یا ذالامن والامان

توزیع و مدیریت کلید و احراز هویت کاربر
Key Distribution and User Authentication (Kerberos, X.509, PKI, ...)

مبتنی بر فصل ۴

ویرایش شده توسط: حمید رضا شهریاری
http://www.aut.ac.ir/shahriari

Remote user authentication principles

- In most computer security contexts, user authentication is the fundamental building block and the primary line of defense
- User authentication is the basis for most types of access control and for user accountability
- RFC 4949 (Internet Security Glossary) defines user authentication as the process of verifying an identity claimed by or for a system entity
  - Identification step
    - Presenting an identifier to the security system
  - Verification step
    - Presenting or generating authentication information that corroborates the binding between the entity and the identifier

NIST Model for Electronic User Authentication

- NIST SP 800-63-2 (*Electronic Authentication Guideline,* August 2013 defines electronic user authentication as the process of establishing confidence in user identities that are presented electronically to an information system
- Systems can use the authenticated identity to determine if the authenticated individual is authorized to perform particular functions
- In many cases, the authentication and transaction or other authorized function take place across an open network such as the Internet
- Equally, authentication and subsequent authorization can take place locally, such as across a local area network

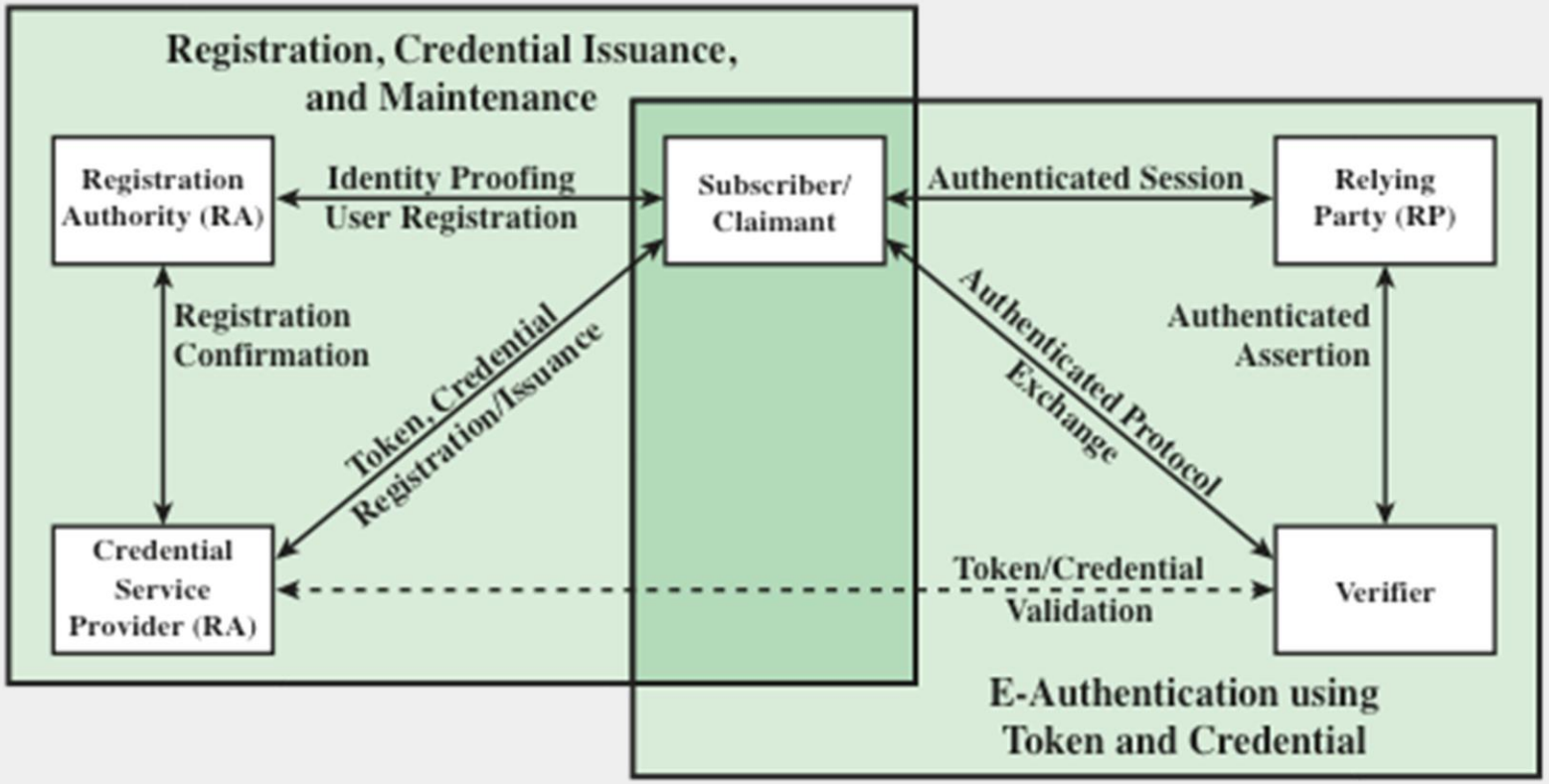


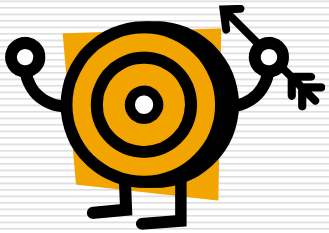
Figure 4.1 The NIST SP 800-63-2 E-Authentication Architectural Model

Means of authentication

- There are four general means of authenticating a user's identity, which can be used alone or in combination
  - Something the individual knows
    - Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions
  - Something the individual possesses
    - Examples include cryptographic keys, electronic keycards, smart cards, and physical keys
    - This type of authenticator is referred to as a *token*
  - Something the individual is (static biometrics)
    - Examples include recognition by fingerprint, retina, and face
  - Something the individual does (dynamic biometrics)
    - Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm

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.

کربروس

- احراز هویت بر اساس رمز نگاری کلید مخفی (متقارن)
- طراحی شده در دانشگاه 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 بعدا با RFC 4120 در سال ۲۰۰۵ جایگزین شد.

نیازمندیها/ویژگیهای عمومی کربروس

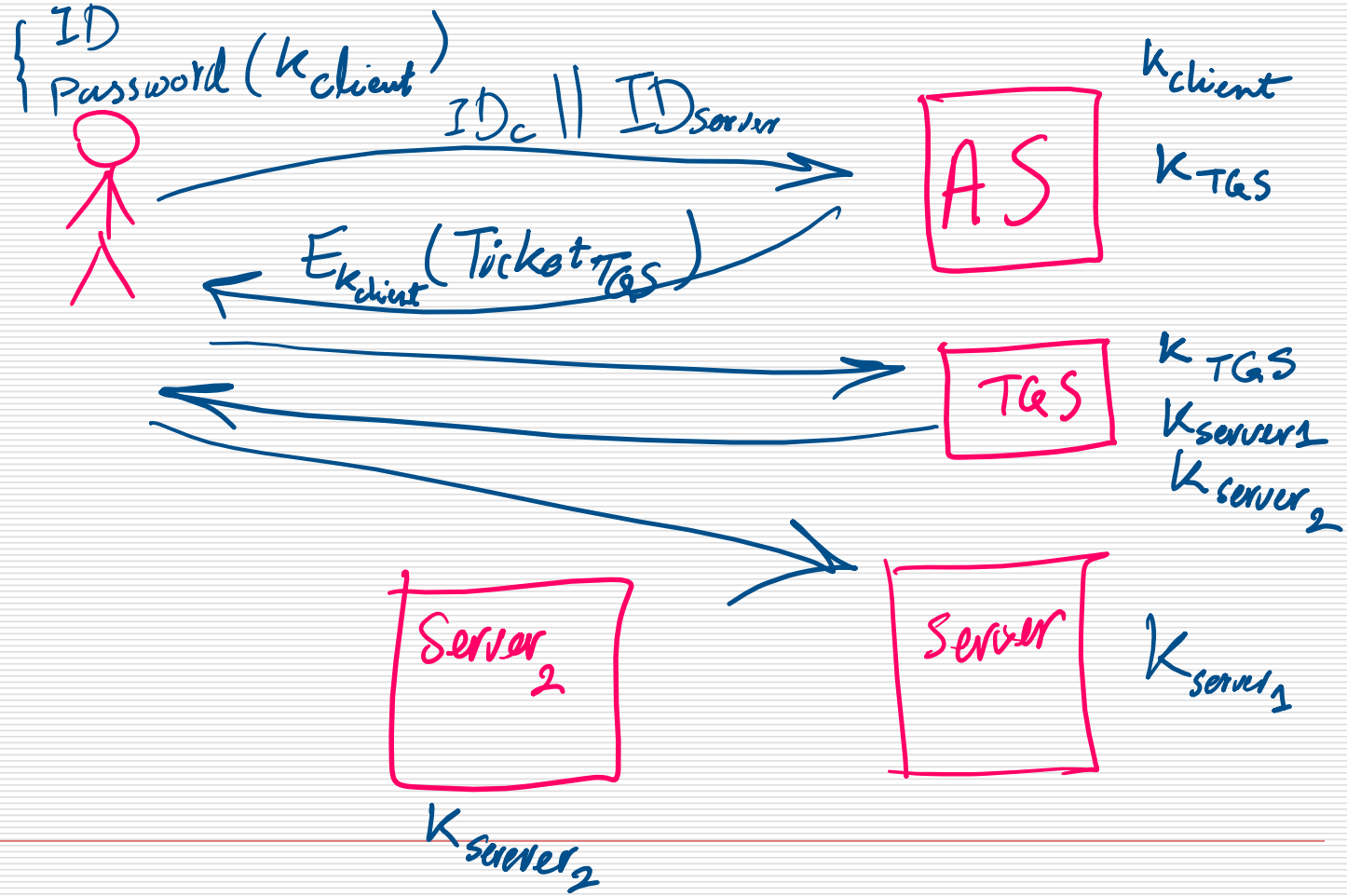
- عمومی‌بودن(Common)
  - در محیط توزیع شده همراه با سرورهای متمرکز و غیر متمرکز
- امنیت (Security)
  - ادعای اصلی
- اطمینان (Reliability)
  - اطمینان از دسترس‌پذیری کارگزار احراز هویت (کربروس)
- شفافیت (Transparency)
  - کاربران باید سیستم را همانند یک سیستم ساده «*شناسه و گذرواژه*» ببینند.
- مقیاس پذیری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

ویژگیهای عمومی کربروس

- چند تعریف
  - دامنه(domain) یا قلمرو (realm) : یک محدوده دسترسی را مشخص می‌کند. به نوعی معادل دامنه‌های تعریف شده در ویندوز می‌باشد.
  - مرکز توزیع کلید: معادل کارگزار کربروس می‌باشد.
  - Principal : به سرویس ها، دستگاه ها، کاربران و همه عناصری که احتیاج به شناساندن خود به کارگزار کربروس دارند، گفته می‌شود.

کربروس

- **برای معرفی کربروس به صورت گام به گام از پروتکل‌های ساده شروع می‌کنیم و سعی می‌کنیم اشکال‌های هر یک را برطرف کنیم تا به کربروس برسیم.**


{ ID
password (K_{client})
$ID_c \| ID_{server}$
$E_{K_{client}}(Ticket_{TGS})$
AS
K_{client}
K_{TGS}
TGS
K_{TGS}
K_{server_1}
K_{server_2}
Server 2
Server
K_{server_1}
K_{server_2}

دیالوگ ساده احراز هویت-۰

فرض: بین AS و هر کارگزار یک کلید مشترک وجود دارد.

