Fundamentals of Information & Network Security ECE 471/571

Lecture #32, 33: Kerberos

Instructor: Ming Li

Dept of Electrical and Computer Engineering

University of Arizona

What is Kerberos?

- Network authentication protocol
- providing strong authentication for client/server applications, using secret-key cryptography.
- A user typed in a password and logged into a workstation. On behalf of the user, the workstation authenticates and accesses resources seamlessly.
- Developed at MIT
- Kerberos V4 and V5 are

widely deployed

- KDC, a database of

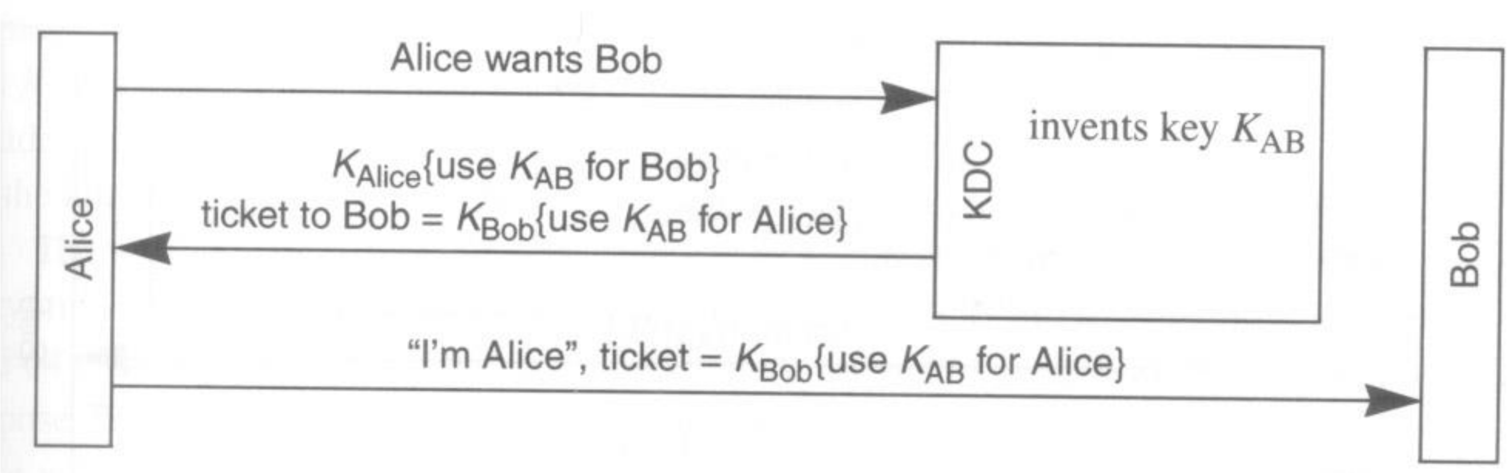
  <principal, key> and

  a library of subroutines

Review: Key Distribution Center (KDC)

- Let K_A be the master key of Alice and K_B the master key of Bob.
- When Alice needs to talk with Bob, she informs KDC, which selects a session key K_{AB} and sends Alice

$$K_A\{K_{AB}, K_B\{\text{Alice}, K_{AB}, \ldots\}\}$$


Alice
Alice wants Bob
KDC
invents key K_{AB}
K_{Alice}\{use K_{AB} for Bob\}
ticket to Bob = K_{Bob}\{use K_{AB} for Alice\}
"I'm Alice", ticket = K_{Bob}\{use K_{AB} for Alice\}
Bob

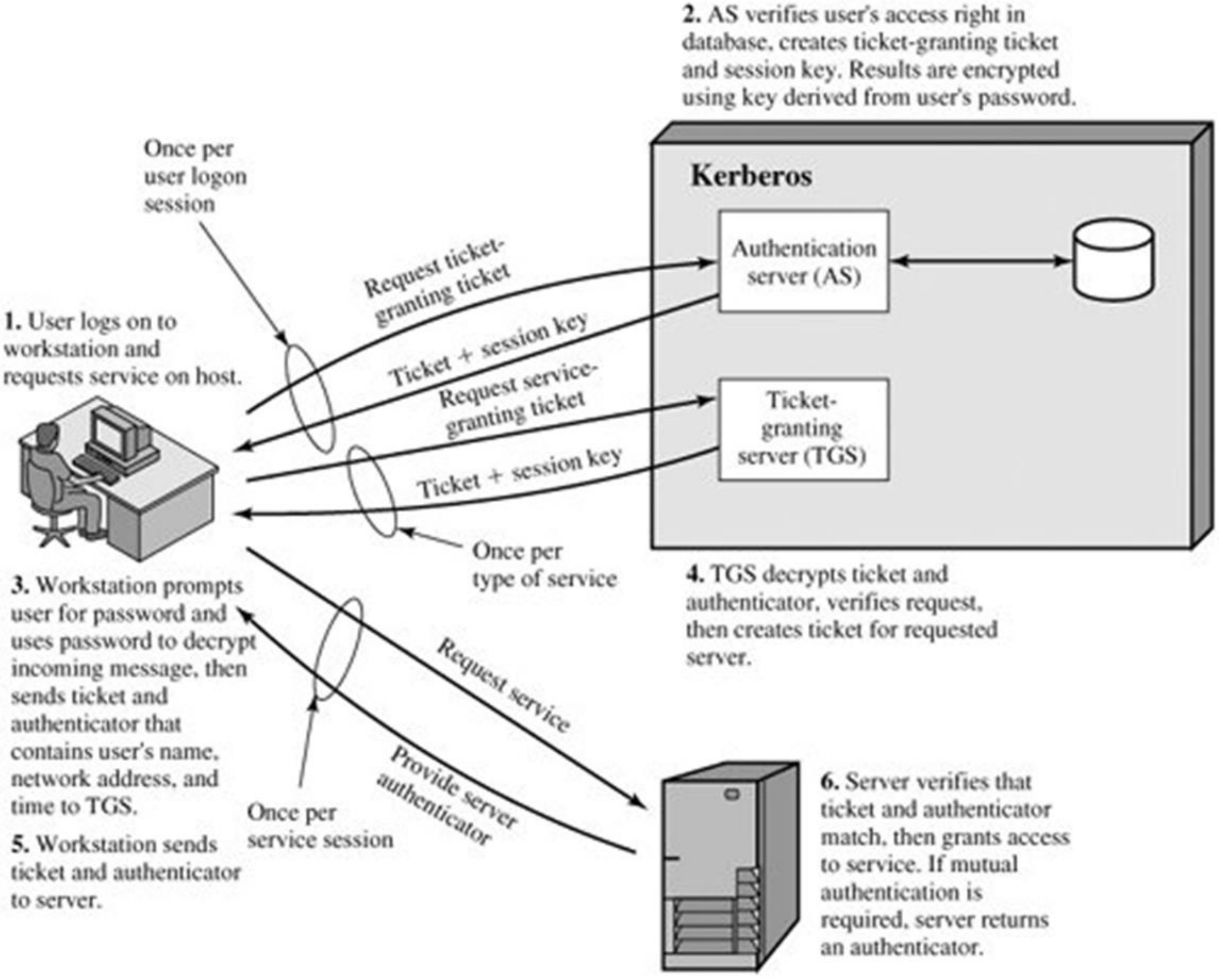
Protocol 11-17. KDC operation (in practice)

Review: Key Distribution Center (KDC)

- K_B\{Alice, K_{AB}, ...\} are called Alice's ticket to Bob
- K_{AB} and K_B\{Alice, K_{AB}, ...\} are called Alice's credential to Bob.
- Alice remembers a password and K_A is a DES key. To bridge the difference, a hash algorithm may be used to convert a password to a key.

