



Cryptography: Key Distribution

Luca Veltri

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

Corso di Sicurezza nelle reti di telecomunicazioni, a.a. 2004/2005

<http://www.tlc.unipr.it/veltri>

Key Distribution

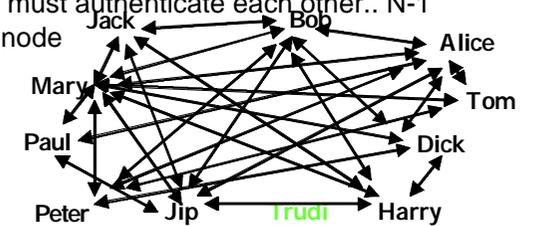
- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
 - often secure system failure due to a break in the key distribution scheme
- Given parties A and B have various **key distribution** alternatives:
 - A can select key and physically 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

Key Distribution

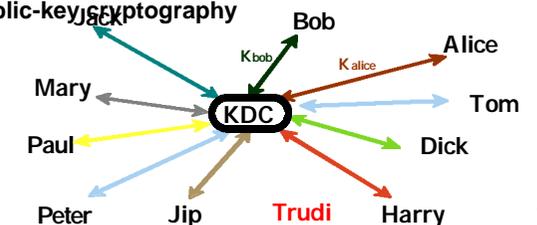
- given parties A and B have various **key distribution** alternatives:
 - A can select key and physically deliver to B
 - third party can select & deliver key to A & B
 - if A & B have communicated previously can use previous key to encrypt a new key
 - if A & B have secure communications with a third party C, C can relay key between A & B

Trusted intermediaries

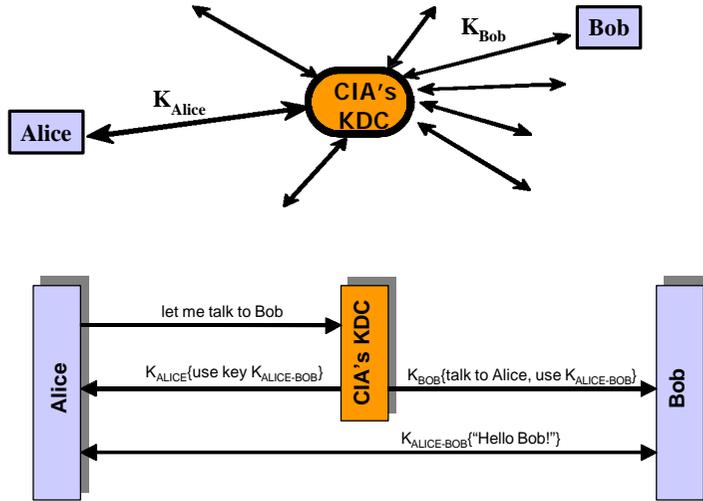
- With N nodes, each node must authenticate each other.. N-1 keys maintained by each node



- Possible solution: Key Distribution Center - KDC
 - similar to CAs for public-key cryptography

