

یا ذالامن والامان

# توزیع و مدیریت کلید Kerberos & X.509

## مبتنی بر فصل ۴

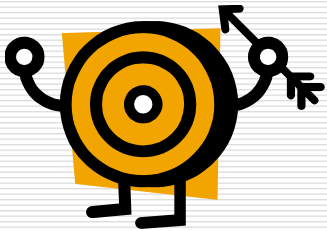
ویرایش شده توسط: حمید رضا شهریاری

<http://ceit.aut.ac.ir/~shahriari>

# اهداف

---

□ آشنایی با چگونگی احراز هویت در الگوریتم کربروس



---

برای دیدن معادل انگلیسی ترجمه‌ها به اسلاید واژه نامه مراجعه نمایید.



# افسانه یونانی

□ سگ سه سر افسانه یونانی : محافظان دروازه های جهنم!  
□ سرها نماد:

- Authentication
- Authorization
- Accounting

□ اگرچه در عمل تنها احراز هویت اعمال شد.



# انگیزه

- محیطهای جدید: به صورت توزیع شده
- در یک محیط توزیع شده سه روش برای امنیت:
  - اعتماد به ایستگاه کاری در معرفی کردن کاربران خود و اعمال سیاست امنیتی مبتنی بر شناسه کاربران
  - نیاز به احراز هویت سیستمهای client توسط کارگزار
  - ولی اعتماد به سیستمهای client نسبت به احراز هویت کاربران خود
  - نیاز به احراز هویت هر یک از کاربران نسبت به سرویس درخواستی و بالعکس



# کربروس

- احراز هویت بر اساس رمز نگاری کلید مخفی (مقارن)
- طراحی شده در MIT
- به جای احراز هویت در هر کارگزار به صورت توزیع شده، یک کارگزار خاص را به احراز هویت اختصاص می‌دهیم
- نسخه ۵ آن در حال استفاده است.
- طی RFC 1510 معرفی شد که بعداً با RFC 4120 در سال ۲۰۰۵ جایگزین شد.

# نیازمندیها/ویژگیهای عمومی کربروس

- عمومی بودن (Common)
  - در محیط توزیع شده همراه با سرورهای متمرکز و غیر متمرکز
- امنیت (Security)
  - ادعای اصلی
- اطمینان (Reliability)
  - اطمینان از دسترس پذیری کارگزار احراز هویت (کربروس)
- شفافیت (Transparency)
  - کاربران باید سیستم را همانند یک سیستم ساده «شناسه و گذرواژه» ببینند.
- مقیاس پذیری (Scalability)
  - قابلیت کار با تعداد زیادی ماشین کاربر و کارگزار

# ویژگیهای عمومی کربروس

---

چند تعریف



■ دامنه: یک محدوده دسترسی را مشخص می کند. به نوعی معادل دامنه های تعریف شده در ویندوز می باشد.

■ مرکز توزیع کلید: معادل کارگزار کربروس می باشد.

■ **Principal**: به سرویس ها، دستگاه ها، کاربران و کلیه عناصری که احتیاج به شناساندن خود به کارگزار کربروس دارند، گفته می شود.



# کربروس

---

□ برای معرفی کربروس به صورت گام به گام از پروتکلهای ساده شروع می کنیم و سعی می کنیم اشکالات هر یک را برطرف کنیم تا به کربروس برسیم.

## • دیالوگ ساده احراز هویت -

فرص: بین AS و هر کارگزار یک کلید مشترک وجود دارد.  
درخواست خدمات توسط کارفرما از کارگزار:

1. **Client → AS:**  $ID_{client} || Pass_{client} || ID_{Server}$
2. **AS → Client:** *Ticket*
3. **Client → Server:**  $ID_{client} || Ticket$

**AS : Authentication Server** کارگزار احراز هویت

**$E_{K_{server}}$ :** Shared key between AS and Server

**Ticket =  $E_{K_{server}} [ID_{client} || Addr_{client} || ID_{server}]$**

# بلیت

---

در واقع نوعی گواهی است که هنگام ورود کاربر به قلمرو کربروس به او داده می‌شود که بیانگر اعتبار او برای دسترسی به منابع شبکه می‌باشد.

# بررسی دیالوگ

□ چرا آدرس کارفرما (Client) در بلیت ذکر میشود؟

■ در غیر این صورت هر شخصی که بلیت را از طریق شنود به دست آورد نیز میتواند از امکانات استفاده کند. اما اکنون تنها خدمات به آدرس ذکر شده در بلیت ارائه میشود.

□ مشکل جعل آدرس



□ چرا شناسه کارفرما ID<sub>client</sub> در گام سوم به صورت رمز نشده ارسال میشود؟

■ زیرا این اطلاعات به صورت رمزنگاری شده در بلیت وجود دارد.

■ اگر شناسه با بلیت مطابقت نداشته باشد خدمات ارائه نمی‌شوند.

## مشکلات دیالوگ ساده احراز هویت -

---

□ ناامنی

■ ارسال کلمه عبور بدون رمزگذاری (به شکل متن واضح)

■ امکان حمله تکرار

□ ناکارایی

■ لزوم تقاضای بلیت جدید برای هر خدمات

# استفاده مجدد از بلیت‌ها

---

□ چرا استفاده مجدد از بلیت‌ها (Tickets) اهمیت دارد؟

■ جلوگیری از تایپ مجدد گذرواژه در یک بازه زمانی کوتاه

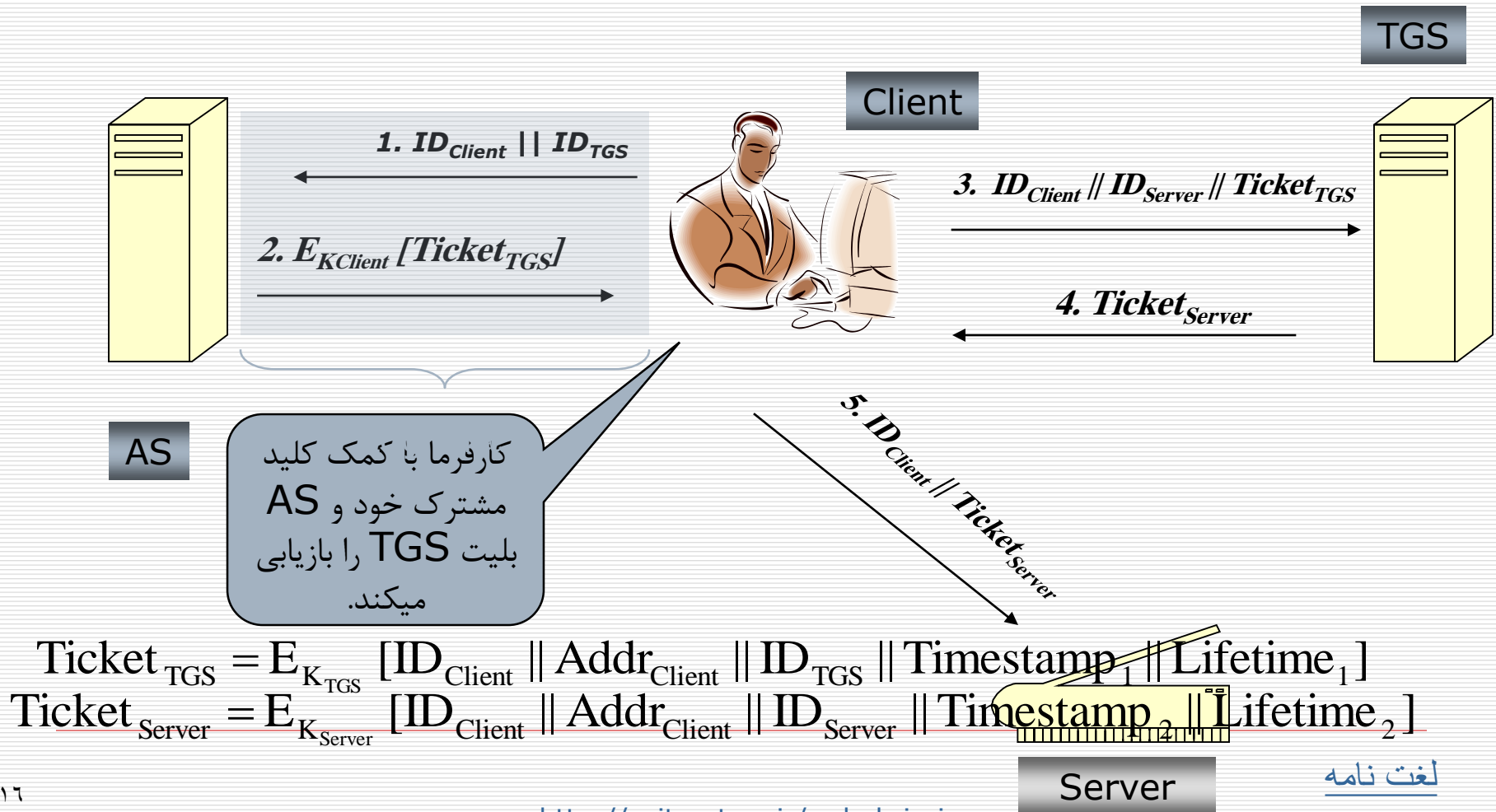
■ شفافیت احراز هویت

□ کاربر متوجه فرآیندهای احراز هویت نمی‌شود.

# افزایش ایمنی-دیالوگ ۱

- استفاده از یک کارگزار جدید با نام کارگزار اعطا کننده بلیت  
■ TGS: Ticket Granting Server
- کارگزار احراز هویت، AS، کماکان وجود دارد.  
■ بلیت «اعطای بلیت» ticket-granting ticket توسط آن صادر می شود.
- اگرچه بلیت های اعطای خدمات توسط TGS صادر میشوند.  
■ بلیت «اعطای خدمات» service-granting ticket
- اجتناب از انتقال **گذرواژه** با رمز کردن پیام کارگزار احراز هویت (AS) به کارفرما توسط کلید مشتق شده از گذرواژه

# افزایش ایمنی - دیالوگ ۱





# افزایش ایمنی - دیالوگ ۱

- پیامهای شماره یک و دو به ازاء هر جلسه Log on رد و بدل میشوند.
- پیامهای شماره سه و چهار به ازاء هر نوع خدمات رد و بدل میشوند.
- پیام شماره پنج به ازاء هر جلسه خدمات رد و بدل میشود.

