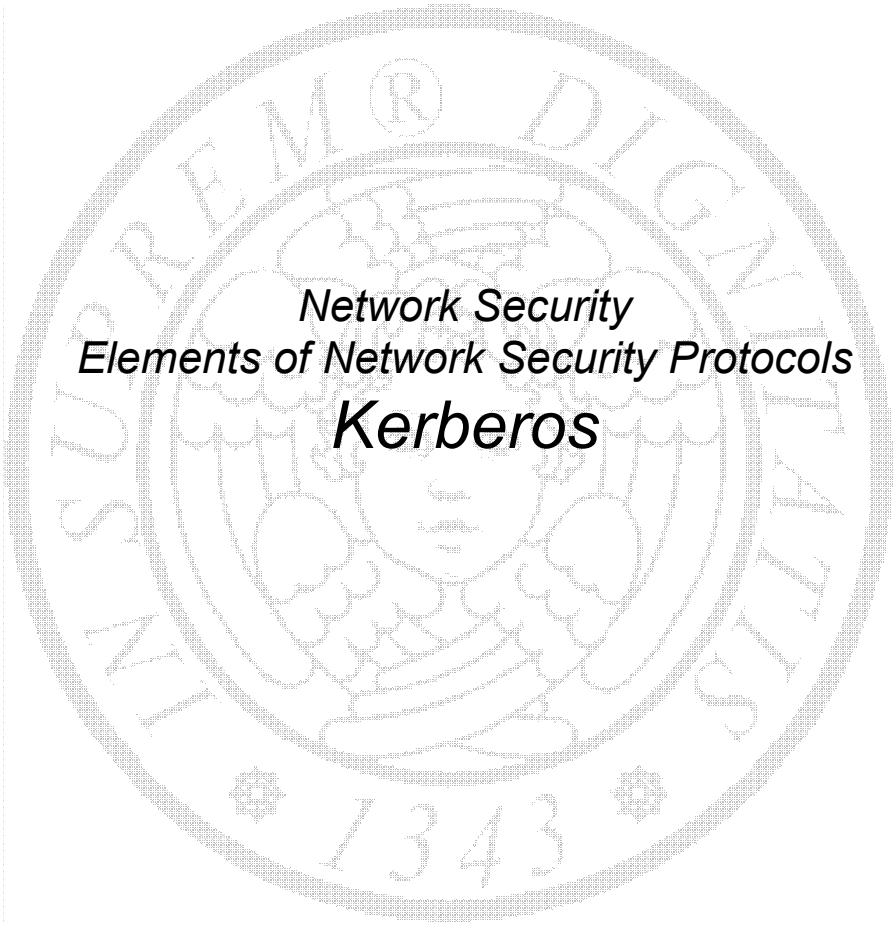


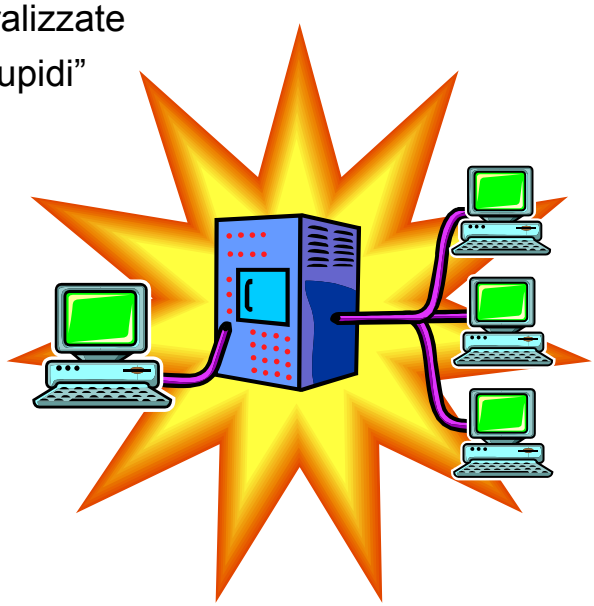
Network Security
Elements of Network Security Protocols
Kerberos

Roadmap

- Architettura e protocollo semplificati
- Architettura completa
  - pre-authentication
  - delegation
    - proxiable tickets
    - forwardable tickets
- Realms

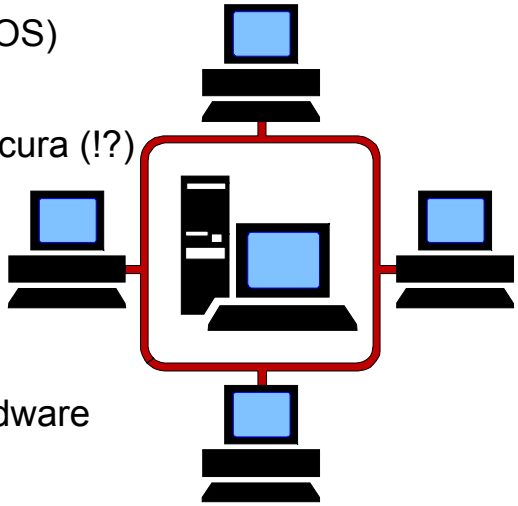
Sistema centralizzato chiuso

- Caratteristiche
  - Informazioni ed elaborazione centralizzate
  - Accesso per mezzo di terminali “stupidi”
  - Unico dominio amministrativo
  - Protezione fisica
  - Utenti noti e fidati (!?)
- Obiettivo
  - Garantire l’integrità del sistema
- Meccanismi
  - Sistema Operativo
  - Meccanismi HW