درخواست خدمات توسط کارفرما از کارگزار:

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

AS : Authentication Server *کارگزار احراز هویت*

$\boldsymbol{E_{Kserver}}$***: Shared key between AS and Server***

$\boldsymbol{Ticket = E_{Kserver}\,[ID_{client} \,||\, Addr_{client} \,||\, ID_{server}]}$

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 آورده 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 ارایه می‌شود.
    - مشکل جعل آدرس

- چرا شناسه کارفرما ID_{client} 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 ارایه نمی‌شوند.

مشکلات دیالوگ ساده احراز هویت-۰

- ناامنی
  - ارسال گذرواژه بدون رمزگذاری (به شکل متن واضح)
  - امکان حمله تکرار (ارسال مجدد یا Replay attack)
- ناکارآیی
  - لزوم تقاضای بلیت جدید برای هر خدمت از سرور جدید

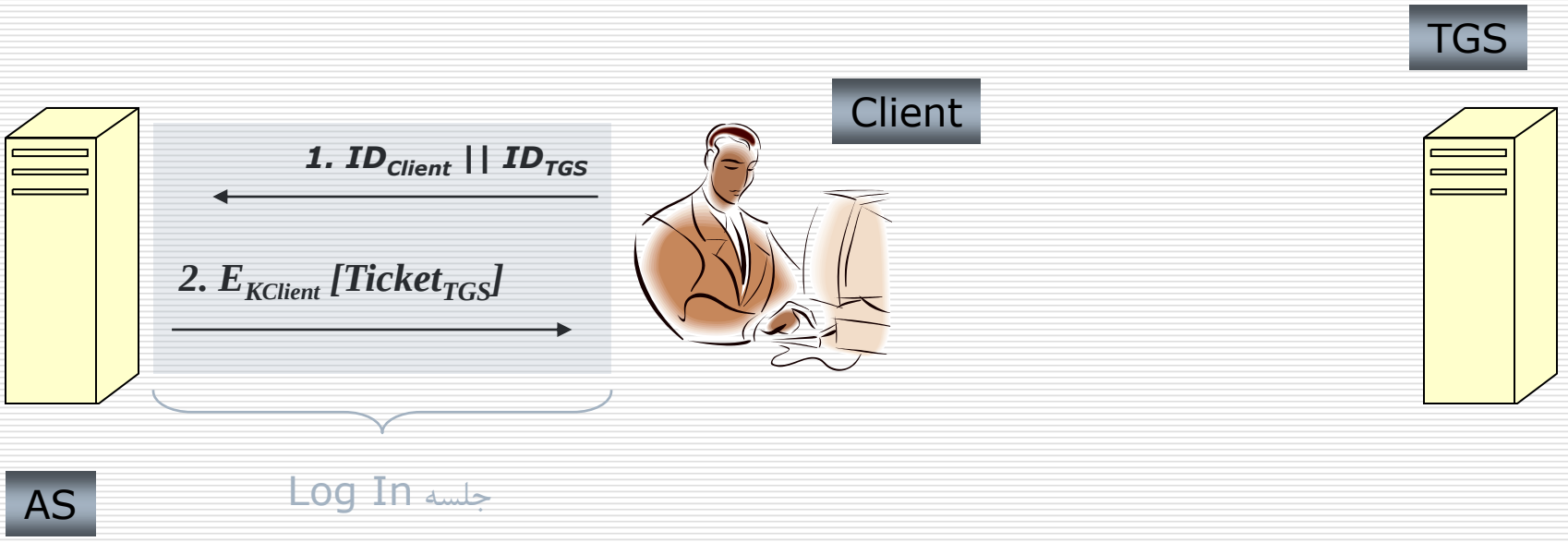
استفاده مجدد از بلیت‌ها

- 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.