1.  $Client \rightarrow AS: ID_{Client} || ID_{TGS}$
2.  $AS \rightarrow Client: E_{K_{Client}} [Ticket_{TGS}]$
3.  $Client \rightarrow TGS: ID_{Client} || ID_{Server} || Ticket_{TGS}$
4.  $TGS \rightarrow Client: Ticket_{Server}$
5.  $Client \rightarrow Server: ID_{Client} || Ticket_{Server}$

# محتوی بلیت‌ها

---

بلیت اعطای بلیت :

$$\text{Ticket}_{\text{TGS}} = E_{K_{\text{TGS}}} [\text{ID}_{\text{Client}} \parallel \text{Addr}_{\text{Client}} \parallel \text{ID}_{\text{TGS}} \parallel \text{Timestamp}_1 \parallel \text{Lifetime}_1]$$

بلیت اعطای خدمات :

$$\text{Ticket}_{\text{Server}} = E_{K_{\text{Server}}} [\text{ID}_{\text{Client}} \parallel \text{Addr}_{\text{Client}} \parallel \text{ID}_{\text{Server}} \parallel \text{Timestamp}_2 \parallel \text{Lifetime}_2]$$

# ویژگی های دیالوگ ۱

- دو بلیت صادر شده ساختار مشابهی دارند. در اساس به دنبال هدف واحدی هستند.
- رمزنگاری  $Ticket_{TGS}$  جهت احراز هویت
- تنها کارفرما می تواند به بلیت رمز شده دسترسی پیدا کند.
- رمز نمودن محتوای بلیتها صحت (Integrity) را فراهم میکند.
- استفاده از مهر زمانی (Timestamp) در بلیتها آنها را برای یک بازه زمانی تعریف شده قابل استفاده مجدد می کند.
- هنوز از آدرس شبکه برای احراز هویت بهره می گیرد.
- چندان جالب نیست زیرا آدرس شبکه جعل (Spoof) می شود.
- با این حال، درجه ای از امنیت مهیا می شود

# نقاط ضعف دیالوگ ۱

---

❑ مشکل زمان اعتبار بلیتها:

■ زمان کوتاه : نیاز به درخواست های زیاد گذرواژه

■ زمان بلند : خطر حمله تکرار

❑ احراز هویت یک سویه : عدم احراز هویت کارگزار توسط کارفرما

■ رسیدن درخواست ها به یک کارگزار غیرمجاز

## کربروس نسخه ۴

---

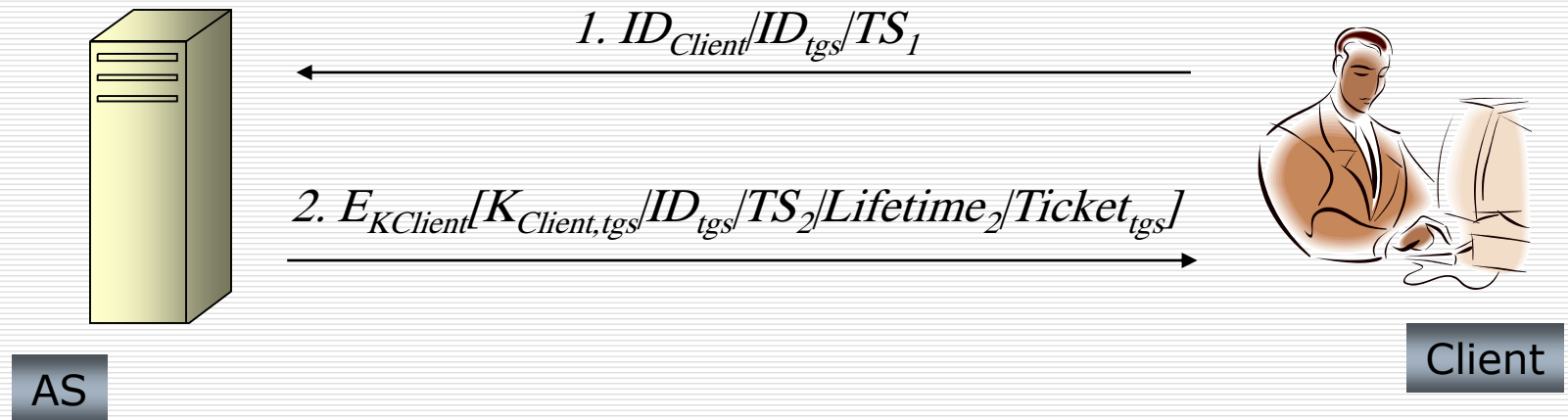
- توسعه یافته پروتکل های قبلی است.
- مشکل حمله تکرار حل شده است.
- احراز هویت دو جانبه (mutual) برقرار میشود.
- کارگزاران و کارفرمایان هر دو از هویت طرف مقابل اطمینان حاصل میکنند

# مقابله با حمله تکرار

---

- **یک نیاز جدید:** کارگزار یا TGS باید اطمینان حاصل نمایند که کاربر بلیت همان کسی است که بلیت برای او صادر شده.
- مفهوم جدیدی به نام اعتبار نامه (Authenticator) ابداع شده است:
  - علاوه بر بلیت‌ها از مفهوم کلید جلسه بهره می‌جوید

# کربروس نسخه ۴ : بررسی الگوریتم-۱

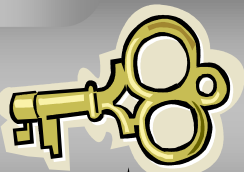


$$Ticket_{tgs} = E_{K_{tgs}}[K_{Client,tgs}/ID_{Client}/Addr_{Client}/ID_{tgs}/TS_2/Lifetime_2]$$

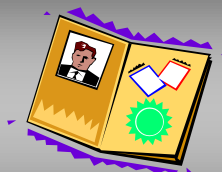
# بلیت TGS

تمامی با کلید  
رمز TGS  
شده اند

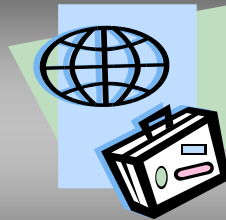
$$Ticket_{tgs} = E_{K_{tgs}}[K_{Client,tgs}/ID_{Client}/Addr_{Client}/ID_{tgs}/TS_2/Lifetime_2]$$



کلید جلسه  
بین  
کارفرما و  
TGS



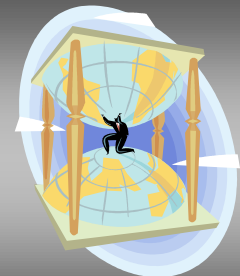
شناسه  
کارفرما



آدرس  
کارفرما



شناسه  
TGS



مهر زمانی  
و  
دوره اعتبار  
بلیت

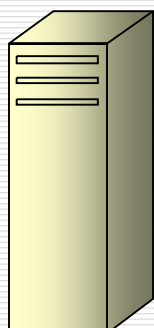


## نتایج این مرحله برای کارفرما

---

- ❑ بدست آوردن امن بلیت “اعطای بلیت” از  $AS$
- ❑ بدست آوردن زمان انقضای بلیت ( $TS_2$ )
- ❑ بدست آوردن **کلید جلسه** امن بین کارفرما و  $TGS$