Sistema distribuito chiuso

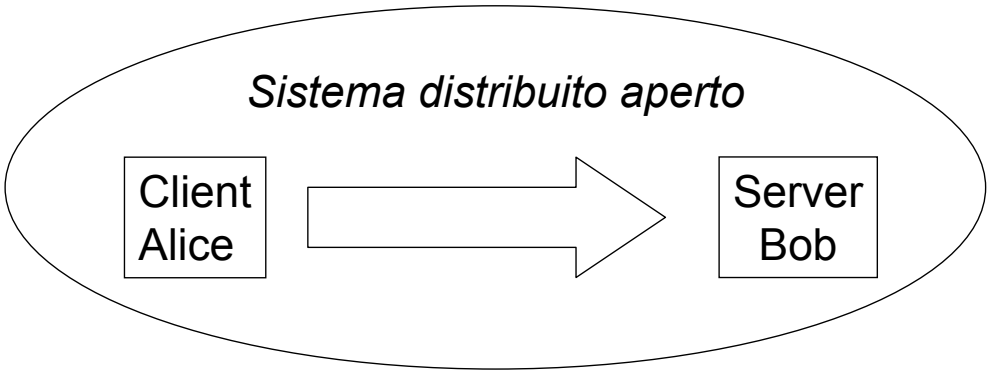
- Caratteristiche
  - Condivisione di risorse (file, stampanti, CPU)
  - Workstation (Windows, Unix, MS-DOS)
  - Unico dominio amministrativo
  - Utenti noti e fidati; la rete (LAN) è sicura (!?)
- Obiettivo
  - Garantire l’integrità del sistema
- Meccanismi
  - Sistema operativo, meccanismi hardware
  - Autenticazione basata sull’indirizzo (address-based authentication)



Sistema distribuito aperto

- Caratteristiche
  - Domini applicativi differenti
  - Utenti sconosciuti o inaffidabili
  - La rete è insicura
- Obiettivo
  - Confidenzialità ed Integrità
- Meccanismi
  - Sistema operativo, meccanismi hardware
  - Crittografia

Lo scenario



- Il server deve consentire solo gli accessi autorizzati
- Il server deve autenticare le richieste di servizio

Approcci all'identificazione dei clienti

A. Affidarsi alle workstation (client) per l'identificazione dei propri utenti ed affidarsi ai server per realizzare una politica di sicurezza basata sull'identità degli utenti (UID)

B. Affidarsi alle workstation per l'identificazione dei propri utenti ma richiedere ai server di autenticare le workstation

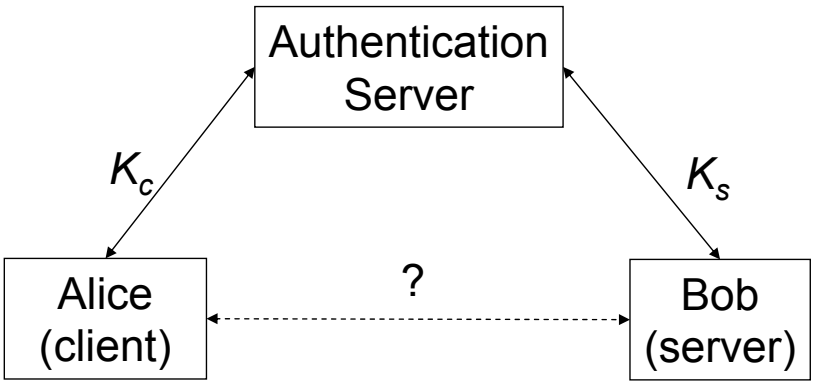
C. Richiedere che l'utente fornisca al server una prova di identità ogni volta che richiede un servizio; richiedere anche al server di fornire una prova della loro identità

sistema distribuito chiuso	A, B
sistema distribuito aperto	C

Kerberos: obiettivi

- **Sicurezza**. Un avversario in ascolto sulla rete non deve essere in grado di acquisire informazioni per impersonare un utente legittimo
- **Affidabilità**. L'indisponibilità di Kerberos implica l'indisponibilità di tutti i servizi, per cui Kerberos deve essere un sistema altamente affidabile
- **Trasparenza**. Il processo di autenticazione deve essere trasparente all'utente ad eccezione dell'inserimento della password
- **Scalabilità**. Kerberos deve essere in grado di gestire un elevato numero di clienti e di server

Kerberos: architettura di base



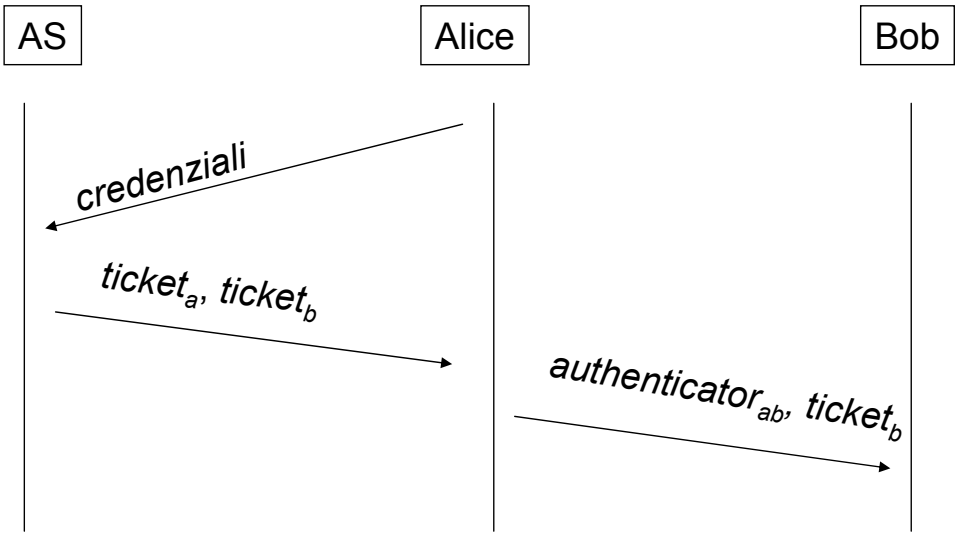
K_c e K_s (*master key*) sono segreti condivisi tra AS e client e server, rispettivamente (ad esempio derivati da password)

