Unit 3

User Authentication Protocols

User Authentication Protocols

- Remote User Authentication Principles
- Remote User Authentication Using Symmetric Encryption
- Kerberos
- Remote User Authentication Using Asymmetric Encryption
- Federated Identity Management

Unit 3 - Key Points

Mutual authentication protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

Kerberos is an authentication service designed for use in a distributed environment.

Kerberos provides a trusted third-party authentication service that enables clients and servers to establish authenticated communication.

Identity management is a centralized, automated approach to provide enterprise-wide access to resources by employees and other authorized individuals.

Identity federation is, in essence, an extension of identity management to multiple security domains.

3.1 Remote User Authentication Principles

- Mutual Authentication
- One-Way Authentication

3.1.1 Mutual Authenentication

Such protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

2 Steps of User Authenentication

- Identification Step
- Verification Step

Four general means of authenticating a user's identity

1. Something the individual knows
    - Password || PINS
2. Something the individual possesses
    - TOKENs
3. Something the individual is
    - Static biometrics - fingerprint || retina || face
4. Something the individual does
    - Dynamic biometrics - Voice Pattern || Handwriting characteristics

Challenges in Mutual Authentication

- **Confidentiality**
  - Masquerade
  - Compromization of Session Keys
  - Prior existence of secret or public keys
- **Timeliness**
  - Replays

4 Types of Replay Attacks

- Simple Replay
- Repetion that can be logged
- Repetition that cannot be detected
- Backward replay without modification

Mitigating Replay Attacks

- Sequence Numbers
- Timestamps
- Challenge / Response

3.1.2 One-Way Authenentication

3.2 Remote User Authentication Using Symmetric Encryption

- Mutual Authentication
- One-Way Authentication

3.2.1 Mutual Authentication

- [NEED78]
- [DENN81, DENN82]
- [KEHN92]