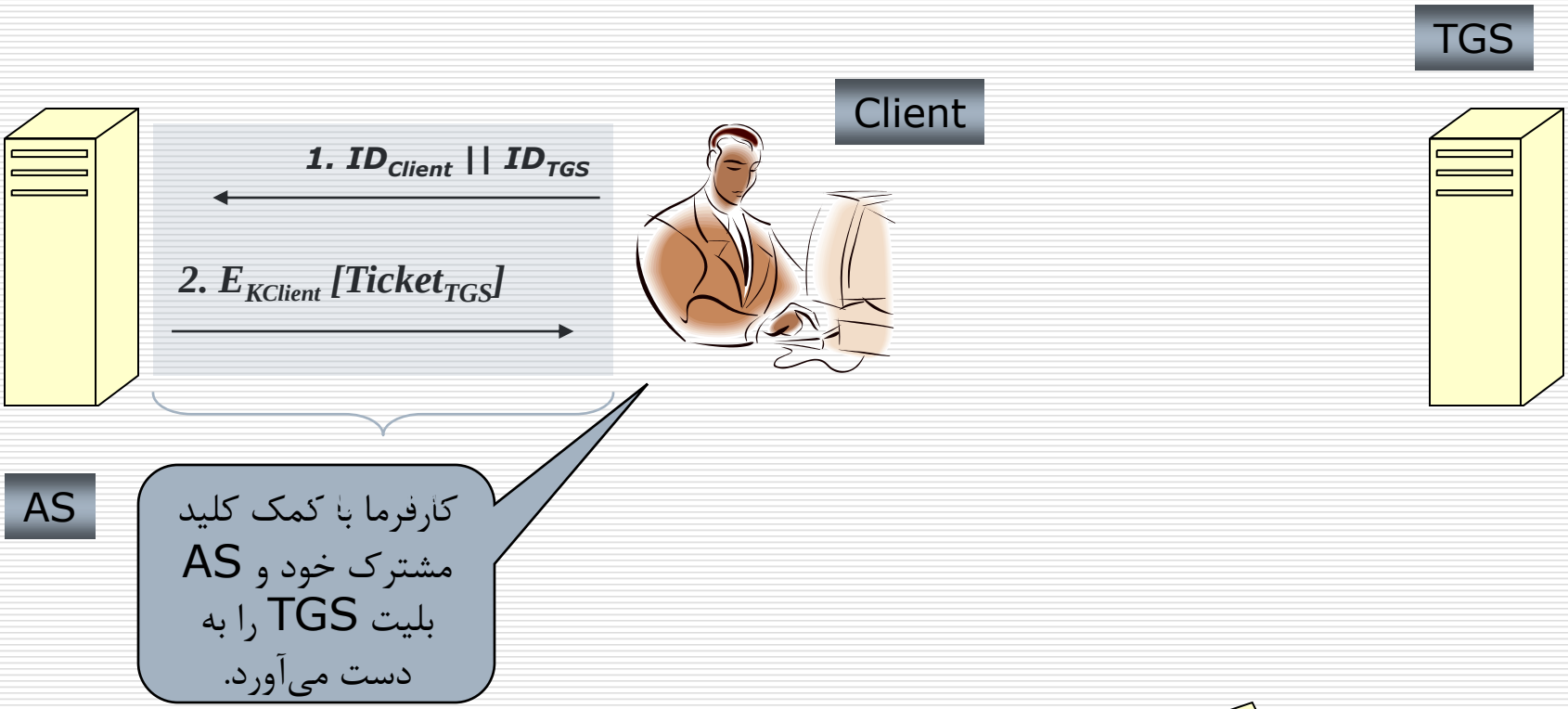
افزایش ایمنی-دیالوگ ۱

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

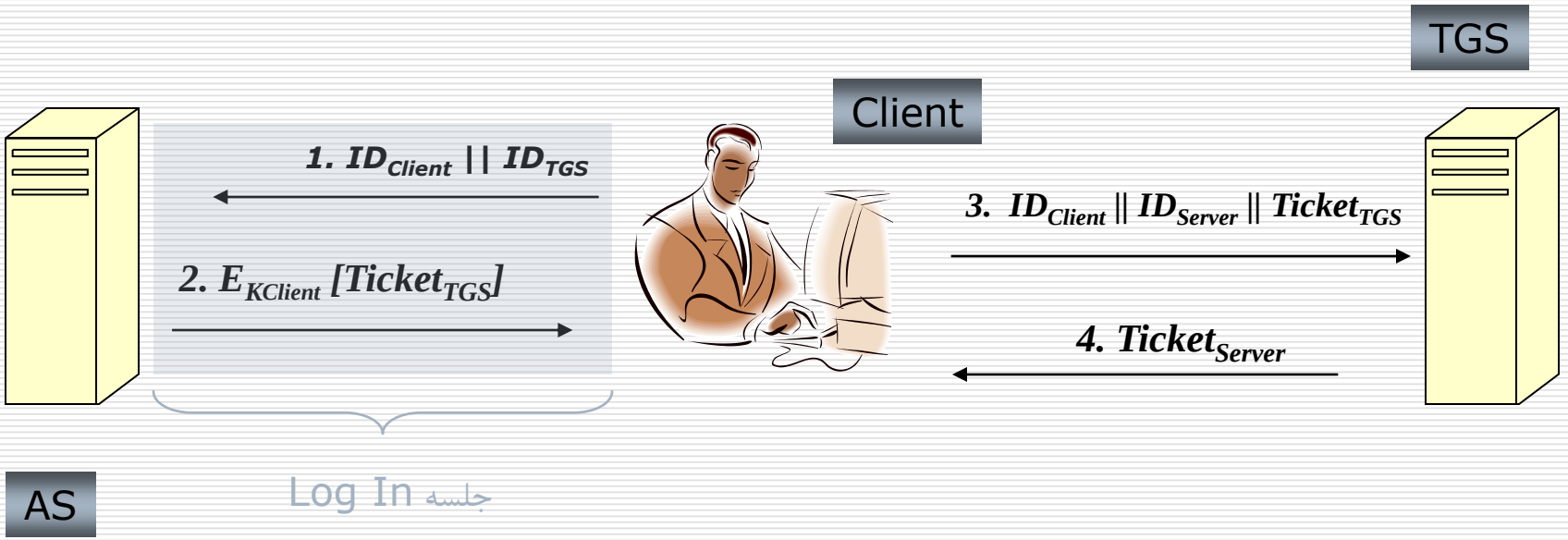
افزایش ایمنی -دیالوگ۱



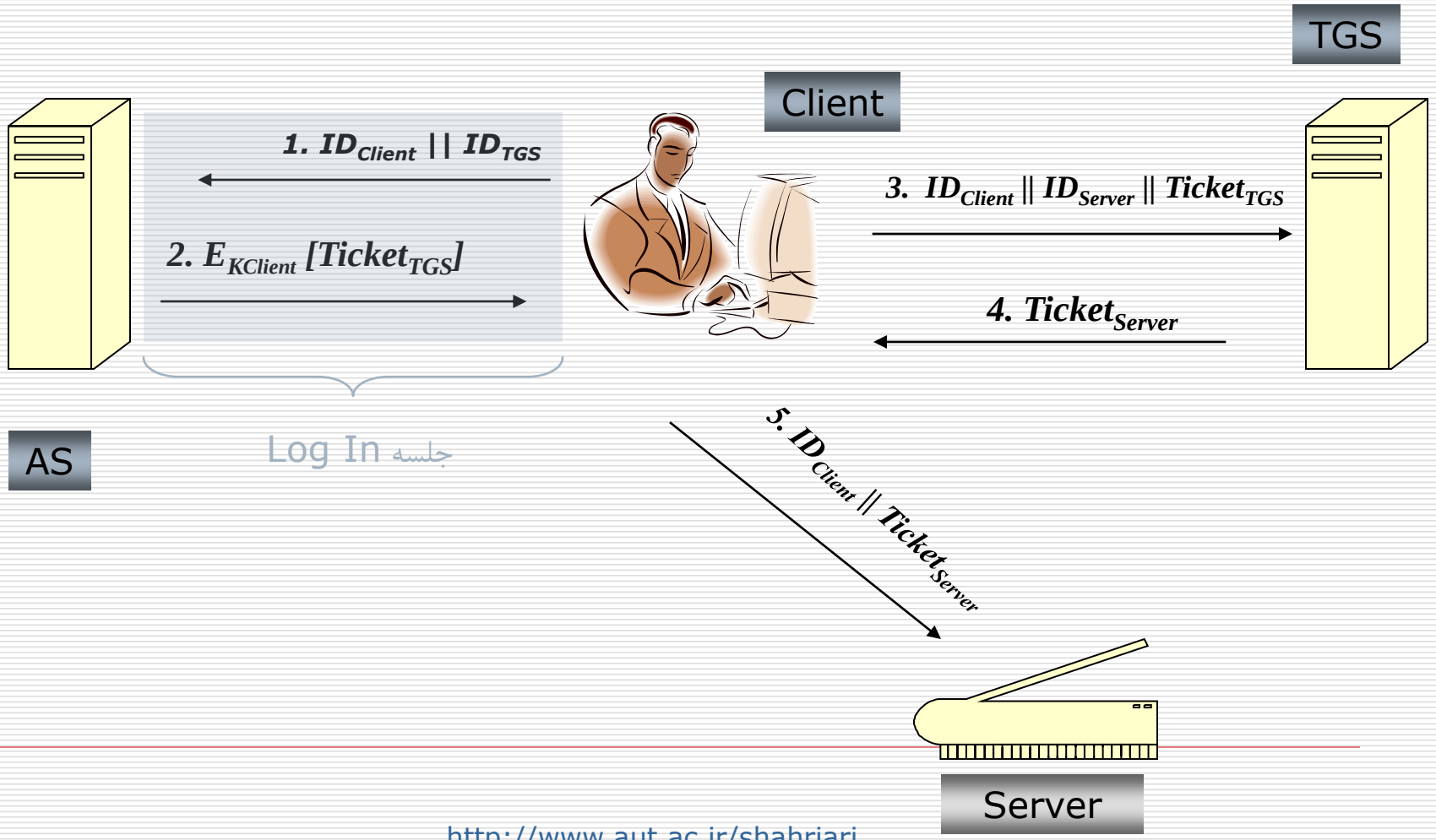
افزایش ایمنی -دیالوگ۱



افزایش ایمنی -دیالوگ۱



افزایش ایمنی -دیالوگ۱



افزایش ایمنی -دیالوگ ۱

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

1. $Client \rightarrow AS: ID_{Client} \,||\, ID_{TGS}$
2. $AS \rightarrow Client: E_{KClient}\,[Ticket_{TGS}]$
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 :

$$Ticket_{TGS} = E_{K_{TGS}} [ID_{Client} \| Addr_{Client} \| ID_{TGS} \| Timestamp_1 \| Lifetime_1]$$

بلیت اعطای خدمات :

$$Ticket_{Server} = E_{K_{Server}} [ID_{Client} \| Addr_{Client} \| ID_{Server} \| Timestamp_2 \| Lifetime_2]$$

ویژگی های دیالوگ ۱

- دو بلیت صادر شده ساختار مشابهی دارند. در اساس به دنبال هدف یکسانی هستند.
- رمزنگاری $Ticket_{TGS}$ جهت احراز هویت
  - تنها کارفرما می‌تواند به بلیت رمزشده 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 می‌کند.
- هنوز از آدرس شبکه برای احراز هویت بهره می‌گیرد.
  - چندان جالب نیست زیرا آدرس شبکه جعل (Spoof) می‌شود.
  - با این حال، درجه ای از امنیت فراهم می‌شود.

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 درخواست‌ها به یک کارگزار غیرمجاز

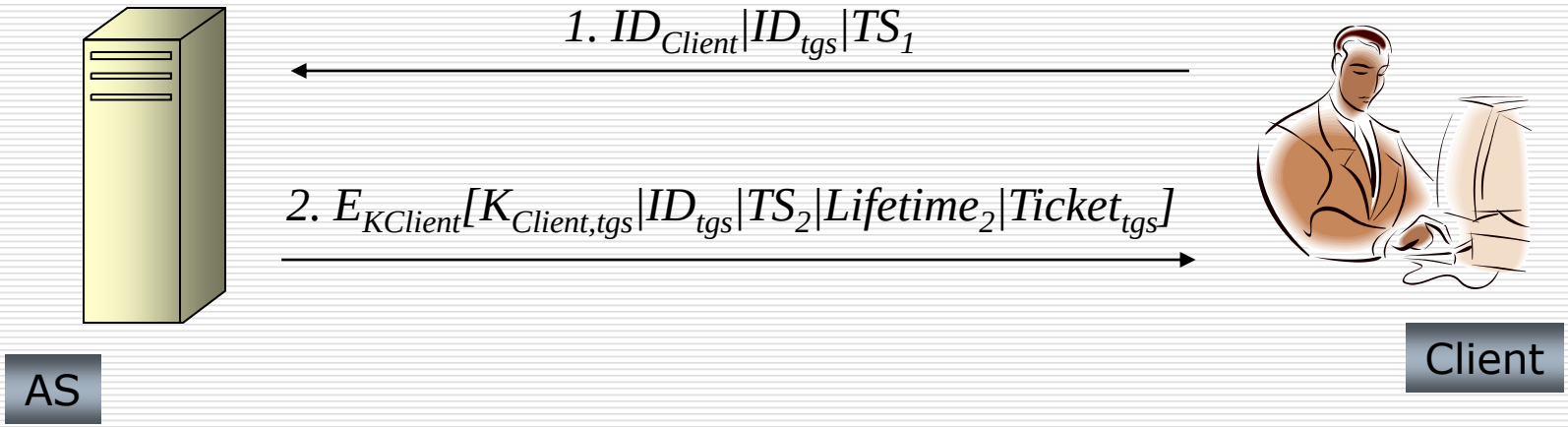
کربروس نسخه ۴

- 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 یابد 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

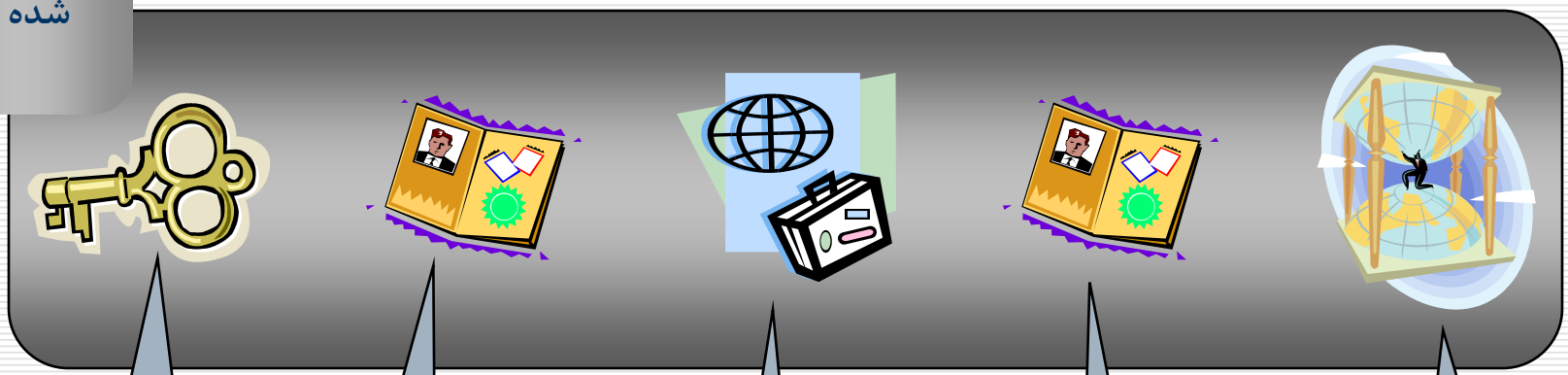


$$Ticket_{tgs}=E_{Ktgs}[K_{Client,tgs}|ID_{Client}|Addr_{Client}|ID_{tgs}|TS_2|Lifetime_2]$$

بلیت TGS

تمامی با کلید رمز TGS شده اند

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



کلید جلسه بین کارفرما و TGS

شناسه کارفرما

آدرس کارفرما

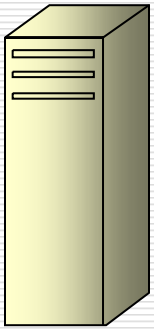
شناسه TGS

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

$Ticket_{Server} =$

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

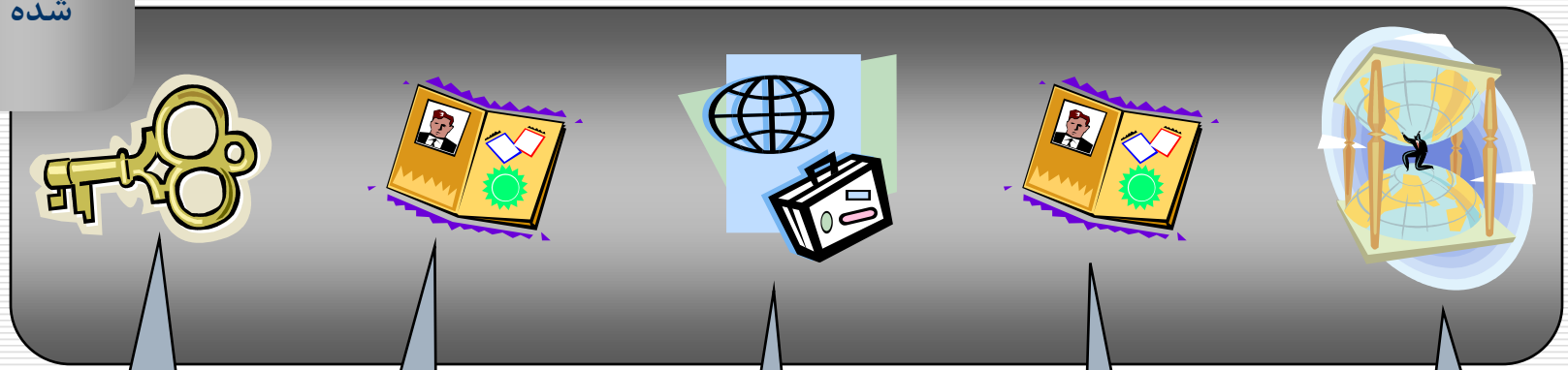
$Authenticator_{Client} =$

$E_{KClient,tgs}[ID_{Client} | Addr_{Client} | TS_3]$

بلیت کارگزار

تمامی با کلید کارگزار رمز شده اند

$$Ticket_{Server} = E_{Kserver}[K_{Client,server}|ID_{Client}|Addr_{Client}|ID_{server}|TS_4|Lifetime_4]$$