# بدست آوردن بلیت «اعطای خدمات»



TGS

۳.  $ID_{server}/Ticket_{tgs}/Authenticator_{Client}$

۴.  $E_{K_{Client,tgs}}[K_{Client,server}/ID_{server}/TS_4/Ticket_{server}]$



Client

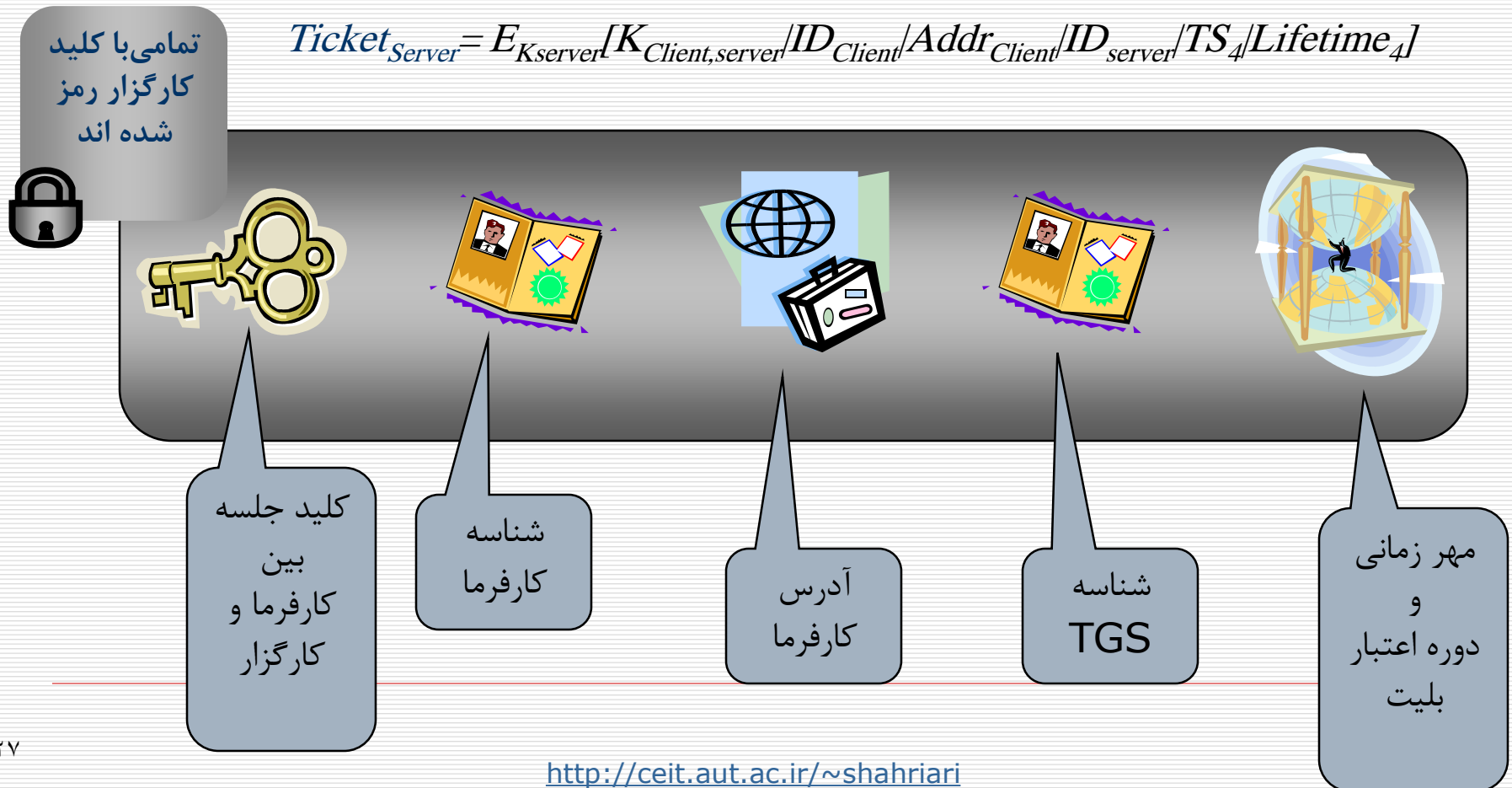
$Ticket_{Server} =$

$E_{K_{server}}[K_{Client,server}/ID_{Client}/Addr_{Client}/ID_{server}/TS_4/Lifetime_4]$

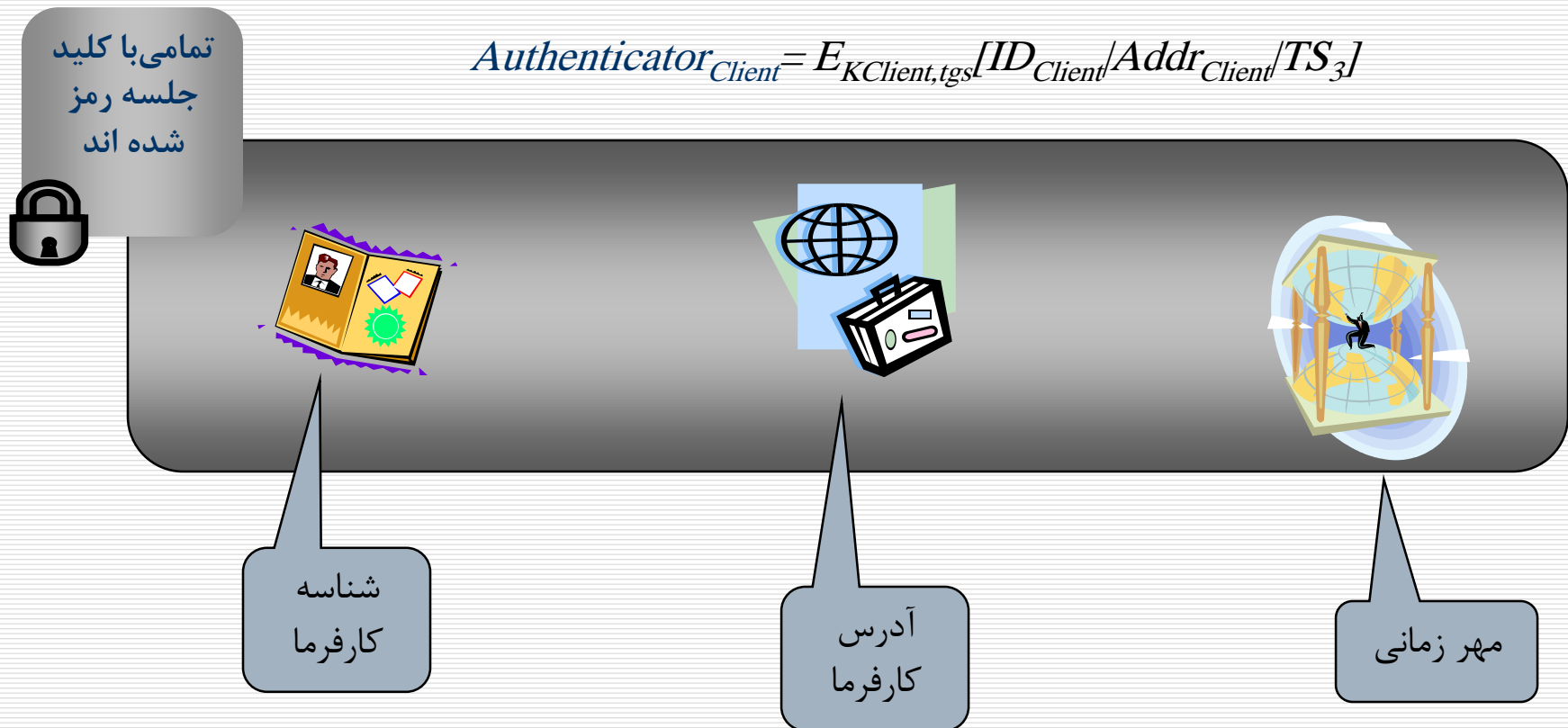
$Authenticator_{Client} =$

$E_{K_{Client,tgs}}[ID_{Client}/Addr_{Client}/TS_3]$

# بلیت کارگزار



# اعتبار نامه کارفرما

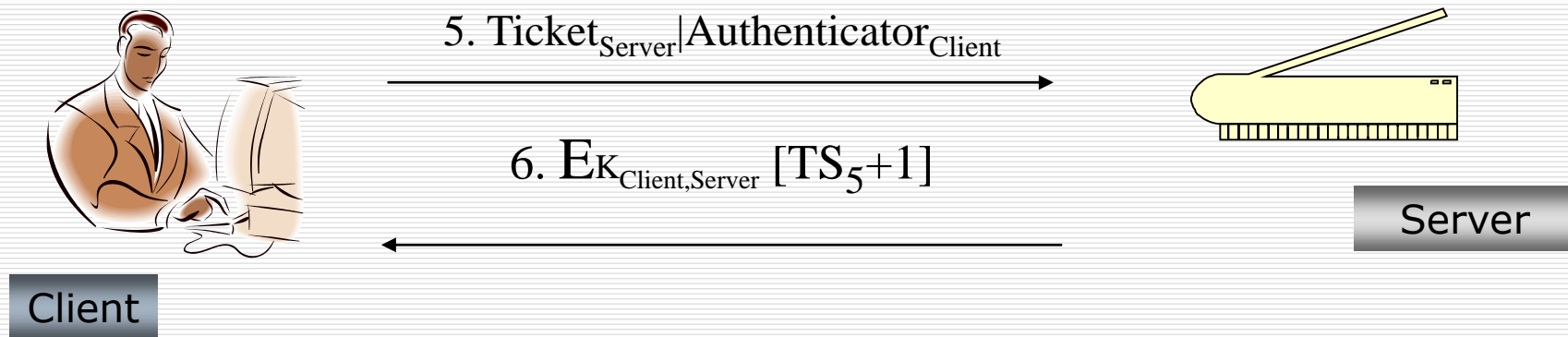


## نتایج این مرحله برای کارفرما

---

- ❑ جلوگیری از حمله تکرار با استفاده از یک اعتبار نامه (Authenticator) یکبار مصرف که عمر کوتاهی دارد.
- ❑ بدست آوردن کلید جلسه برای ارتباط با سرور V

# دستیابی به خدمات سرور



# نتایج این مرحله برای کارفرما

---

□ احراز هویت کارگزار در گام ششم با برگرداندن پیغام رمزشده

□ جلوگیری از بروز حمله تکرار

## کربروس نسخه ۴ : شمای کلی

### (a) Authentication Service Exchange: to obtain ticket-granting ticket

(1)  $C \rightarrow AS: ID_C \parallel ID_{tgs} \parallel TS_1$

(2)  $AS \rightarrow C: E_{K_c}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$

$$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$$

### (b) Ticket-Granting Service Exchange: to obtain service-granting ticket

(3)  $C \rightarrow TGS: ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4)  $TGS \rightarrow C: E_{K_{c,tgs}}[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$