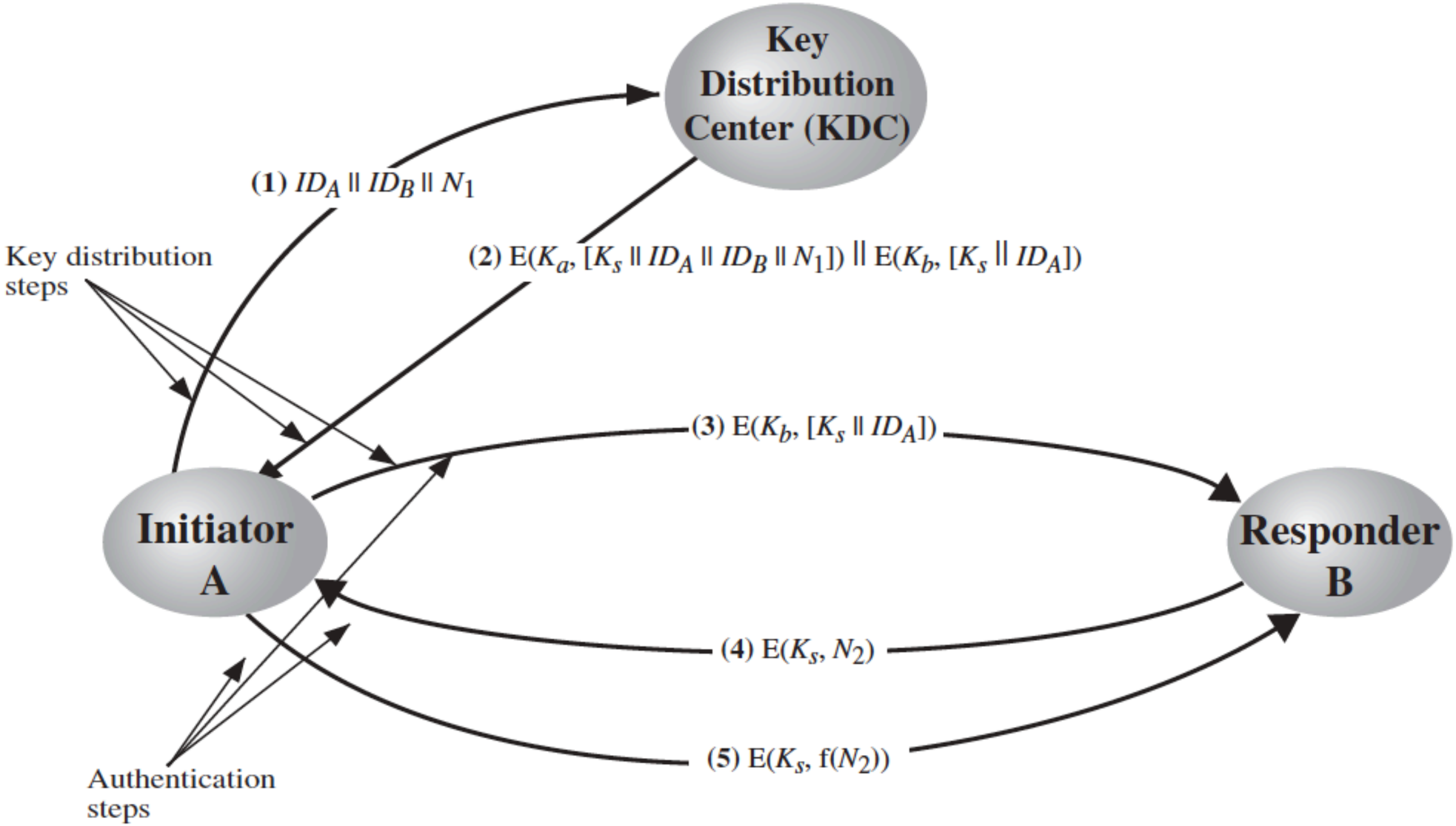


Figure 14.3 Key Distribution Scenario

[NEED78]

1. $\mathrm{A} \rightarrow \mathrm{B}$: $ID_A \| N_a$
2. $\mathrm{B} \rightarrow \mathrm{KDC}$: $ID_B \| N_b \| \mathrm{E}(\mathrm{K}_b, [ID_A \| N_a \| T_b])$
3. $\mathrm{KDC} \rightarrow \mathrm{A}$: $\mathrm{E}(K_a, [ID_B \| N_a \| K_s \| T_b]) \| \mathrm{E}(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $\mathrm{A} \rightarrow \mathrm{B}$: $\mathrm{E}(K_b, [ID_A \| K_s \| T_b]) \| \mathrm{E}(K_s, N_b)$

[DENN81, DENN82]

1. $\mathrm{A} \rightarrow \mathrm{KDC}: \quad ID_A \| ID_B$
2. $\mathrm{KDC} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_a, [K_s \| ID_B \| T \| \mathrm{E}(K_b, [K_s \| ID_A \| T])])$
3. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_b, [K_s \| ID_A \| T])$
4. $\mathrm{B} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_s, \mathrm{N}_1)$
5. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_s, \mathrm{f}(N_1))$

[KEHN92]

1. $\mathrm{A} \rightarrow \mathrm{KDC}: \quad ID_A \| ID_B \| N_1$
2. $\mathrm{KDC} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_a, [K_s \| ID_B \| N_1 \| \mathrm{E}(K_b, [K_s \| ID_A])])$
3. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_b, [K_s \| ID_A])$
4. $\mathrm{B} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_s, N_2)$
5. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_s, \mathrm{f}(N_2))$

3.2.2 One-Way Authentication

1. $A \rightarrow KDC: \quad ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A]) \| E(K_s, M)$