کلید جلسه بین کارفرما و کارگزار

شناسه کارفرما

آدرس کارفرما

شناسه TGS

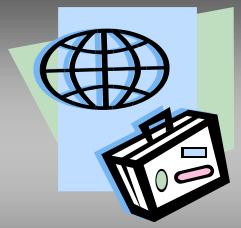
مهر زمانی و دوره اعتبار بلیت

اعتبار نامه کارفرما

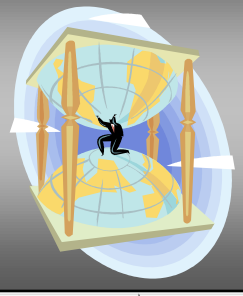
تمامی با کلید جلسه رمز شده اند

$$Authenticator_{Client} = E_{KClient,tgs}[ID_{Client}|Addr_{Client}|TS_3]$$

شناسه کارفرما



آدرس کارفرما



مهر زمانی

نتایج این مرحله برای کارفرما

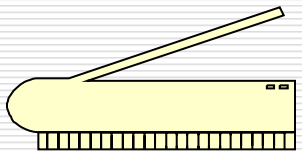
- جلوگیری از حمله تکرار با استفاده از یک اعتبار نامه (*Authenticator*) یک بار مصرف که عمر کوتاهی دارد.
- بدست آوردن کلید جلسه برای ارتباط با سرور ($K_{c,s}$)

دستیابی به خدمات سرور

Client

5. $Ticket_{Server}|Authenticator2_{Client}$

6. $E_{K_{Client,Server}}[TS_5+1]$



Server

اعتبارنامه جدیدی که توسط کلاینت برای ارائه به سرور ساخته می شود:

$$Authenticator2_{Client} = E_{KClient,s}[ID_{Client}|Addr_{Client}|TS_5]$$

نتایج این مرحله برای کارفرما

- احراز هویت کارگزار در گام ششم با برگرداندن پیام رمزشده
- جلوگیری از بروز حمله تکرار

کربروس نسخه ۴: شمای کلی

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

(1) C → AS: $ID_c \| ID_{tgs} \| TS_1$

(2) AS → C: $E_{K_c}\left[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}\right]$

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

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

(3) C → TGS: $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C: $E_{K_{c,tgs}}\left[K_{c,v} \| ID_v \| TS_4 \| Ticket_v\right]$

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

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

$$Authenticator_c = E_{K_{tgs}}\left[ID_C \| AD_C \| TS_3\right]$$

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

(5) C → V: $Ticket_v \| Authenticator_c$

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

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

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

Rationale for the Elements of the Kerberos Version 4 Protocol

Message (1)	Client requests ticket-granting ticket.
ID_C	Tells AS identity of user from this client.
ID_{tgs}	Tells AS that user requests access to TGS.
TS_1	Allows AS to verify that client's clock is synchronized with that of AS.
Message (2)	AS returns ticket-granting ticket.
K_c	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2).
$K_{c,tgs}$	Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}	Confirms that this ticket is for the TGS.
TS_2	Informs client of time this ticket was issued.
$Lifetime_2$	Informs client of the lifetime of this ticket.
$Ticket_{tgs}$	Ticket to be used by client to access TGS.

(a) Authentication Service Exchange

Table 4.2 Rationale for the Elements of the Kerberos Version 4 Protocol
(page 2 of 3)

Message (3)	Client requests service-granting ticket.
ID_V	Tells TGS that user requests access to server V.
$Ticket_{tgs}$	Assures TGS that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket .
Message (4)	TGS returns service-granting ticket.
$K_{c,tgs}$	Key shared only by C and TGS protects contents of message (4).
$K_{c,v}$	Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key.
ID_V	Confirms that this ticket is for server V.
TS_4	Informs client of time this ticket was issued.
$Ticket_V$	Ticket to be used by client to access server V.
$Ticket_{tgs}$	Reusable so that user does not have to reenter password.
K_{tgs}	Ticket is encrypted with key known only to AS and TGS, to prevent Tampering.
$K_{c,tgs}$	Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_{tgs}	Assures server that it has decrypted ticket properly.
TS_2	Informs TGS of time this ticket was issued.
$Lifetime_2$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay.
$K_{c,tgs}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_C	Must match address in ticket to authenticate ticket.
TS_3	Informs TGS of time this authenticator was generated.

Rationale for the Elements of the Kerberos Version 4 Protocol

Message (5)	Client requests service.
$Ticket_V$	Assures server that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (6)	Optional authentication of server to client.
$K_{c,v}$	Assures C that this message is from V.
$TS_5 + 1$	Assures C that this is not a replay of an old reply.
$Ticket_v$	Reusable so that client does not need to request a new ticket from TGS for each access to the same server.
K_v	Ticket is encrypted with key known only to TGS and server, to prevent tampering.
$K_{c,v}$	Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_V	Assures server that it has decrypted ticket properly.
TS_4	Informs server of time this ticket was issued.
$Lifetime_4$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay.
$K_{c,v}$	Authenticator is encrypted with key known only to client and server, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_c	Must match address in ticket to authenticate ticket.
TS_5	Informs server of time this authenticator was generated.

(c) Client/Server Authentication Exchange

کربروس نسخه ۴: شمای کلی

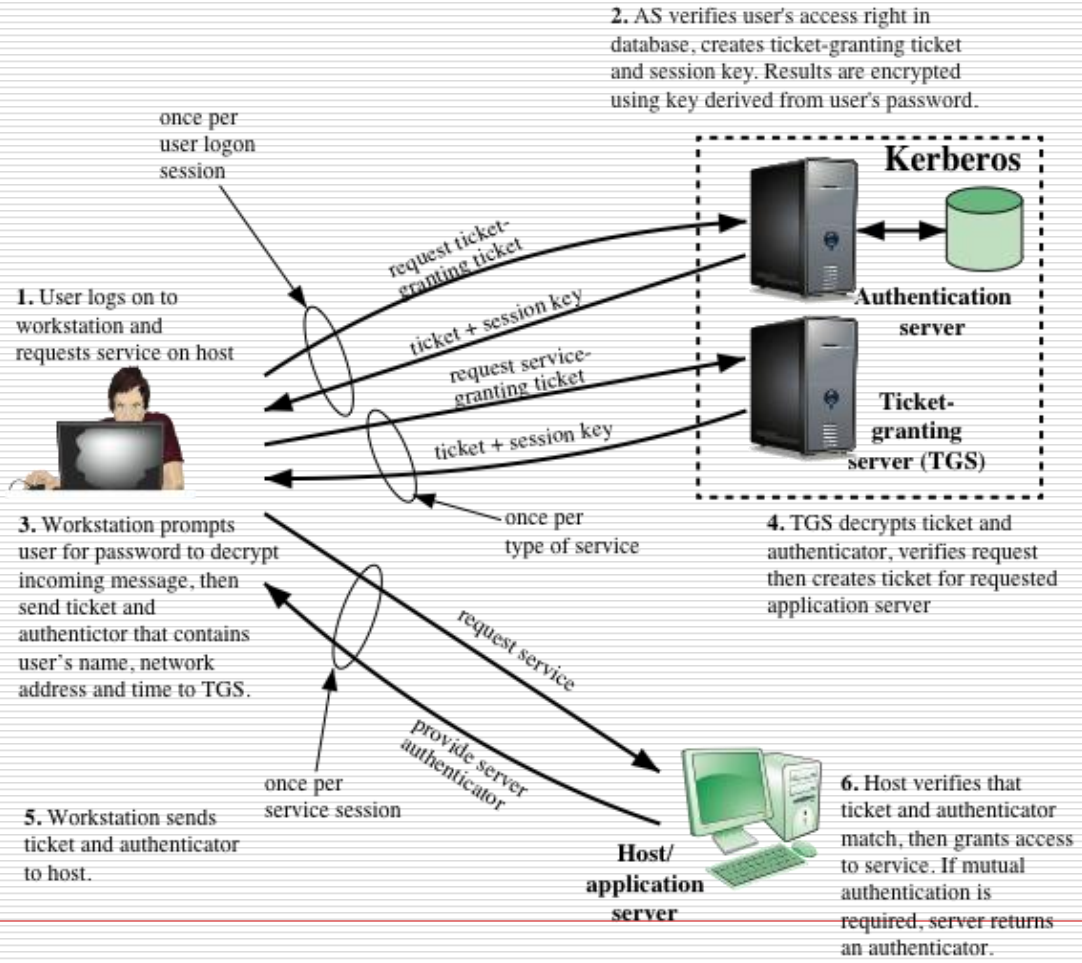


Figure 4.2 Overview of Kerberos

قلمرو کربروس (realm)

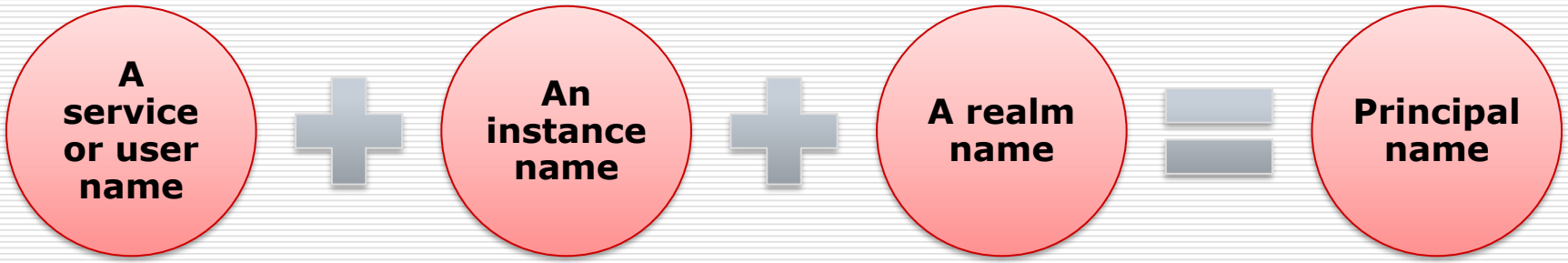
A Kerberos environment consists of:

- **A Kerberos server**
- **A number of clients**
- **A number of application servers**

- قلمرو کربروس از بخش‌های زیر تشکیل شده است:
  - کارگزار کربروس
  - کارفرمایان
  - کارگزاران برنامه‌های 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.

