Cryptography and Network Security

KERBEROS

Session Meta Data

Author	**Dr T Sree Sharmila**
Reviewer	
Version Number	1.0
Release Date	1 August 2018

Revision History

Revision Date	Details	Version no.
		1.0

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Kerberos - Introduction

- Kerberos is an authentication service developed as part of Project Athena at MIT, and is one of the best known and most widely implemented **trusted third party** key distribution systems.
- Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.
- Unlike most other authentication schemes, Kerberos relies exclusively on symmetric encryption, making no use of public-key encryption.
- Two versions of Kerberos are in common use: v4 & v5.

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Kerberos Motivation

- Without knowledge of identity of person requesting an operation difficult to decide if it should be allowed.
- Traditional authentication methods are not suitable for use in computer networks where attackers can monitor network traffic and intercept passwords.
- Use of strong authentication methods is imperative.

Kerberos Motivation

In a common distributed architecture
Three approaches to security envisaged:

- Rely on individual client work stations to assure identity of user.
- Require client systems to authenticate themselves to servers.
- Require user to prove identity for each service invoked.

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Kerberos Requirements

- its first report identified requirements as:
  - secure
  - reliable
  - transparent
  - Scalable
- Clients and servers trust Kerberos to mediate their mutual authentication

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Kerberos Version 4

- Uses DES, in a rather elaborate protocol, to provide authentication
- Uses an Authentication Server (AS)
  - Knows all user passwords, and stores in a DB
  - Shares a unique secret key with each server
  - Send an encrypted ticket granting ticket
  - TGT contains a lifetime and timestamp
- Uses a Ticket Granting Server (TGS)
  - Issues tickets to users authenticated by AS
  - Encrypted with a key only known by AS and TGS
  - Returns a service granting ticket
- Service granting ticket contains timestamp and lifetime