3.3 Kerberos

Versions

Kerberos v4
Kerberos v5

Kerberos V4

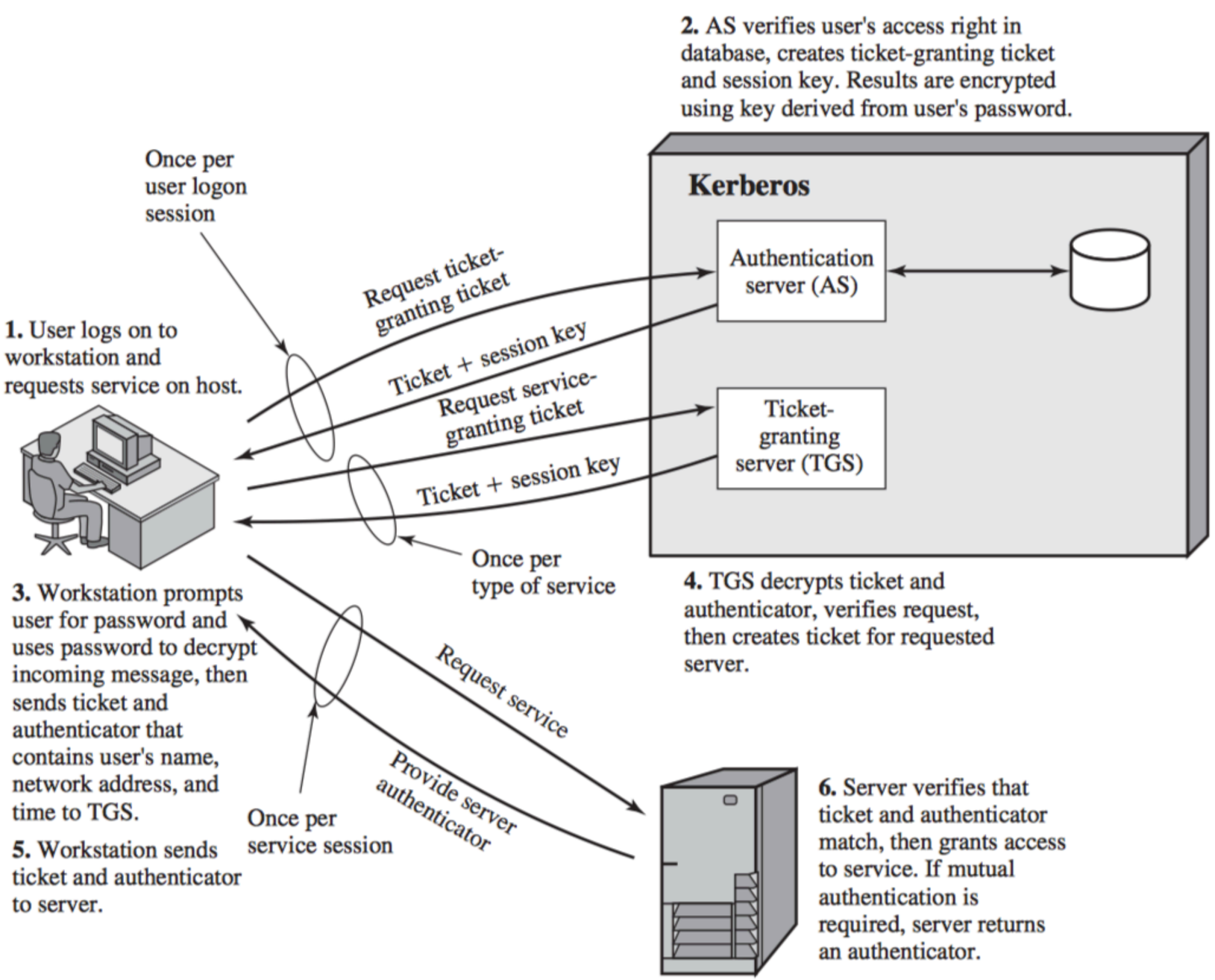


Figure 15.1 Overview of Kerberos

Table 15.1 Summary of Kerberos Version 4 Message Exchanges

(1) $C \rightarrow AS \quad ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

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

(3) $C \rightarrow TGS \quad ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C \quad E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

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

(5) $C \rightarrow V \quad Ticket_v \| Authenticator_c$

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

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

Table 15.2 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

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.