$$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$$

$$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$$

$$Authenticator_c = E_{K_{tgs}}[ID_C \parallel AD_C \parallel TS_3]$$

### (c) Client/Server Authentication Exchange: to obtain service

(5)  $C \rightarrow V: Ticket_v \parallel Authenticator_c$

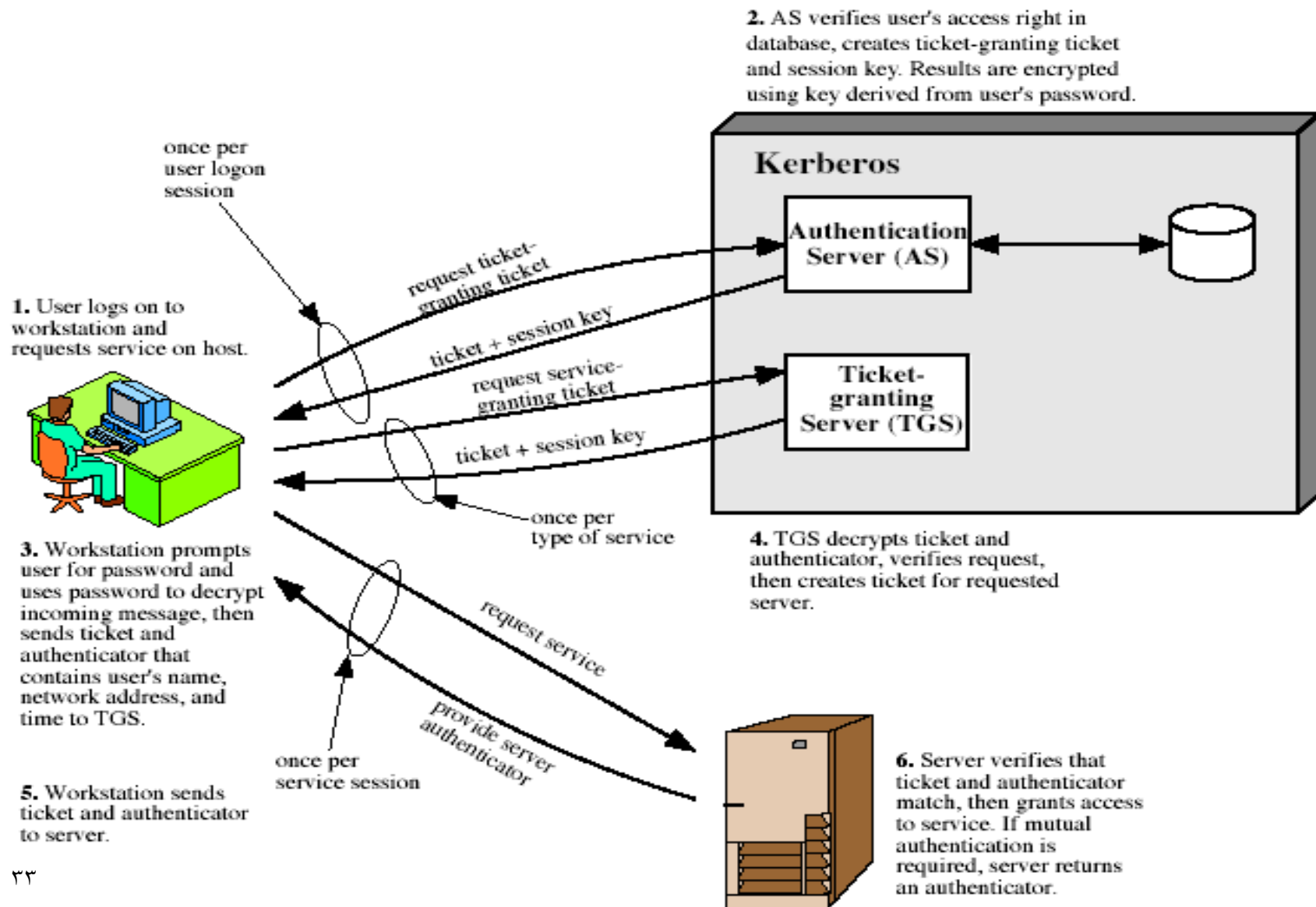
(6)  $V \rightarrow C: E_{K_{c,v}}[TS_5 + 1]$  (for mutual authentication)

$$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$$

$$Authenticator_c = E_{K_{c,v}}[ID_C \parallel AD_C \parallel TS_5]$$



# کربروس نسخه ۴ : شمای کلی



# قلمرو کربروس (realm)

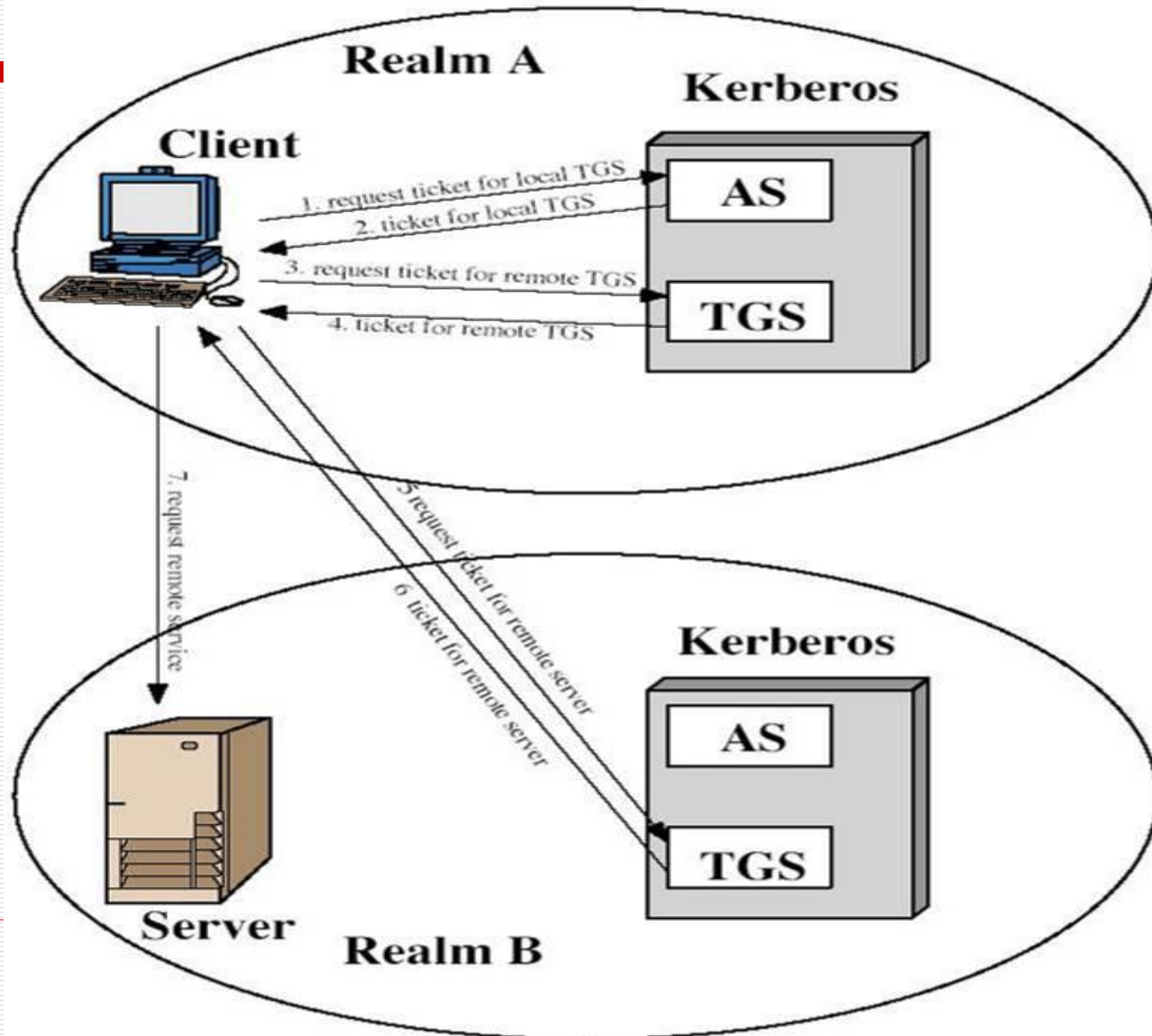
---

- قلمرو کربروس از بخشهای زیر تشکیل شده است:
  - کارگزار کربروس
  - کارفرمایان
  - کارگزاران کاربردی Application Servers
- کارگزار کربروس گذرواژه تمام کاربران را در پایگاه داده خود دارد.
- کارگزار کربروس با هر کارگزار کاربردی کلیدی مخفی به اشتراک گذاشته است.
- معمولاً هر قلمرو معادل یک حوزه مدیریتی است.

# احراز هویت بین قلمرویی (InterRealm)

- امکان این که کاربران بتوانند از خدمات موجود در قلمروهای دیگر استفاده کنند.
- کارگزاران کربروس هر قلمرو، یک کلید مخفی با کارگزاران کربروس قلمرو همکار مقابل به اشتراک می گذارند.
- وجود  $N$  قلمرو همکار نیازمند  $N(N-1)/2$  کلید مخفی است.
- دو کارگزار کربروس همدیگر را ثبت نام می کنند.

# احراز هویت بین قلمرویی



# کربروس نسخه ۵

---

## مشخصات

- در اواسط ۱۹۹۰ مطرح شد
- نقص‌ها و کمبودهای نسخه قبلی را برطرف کرده است
- به عنوان استاندارد اینترنتی **RFC 1510** در نظر گرفته شده است.
- ویندوز ۲۰۰۰ از استاندارد اینترنتی کربروس نسخه ۵ به عنوان روش اصلی احراز هویت کاربران استفاده می‌کند.

## مشکلات Kerberos v4 و نحوه رفع آنها در نسخه ۵

□ وابستگی به یک سیستم رمزنگاری خاص (DES)

+ در نسخه ۵ می توان از هر الگوریتم متقارن استفاده کرد

□ وابستگی به IP

+ در نسخه ۵ می توان از هر آدرس شبکه (مثلا OSI یا IP) استفاده کرد

□ محدود بودن زمان اعتبار بلیتها

+ در نسخه ۵ این محدودیت وجود ندارد



## مشکلات Kerberos v4 و نحوه رفع آنها در نسخه ۵

---

- امکان انتقال اعتبار یک کاربر به یک سرور دیگر وجود ندارد
- + مثلاً **DBMS** نیاز دارد برای پاسخ دادن به پرس و جوی کاربر، برخی داده‌ها را از یک پایگاه داده دیگر بگیرد.
- با افزایش تعداد قلمروها، تعداد کلیدها بصورت تصاعدی افزایش می‌یابد
- + در نسخه ۵ این مشکل حل شده است.