Overview of Kerberos

- Purpose: authentication in distributed systems



1. User logs on to workstation and requests service on host.

2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.

5. Workstation sends ticket and authenticator to server.

6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Kerberos
Authentication server (AS)
Ticket-granting server (TGS)
Once per user logon session
Request ticket-granting ticket
Ticket + session key
Request service-granting ticket
Ticket + session key
Once per type of service
Request service
Provide server authenticator
Once per service session

Configuration

- Kerberos server: KDC
- Each principal has its master key, shared with KDC.
  - Human user: derived from password
  - Machine: pre-configured
- KDC has a master key, known only by itself.
- KDC keeps a database of <principal, key>
- Based on secret-key cryptography - DES. V5 theoretically can use other encryption algorithms.

Session key

- Login session
- Problem
  - K_A is the long-term authentication key, should the workstation remember K_A for the whole login session?
- Solution: Session key
  - Instead of letting the workstation to keep K_A for the entire session, it is more secure to use K_A only at the beginning to negotiate a session key S_A for the entire login session.

Ticket-Granting Ticket

- When Alice logs on, KDC sends the workstation

  $K_A\{S_A, K_{KDC}\{Alice, S_A, \ldots\}\}$,

  where K_{KDC} is the master key of KDC.
- $K_{KDC}\{Alice, S_A, \ldots\}$ is called a ticket-granting ticket (TGT).

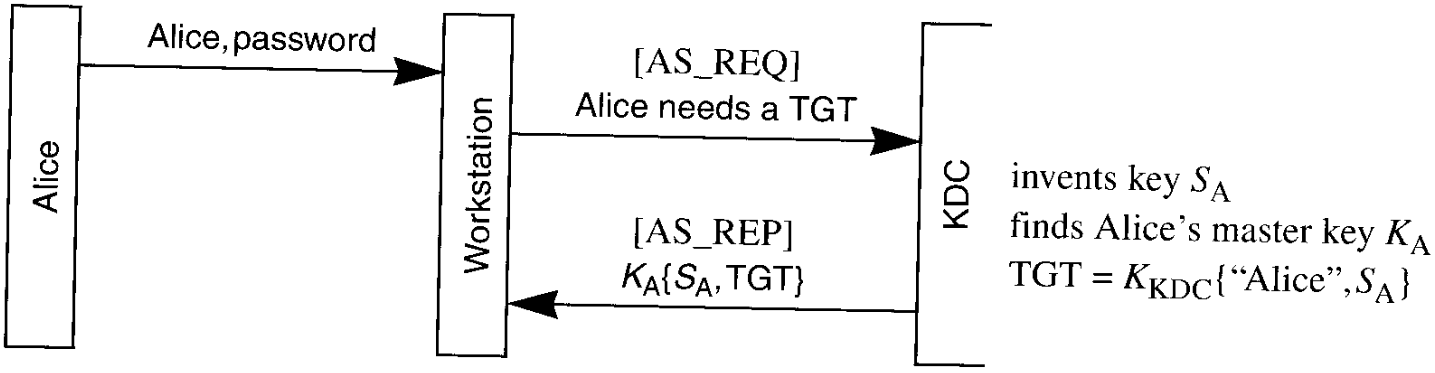
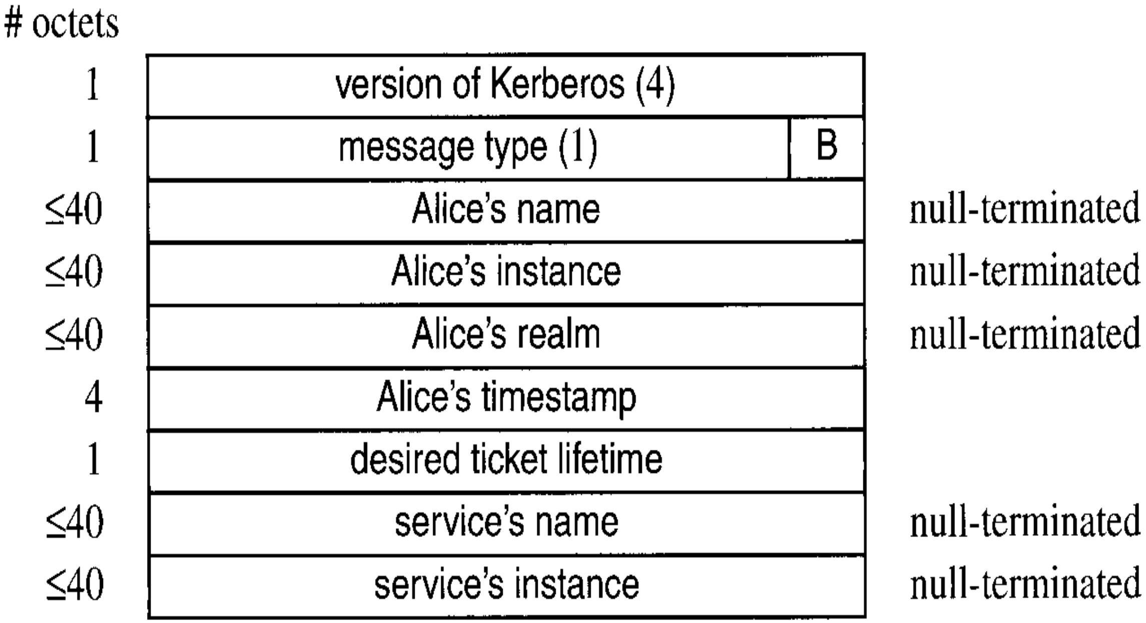

Alice
Alice, password
Workstation
[AS_REQ]
Alice needs a TGT
KDC
[AS_REP]
$K_A\{S_A, \mathrm{TGT}\}$
invents key S_A
finds Alice's master key K_A
TGT = $K_{KDC}\{$"Alice", $S_A\}$