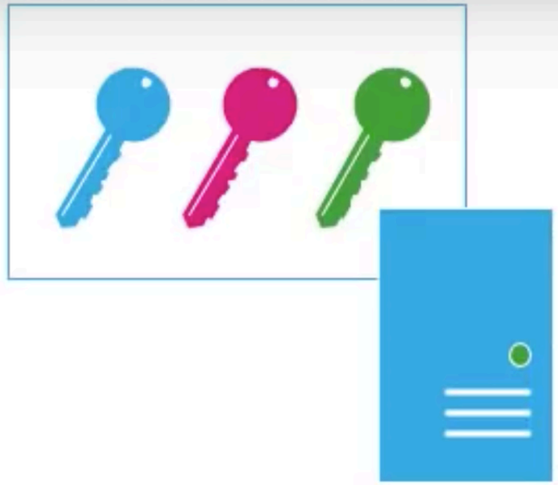
$K_{c,tgs}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_C	Must match address in ticket to authenticate ticket.
TS_3	Informs TGS of time this authenticator was generated.

(b) Ticket-Granting Service Exchange

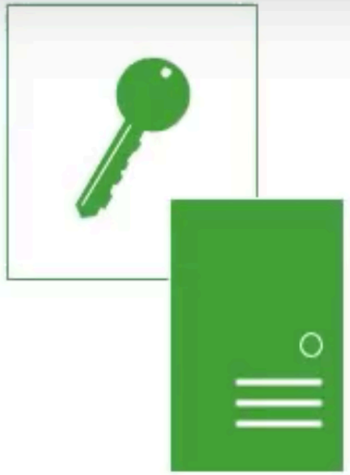
Message (5)	Client requests service.
$Ticket_V$	Assures server that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (6)	Optional authentication of server to client.
$K_{c,v}$	Assures C that this message is from V.
$TS_5 + 1$	Assures C that this 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

Step by step



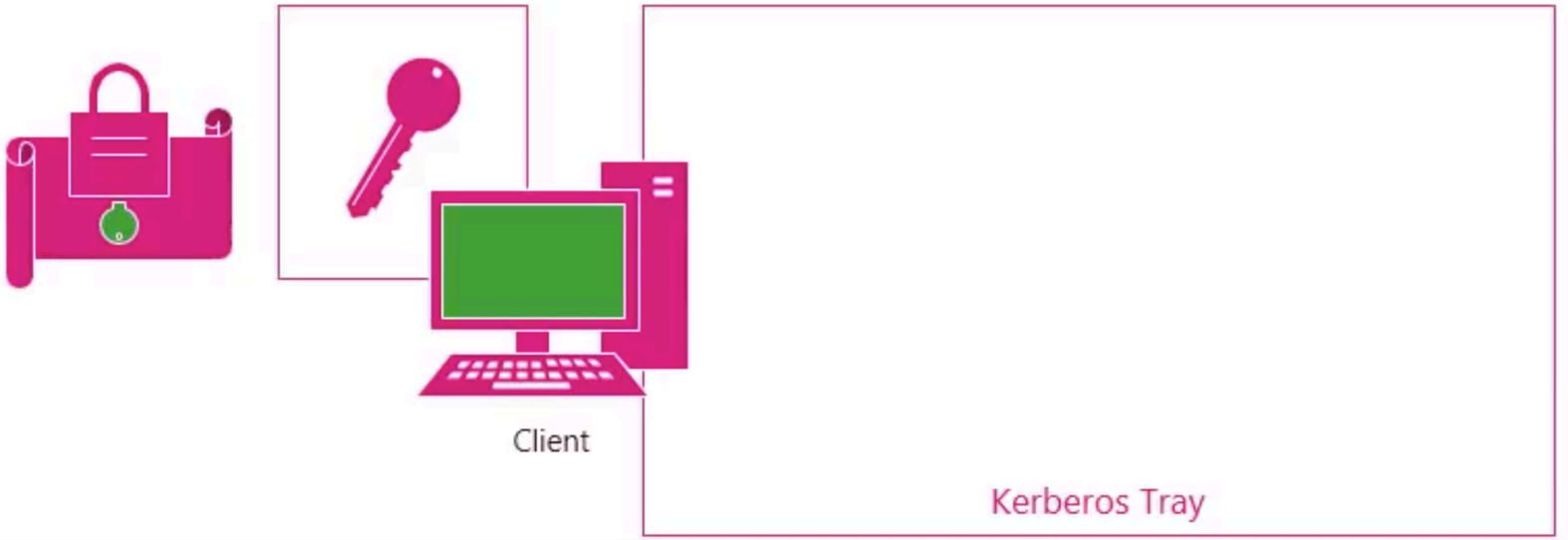
Key Distribution Center
(KDC – Domain Controller)