Obiettivo primario: autenticazione mutua di client e server

Obiettivi secondari:

- Stabilire una chiave condivisa tra client e server
- Provare al server che il client è attivo e viceversa
- …

Kerberos: l'idea



- $\boldsymbol{ticket_a} = E_{Ka}(A||K_{ab}||\ldots)$; $\boldsymbol{ticket_b} = E_{Kb}(A||K_{ab}||\ldots)$; $\boldsymbol{authenticator_{ab}} = E_{Kab}(t_a||\ldots)$
- il ***ticket*** contiene la chiave di sessione K_{ab} cifrata con la master key (K_a, K_b)
- L'***authenticator*** assicura a *B* la presenza di *A* e viceversa

Kerberos 5: authentication protocol (simpl.)

KRB_AS_REQ	$M1$	$A \rightarrow AS:$	A,B,t,L,N_a,WS
KRB_AS_RESP	$M2$	$AS \rightarrow A:$	$\{A,B,t,L,K_{ab},WS\}_{K_b}, \{B,t,L,N_a,K_{ab}\}_{K_a}$
KRB_AP_REQ	$M3$	$A \rightarrow B:$	$\{A,B,t,L,K_{ab},WS\}_{K_b}, \{A,t_a,\cdots\}_{K_{ab}}$
KRB_AP_RESP	$M4$	$B \rightarrow A:$	$\{t_a,\cdots\}_{K_{ab}}$

$authenticator_{ab}$

$ticket_b$ $ticket_a$

Kerberos 5: authentication protocol (simpl.)

KRB_AS_REQ	$M1$	$A \rightarrow AS:$	A,B,t,L,N_a,WS
KRB_AS_RESP	$M2$	$AS \rightarrow A:$	$\{A,B,t,L,K_{ab},WS\}_{K_b}, \{B,t,L,N_a,K_{ab}\}_{K_a}$
KRB_AP_REQ	$M3$	$A \rightarrow B:$	$\{A,B,t,L,K_{ab},WS\}_{K_b}, \{A,t_a,subkey_a\}_{K_{ab}}$
KRB_AP_RESP	$M4$	$B \rightarrow A:$	$\{t_a,subkey_b\}_{K_{ab}}$

- ***L*** intervallo di validità del ticket, permette ad Alice di riutilizzare lo stesso ticket per eseguire più autenticazioni con Bob senza dover interagire con AS e quindi evitare i messaggi M1 ed M2
- Marca temporale t_a generata da *Alice*. *Bob* verifica che sia "recente". Per ogni autenticazione, Alice genera un nuovo autenticatore usando la stessa K_{ab} ma con un diverso t_a
- Identificatore della workstation del client ***WS*** permette al server di controllare quali computer possono usare i ticket.
- Le chiavi $subkey_a$ e $subkey_b$ possono essere utilizzate per l'espletamento del servizio.

Kerberos 5 (protocollo ideale)

Ipotesi

$$A \mid\equiv A \overset{K_a}{\leftrightarrow} AS \qquad B \mid\equiv B \overset{K_b}{\leftrightarrow} AS$$

$$AS \mid\equiv A \overset{K_a}{\leftrightarrow} AS \qquad AS \mid\equiv B \overset{K_b}{\leftrightarrow} AS$$

$$AS \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

$$A \mid\equiv \left(AS \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B \right) \qquad B \mid\equiv \left(AS \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B \right)$$

$$A \mid\equiv \#(t)$$
$$B \mid\equiv \#(t) \qquad B \mid\equiv \#(t_a)$$

Protocollo

$$M2 \quad AS \rightarrow A: \quad \left\{ t, N_a, \left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right\}_{K_a}, \left\{ t, \left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right\}_{K_b}$$

$$M3 \quad A \rightarrow B: \quad \left\{ t, \left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right\}_{K_b}, \left\{ t_a, \left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right\}_{K_{ab}} \text{ from } A$$

$$M4 \quad B \rightarrow A: \quad \left\{ t_a, \left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right\}_{K_{ab}} \text{ from } B$$