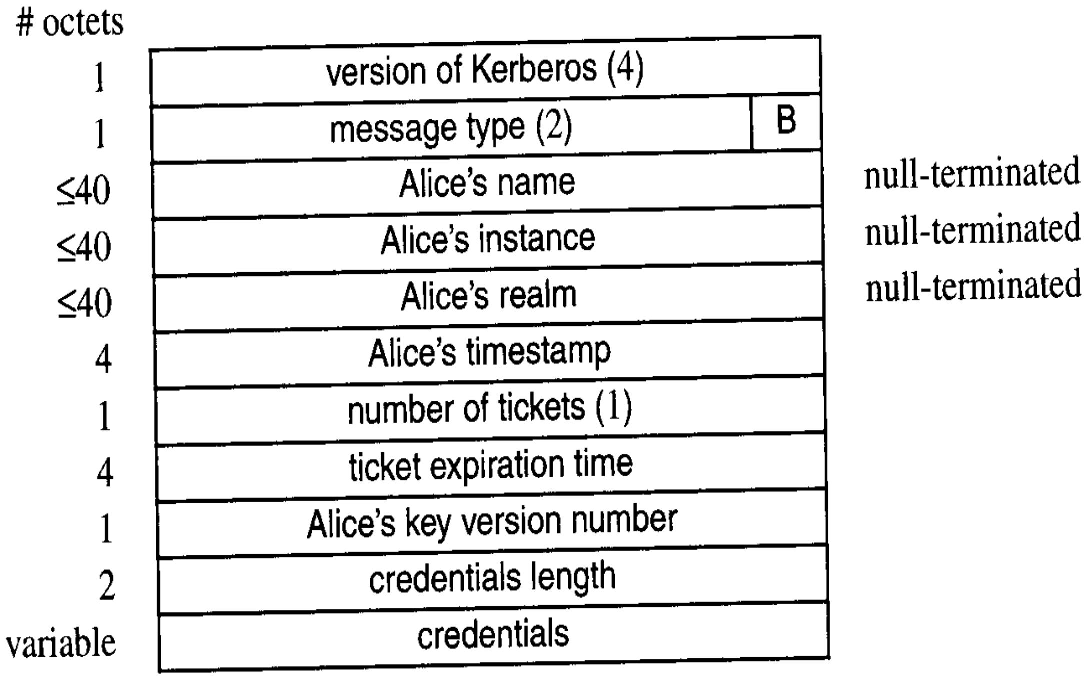
Figure 13-1. Obtaining a TGT

AS_REQ



# octets			
1	version of Kerberos (4)		
1	message type (1)	B	
≤40	Alice’s name		null-terminated
≤40	Alice’s instance		null-terminated
≤40	Alice’s realm		null-terminated
4	Alice’s timestamp		
1	desired ticket lifetime		
≤40	service’s name		null-terminated
≤40	service’s instance		null-terminated

AS_REP



# octets			
1	version of Kerberos (4)		
1	message type (2)	B	
≤40	Alice's name		null-terminated
≤40	Alice's instance		null-terminated
≤40	Alice's realm		null-terminated
4	Alice's timestamp		
1	number of tickets (1)		
4	ticket expiration time		
1	Alice's key version number		
2	credentials length		
variable	credentials		

Obtaining Services from a Remote Node

- Before Alice talks to Bob, K_{KDC}\{Alice, S_A, ...\} is used to authenticate Alice to KDC, which then sends Alice S_A\{K_{AB}, K_B\{Alice, K_{AB}, …\}\}
- Essentially, TGT informs the KDC to use session key S_A instead of Alice's master key K_A

- Step 1: Alice uses TGT to obtain a ticket
- Step 2: Alice uses ticket to log into remote node

Step 1: Getting a Ticket to Remote Node

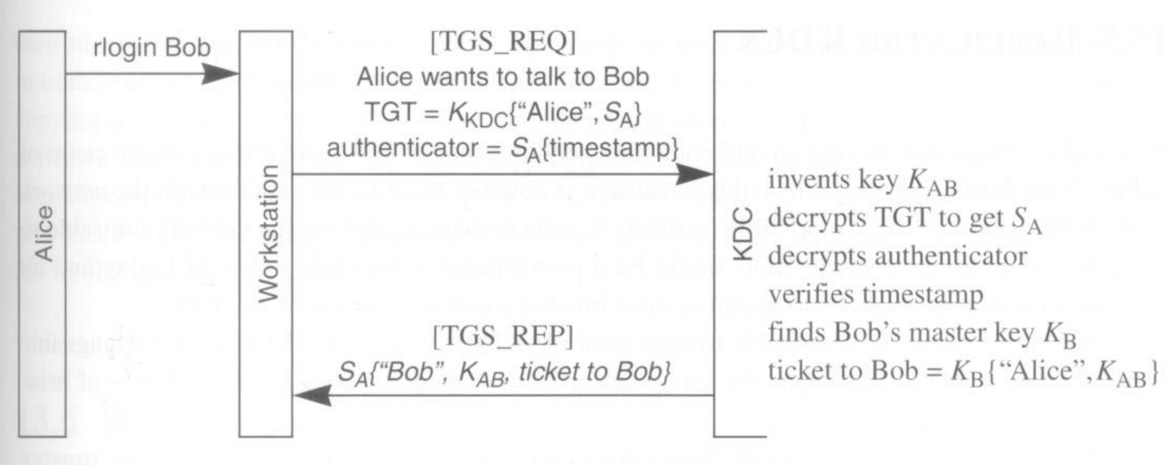
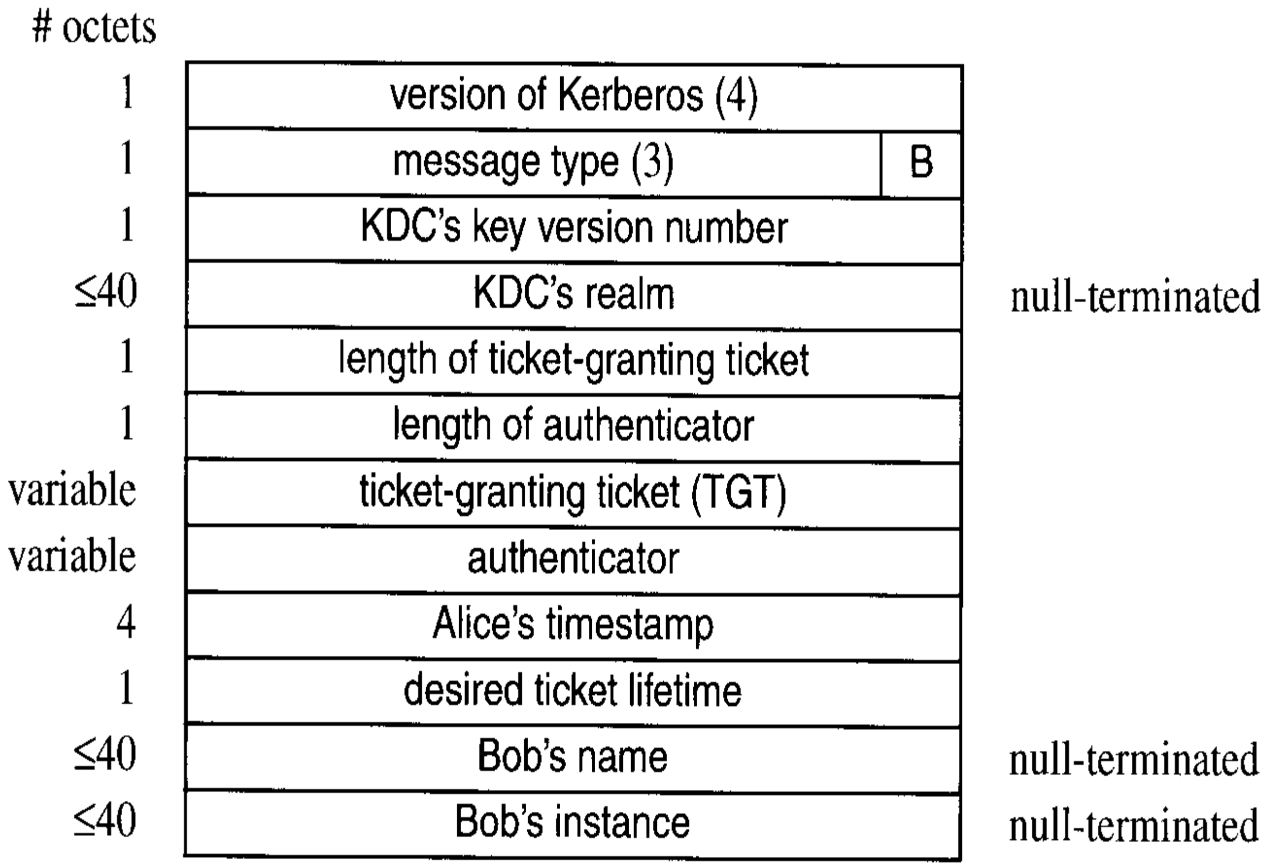