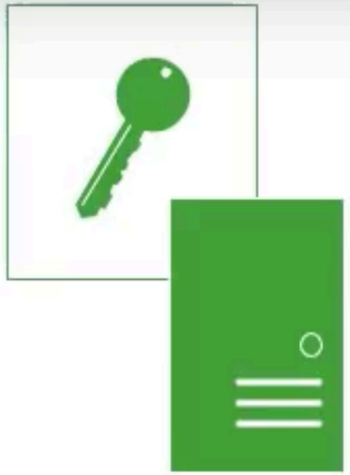
File Server

Key Distribution Center
(KDC – Domain Controller)

File Server




Key Distribution Center
(KDC – Domain Controller)


File Server

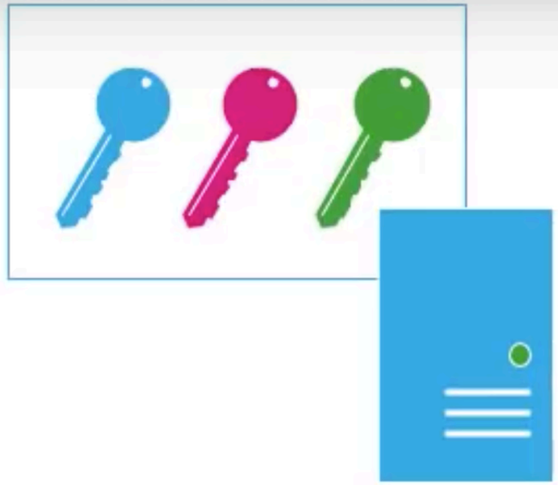

Client
Kerberos Tray

Key Distribution Center
(KDC – Domain Controller)

File Server

Client

Kerberos Tray



Key Distribution Center
(KDC – Domain Controller)

File Server


Key Distribution Center
(KDC – Domain Controller)


File Server

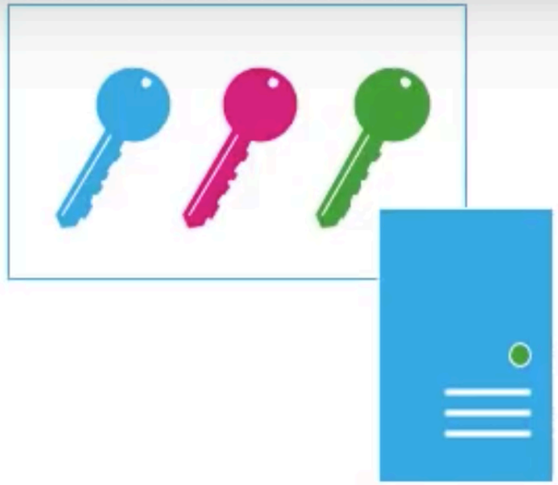

Client
Kerberos Tray

Key Distribution Center
(KDC – Domain Controller)

File Server

Client

Kerberos Tray



Key Distribution Center
(KDC – Domain Controller)

File Server

Key Distribution Center
(KDC – Domain Controller)

File Server

Client

Kerberos Tray

Shortcomings of Kerberos V4

- Environment Shortcomings
- Technical Deficiencies

Environment Shortcomings

1. Encryption system dependence
2. Internet protocol dependence
3. Message Byte Ordering
4. Ticket Lifetime
5. Authentication Forwarding
6. Inter-Realm Authentication

Encryption system dependence

- DES Dependency of v4

Internet protocol dependence

- IP Protocol Addresses only

Message Byte Ordering

- Did not follow convention of byte ordering/ was ambigious
- ASN.1 - Abstract Syntax Notation One
- BER - Basic Encoding Rules

Ticket Lifetime

- 8 bit life time
- Unit of 5 mins
- total (2^8)*5 = 1280 mins ~= 21 hours
- Explicit start and end time in v5