## کربروس نسخه ۵: شمای کلی

(a) Authentication Service Exchange: to obtain ticket-granting ticket	
(1) $C \rightarrow AS$ :	$Options \parallel ID_c \parallel Realm_c \parallel ID_{tgs} \parallel Times \parallel Nonce_1$
(2) $AS \rightarrow C$ :	$Realm_c \parallel ID_c \parallel Ticket_{tgs} \parallel E_{K_c} [K_{c,tgs} \parallel Times \parallel Nonce_1 \parallel Realm_{tgs} \parallel ID_{tgs}]$
	$Ticket_{tgs} = E_{K_{tgs}} [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket	
(3) $C \rightarrow TGS$ :	$Options \parallel ID_v \parallel Times \parallel Nonce_2 \parallel Ticket_{tgs} \parallel Authenticator_c$
(4) $TGS \rightarrow C$ :	$Realm_c \parallel ID_c \parallel Ticket_v \parallel E_{K_{c,tgs}} [K_{c,v} \parallel Times \parallel Nonce_2 \parallel Realm_v \parallel ID_v]$
	$Ticket_{tgs} = E_{K_{tgs}} [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
	$Ticket_v = E_{K_v} [Flags \parallel K_{c,v} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
	$Authenticator_c = E_{K_{c,tgs}} [ID_c \parallel Realm_c \parallel TS_1]$
(c) Client/Server Authentication Exchange: to obtain service	
(5) $C \rightarrow TGS$ :	$Options \parallel Ticket_v \parallel Authenticator_c$
(6) $TGS \rightarrow C$ :	$E_{K_{c,v}} [TS_2 \parallel Subkey \parallel Seq\#]$
	$Ticket_v = E_{K_v} [Flags \parallel K_{c,v} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
	$Authenticator_c = E_{K_{c,v}} [ID_c \parallel Realm_c \parallel TS_2 \parallel Subkey \parallel Seq\#]$



# پیاده سازی های موجود

---

□ دانشگاه MIT : اولین پیاده سازی کربروس که هنوز به عنوان مرجع مورد استفاده قرار می گیرد

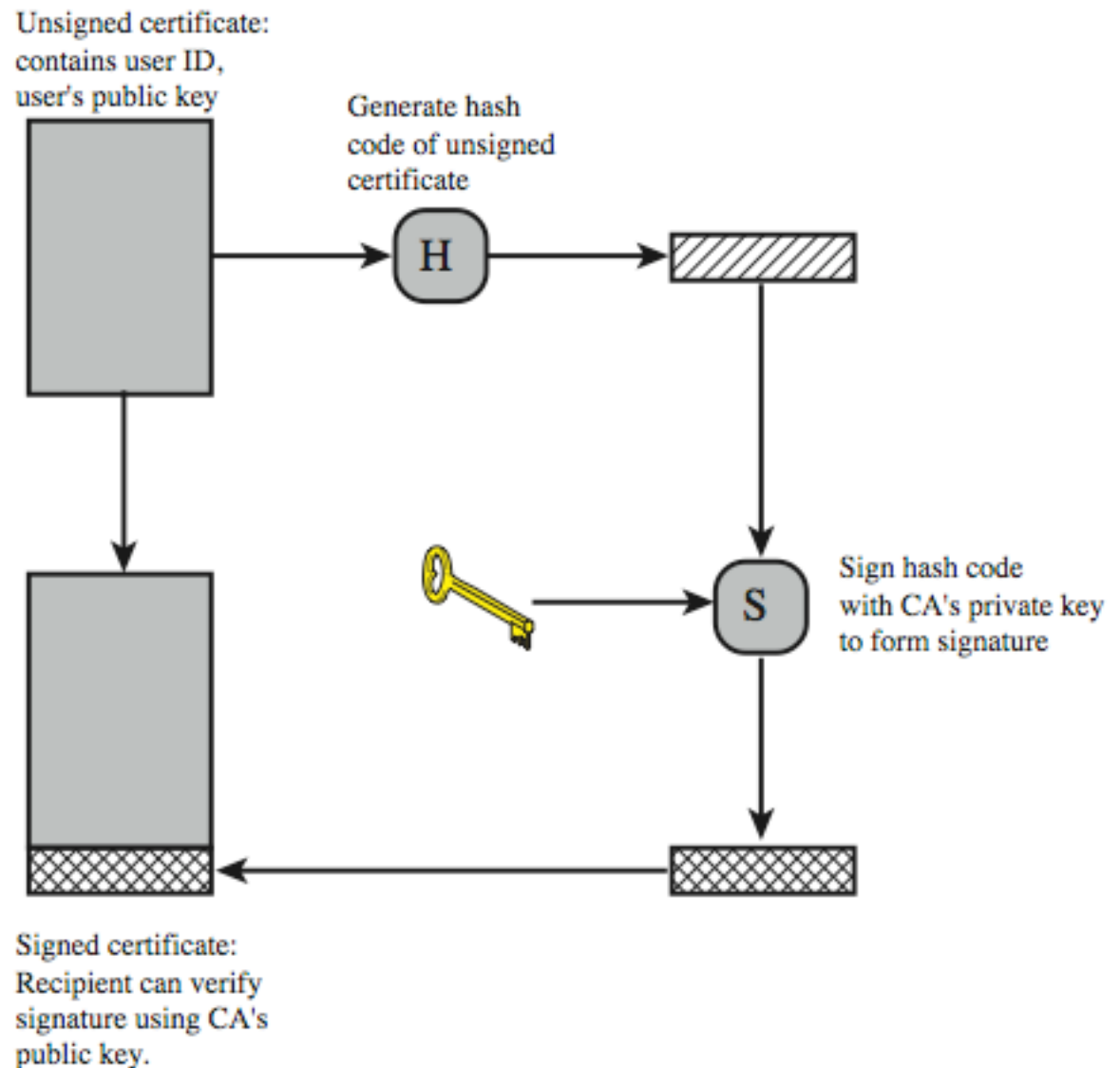
■ <http://web.mit.edu/kerberos/>

□ Active Directory : پیاده سازی ارائه شده توسط مایکروسافت

---

# توزیع کلید عمومی و گواهی های کلید عمومی

# X.509 Certificate Use

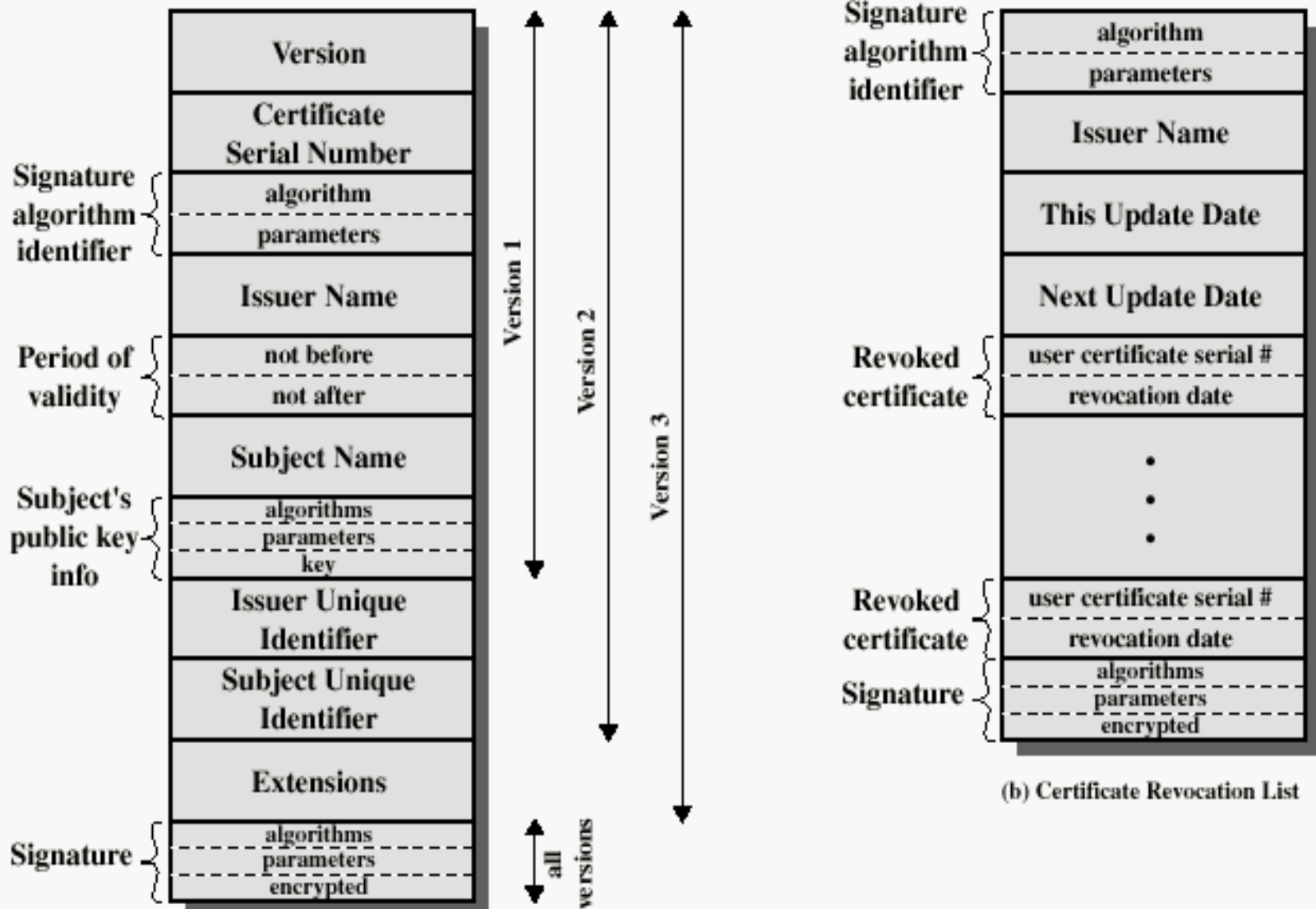


# X.509 Authentication Service

---

- ❑ Distributed set of servers that maintains a database about users.
- ❑ Each certificate contains the public key of a user and is signed with the private key of a CA.
- ❑ Is used in S/MIME, IP Security, SSL/TLS and SET.
- ❑ RSA is recommended to use.