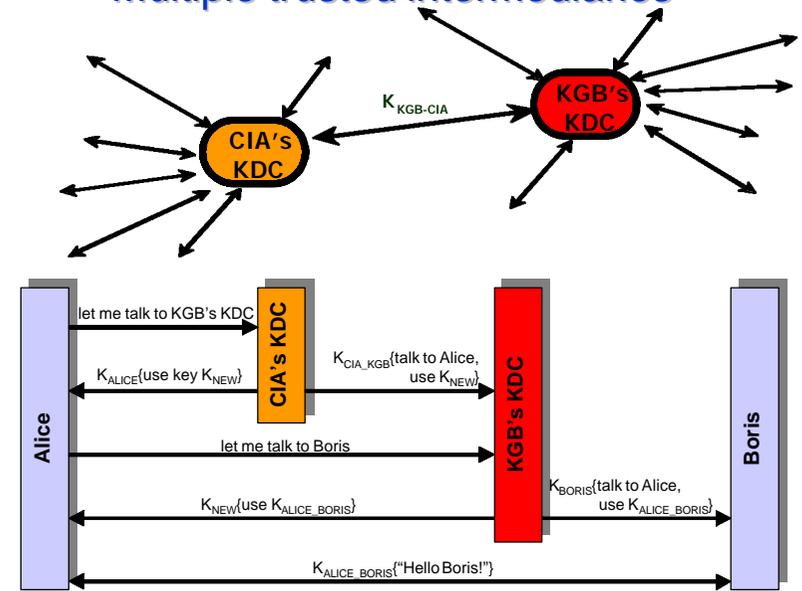


Trusted intermediaries



5

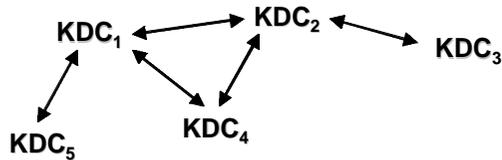
Multiple trusted intermediaries



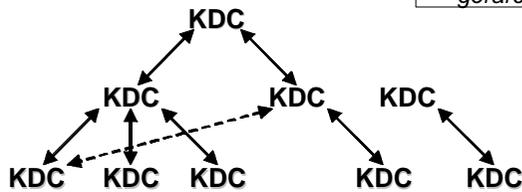
6

Multiple trusted intermediaries

a ragnatela

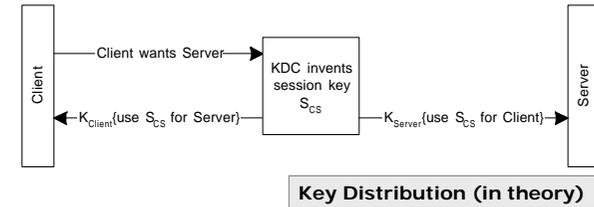


gerarchico

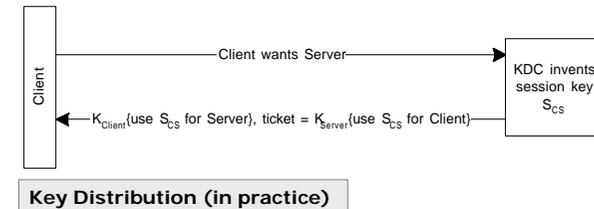


7

Key Distribution in practice



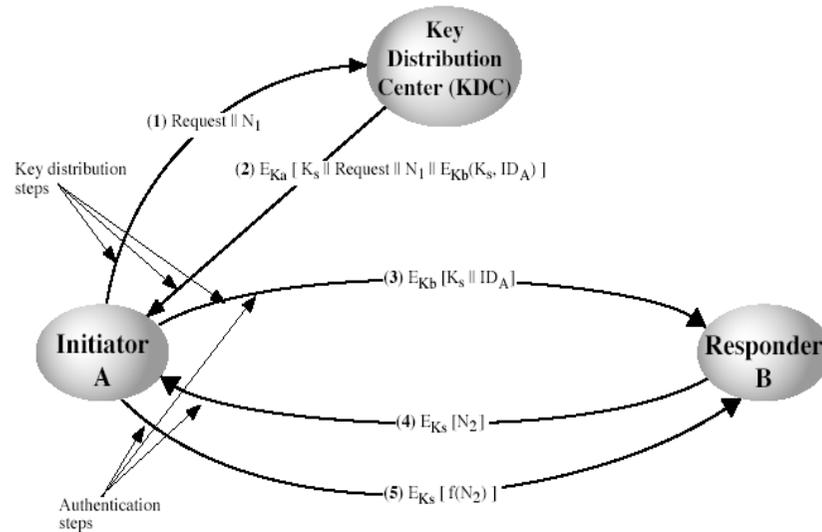
Key Distribution (in theory)



Key Distribution (in practice)

8

Key Distribution



9

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 $\parallel N_1$	$= ID_A \parallel ID_B \parallel N_1$
2. KDC → A : $E_{K_A}[K_S \parallel \text{req} \parallel N_1 \parallel \text{ticket}]$	$= E_{K_A}[K_S \parallel ID_B \parallel N_1 \parallel E_{K_B}[K_S \parallel ID_A]]$
3. A → B : ticket	$= E_{K_B}[K_S \parallel ID_A]$
4. B → A : challenge	$= E_{K_S}[N_2]$
5. A → B : response	$= E_{K_S}[f(N_2)]$

10

Key Distribution

- 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 *nounce* 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
 - B autentica A con msg 3
 - A autentica B con msg 4 e 5
- $f(x)$ è una qualsiasi funzione di x , e.g. $f(x)=x+1$

11

Key Distribution Issues

- hierarchies of KDC's required for large networks, but must trust each other
- use of decentralized key distribution
- session key lifetimes should be limited for greater security
- controlling purposes keys are used for

12

Kerberos



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)

14

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 client↔KDC communications use the short-term client key**