Authentication Forwarding

- No forwarding of credentials
- Example – Printing a File on a network

Inter-Realm Authentication

- Lack of interoperability
- N Realms = (N^2) Kerberos-to-kerberos relationships

Technical Deficiencies

1. Double Encryption
2. PCBC Encryption
3. Session Key
4. Password Attacks

Double Encryption

- Redundant Double encryption
- Removed in v5

PCBC Encryption mode in DES

- Propagating Cipher Block Chaining
- Non-standard

Session Key

- Possible threat of replay attack
- Use of sub-session key between client and server

Password Attacks

- Vulnerable to password attack
- Bruteforce or dictionary attacks

Kerberos V5

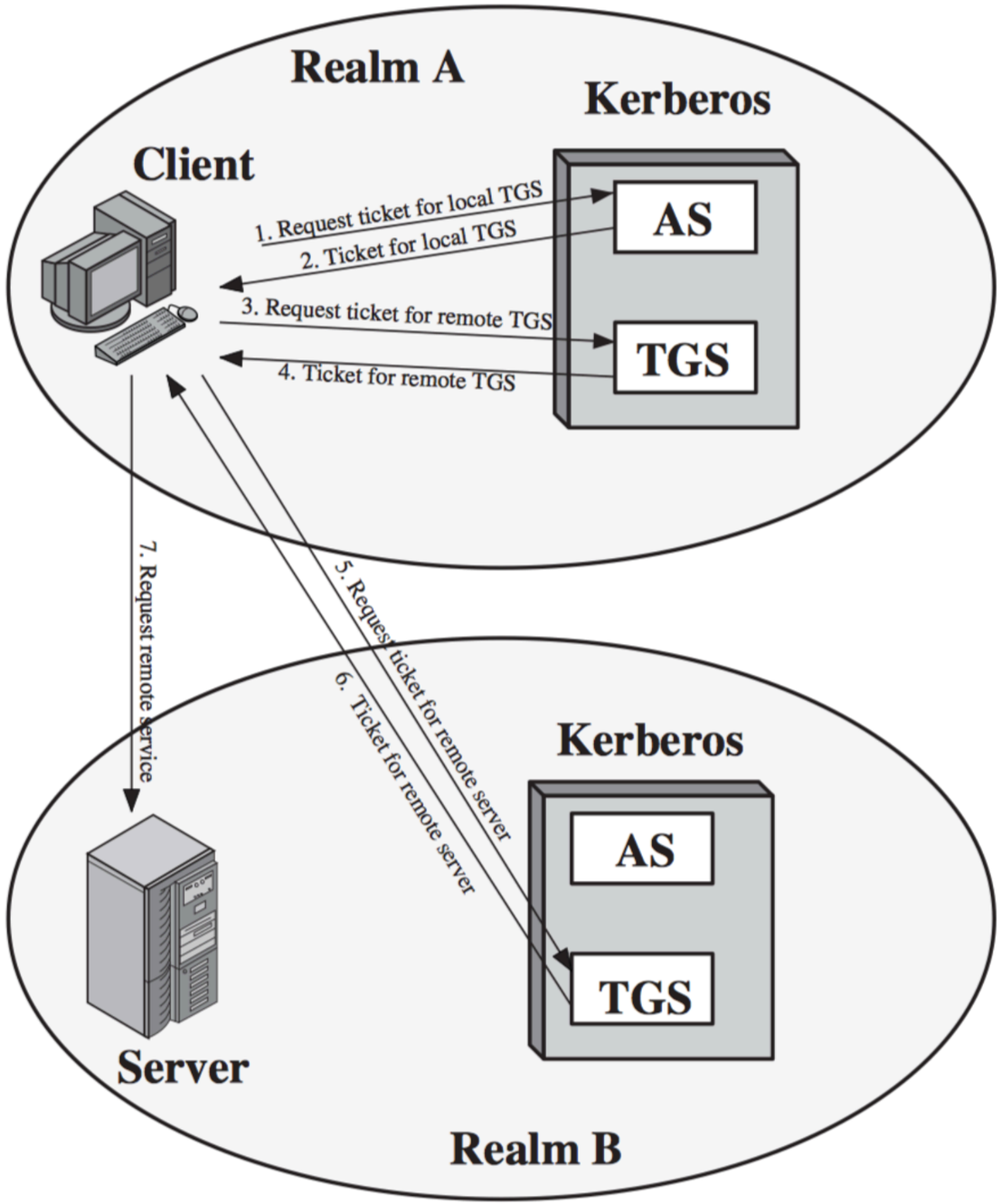


Figure 15.2 Request for Service in Another Realm

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

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

(3) $C \rightarrow TGS$: $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C$: $E(K_{c,tgs}, [K_{c,tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}])$

(5) $C \rightarrow TGS_{rem}$: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$

(6) $TGS_{rem} \rightarrow C$: $E(K_{c,tgsrem}, [K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}])$

(7) $C \rightarrow V_{rem}$: $Ticket_{vrem} \| Authenticator_c$

Table 15.3 Summary of Kerberos Version 5 Message Exchanges

(1)	C → AS	$\text{Options} \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$
(2)	AS → C	$Realm_C \| ID_C \| Ticket_{tgs} \| \text{E}(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$
		$Ticket_{tgs} = \text{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

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

(3)	C → TGS	$\text{Options} \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$
(4)	TGS → C	$Realm_c \| ID_C \| Ticket_v \| \text{E}(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$
		$Ticket_{tgs} = \text{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$
		$Ticket_v = \text{E}(\text{K}_v, [\text{Flags} \| \text{K}_{c,v} \| \text{Realm}_c \| \text{ID}_C \| \text{AD}_C \| \text{Times}])$