Analisi del protocollo

dopo il messaggio M2

$$A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

dopo il messaggio M3

$$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

key authentication

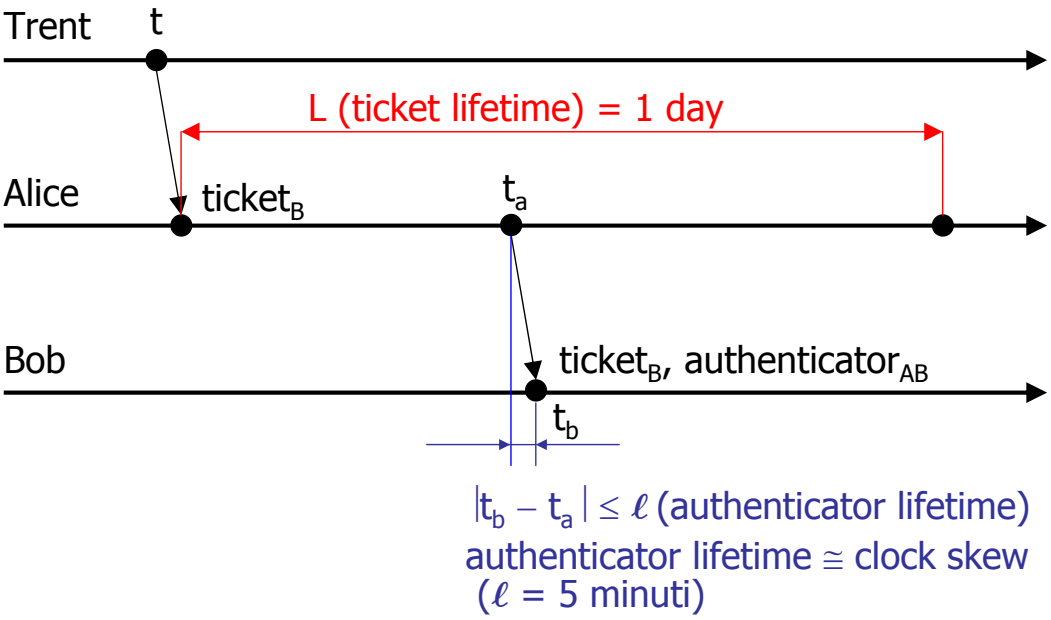
$$B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

dopo il messaggio M4

$$A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

key confirmation

Kerberos: ticket e authenticator lifetime



Kerberos: considerazioni

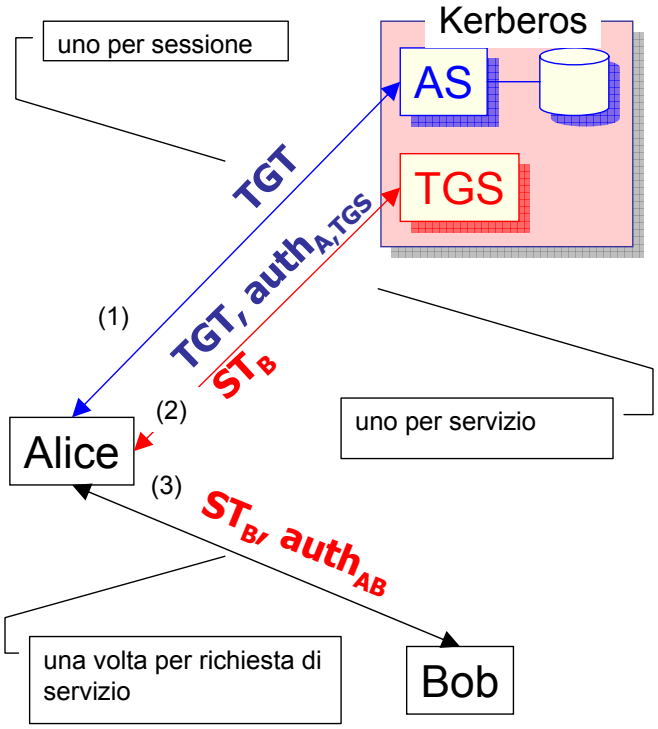
- Kerberos utilizza i timestamp e quindi richiede che i clock siano sincronizzati
  - Se i clock non sono sincronizzati ed i timestamp non sono cached allora è possibile un replay attack
  - In Kerberos 5, ℓ = 5 minuti: perció un authenticator può essere replicato in questa finestra
- Se le chiavi K_a e K_b derivano da password, allora lo schema è tanto sicuro quanto la loro segretezza o la loro resistenza ad un password-guessing attack

Kerberos: architettura completa

Un utente utilizza molti servizi per cui

- la password/master key viene inserita dall'utente per ogni nuova autenticazione e rimossa subito dopo (poco usabile)
- la password/master key viene memorizzata sulla workstation dell'utente per un lungo periodo di tempo (poco sicuro)

Kerberos: architettura completa



"Every problem in computer science can be solved by adding a level of indirection"
(D. Wheelar, Cambridge)

1. Authentication server rilascia il ticket granting ticket (TGT) per interagire con il Ticket Granting Server (TGS)
2. Il TGS rilascia ad Alice il service ticket (ST_B) per interagire con il server Bob

Ticket Granting Service

Fase 1

Alice interagisce con AS e riceve TGT, il ticket granting ticket, un ticket per il server TGS

Fase 2

Alice interagisce con TGS e riceve ST_b, service ticket, un ticket per il server Bob