Alice
rlogin Bob
Workstation
[TGS_REQ]
Alice wants to talk to Bob
TGT = K_{KDC}{"Alice", S_A}
authenticator = S_A{timestamp}
KDC
invents key K_{AB}
decrypts TGT to get S_A
decrypts authenticator
verifies timestamp
finds Bob's master key K_B
ticket to Bob = K_B{"Alice", K_{AB}}
[TGS_REP]
S_A{"Bob", K_{AB}, ticket to Bob}

Figure 13-2. Getting a ticket to Bob for Alice

Alice

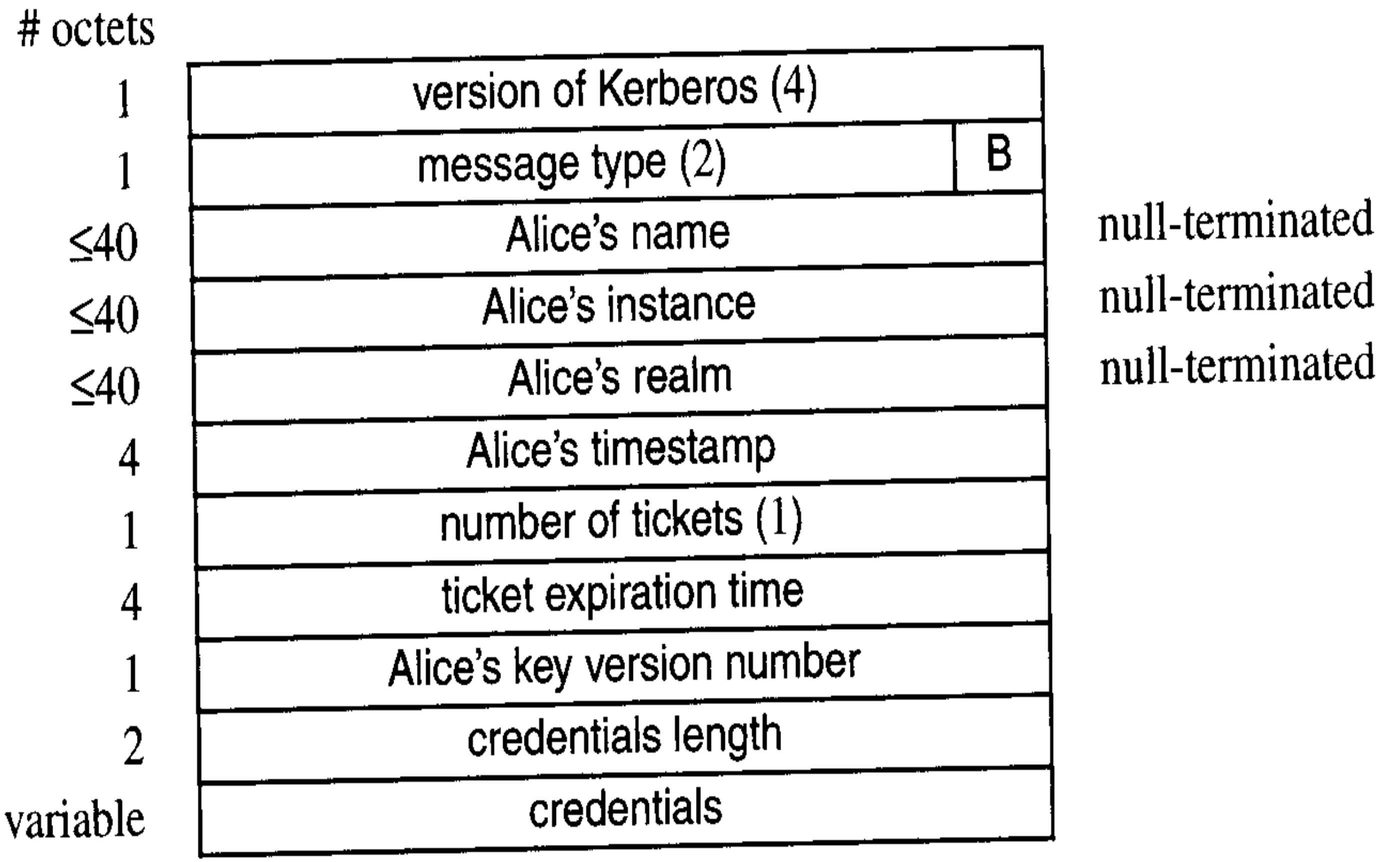
Ticket granting server

TGS_REQ



# octets			
1	version of Kerberos (4)		
1	message type (3)	B	
1	KDC's key version number		
≤40	KDC's realm		null-terminated
1	length of ticket-granting ticket		
1	length of authenticator		
variable	ticket-granting ticket (TGT)		
variable	authenticator		
4	Alice's timestamp		
1	desired ticket lifetime		
≤40	Bob's name		null-terminated
≤40	Bob's instance		null-terminated