Kerberos principal

- A service or user that is known to the Kerberos system
- Each Kerberos principal is identified by its principal name

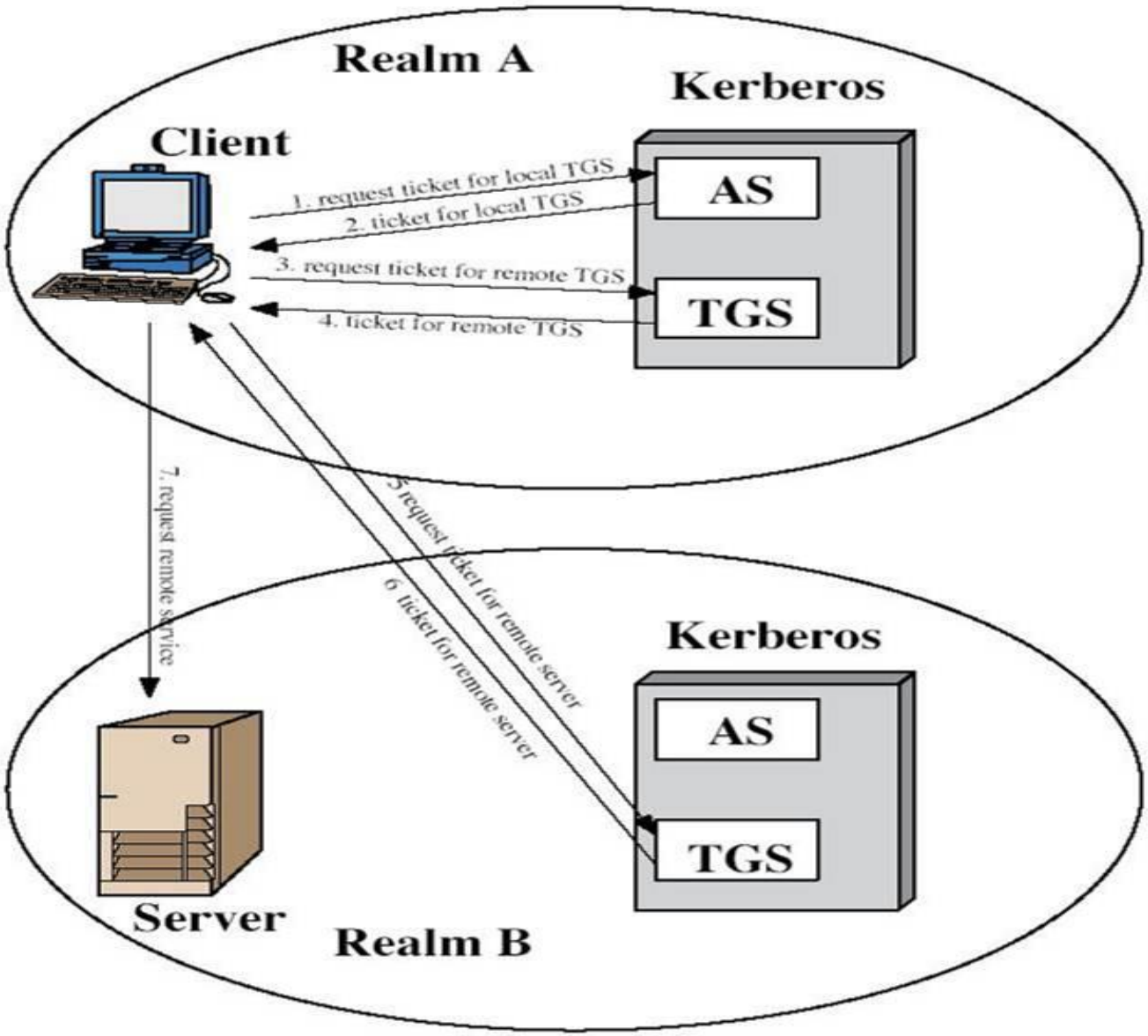


Principal names consist of three parts

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 می‌گذارند.
  - وجود ***N*** قلمرو همکار نیازمند ***N(N-1)/2*** کلید مخفی است.
  - دو کارگزار کربروس همدیگر را ثبت نام می‌کنند.

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 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 می‌توان از هر الگوریتم متقارن استفاده کرد
- وابستگی به IP
  + در نسخه ۵ می‌توان 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
  + مثلا 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) $C \rightarrow AS$: $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) $AS \rightarrow C$: $Realm_c \| ID_c \| Ticket_{tgs} \| E_{Kc}[K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{c,tgs} \| Realm_c \| ID_c \| AD_c \| Times]$

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

(3) $C \rightarrow TGS$: $Options \| ID_v \| Times \| \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C$: $Realm_c \| ID_c \| Ticket_v \| E_{K_{c,tgs}}[K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v]$

$Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{c,tgs} \| Realm_c \| ID_c \| AD_c \| Times]$

$Ticket_v = E_{K_v}[Flags \| K_{c,v} \| Realm_c \| ID_c \| AD_c \| Times]$

$Authenticator_c = E_{K_{c,tgs}}[ID_c \| Realm_c \| TS_1]$

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

(5) $C \rightarrow TGS$: $Options \| Ticket_v \| Authenticator_c$

(6) $TGS \rightarrow C$: $E_{K_{c,v}}[TS_2 \| Subkey \| Seq\#]$

$Ticket_v = E_{K_v}[Flags \| K_{c,v} \| Realm_c \| ID_c \| AD_c \| Times]$

$Authenticator_c = E_{K_{c,V}}[ID_c \| Realm_c \| TS_2 \| Subkey \| Seq\#]$

پیاده سازی های موجود

- دانشگاه MIT : اولین پیاده سازی کربروس که هنوز به عنوان مرجع مورد استفاده قرار می‌گیرد
  - http://web.mit.edu/kerberos/

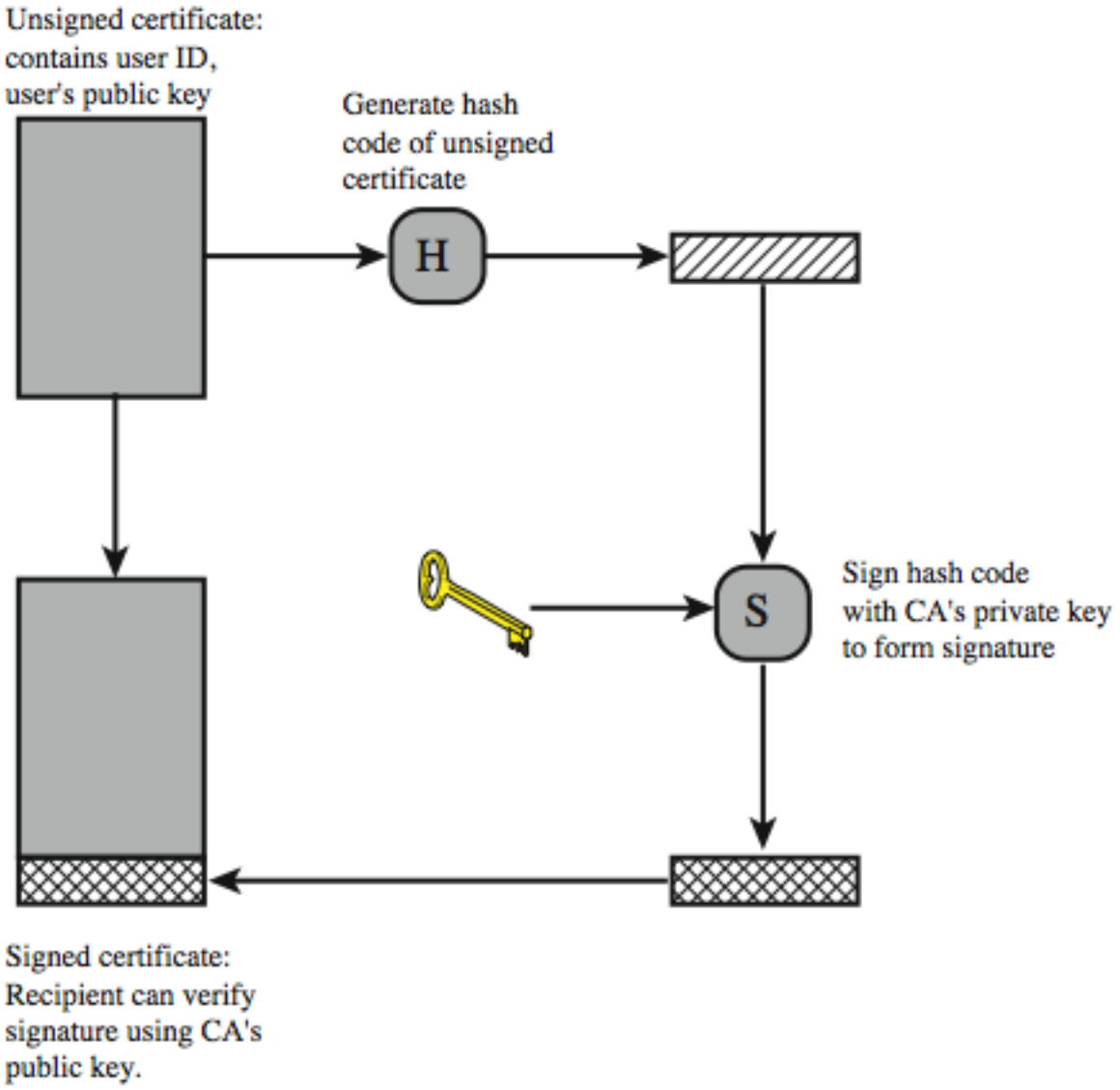
- Active Directory : پیاده سازی ارائه شده توسط مایکروسافت

مشکلات و محدودیتها

- گلوگاه شکست (Single point of failure)
  - در صورت از کار افتادن سرور مرکزی هیچ کاربری نمی‌تواند وارد شود.
- وابستگی به زمان و نیاز به همگامی زمانی
- پروتکل مدیریتی استانداردی وجود ندارد (مدیریت کاربران، گذرواژه‌ها و ...)
- هر سروری که نیاز به نام میزبان دارد، نیاز به اشتراک کلید با کارگزار کربروس دارد که در کلاسترینگ و نیز مجازی سازی مشکل ایجاد می‌کند.