Fase 3

Alice usa ST_b per autenticare ed autenticarsi con Bob

Interazione con TGS

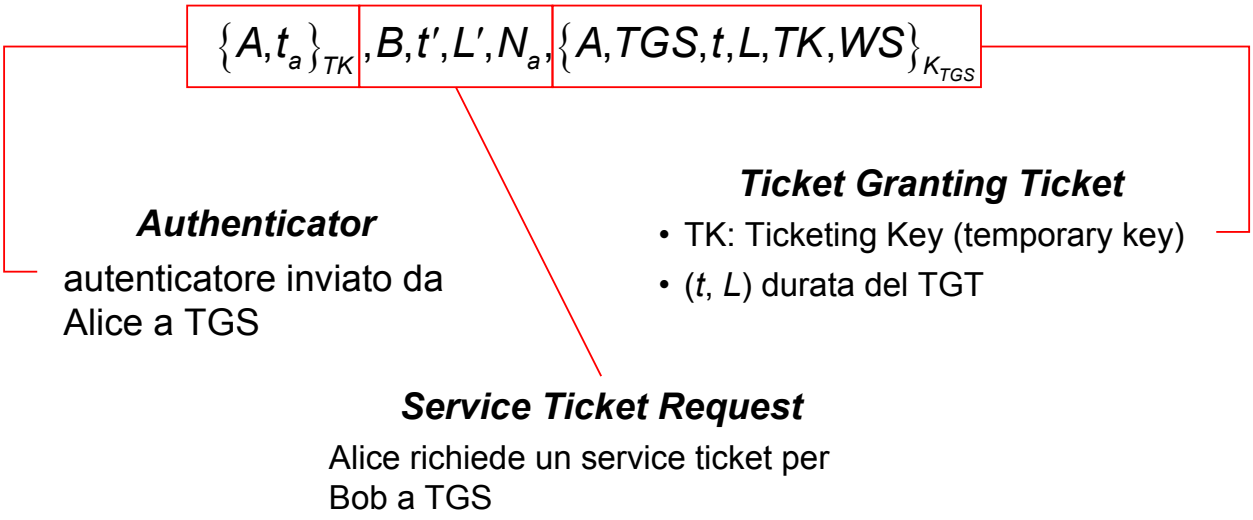
Fase 2: Messaggio KRB_TGS_REQ

$$\{A,t_a\}_{TK},B,t',L',N_a,\{A,TGS,t,L,TK,WS\}_{K_{TGS}}$$

Fase 2: Messaggio KRB_TGS_RESP

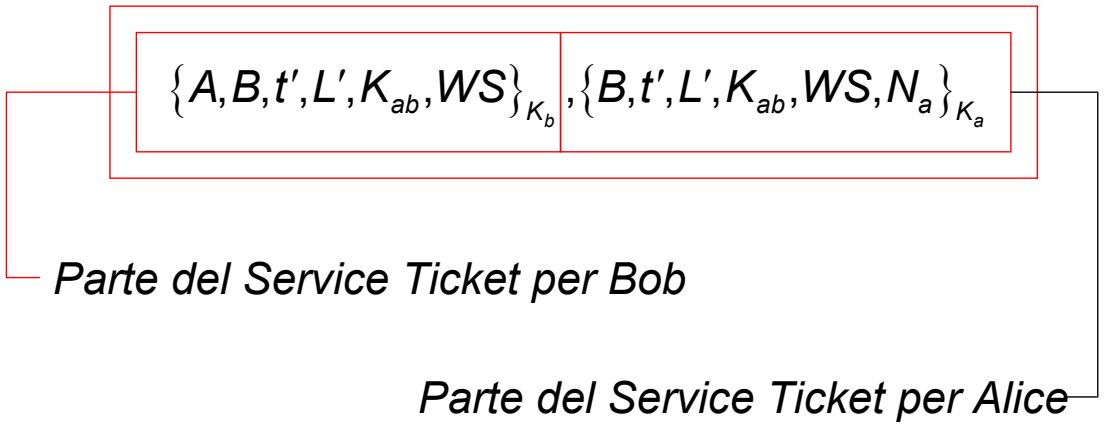
$$\{A,B,t',L',K_{ab},WS\}_{K_b},\{B,t',L',K_{ab},WS,N_a\}_{K_a}$$