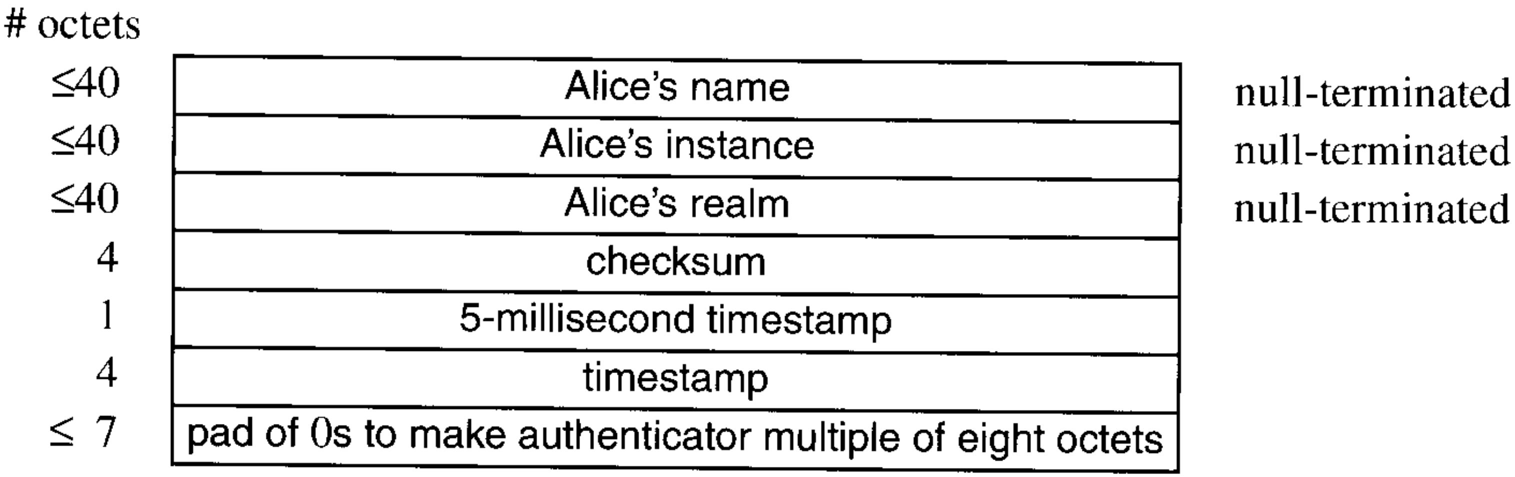
TGS_REP (Also AS_REP)



# octets			
1	version of Kerberos (4)		
1	message type (2)	B	
≤40	Alice's name		null-terminated
≤40	Alice's instance		null-terminated
≤40	Alice's realm		null-terminated
4	Alice's timestamp		
1	number of tickets (1)		
4	ticket expiration time		
1	Alice's key version number		
2	credentials length		
variable	credentials		

Authenticator

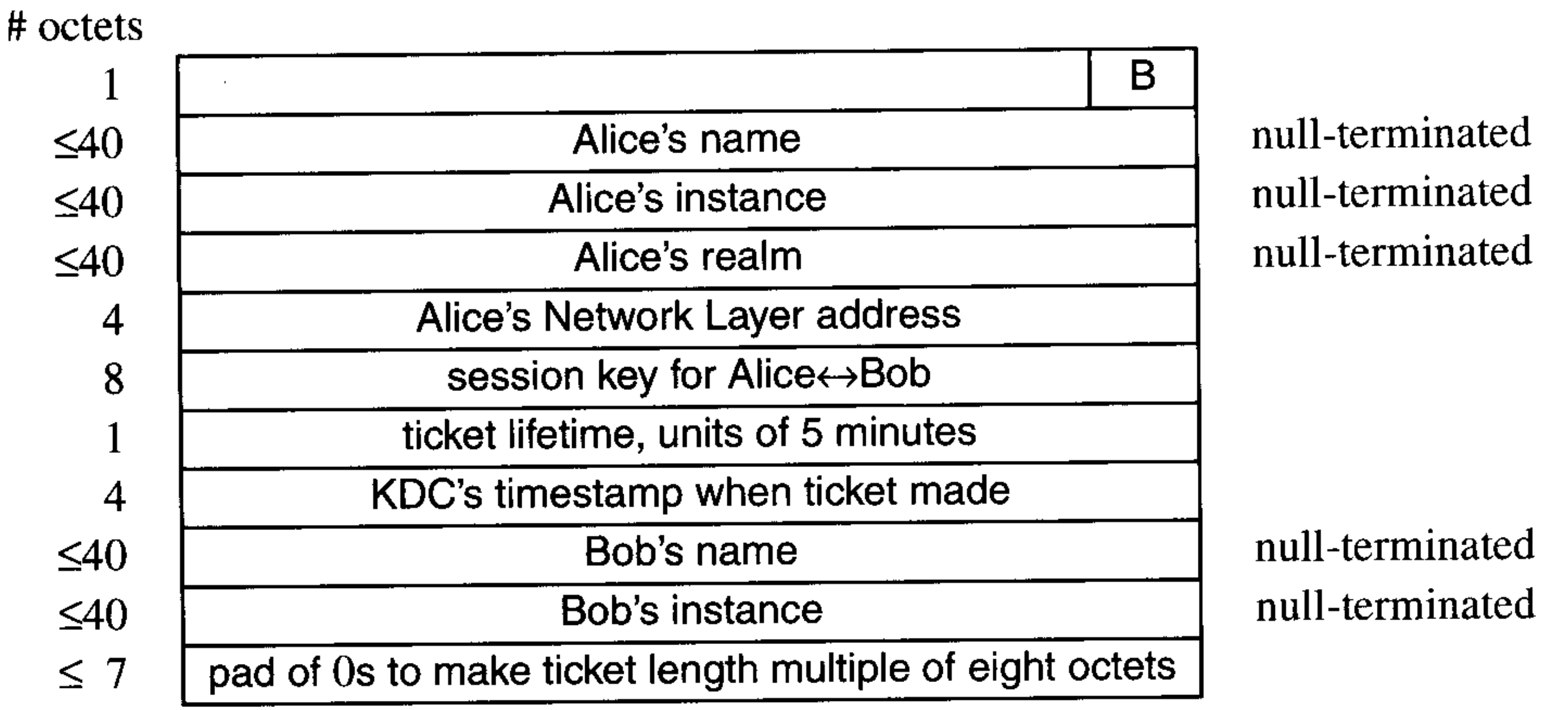
- Encrypted with the session keys shared between the two parties



