 servono per scambiare la chiave di sessione $K_S$
- I passi 4 e 5 (insieme al 3) svolgono il ruolo di autenticazione
- $f(x)$ è una qualsiasi funzione di $x$, e.g. $f(x) = x + 1$

---

**Kerberos**

- Designed at MIT based on late-70 ’s work by Needham and Schroeder
- Provides centralised private-key third-party authentication in a distributed network
  - allows users access to services distributed through network without needing to trust all workstations
  - relies on key distribution center (KDC) to perform mediated authentication
  - relies on conventional encryption
- KDC shares a key with each client and server
- Currently in use version 4 and 5 (Kerberos V4 and Kerberos V5)
- Version 4 makes use of DES
- Implemented in MS Windows2000 and linux (PAM)

---

**Kerberos**

- When a client wants to connect to a server
  - KDC sends to client
    - Session key encrypted with clients key
    - Session key + client ID encrypted with servers key (the ticket)
  - User forwards the latter (the ticket) to the server
  - User decrypts session key, server decrypts ticket to recover client ID and session key
    - Only the client can recover the client-encrypted session key
    - Only the server can recover the server-encrypted session key
- To avoid long-term password storage within the client workstation, the KDC generates a short-term client key
  - KDC sends the short-term client key encrypted with the user’s password to the client
  - Future clients↔KDC communications use the short-term client key
Kerberos (cont.)

- The KDC sends also a ticket-granting ticket (TGT) to the client
  - TGT contains the client short-term key and other user’s information (user’s name, expiration time, etc.) encrypted with the KDC master key
- The TGT is used in the successive client-KDC communications to inform the KDC about the short-term client key
- The KDC separates the authentication server and ticket-granting server
- The KDC is composed by two entities
  - an Authentication Server (KDC/AS)
    - users initially negotiate with AS to identify self
    - AS issues the ticket granting ticket TGT to talk to the KDC/TGS
  - a Ticket Granting server (KDC/TGS)
    - users subsequently request access to other services from TGS on basis of users TGT

AS and TGS

- Due servizi:
  - Authentication Service (AS)
    - KDC/AS emana i “Ticket Granting Ticket” (TGT) validi per ottenere il “ticket-granting service” nel reame
    - prima di ottenere ticket per i servizi, si deve ottenere un TGT dall’AS
  - Ticket-Granting Service (TGS)
    - KDC/TGS emana ticket validi per accedere ad altri servizi nel reame o in TGS di reami di fiducia
    - per l’accesso ad un servizio, si deve contattare il TGS, presentare un TGT, e richiedere un ticket

Kerberos v4 Overview

1. User keys on to workstation and requests service-on-host.
2. Authentication Server (AS) verifies user’s access rights in database, creates ticket granting ticket and session key. Results are encrypted using key derived from user’s password.
3. Authorization grants user access to KDC/AS and uses password to decrypt ticket granting message, then sends ticket and authentication 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. Server verifies that ticket and authentication match, then grants access to service, if mutual authentication is required, server returns an authenticator.

Authentication Service

KDC

Alice wants TGS, K_{AS\text{Alice}}(Alice, time)

\[ K_{SA\text{Alice}}(\text{use } S_{\text{Alice} \text{for TGS}}, \text{TGT} = K_{TGS}(\text{use } S_{\text{Alice} \text{for Alice}}) \]

KRB_AS_REQ

KRB_AS_REP

\[ K_{\text{KDC}} \text{ AS invents session key } S_{\text{Alice}}]
Ticket-Granting Service

Alice wants Bob, $S_{KDC} = \{\text{Alice, time}\}, \text{TGT} = K_{TGS}$ (use $S_{Alice}$ for Alice)

KDC

TGS invents session key $S_{AB}$

$S_{KDC} = \{\text{use } S_{AB} \text{ for Bob}, \text{ticket } = K_{TGT} \text{ for Alice}\}$

KRB_TGS_REQ

KRB_TGS REP

Client/Server exchange

Alice (client)

$S_{AB} = \{\text{Alice, time}\}, \text{ticket } = K_{TGT}$ (use $S_{AB}$ for Alice)

Bob (server)

$S_{AB} = \{\text{time}\}$

KRB_AP_REQ

KRB_AP_REP

Used as authentication info (A $\rightarrow$ B)

Optionally, for mutual authentication (B $\rightarrow$ A)

Kerberos Realms

- A Kerberos environment consists of:
  - A Kerberos server
  - A number of clients, all registered with server
  - Application servers, sharing keys with server
- This is termed a realm
  - Typically a single administrative domain
- If have multiple realms, their Kerberos servers must share keys and trust

Kerberos V5

- Developed in mid 1990’s
- Provides improvements over v4
  - Addresses environmental shortcomings
    - Encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
  - And technical deficiencies
    - Double encryption, non-std mode of use, session keys, password attacks
- Improvements:
  - Extended ticket lifetimes (V4 max = 21 hours)
  - Allowed delegation of rights
  - Allowed hierarchical realms
  - Added algorithms other than DES
  - V4 used ad hoc encoding, V5 used ASN.1
- Specified as Internet standard RFC 1510