توزیع کلید عمومی و گواهی های کلید عمومی

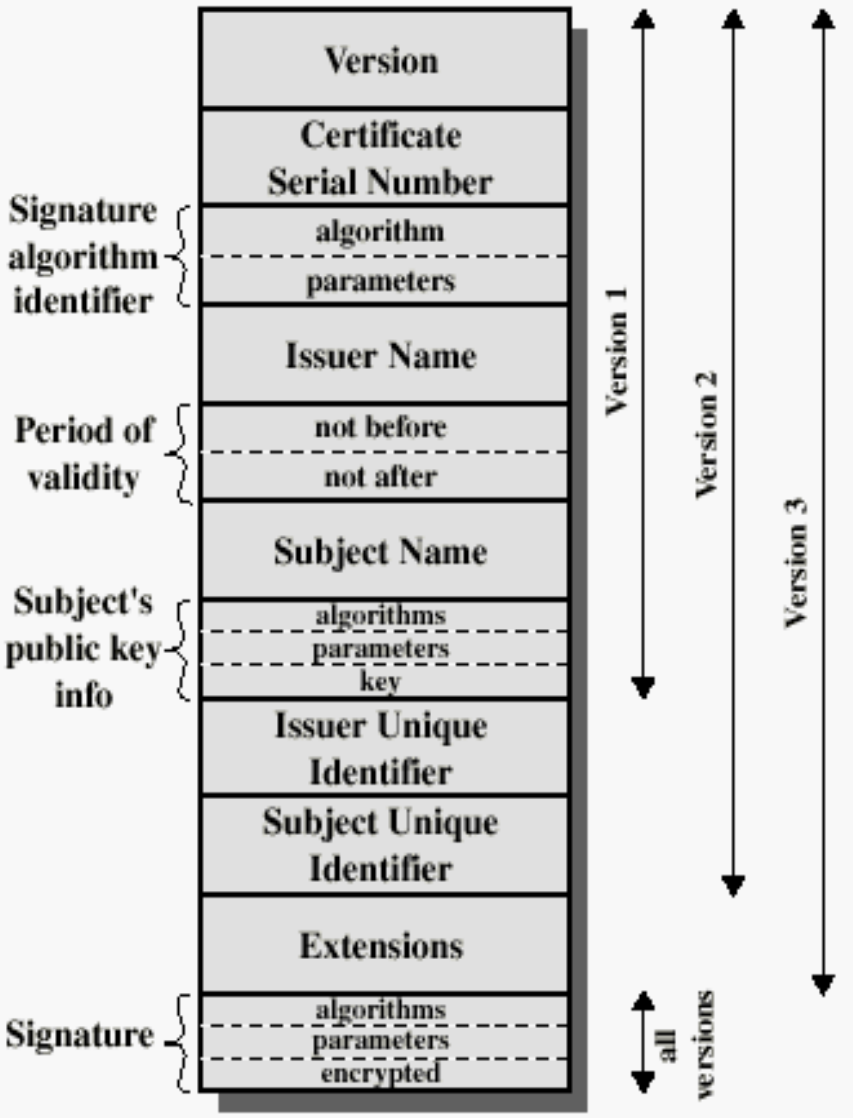
X.509 Certificate Generation



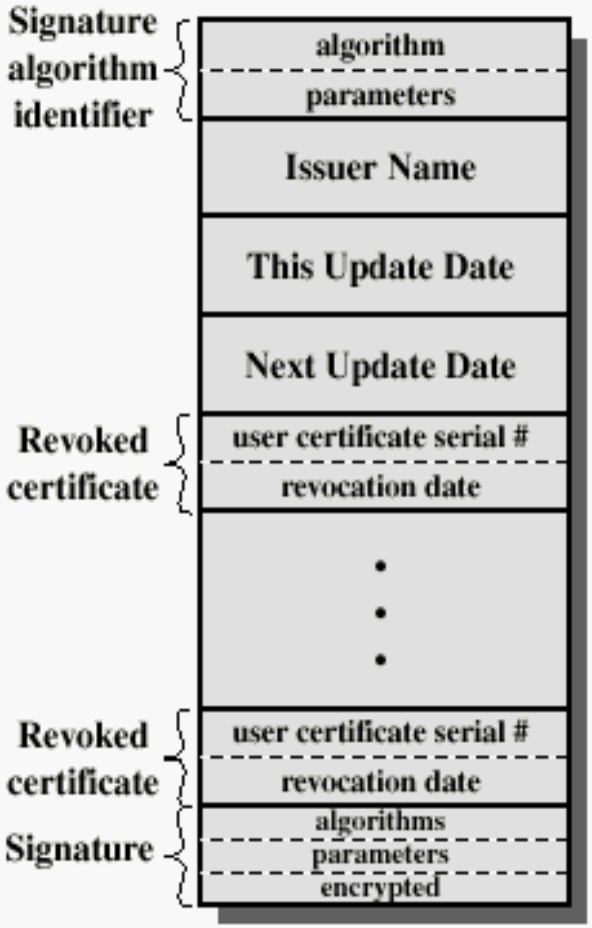
X.509 Authentication Service

- 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

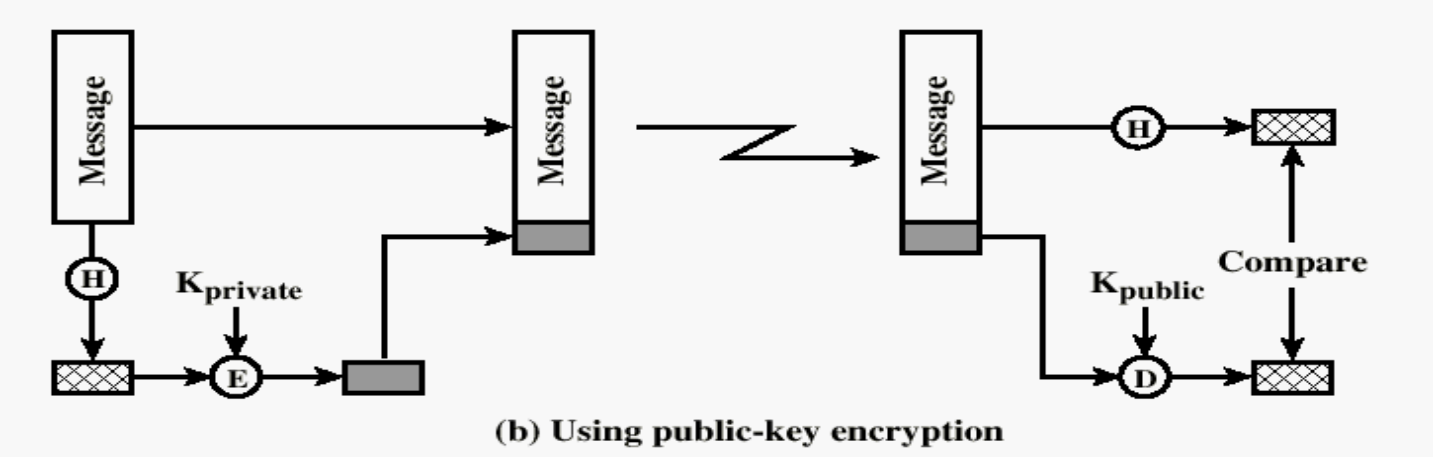


(a) X.509 Certificate



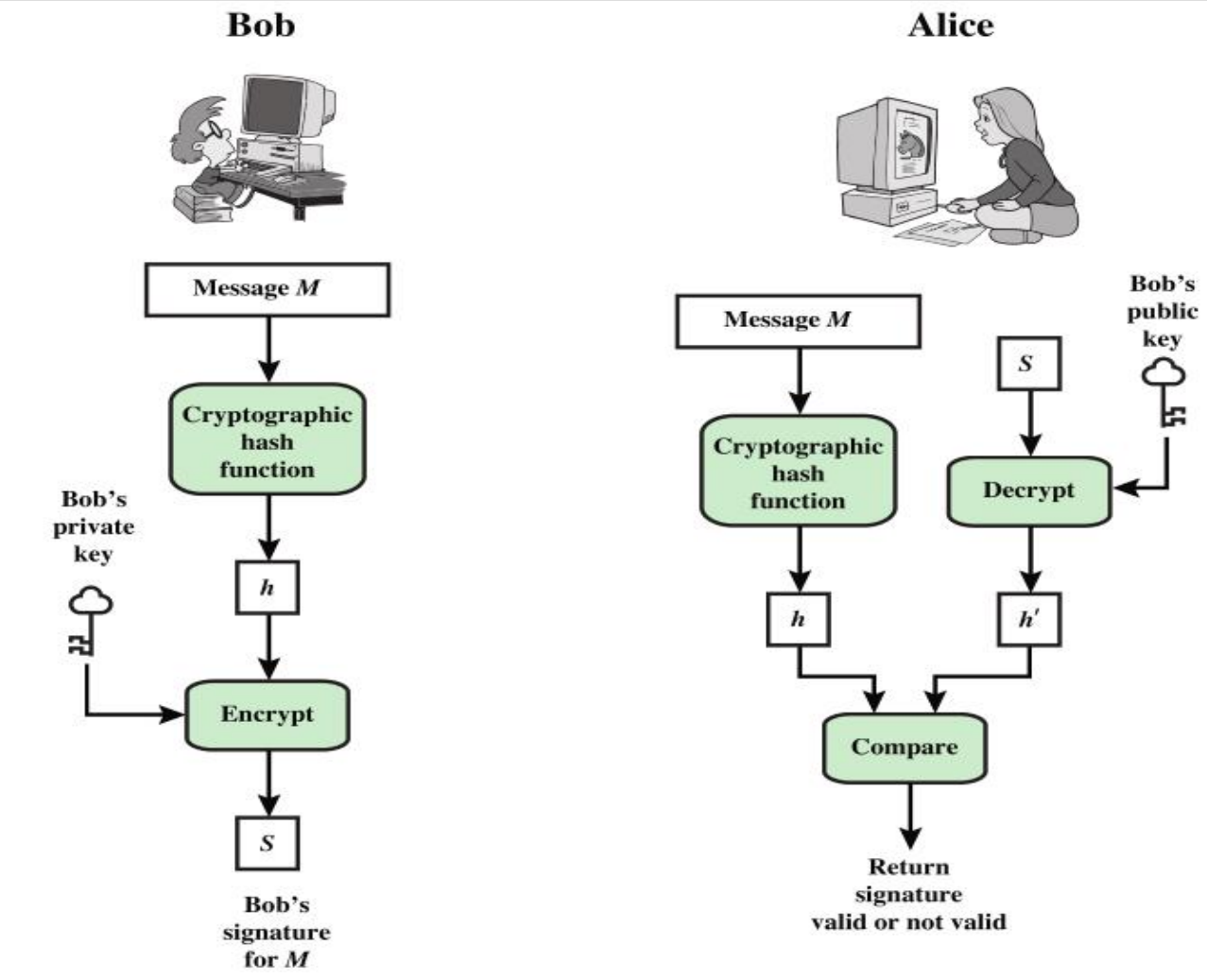
(b) Certificate Revocation List

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 <<X>>
  - 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 <<A>> ? X2 <<B>>
- Solution:
  - Two CAs securely exchange public keys:
    - X1 <<X2>>
    - X2 <<X1>>
  - Now: How can A acquire B's public key?

X.509 CA Hierarchy

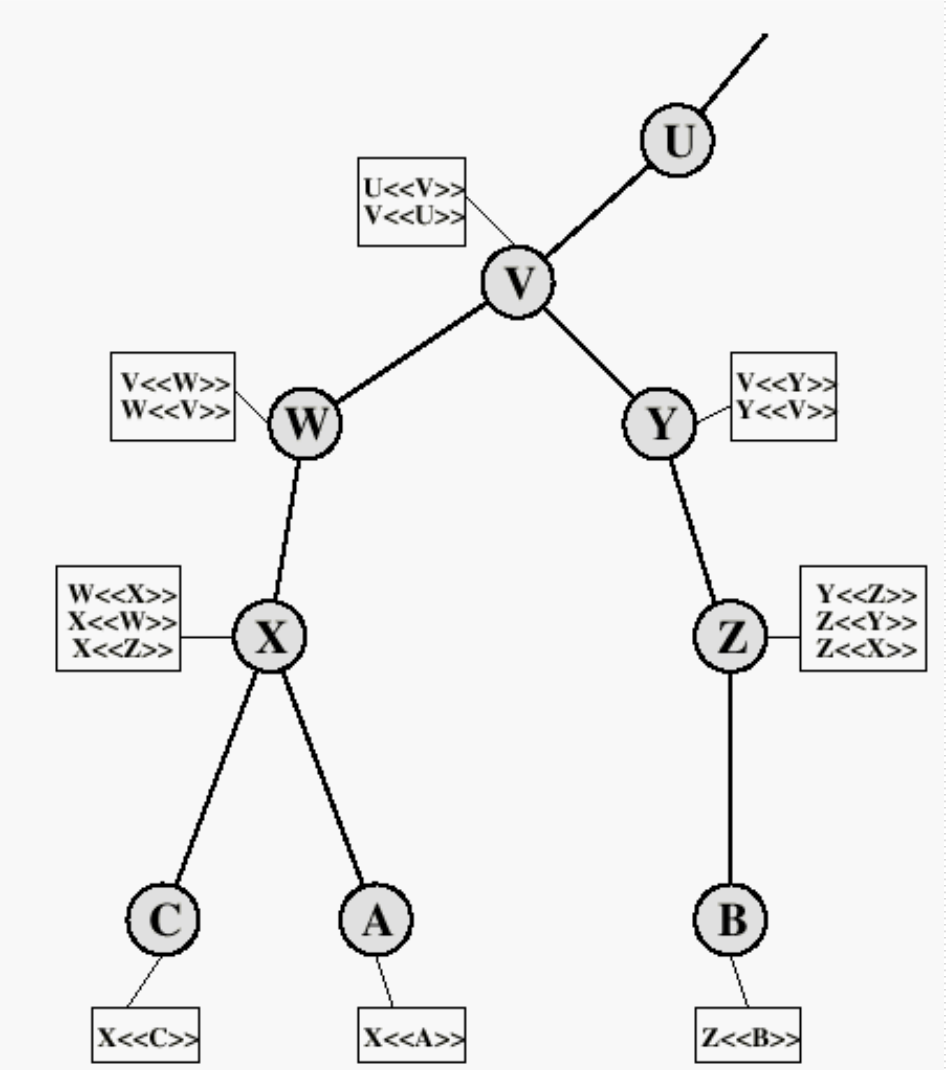
- Two types of certificates:
  - Forward certificates: Certificates of X generated by other CAs,
  - Reverse certificates: Certificates generated by X that are the certificates of other CAs.