# octets		
≤40	Alice's name	null-terminated
≤40	Alice's instance	null-terminated
≤40	Alice's realm	null-terminated
4	checksum	
1	5-millisecond timestamp	
4	timestamp	
≤ 7	pad of 0s to make authenticator multiple of eight octets	

Tickets

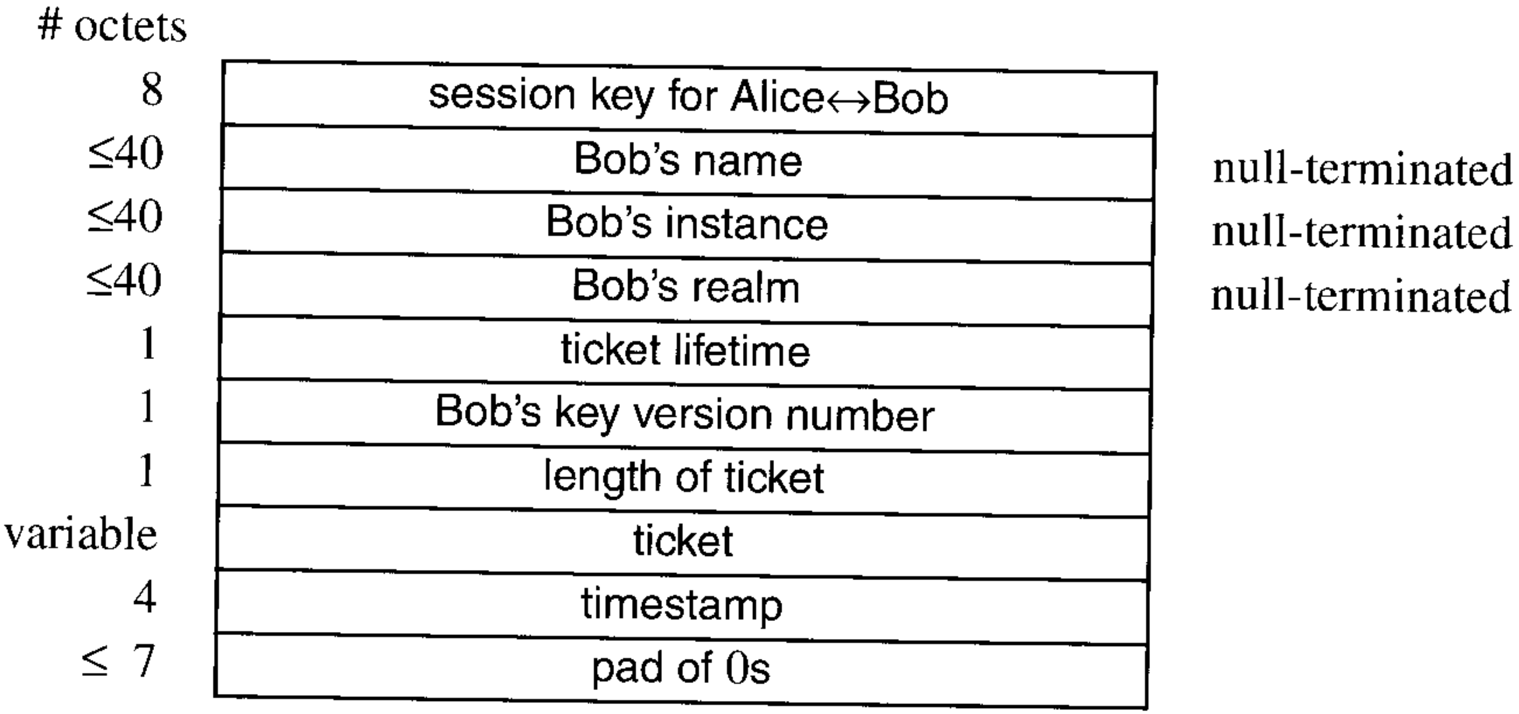
- Encrypted by KDC with receiver (Bob)'s master key, given to sender (Alice)



# octets		
1	B	
≤40	Alice's name	null-terminated
≤40	Alice's instance	null-terminated
≤40	Alice's realm	null-terminated
4	Alice's Network Layer address	
8	session key for Alice↔Bob	
1	ticket lifetime, units of 5 minutes	
4	KDC's timestamp when ticket made	
≤40	Bob's name	null-terminated
≤40	Bob's instance	null-terminated
≤ 7	pad of 0s to make ticket length multiple of eight octets	

Credentials

- Encrypted by KDC with requester (Alice)'s session key



# octets		
8	session key for Alice↔Bob	
≤40	Bob's name	null-terminated
≤40	Bob's instance	null-terminated
≤40	Bob's realm	null-terminated
1	ticket lifetime	
1	Bob's key version number	
1	length of ticket	
variable	ticket	
4	timestamp	
≤ 7	pad of 0s	