Kerberos v4

- A Simple Authentication Dialogue

1) C -> AS: $ID_C||P_C||ID_V$
2) AS->C: Ticket
3) C->V: $ID_C||$ Ticket

Ticket = $E_{KV}[ID_C||AD_C||ID_V]$

A More Secure Authentication Dialogue

- **once per logon session**

1) C -> AS: $ID_C||ID_{tgs}$
2) AS->C: $E_{Kc}[Ticket_{tgs}]$

- **once per type of service**

3) C->TGS: $ID_C||ID_V||Ticket_{tgs}$
4) TGS->C: $Ticket_V$

- **once per service session**

5) C->V: $ID_C||Ticket_V$

$Ticket_{tgs} = E_{Ktgs}[ID_C||AD_C||ID_{tgs}||TS_1||Lifetime_1]$

$Ticket_v = E_{Kv}[ID_C||AD_C||ID_v||TS_2||Lifetime_2]$

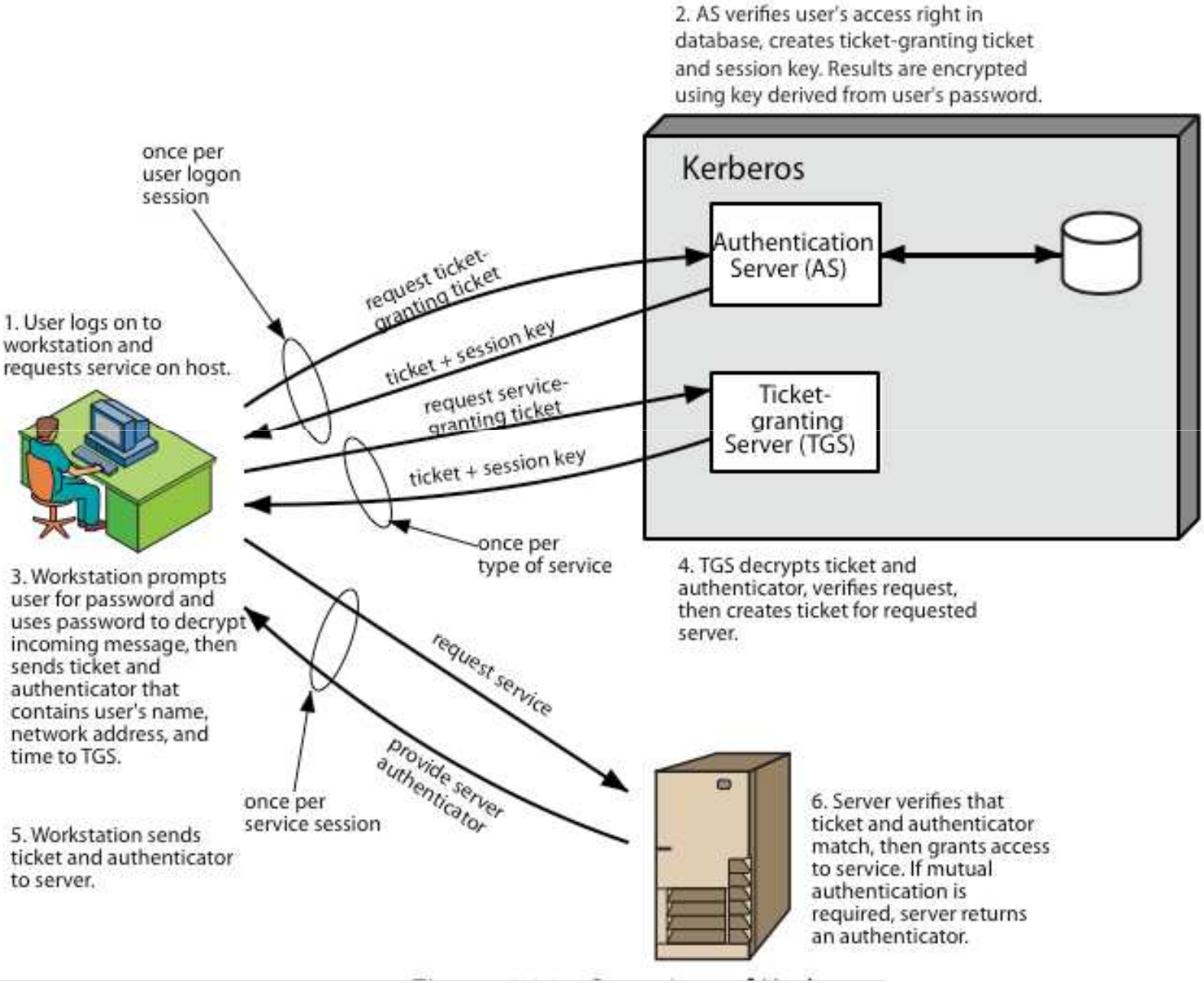
Kerberos v4 Overview

- a basic third-party authentication scheme
- have an Authentication Server (AS)
  - users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- have a Ticket Granting server (TGS)
  - users subsequently request access to other services from TGS on basis of users TGT

Kerberos v4 Dialogue

1. obtain ticket granting ticket from AS
   - once per session
2. obtain service granting ticket from TGT
   - for each distinct service required
3. client/server exchange to obtain service
   - on every service request

Kerberos 4 Overview



summary of Kerberos version 4 message exchanges

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) C → AS: $ID_c \| ID_{tgs} \| TS_1$
(2) AS → C: $E_{K_c}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}]$
$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket
(3) C → TGS: $ID_v \| Ticket_{tgs} \| Authenticator_c$
(4) TGS → C: $E_{K_{c,tgs}}[K_{c,v} \| ID_v \| TS_4 \| Ticket_v]$
$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2]$
$Ticket_v = E_{K_v}[K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4]$
$Authenticator_c = E_{K_{tgs}}[ID_C \| AD_C \| TS_3]$
(c) Client/Server Authentication Exchange: to obtain service
(5) C → V: $Ticket_v \| Authenticator_c$
(6) V → C: $E_{K_{c,v}}[TS_5 + 1]$ (for mutual authentication)
$Ticket_v = E_{K_v}[K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4]$
$Authenticator_c = E_{K_{c,v}}[ID_C \| AD_C \| TS_5]$

Rationale for the Elements of the Kerberos Version 4 Protocol

(a) Authentication Service Exchange	
Message (1)	Client requests ticket-granting ticket
ID_C:	Tells AS identity of user from this client
$IDtgs$:	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
E_{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

(b) Ticket-Granting 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
$E_{K_{tgs}}$	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
$E_{K_{c,tgs}}$	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_2:	Informs TGS of time this authenticator was generated

(c) Client/Server Authentication 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
$E_{K_{cv}}$:	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
E_{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
$E_{K_{cv}}$:	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