		$Authenticator_c = \text{E}(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$

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

(5)	C → V	$\text{Options} \| \text{Ticket}_v \| \text{Authenticator}_c$
(6)	V → C	$\text{E}_{K_{c,v}}[\text{TS}_2 \| \text{Subkey} \| \text{Seq}\neq]$
		$\text{Ticket}_v = \text{E}(\text{K}_v, [\text{Flag} \| \text{K}_{c,v} \| \text{Realm}_c \| \text{ID}_C \| \text{AD}_C \| \text{Times}])$
		$Authenticator_c = \text{E}(K_{c,v}, [ID_C \| Relam_c \| TS_2 \| Subkey \| Seq\neq])$

(c) Client/Server Authentication Exchange to obtain service

https://goo.gl/v7HTXh

Remote User Authentication Using Asymmetric Encryption

- Mutual Authentication
- One-Way Authentication

A protocol using timestamps is provided in [DENN81]:

1. $A \rightarrow AS: \quad ID_A \| ID_B$
2. $AS \rightarrow A: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$
3. $A \rightarrow B: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

Asymmetric

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_A \| ID_B) \| N_b]))$
7. $A \rightarrow B: \quad E(K_s, N_b)$

Symmetric [KEHN92]

1. $\mathrm{A} \rightarrow \mathrm{KDC}: \quad ID_A \| ID_B \| N_1$
2. $\mathrm{KDC} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_a, [K_s \| ID_B \| N_1 \| \mathrm{E}(K_b, [K_s \| ID_A])])$
3. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_b, [K_s \| ID_A])$
4. $\mathrm{B} \rightarrow \mathrm{A}: \quad \mathrm{E}(K_s, N_2)$
5. $\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(K_s, \mathrm{f}(N_2))$

Federated Identity Management

- Identity Management
- Identity Federation

Federated identity management is a relatively new concept dealing with the use of a common identity management scheme across multiple enterprises and numerous applications and supporting many thousands, even millions, of users.

Identity Management

Identity management is a centralized, automated approach to provide enterprisewide access to resources by employees and other authorized individuals.