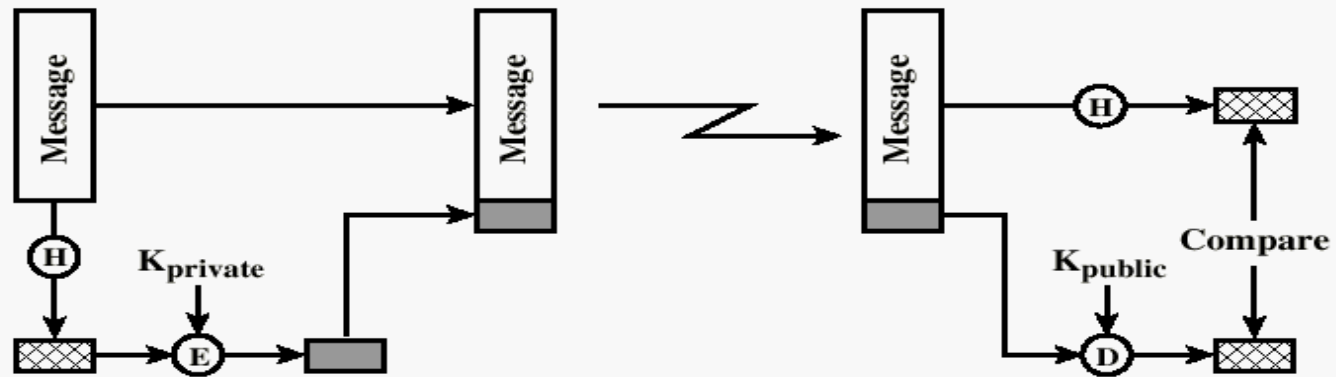
# X.509 Formats



(a) X.509 Certificate

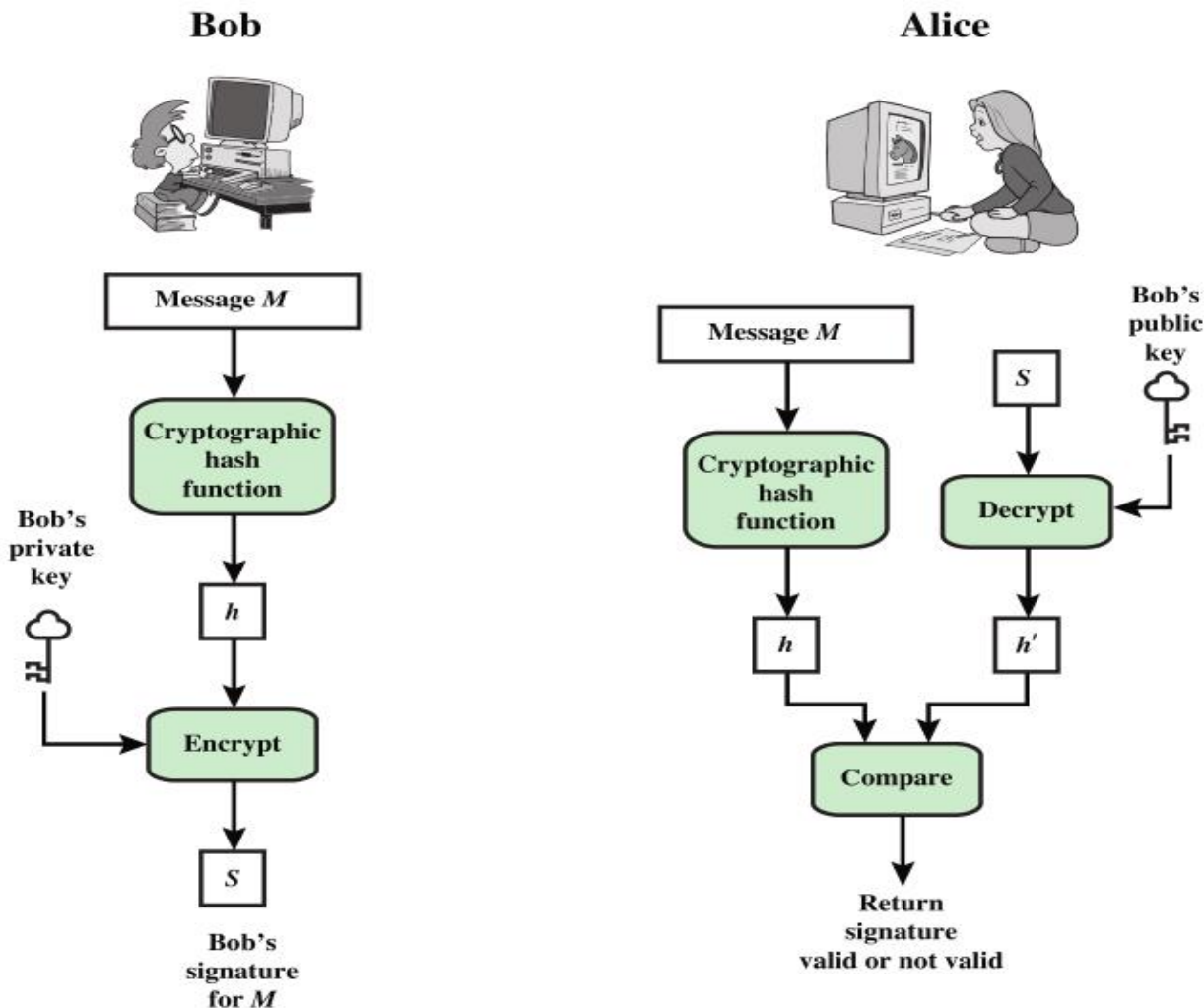
<http://ceit.aut.ac.ir/~shahriari>

# Typical Digital Signature Approach



(b) Using public-key encryption

# Essentials elements of Digital Signature



# Obtaining a User's Certificate

---

- Characteristics of certificates generated by CA:
  - Any user with access to the public key of the CA can recover the user public key that was certified.
  - No part other than the CA can modify the certificate without this being detected.



# X.509 notations

---

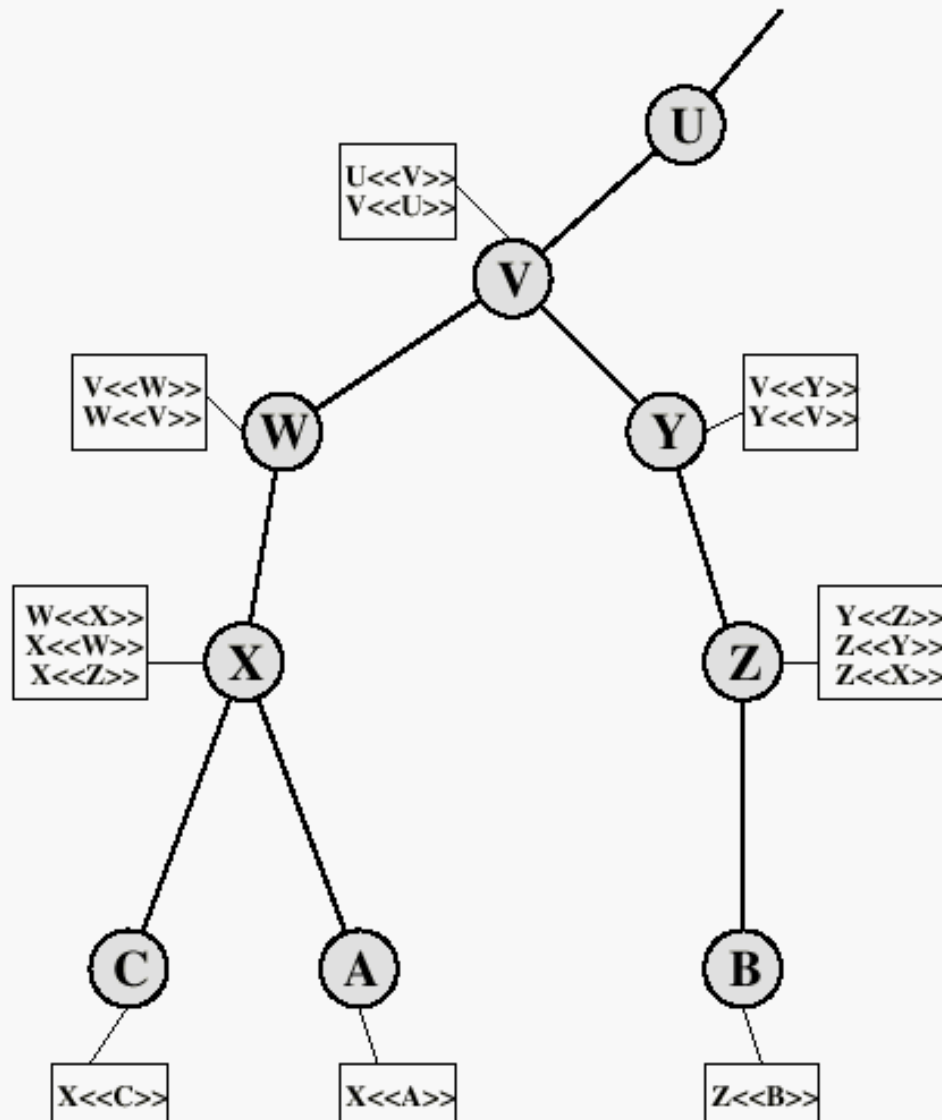
- $Y \llbracket X \rrbracket$ 
  - The certificate of user X issued by certificate authority Y
- $Y\{I\}$ 
  - The signing of I by Y

# Problem!

---

- How two users with different CAs, can authenticate each other?
  - $X1 \ll A \gg ? X2 \ll B \gg$
- Solution:
  - Two CAs securely exchange public keys:
    - $X1 \ll X2 \gg$
    - $X2 \ll X1 \gg$
  - Now: How can A acquire B's public key?