Messaggio KRB_TGS_REQ



Messaggio KRB_TGS_RESP

Service Ticket rilasciato ad Alice per interagire con Bob

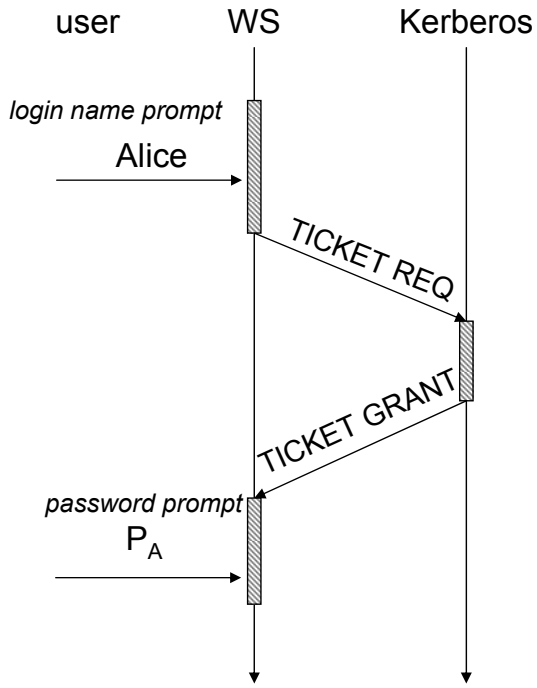


User and Workstation Authentication

Kerberos autentica gli *utenti* rispetto ai servizi di rete ma

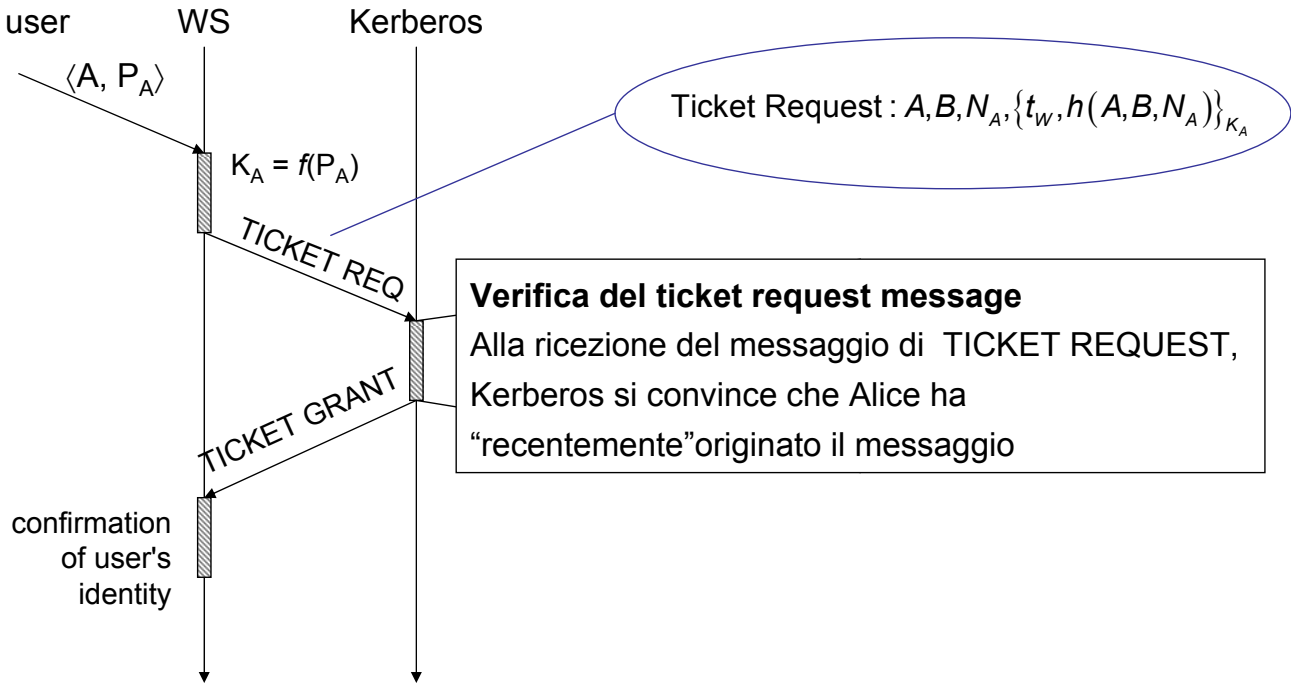
- non autentica gli utenti rispetto al KDC
  - chiunque può contattare KDC e richiedere ed ottenere un ticket per Alice;
  - Kerberos garantisce che nessuno eccetto Alice possa utilizzare tale ticket;
  - Un avversario può sfruttare questa possibilità per lanciare un password attack
- non autentica gli utenti rispetto alle workstation (*indirect authentication*)
  - Una WS è un mezzo attraverso il quale gli utenti accedono ai servizi
  - Le WS sono uniform,, interchangeable, thin client;

Traditional Kerberos authentication

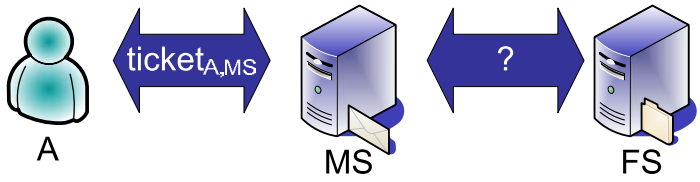


- Un avversario può collezionare ticket e poi utilizzarli per montare un password attack
- La WS non autentica Alice

Kerberos pre-authentication



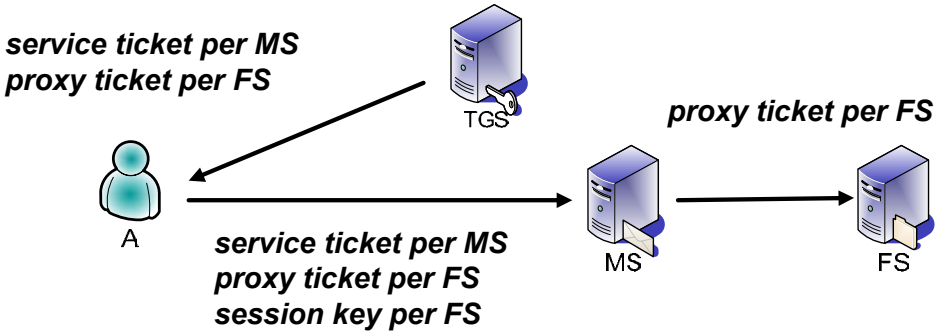
Delegation



- Il Mail Server MS deve interagire con il File System FS *per conto* dell'utente secondo il *principio del minimo privilegio*
- Kerberos fornisce due meccanismi che permettono ad Alice di *delegare* MS
  - proxy tickets
  - forwardable TGT

Proxy ticket

Permette di richiedere un service ticket legato ad un indirizzo diverso da quello del richiedente



Proxy Ticket: $\{K_{A,FS}, N_A, t, L, FS\}_{K_a}, \{K_{A,FS}, MS, t, L\}_{K_{fs}}$