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

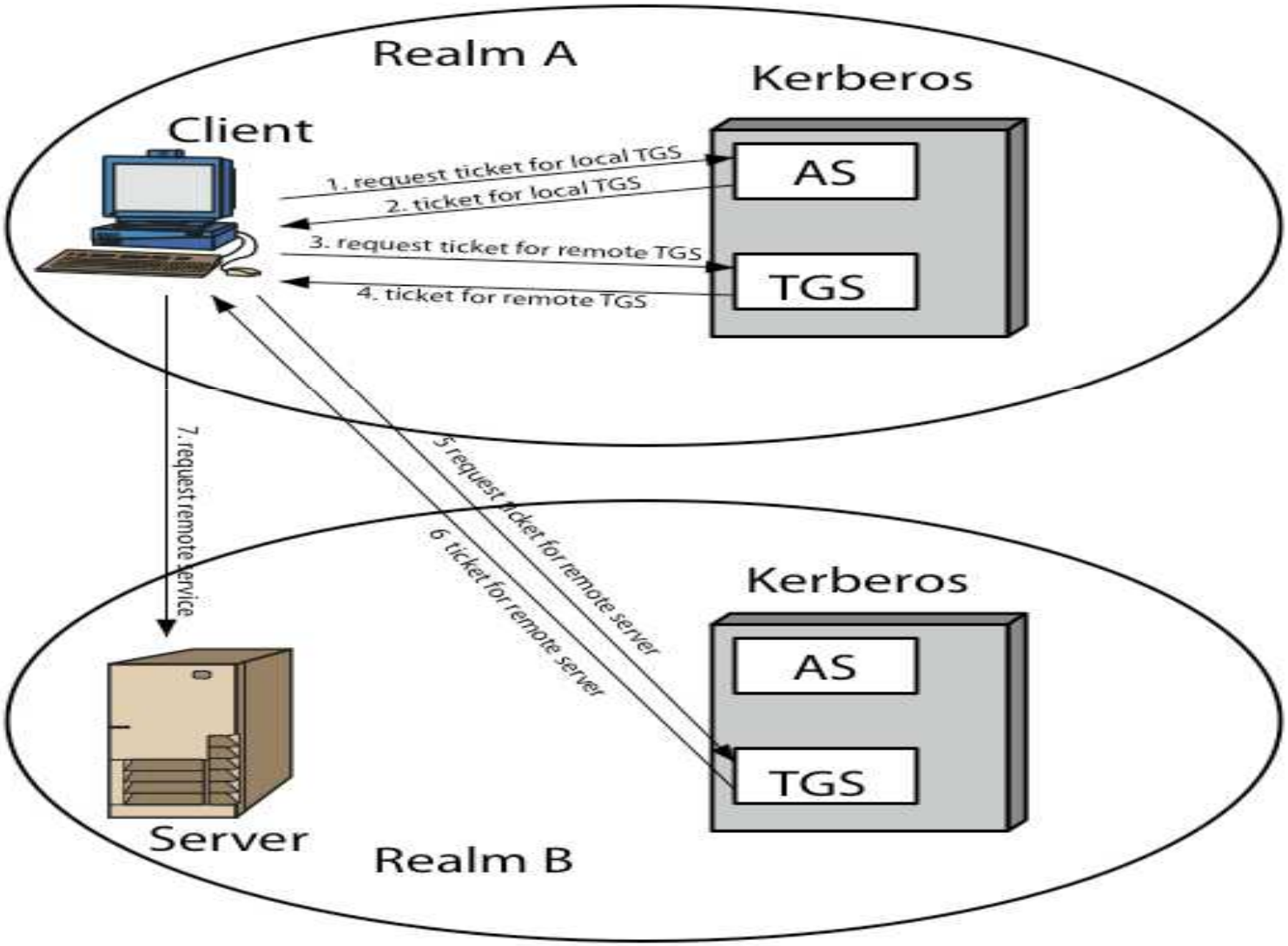
Kerberos Realms

- A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers is referred to as a Kerberos realm.
- A Kerberos realm is a set of managed nodes that share the same Kerberos database, and are part of the same administrative domain.
- If have multiple realms, their Kerberos servers must share keys and trust each other.