- A acquires B certificate using chain:

X<<W>>, W<<V>>, V<<Y>>, Y<<Z>>, Z<<B>>

- B acquires A certificate using chain:

Z<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>



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.

Public Key Infrastructure (PKI)

- RFC 4949 (Internet Security Glossary) defines public-key infrastructure (PKI) as:
  - The set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.
- Objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys.

Public Key Infrastructure (PKI)

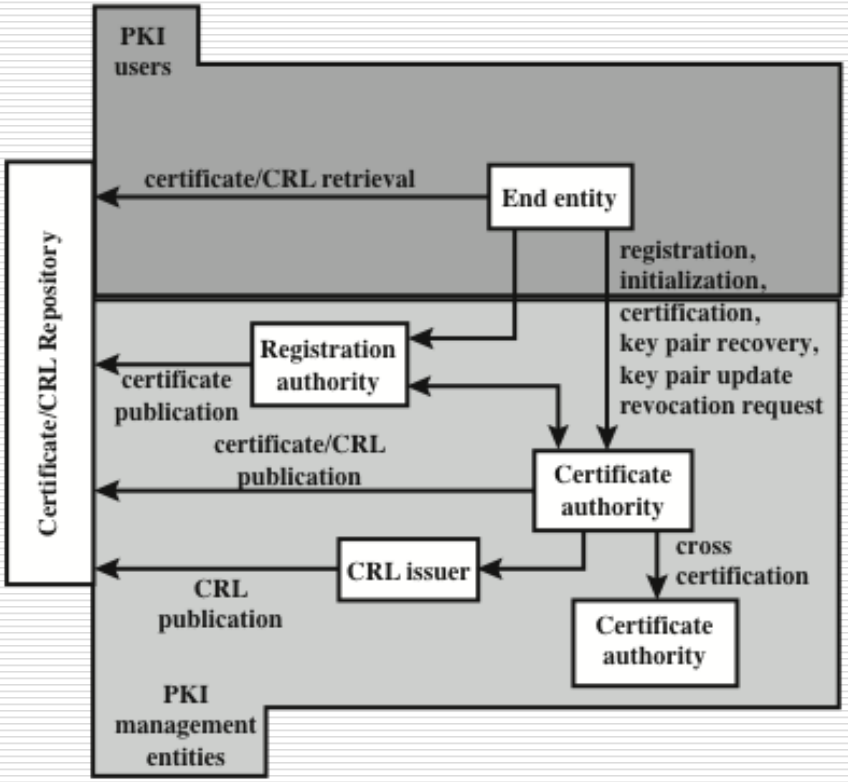


Figure 4.7 PKIX Architectural Model

Identity Management System

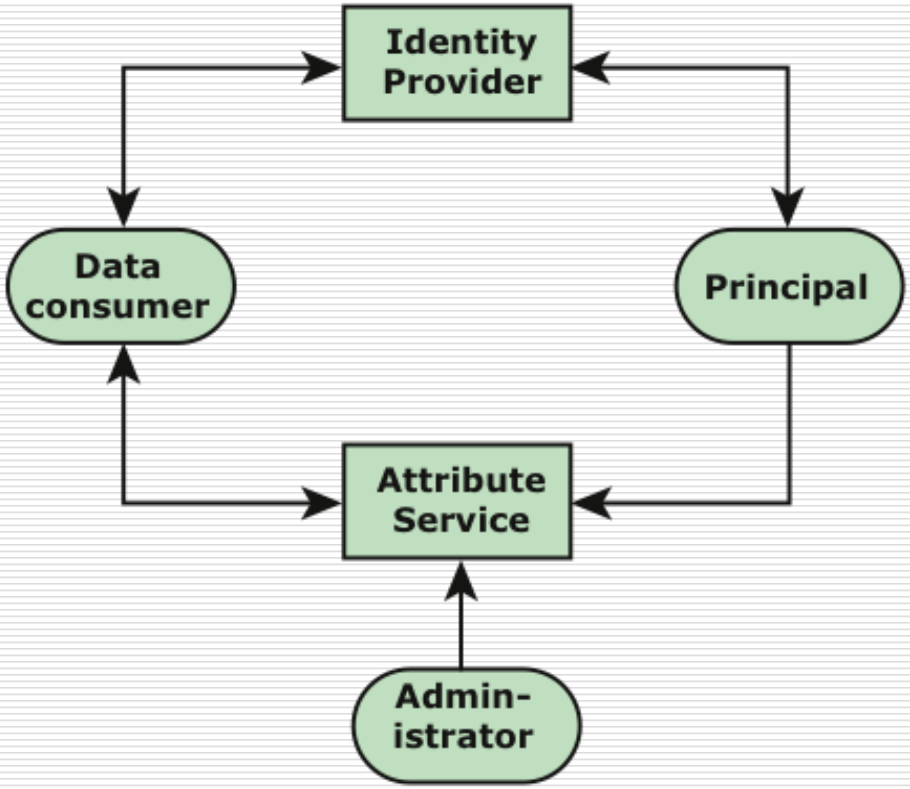


Figure 4.8 Generic Identity Management System

PKIX Management functions

- Functions that potentially need to be supported by management protocols:
  - Registration
  - Initialization
  - Certification
  - Key pair recovery
  - Key pair update
  - Revocation request
  - Cross certification
- Alternative management protocols:
  - Certificate management protocols (CMP)
    - Designed to be a flexible protocol able to accommodate a variety of technical, operational, and business models
  - Certificate management messages over CMS (CMC)
    - Is built on earlier work and is intended to leverage existing implementations

IDENTITY MANAGEMENT

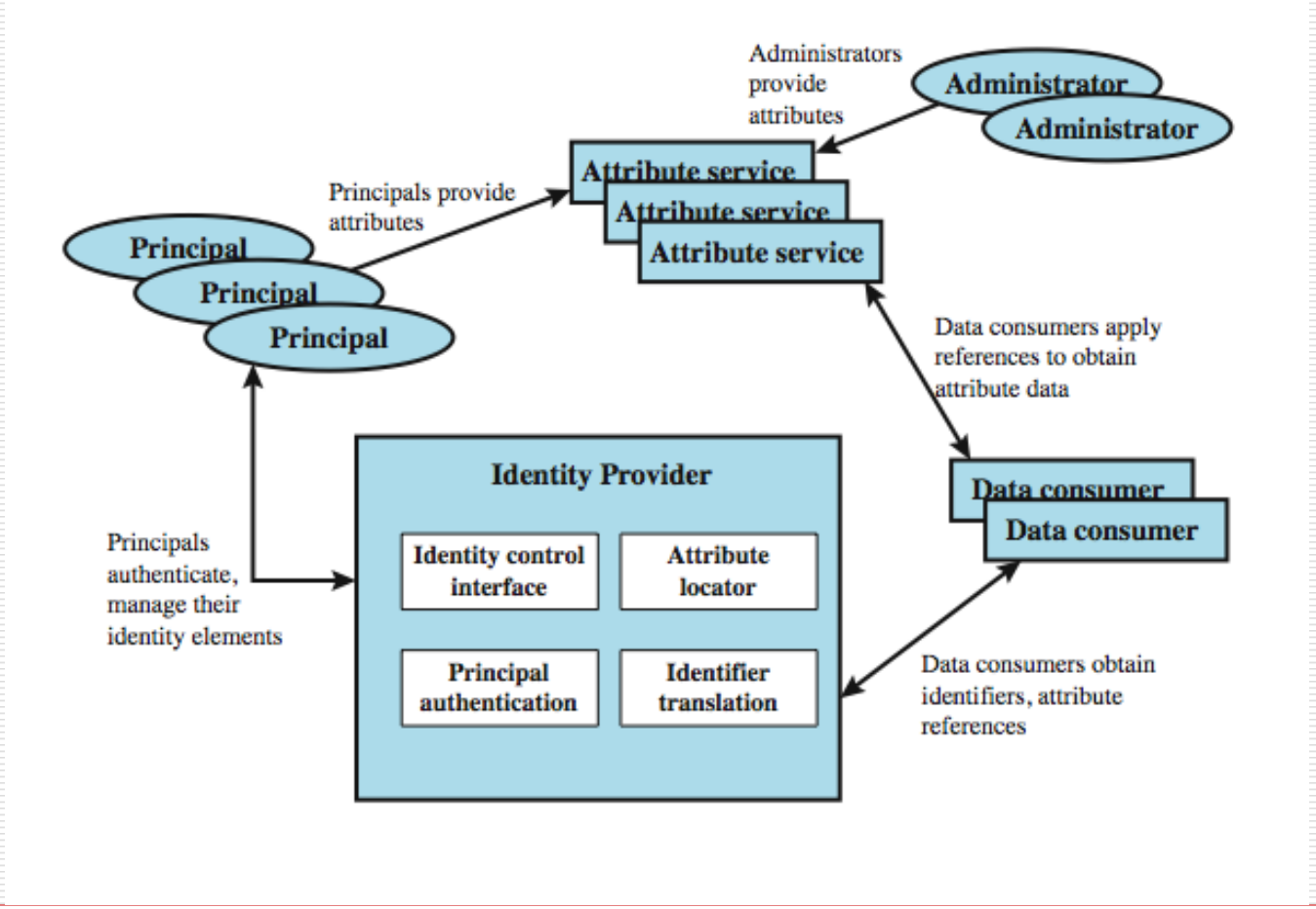
Federated Identity Management

- Federated identity management:
  - dealing with the use of a common identity management scheme across multiple enterprises and numerous applications
- use of common identity management scheme
  - across multiple enterprises & numerous applications
  - supporting many thousands, even millions of users
- Kerberos contains many of these elements