Step 2: Logging into Remote Node

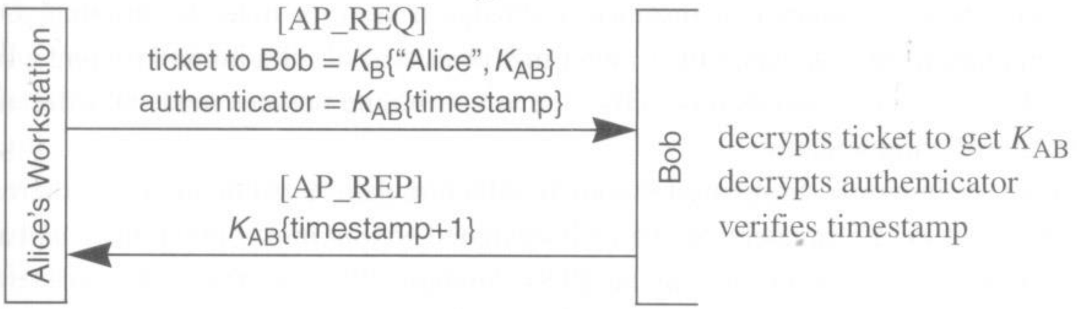
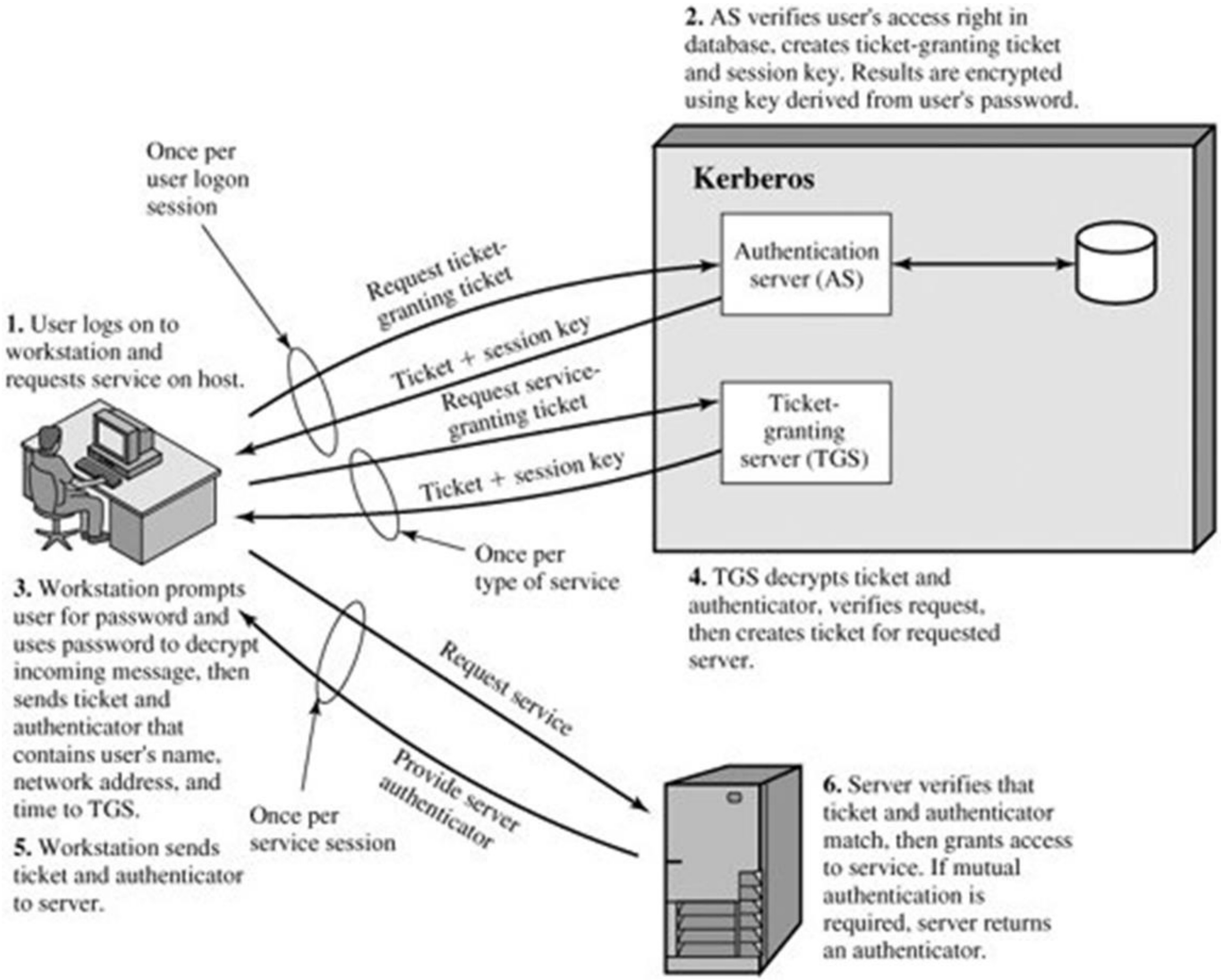

Alice's Workstation
[AP_REQ]
ticket to Bob = K_B{"Alice", K_{AB}}
authenticator = K_{AB}{timestamp}
Bob
decrypts ticket to get K_{AB}
decrypts authenticator
verifies timestamp
[AP_REP]
K_{AB}{timestamp+1}

Figure 13-3. Logging into Bob from Alice's workstation

Remote server

Summary



2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

Kerberos

Authentication server (AS)

Ticket-granting server (TGS)

Once per user logon session

Request ticket-granting ticket

Ticket + session key

Request service-granting ticket

Ticket + session key

Once per type of service

1. User logs on to workstation and requests service on host.

3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.

Request service

Provide server authenticator

Once per service session

5. Workstation sends ticket and authenticator to server.

6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Replicated KDCs

- Purposes
  - Prevent single point failure
  - Prevent performance bottleneck
- Multiple KDCs
  - One master copy for read/write
  - Multiple replicas for read only
  - All having the same database and the same master key
- Updating KDC database
  - KDC’s database is transferred in clear
  - Privacy: keys are stored as ciphertext encrypted by KDC’s master key
  - Integrity: a cryptographic hash of the database file and a timestamp

Realms

- To scale to a large network including multiple administrations, the principals are divided into realms. Each realm has its own KDC.
- The KDCs of other realms are treated as resources (principals) of a local realm.