Kerberos Realms

- A Kerberos Realm
  - Set of managed nodes that share the same Kerberos database
- this is termed a realm
  - typically a single administrative domain
- Kerberos server in each realm shares a secret key with one another
- There must be trust between the servers
- i.e. each server are registered with one another

Request for Service in Another Realm



summary of Kerberos realm message exchanges

1) C -> AS: $ID_C||ID_{tgs}||TS_1$
2) AS->C: $E_{Kc}[K_{c,tgs}||ID_{tgs}||TS_2||Lifetime_2||Ticket_{tgs}]$
3) C->TGS: $ID_{tgsrem}||Ticket_{tgs}||Authenticator_c$
4) TGS->C: $E_{Kc,tgs}[K_{c,tgsrem}||ID_{tgsrem}||TS_4||Ticket_{tgsrem}]$
5) C->TGS_{rem}: $ID_{vrem}||Ticket_{tgsrem}||Authenticator_c$
6) C->V_{rem}: $Ticket_{vrem}||Authenticator_c$

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Kerberos Version 5

- developed in mid 1990's
- provides improvements over v4
  - addresses environmental shortcomings
    - encryption algorithm, network protocol, byte order, ticket lifetime, authentication forwarding, inter-realm authentication
  - and technical deficiencies
    - double encryption, non-standard mode of use, session keys, password attacks
- specified as Internet standard RFC 1510

summary of Kerberos version 5 messaae exchanaes

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

(1) C → AS: $\text{Options} \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) AS → C: $Realm_c \| ID_C \| Ticket_{tgs} \| E_{K_c}\left[K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}\right]$

$$Ticket_{tgs} = E_{K_{tgs}}\left[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times\right]$$

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

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

(4) TGS → C: $Realm_C \| ID_C \| Ticket_V \| E_{K_{c,tgs}}\left[K_{C,V} \| Times \| Nonce_2 \| Realm_V \| ID_V\right]$

$$Ticket_{tgs} = E_{K_{tgs}}\left[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times\right]$$

$$Ticket_v = E_{K_v}\left[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times\right]$$

$$Authenticator_c = E_{K_{c,gs}}\left[ID_C \| Realm_c \| TS_1\right]$$

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

(5) C → V: $\text{Options} \| \text{Ticket}_v \| \text{Authenticator}_c$

(6) V → C: $E_{K_{C,V}}\left[\text{TS}_2 \| \text{Subkey} \| \text{Seq\#}\right]$

$$Ticket_v = E_{K_v}\left[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times\right]$$

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

Kerberos Version 5 Flags

INITIAL	This ticket was issued using the AS protocol and not issued based on a ticket-granting ticket.
PRE-AUTHENT	During initial authentication, the client was authenticated by the KDC before a ticket was issued.
HW-AUTHENT	The protocol employed for initial authentication required the use of hardware expected to be possessed solely by the named client.
RENEWABLE	Tells TGS that this ticket can be used to obtain a replacement ticket that expires at a later date.
MAY-POSTDATE	Tells TGS that a postdated ticket may be issued based on this ticket-granting ticket.
POSTDATED	Indicates that this ticket has been postdated; the end server can check the authtime field to see when the original authentication occurred.
INVALID	This ticket is invalid and must be validated by the KDC before use.
PROXIABLE	Tells TGS that a new service-granting ticket with a different network address may be issued based on the presented ticket.
PROXY	Indicates that this ticket is a proxy.
FORWARDABLE	Tells TGS that a new ticket-granting ticket with a different network address may be issued based on this ticket-granting ticket.
FORWARDED	Indicates that this ticket has either been forwarded or was issued based on authentication involving a forwarded ticket-granting ticket.

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Summary

- have discussed:
  - Kerberos V4
  - Kerberos Realms
  - Kerberos V5

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

Test your understanding

1) What problem was Kerberos designed to address?
2) What are three threats associated with user authentication over a network or internet?
3) List three approaches to secure user authentication in a distributed environment.
4) What four requirements were defined for Kerberos?
5) In Kerberos, when Bob receives a Ticket from Alice, how does he know it came from Alice?
6) In Kerberos, what does the Ticket contain that allows Alice and Bob to talk securely?

Agenda

- Introduction
- Motivation
- Requirement
- Kerberos V4
- Kerberos Realms
- Kerberos V5
- Summary
- Test your understanding
- References

References

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002.