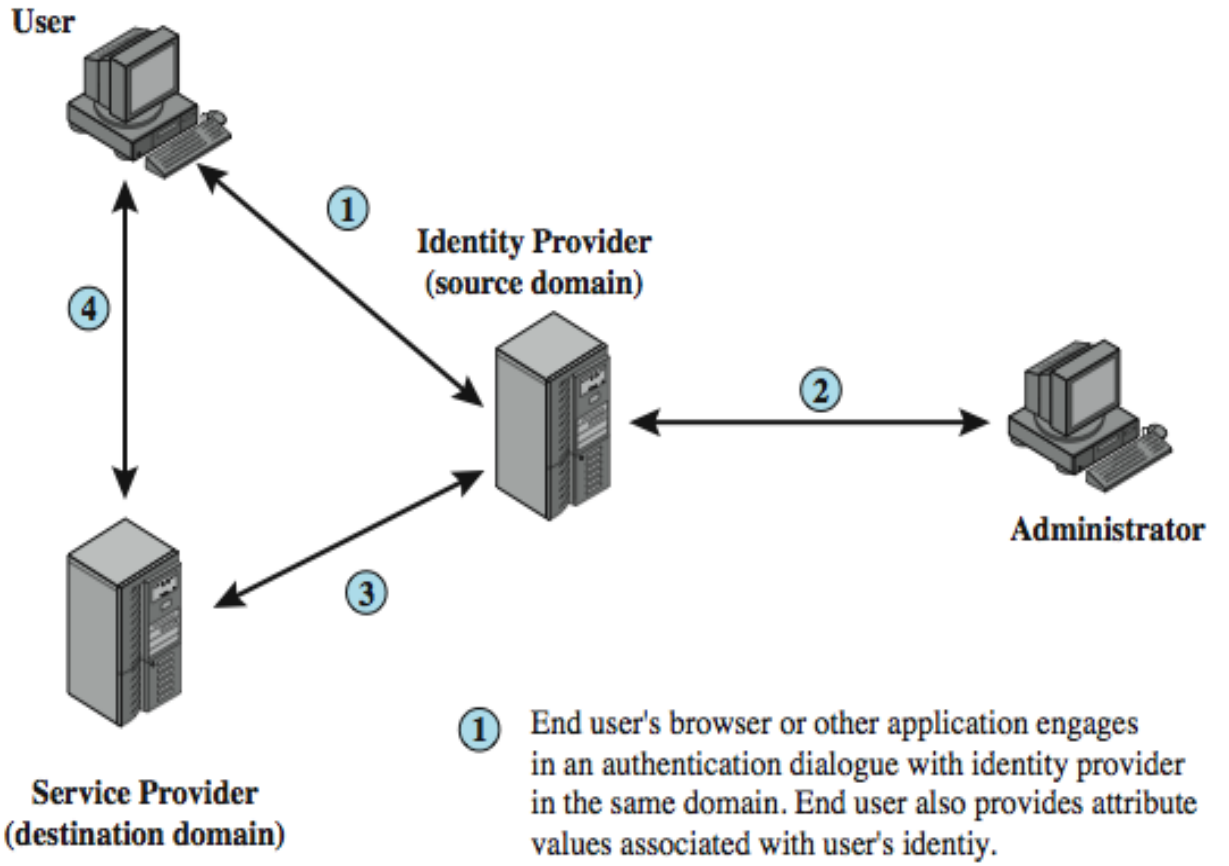
Federated Identity Management

- Principal elements are:
  - Authentication: confirmating user corresponds to the user name provided.
  - Authorization: granting access to services/resources given user authentication.
  - Accounting: process for logging access and authorization.
  - Provisioning: enrollment of users in the system.
  - Workflow automation: movement of data in a business process.
  - Delegated administration: use of role-based access control to grant permissions.
  - Password synchronization: Creating a process for single sign-on (SSO) or reduced sign-on (RSO).
  - Self-service password reset: enable user to modify their password
  - Federation: process where authentication and permission will be passed on from one system to another, usually across multiple enterprises, reducing the number of authentications needed by the user.
  - Kerberos contains a number of the elements of an identity management system.

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 identiy.
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

The Extensible Markup Language (XML)

- Appear similar to HTML documents that are visible as Web pages, but provide greater functionality
- Includes strict definitions of the data type of each field
- Provides encoding rules for commands that are used to transfer and update data objects

The Simple Object Access Protocol (SOAP)

- Minimal set of conventions for invoking code using XML over HTTP
- Enables applications to request services from one another with XML-based requests and receive responses as data formatted with XML

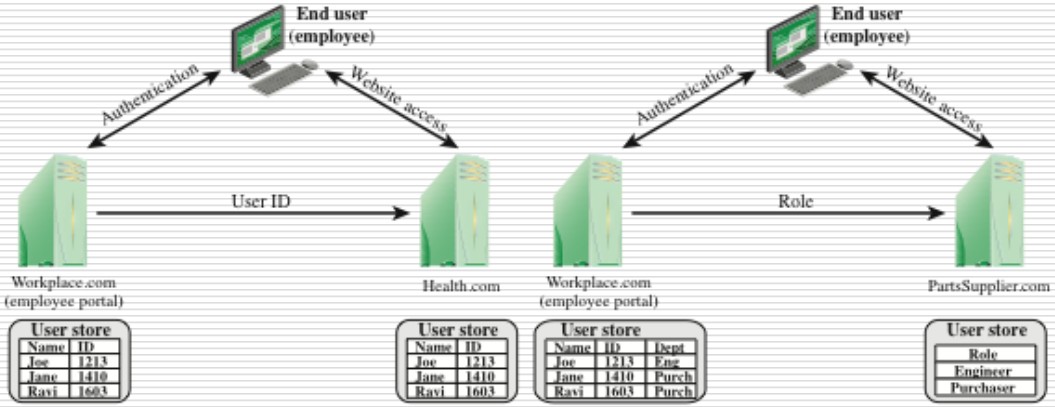
WS-Security

- A set of SOAP extensions for implementing message integrity and confidentiality in Web services
- Assigns security tokens to each message for use in authentication

Security Assertion Markup Language (SAML)

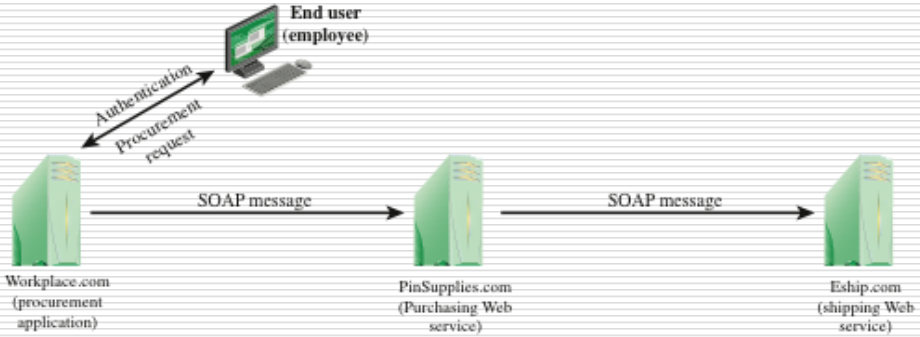
- An XML-based language for the exchange of security information between online business partners
- Conveys authentication information in the form of assertions about subjects

Federated Identity Examples



(a) Federation based on account linking

(b) Federation based on roles



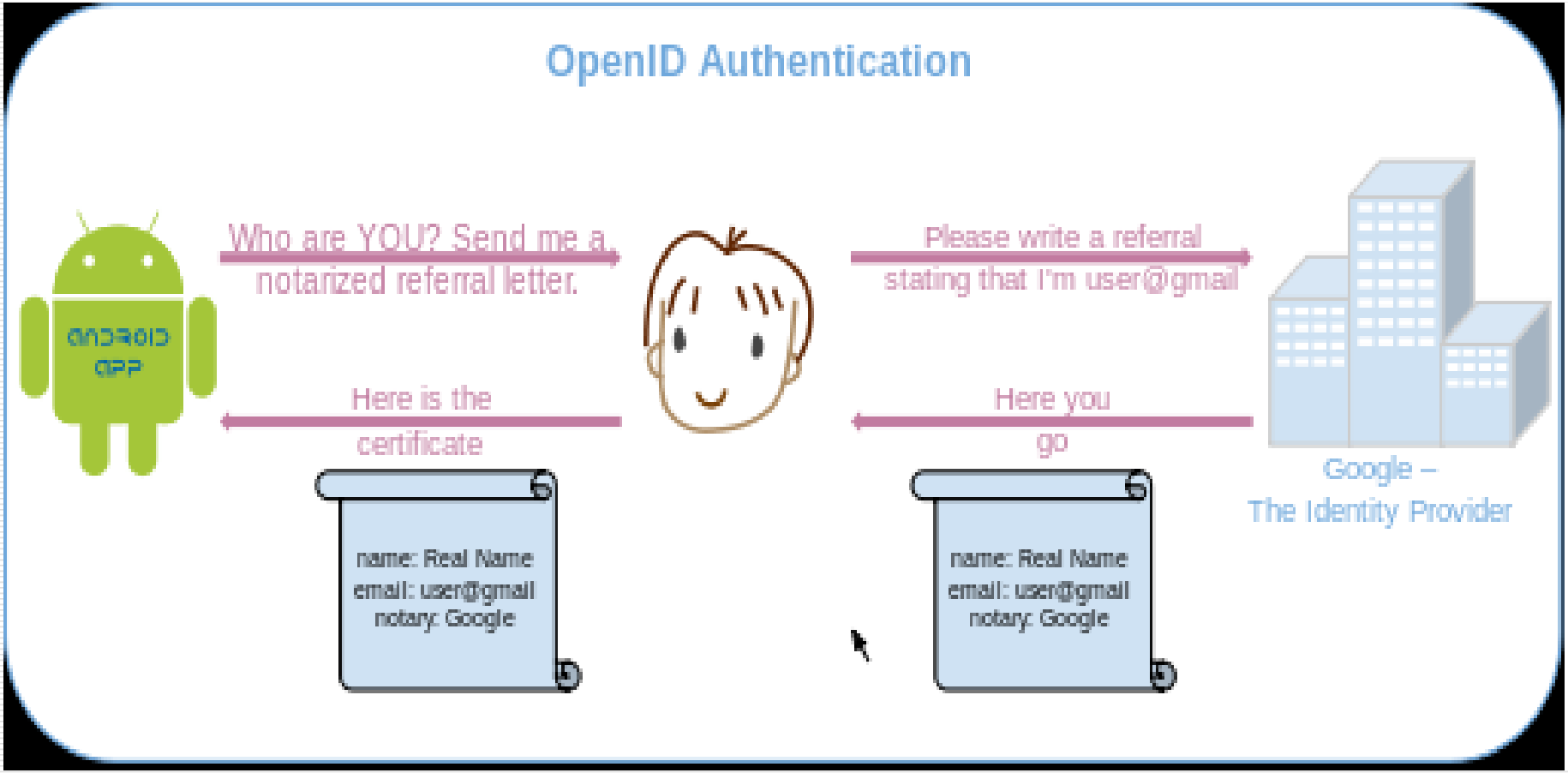
(c) Chained Web Services

Figure 4.10 Federated Identity Scenarios

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
android app
Who are YOU? Send me a notarized referral letter.
Here is the certificate
name: Real Name
email: user@gmail
notary: Google
Please write a referral stating that I'm user@gmail
Here you go
name: Real Name
email: user@gmail
notary: Google
Google –
The Identity Provider

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	عناصری 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.