15

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

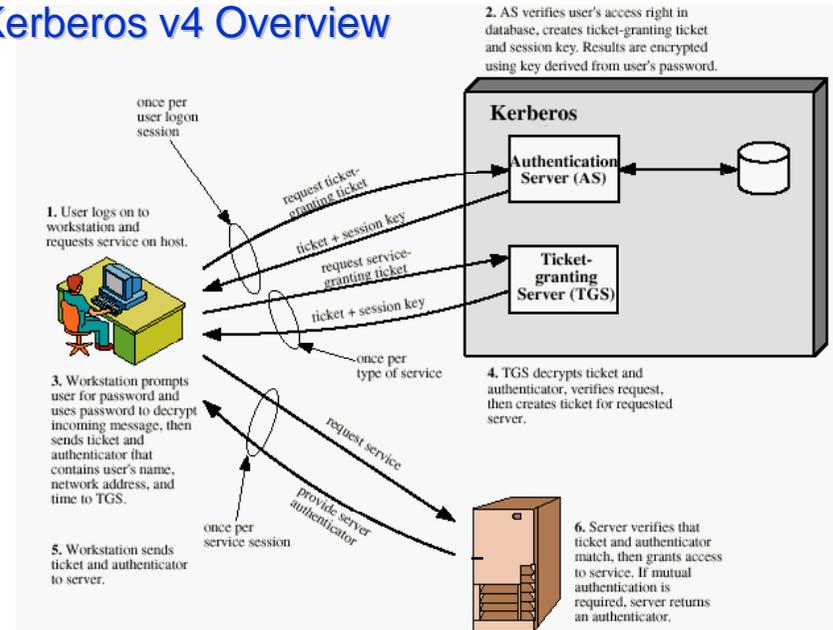
16

Kerberos KDC: AS and TGS

- Il KDC è un singolo processo che fornisce i due servizi:
 - Authentication Service (AS). **Emana "Ticket Granting Tickets" (TGTs) 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). **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.**

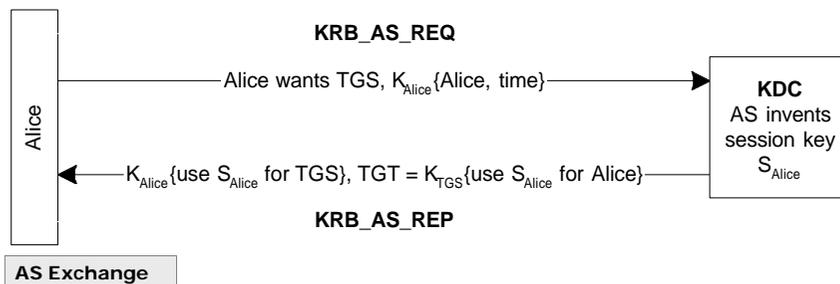
17

Kerberos v4 Overview



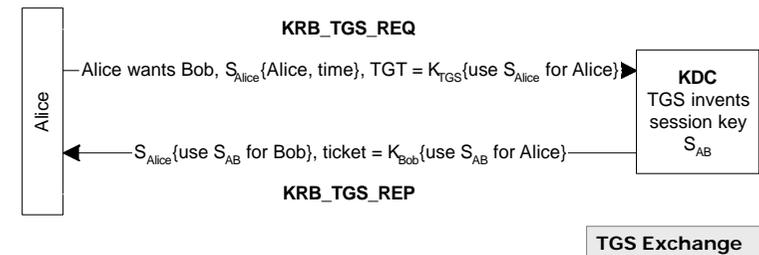
18

Authentication Service



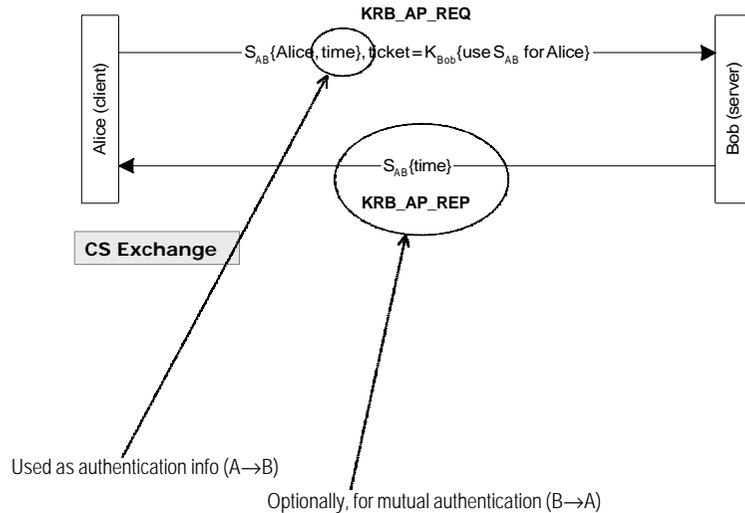
19

Ticket-Granting Service



20

Client/Server exchange



21

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

22

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

23