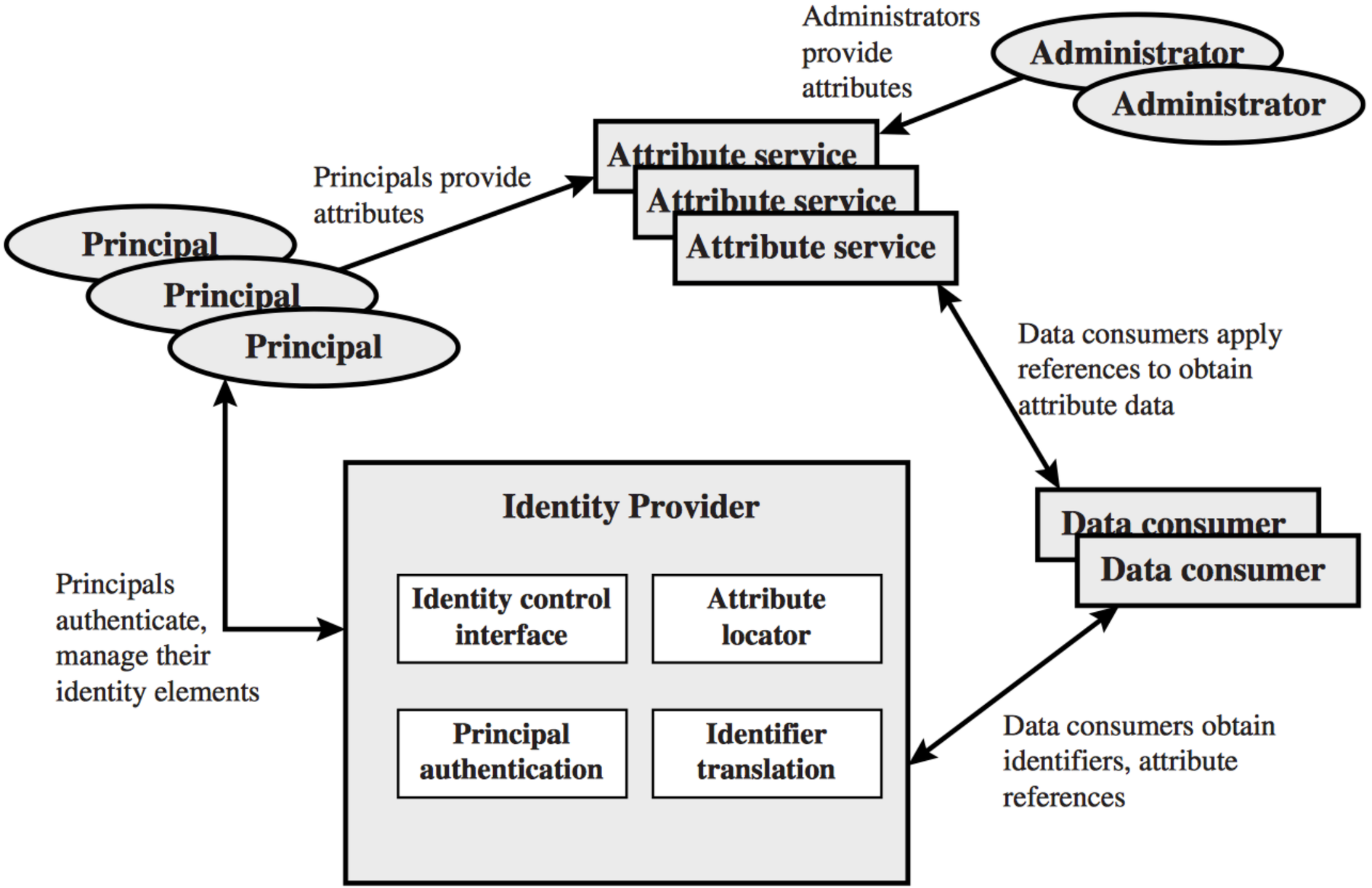


Figure 15.3 Generic Identity Management Architecture

Principal Elements of Identity Management System

- Authentication
- Authorization
- Accounting
- Provisioning
- Workflow Automation
- Delegated Administration
- Password Synchronization
- Self-service Password Reset
- Federation

Identity Federation

Identity federation is, in essence, an extension of identity management to multiple security domains. Such domains include autonomous internal business units, external business partners, and other third-party applications and services.

The goal is to provide the sharing of digital identities so that a user can be authenticated a single time and then access applications and resources across multiple domains.