Inter-realm Authentication

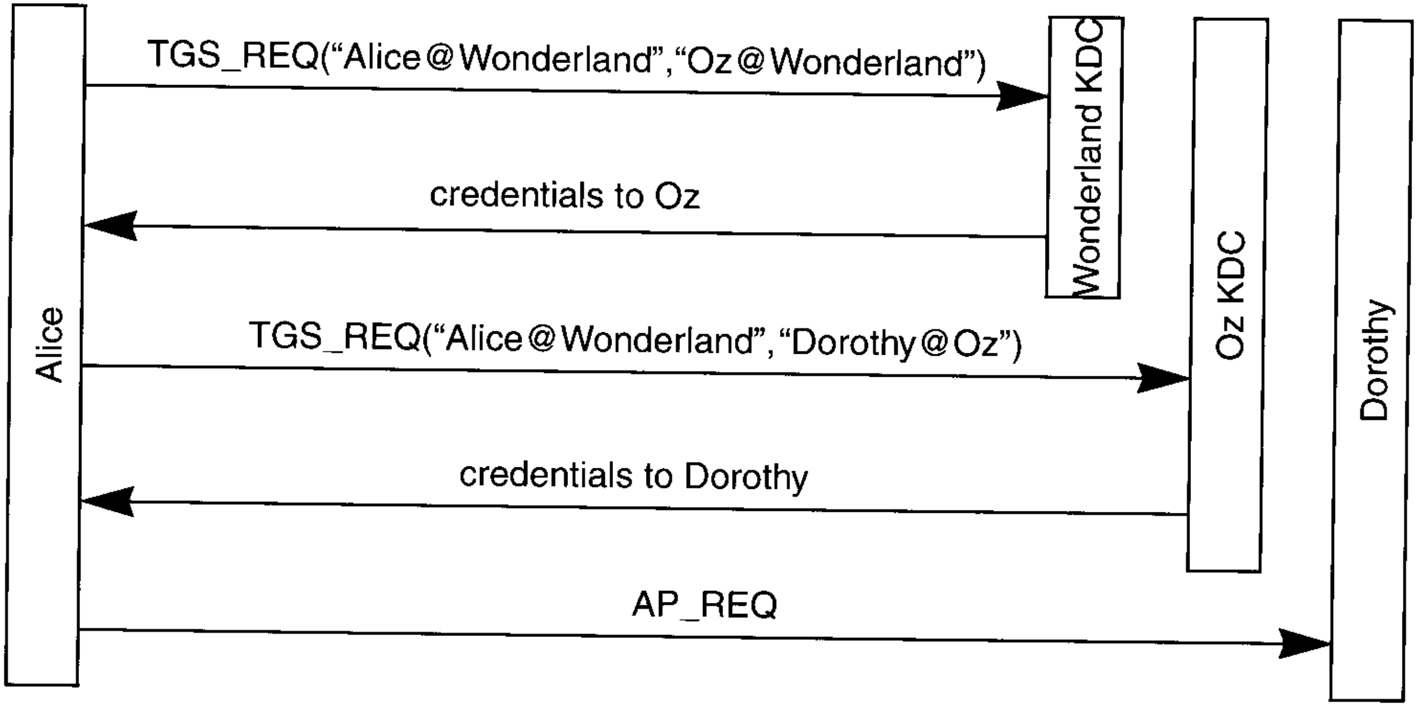
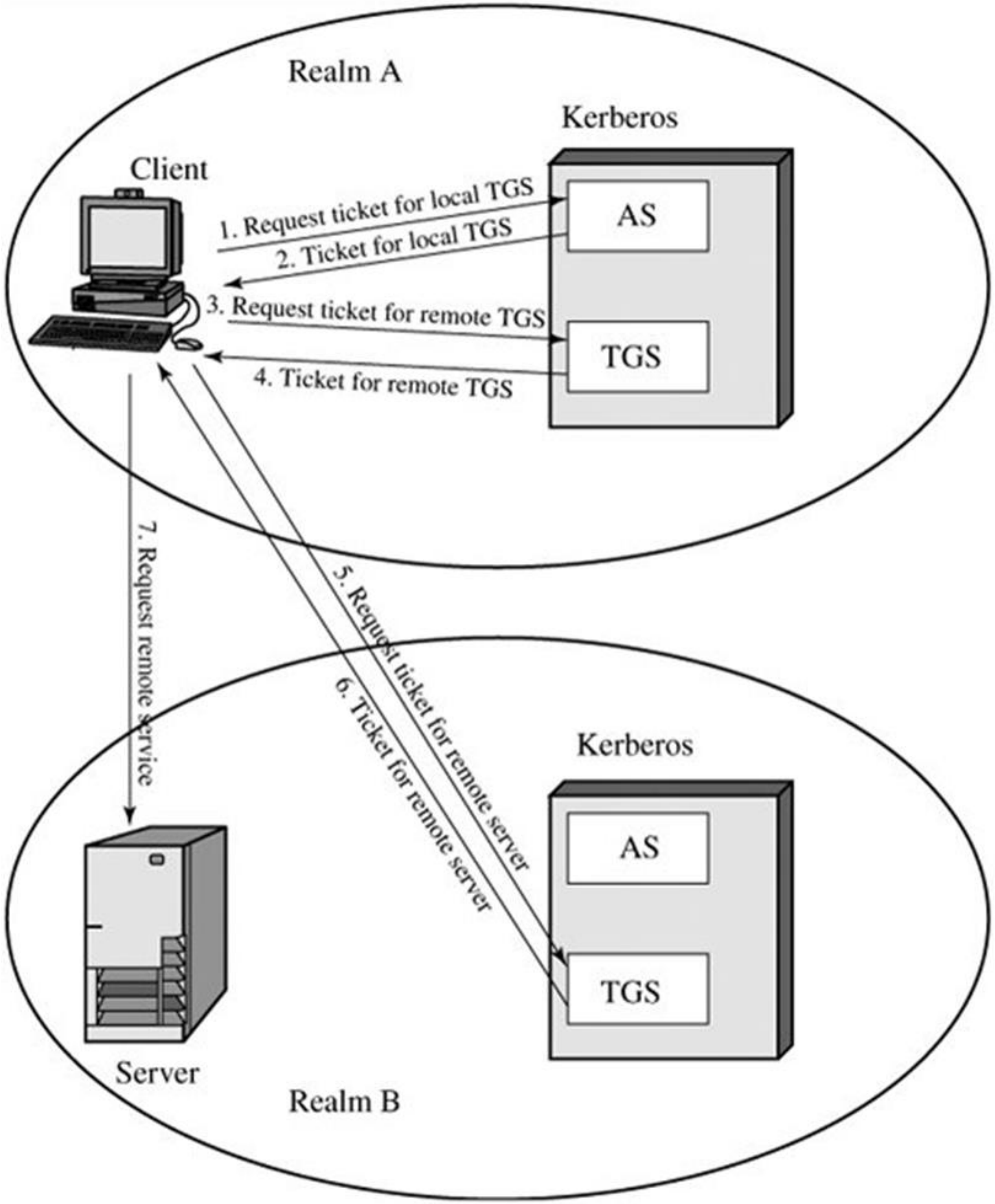

Alice
Wonderland KDC
Oz KDC
Dorothy
TGS_REQ("Alice@Wonderland","Oz@Wonderland")
credentials to Oz
TGS_REQ("Alice@Wonderland", "Dorothy@Oz")
credentials to Dorothy
AP_REQ

Figure 13-4. Interrealm authentication


Realm A
Kerberos
Client
AS
1. Request ticket for local TGS
2. Ticket for local TGS
3. Request ticket for remote TGS
TGS
4. Ticket for remote TGS
7. Request remote service
5. Request ticket for remote server
6. Ticket for remote server
Kerberos
AS
TGS
Server
Realm B

Inter-realm Authentication

- Kerberos V4 does not allow authentication through a chain of KDCs.
  - Reason: A rogue KDC can impersonate other realms
- Kerberos V5 does.
  - Hierarchy of realms

Kerberos V4 vs. V5

- Encryption system: V4 requires DES, V5 can use any
- Internet protocol: V4 requires IP, V5 can use other types
- Message byte ordering: V4 uses B BIT, all message structures are defined using Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules (BER) in V5 providing unambiguous byte ordering
- Ticket lifetime: 21 hours in V4 (encoded in a 1-octet quantity), V5 tickets include explicit start and end time allowing arbitrary lifetimes

Kerberos V4 vs. V5

- Authentication forwarding/delegation: V4 does not allow and V5 allows
- Inter-realm authentication: no chaining in V4 (N realms require O(N2) Kerberos-to-Kerberos relationships), V5 supports KDC hierarchy
- Session keys: negotiation of sub-session keys is supported in V5 for different sessions of the same service type
- Privacy + integrity: V4 uses PCBC, V5 uses explicit integrity mechanisms (e.g., hash) with CBC encryption
- Password attacks: both versions are vulnerable

Readings

- Chapter 15.3 of textbook, or Chapter 13 of Kaufman’s book for Kerberos
- Chapter 20 of textbook for IPSec