# X.509 CA Hierarchy



# Revocation of Certificates

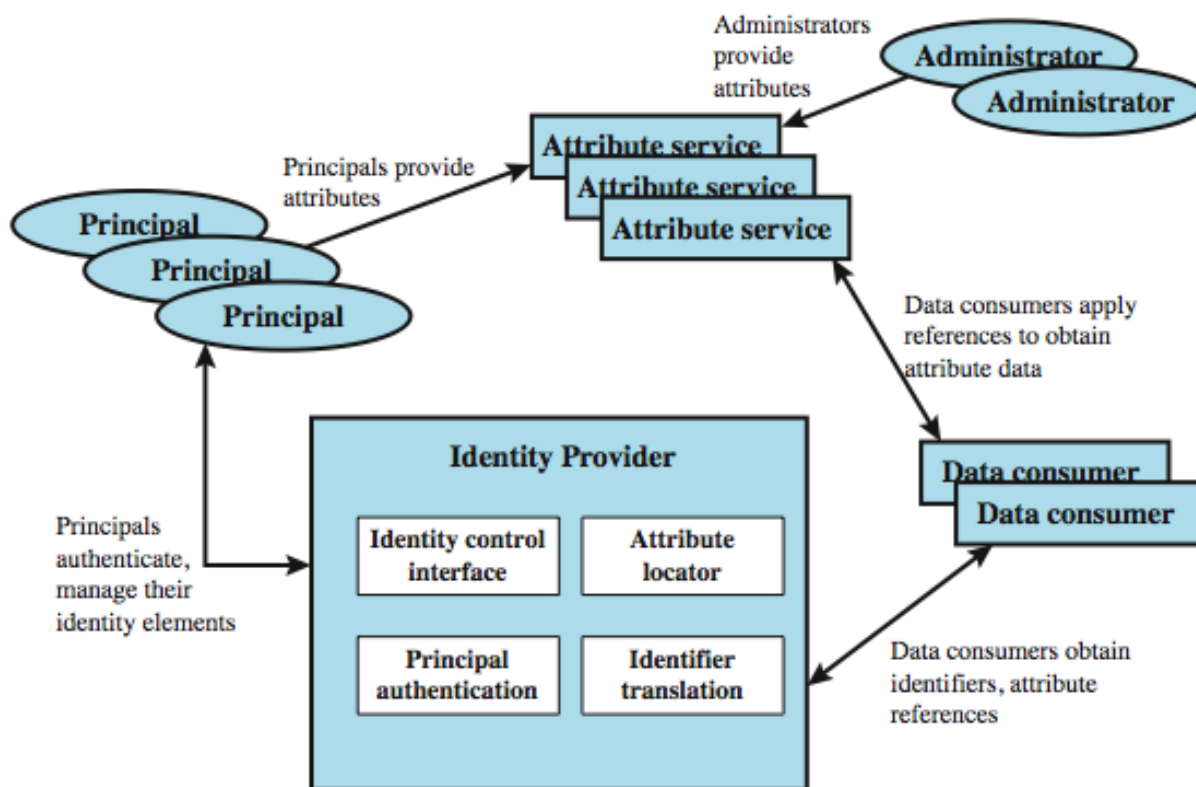
---

- Reasons for revocation:
  - The users secret key is assumed to be compromised.
  - The user is no longer certified by this CA.
  - The CA's certificate is assumed to be compromised.

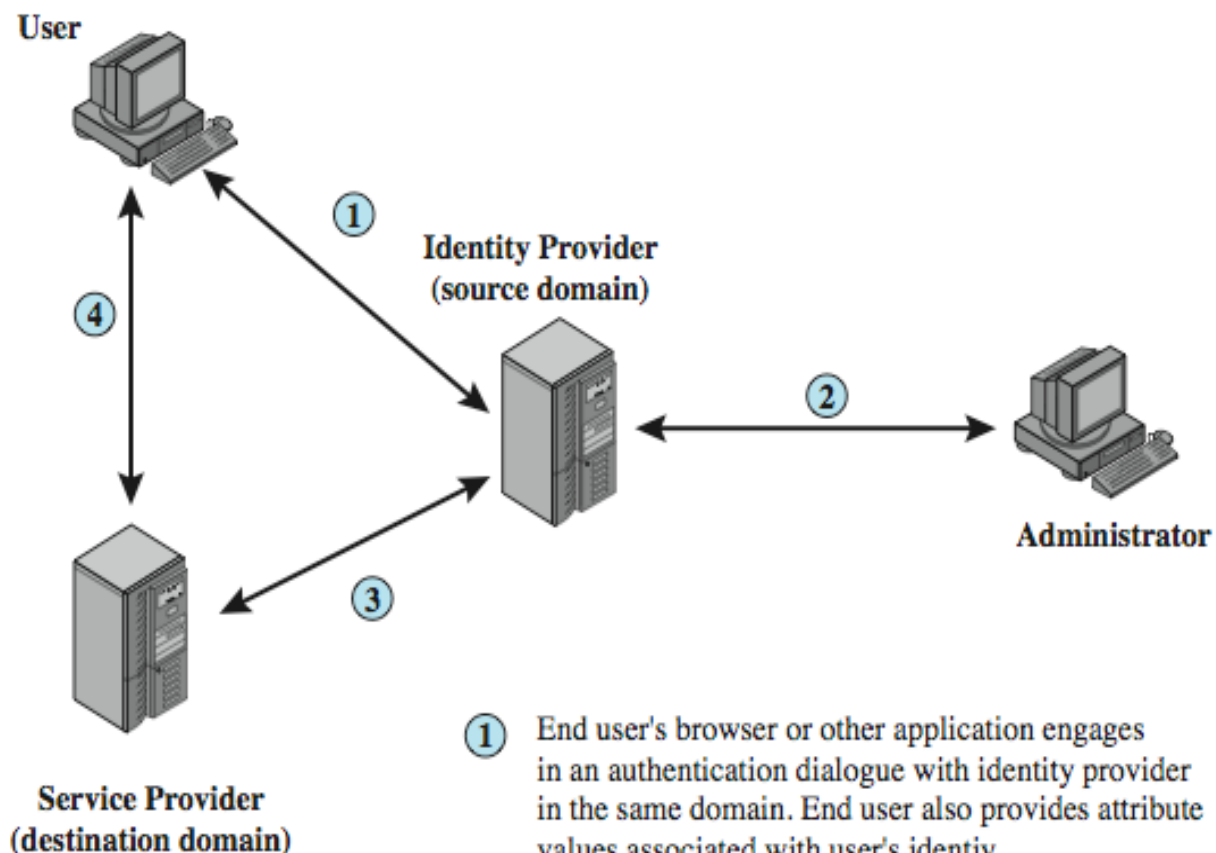
# Federated Identity Management

- use of common identity management scheme
  - across multiple enterprises & numerous applications
  - supporting many thousands, even millions of users
- principal elements are:
  - authentication, authorization, accounting, provisioning, workflow automation, delegated administration, password synchronization, self-service password reset, federation
- Kerberos contains many of these elements

# Identity Management



# Identity Federation



- 1** End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.
- 2** Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.
- 3** A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.
- 4** Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

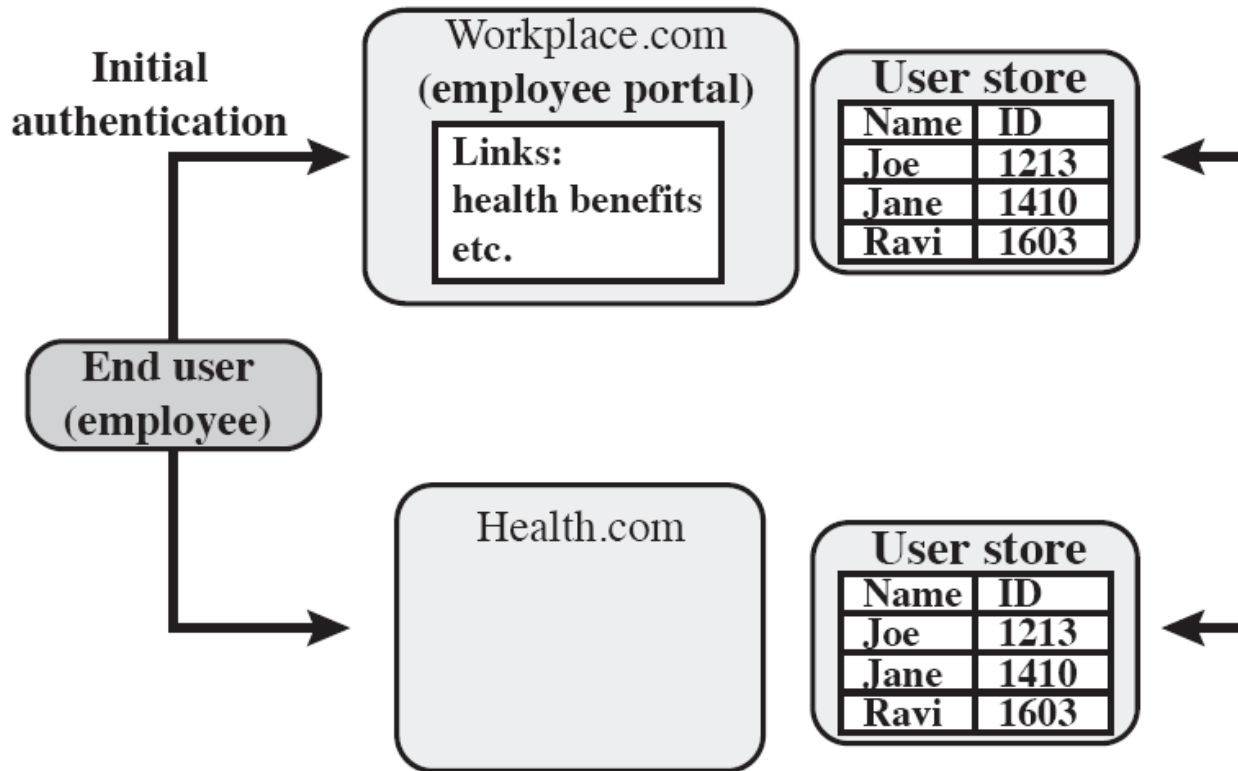
# Standards Used

---

- Security Assertion Markup Language (SAML)
  - XML-based language for exchange of security information between online business partners
- part of OASIS (Organization for the Advancement of Structured Information Standards) standards for federated identity management
  - e.g. WS-Federation for browser-based federation
- need a few mature industry standards