Proxy ticket per FS richiesto da A ma rilasciato a MS

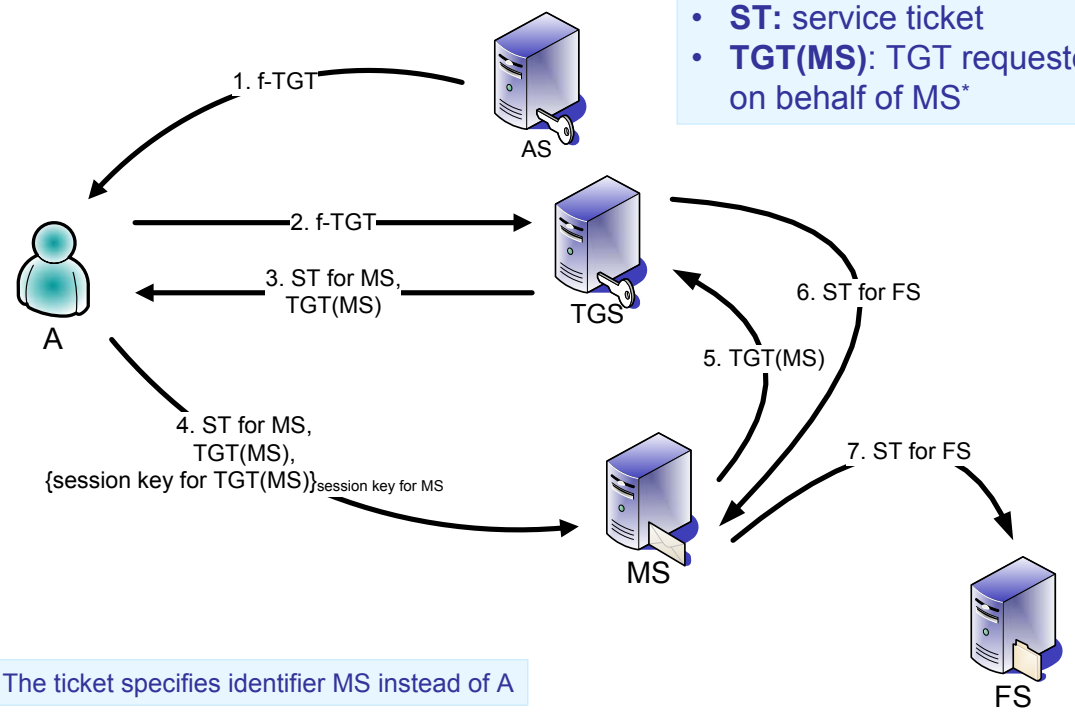
$\{\text{session key for FS}\}_{\text{session key for MS}} \equiv \{\cdots, K_{A,FS}, \cdots\}_{K_{a,ms}}$

Proxy ticket

- Questa soluzione richiede che il software di Alice conosca in anticipo tutti i proxy ticket di cui ha bisogno oppure che sia capace di negoziarli con il server MS volta per volta
- I forwardable ticket permettono di risolvere questo problema permettendo al server delegato MS di chiedere i ticket di cui ha bisogno

Forwardable ticket

- **f-TGT: *forwardable*** TGT
- **ST:** service ticket
- **TGT(MS)**: TGT requested by A on behalf of MS*



* The ticket specifies identifier MS instead of A

Forwardable ticket

- **[…]**
- **Step 3**. Il TGS ritorna ad Alice
  1. un *service ticket* per MS
     {…, $K_{a,ms}$, …}$_{Ka}$, {…, $K_{a,ms}$,…}$_{Kms}$
  2. un *TGT* contenente la *ticketing key* associata con MS invece che Alice
     {…, TK_a,…}$_{Ka}$, {…, TK_a, MS, …}$_{Ktgs}$.
- **Step 4.** Alice inoltra i due ticket al mail server insieme alla ticketing key TK_a cifrata con $K_{a,ms}$
- **[…]**

La ticketing key è associata ad MS invece che ad A

Proxy vs forwardable ticket

Proxy ticket

- (PRO) Il client controlla quali permessi delegare al server.
- (CON) Il client deve conoscere di quali ticket ci sarà bisogno.

Forwardable ticket

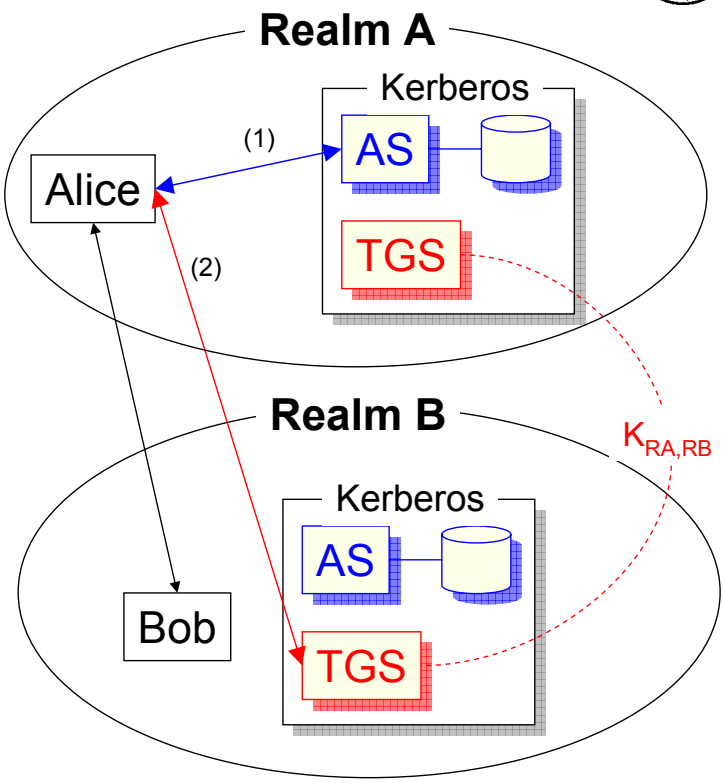
- (PRO) Il server (implementazione) determina di quali ticket ha bisogno.
- (CON) Un server compromesso può abusare di tutti i ticket che vuole.