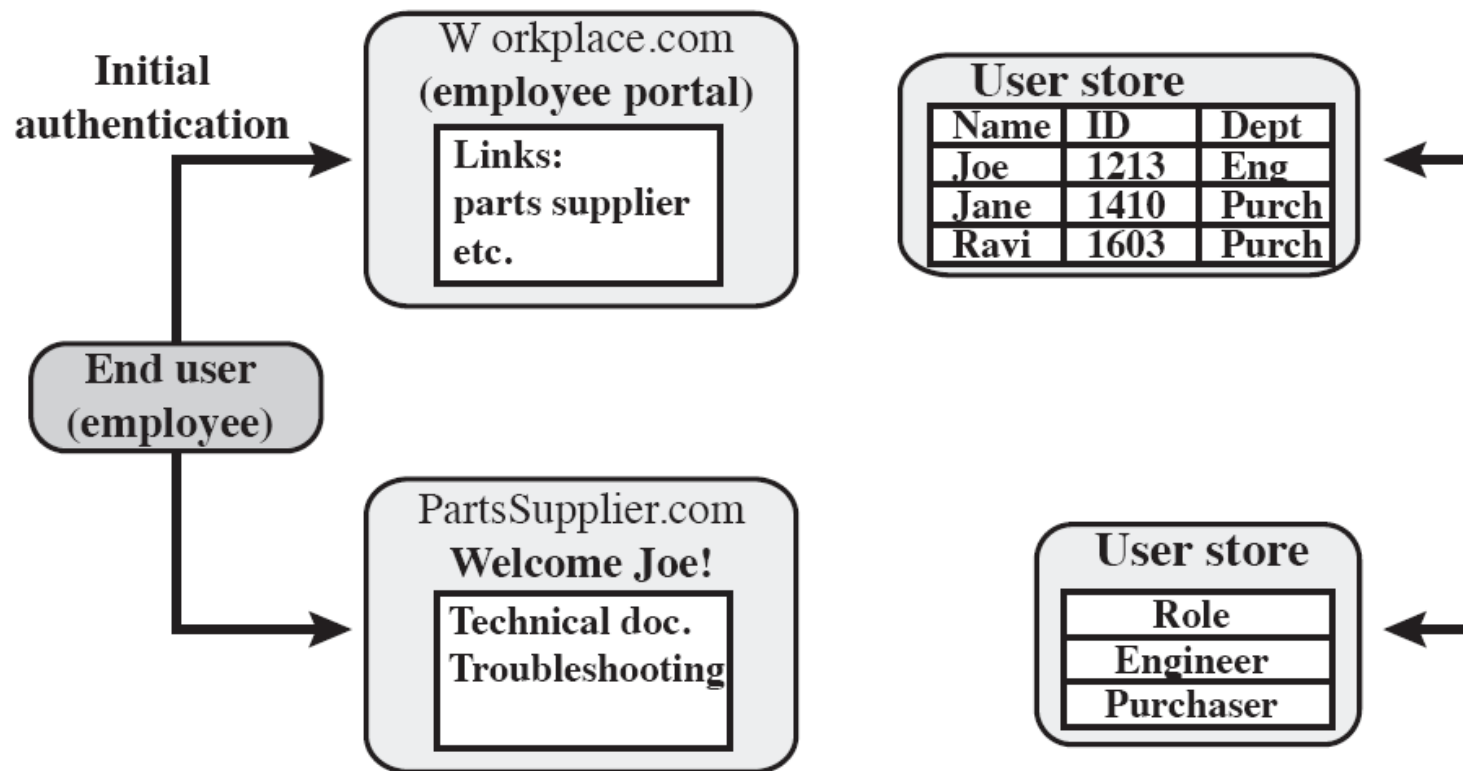


# Federated Identity Examples



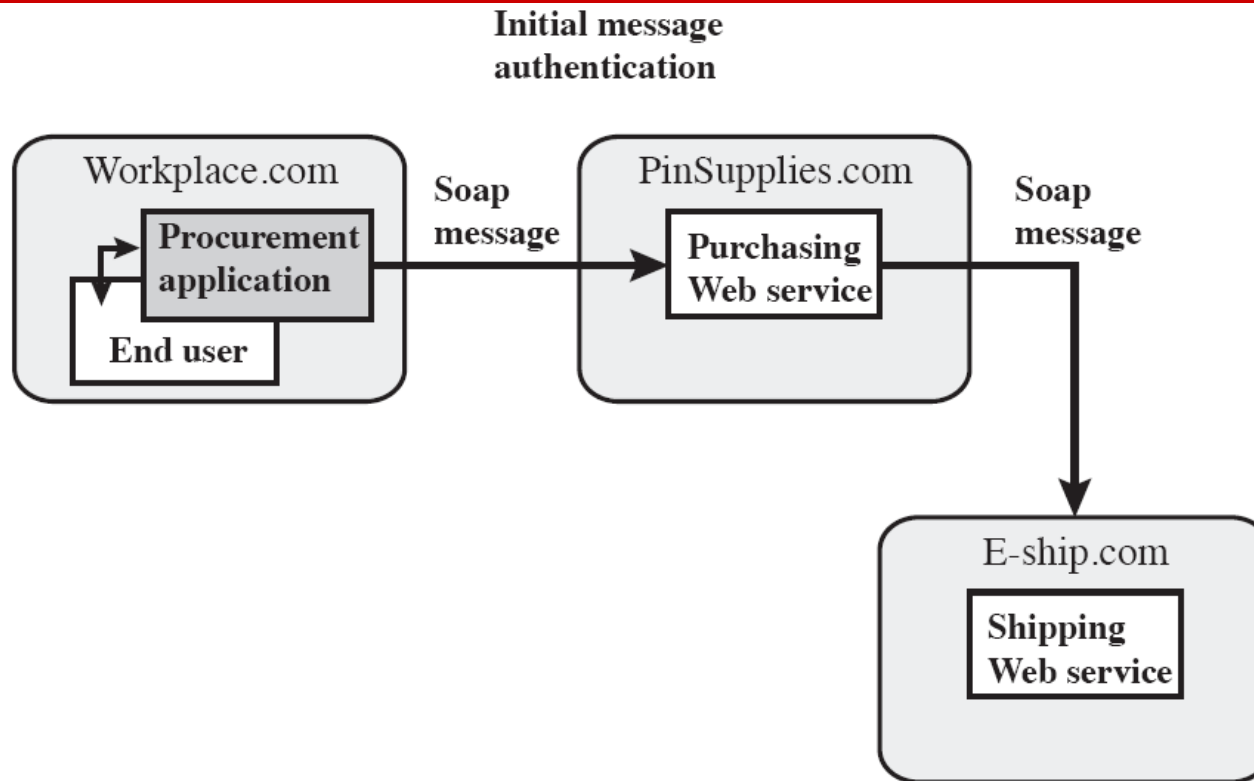
(a) Federation based on account linking

# Federated Identity Examples



(b) Federation based on roles

# Federated Identity Examples



(b) Chained Web Services

# OpenID

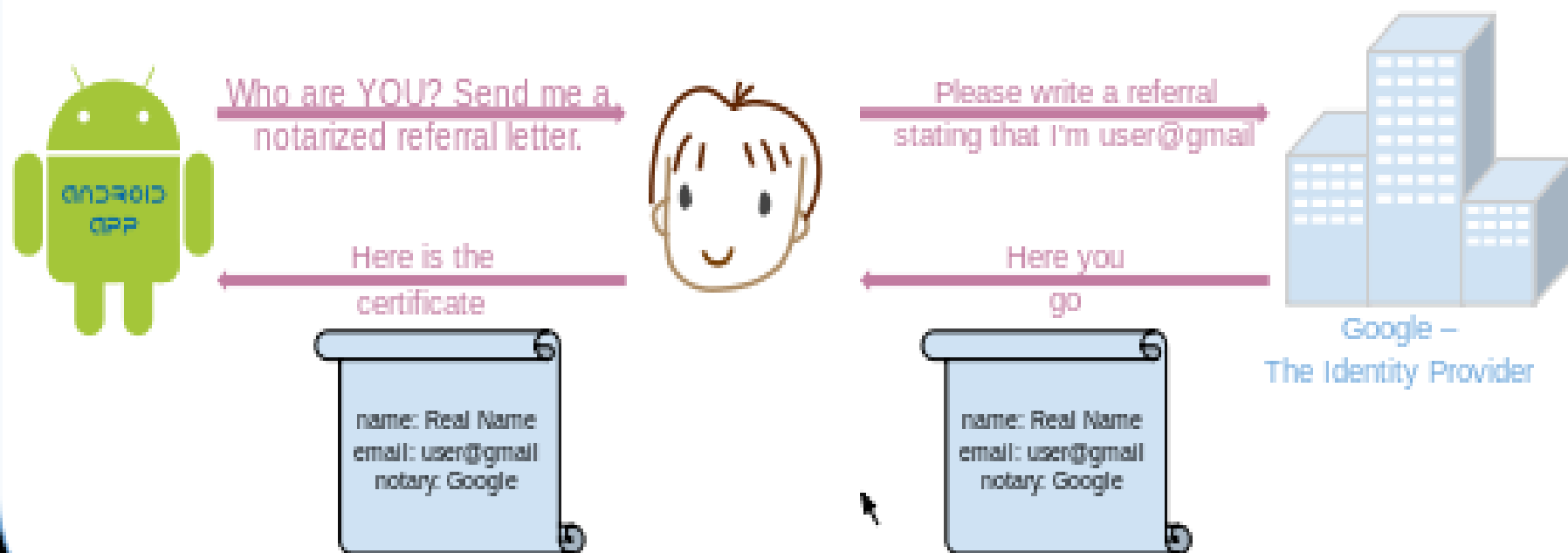
---

## □ OpenID

- open standard
- describes how users can be authenticated in a decentralized manner
- eliminating the need for services to provide their own ad hoc systems
- allowing users to consolidate their digital identities.

# OpenID

## OpenID Authentication



# واژه نامه

Authentication	احراز هویت
Reliability	قابلیت اطمینان
Authenticator	اعتبار نامه
ticket-granting ticket	بلیت "اعطای بلیت"
service-granting ticket	بلیت اعطای خدمات
Tickets	بلیت‌ها
Distributed	توزیع شده
Register	ثبت نام
Integrity	صحت
Spoof	جعل
Address Spoofing	جعل آدرس
service session	جلسه خدمات
Replay Attack	حمله تکرار
Administrative Domain	حوزه مدیریتی

Services	خدمات
Realm	دامنه
Transparency	شفافیت
ID	شناسه
Principal	عنصری که شناسانده میشوند
Realm	قلمرو
Password	گذر واژه
Certificate	گواهی
Plain Text	متن واضح
Key Management	مدیریت کلید
KDC: Key distribution Center	مرکز توزیع کلید
Scalability	مقیاس پذیری
Timestamp	مهر زمانی
service type	نوع خدمات
AS: Authentication Server	کارگزار احراز هویت،
Server	کارگزار
TGS: Ticket Granting Server	کارگزار اعطا کننده بلیت
Session Key	

# پیوست

---

## □ AAA (Authentication, Authorization, and Accounting)

The process of providing and tracking access to network resources. Authentication involves the mechanism to verify user identity. Once identified, Authorization grants the user access privileges to system and network resources. Accounting keeps a history of system and network resource utilization and the users involved.