Limitazioni alla delega

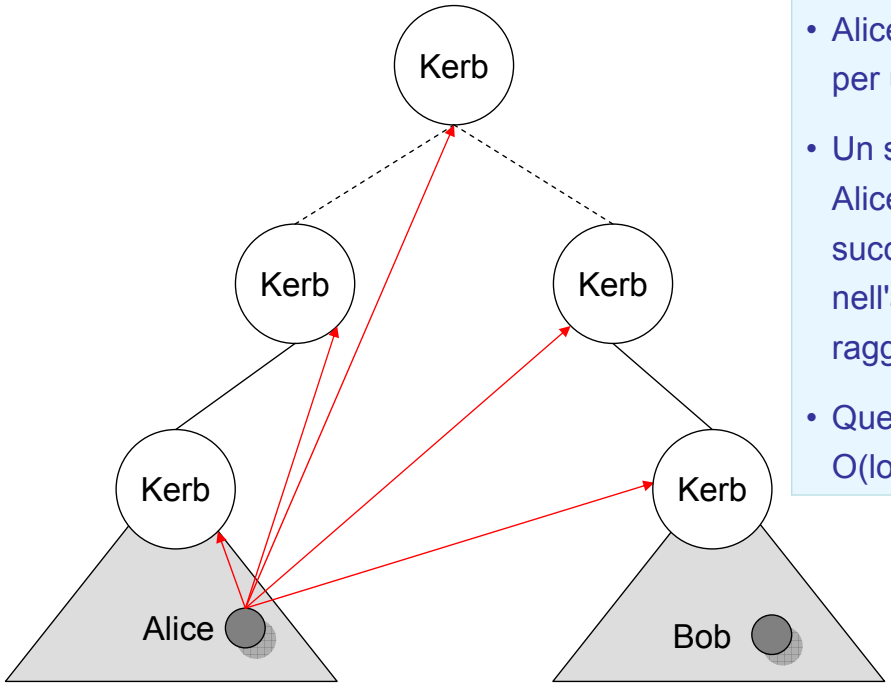
- Un ticket ha un tempo di vita massimo (lifetime)
- Un ticket specifica un insieme massimo di permessi (capability)

Realms e referral tickets

- Server e client possono appartenere a domini amministrativi (*realm*) differenti
- Kerberos autentica i client ai server del proprio realm
- Alice vuole autenticarsi a Bob di un altro realm
  - Alice chiede a Kerberos A un ***referral TGT*** per il realm B (1).
  - Kerberos A crea un referral TGT usando una ***inter-realm key*** ($K_{RA,RB}$)
  - Alice usa il referral TGT per richiedere a Kerberos B un service ticket per Bob (2).



Organizzazione gerarchica dei realms



- Alice richiede un referral ticket per un foreign server Bob.
- Un server Kerberos dà ad Alice un ticket per il nodo successivo (su o giù nell'albero) finché Alice non raggiunge il realm di Bob
- Questo processo richiede O(log n) operazioni

Attacking a Kerberos Network

- Intrusion tolerance
  - Kerberos ha un approccio pratico: ammette le intrusioni ma cerca di limitarne gli effetti
    - workstation: i danni sono limitati alla sola workstation (ed i suoi utenti)
    - server: i danni si estendono a tutti i clienti del server; è bene distribuire i servizi su più server
    - KDC: il sistema è completamente compromesso
- Clock synchronization
  - è un requisito cruciale

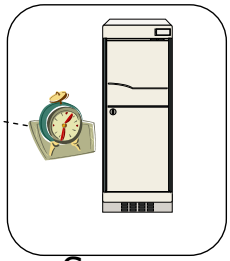
Kerberos e clock synchronization

L'avversario ha ottenuto una "vecchia" chiave di sessione *K* ed il relativo ticket *t*

Time server

NTP

possibili obiettivi



Server

se l'avversario riesce a "rimettere indietro" (alterare) il clock del server allora può riutilizzare la chiave *K*

Kerberos e certificati

L'utilizzo dei certificati elimina il bisogno di segreti condivisi (es., K_a) basati su password riutilizzabili

Procedura PKINIT

$$M1. \quad A \rightarrow T \quad S_A(A, B, N_A), certificate_A$$
$$M2. \quad T \rightarrow A \quad ticket_B, E_{e_A}(S_T(K, N_A, L, B))$$

Alice è in possesso di $certificate_T$

Windows 2000 incorpora PKINIT nel suo ambiente di autenticazione basato su Kerberos

Kerberos

- Kerberos si basa sul protocollo di Needham-Schroeder (1978)
- Kerberos è stato sviluppato al MIT nel 1980
- Kerberos V4 e Kerberos V5 (RFC 1510)
- Kerberos fa parte di OSF DCE e di Windows 2000 (e successivi)
- In Windows 2000, Kerberos ha sostituito Windows NT domain authentication mechanism