

Distributed File Systems Security with Kerberos

Austin Godber

godber@asu.edu

<http://uberhip.com/godber/cse531/>

Distributed Filesystems

- NFS
 - host and user based security, Kerberos with RPCSEC or V4
- AFS
 - Variants
 - CODA
 - Intermezzo
 - Kerberos
- SMB
 - Kerberos (with Domain Authentication), Other Methods

NFS

- Stateless Server (V2,V3) Stateful in new V4 Proposal
- Block Level Transfer
 - 8k
 - Read-ahead
- Data Caching
- Security
 - Mount level – Server name or IP (/etc/exports)
 - File Level – Unix User ID (UID) and UNIX Permissions

AFS

- Data Caching (64k Blocks)
- Read Only Replication
- Security
 - Kerberos – for authentication
 - ACLs – for Access Control

SMB

- Data Caching
- Read Ahead, Write Behind
- File Change Notification
- Replicated Virtual Volumes
- Security
 - Kerberos – for authentication
 - Password – cleartext or Challenge/Response

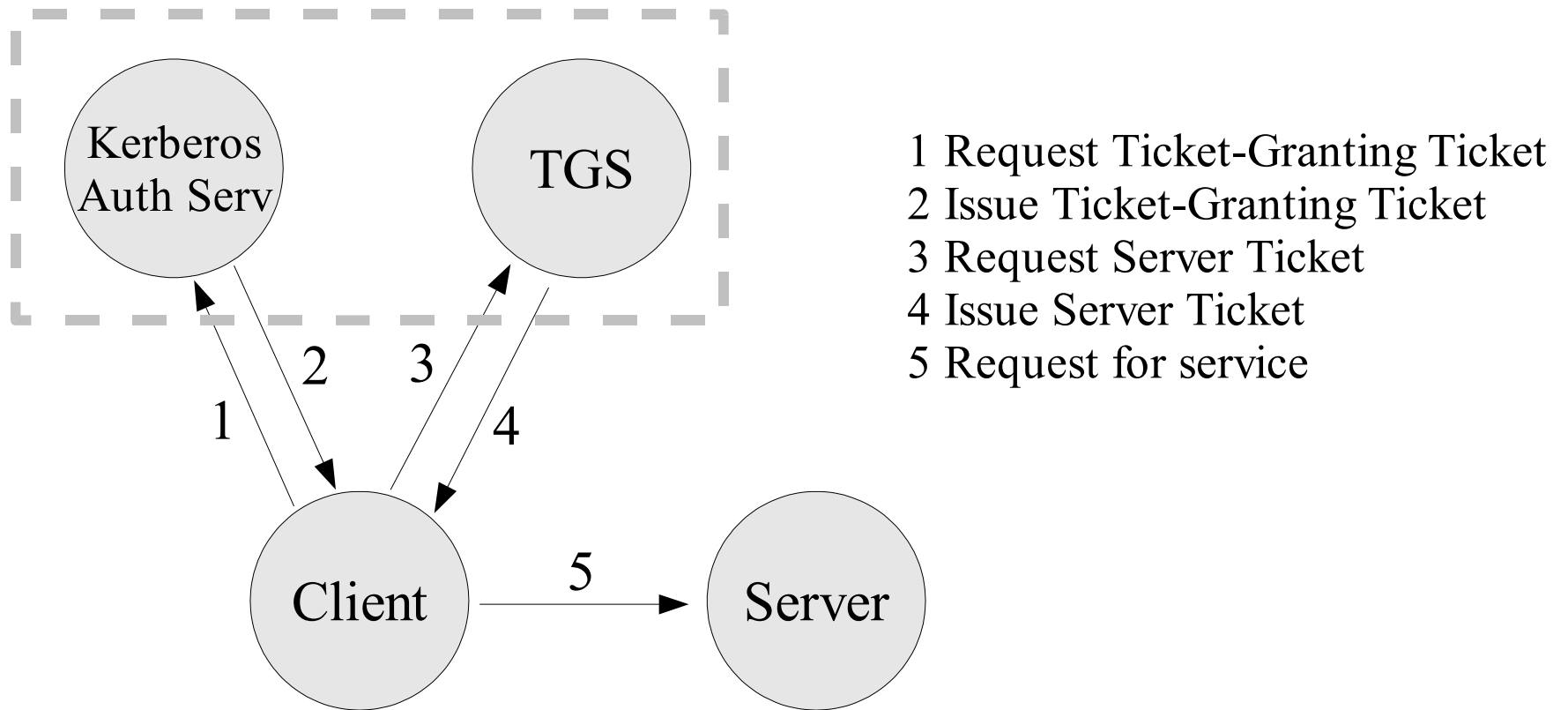
Crypto Primer

- Symmetric Key Cryptography
 - Alice (A) and Bob (B) – partake in transaction
 - Trent (T) – trusted third party
 - Mallory (M) – Malicious User
- Example Notation
$$\{X, Y\}_{K_A} = C$$

Kerberos

- Trusted 3rd Party Authentication Protocol
- Based on symmetric key cryptography (DES)
 - Based on Needham-Schroeder Authentication and Key Exchange Protocol
- Kerberos shares a different secret key with every network entity.
- Times must be synchronized between participants (NTP)
- MIT Project Athena Early 1980's

Kerberos - Cartoon



Kerberos Protocol Definitions

- s = server; c = client
- a = client's network address
- v = beginning and ending validity time for ticket
- t = timestamp; K_X = X's secret key
- $K_{X,Y}$ = session key for X and Y
- $\{m\}K_X$ = m encrypted with X's secret key
- Ticket: $T_{c,s} = s, \{c, a, v, K_{c,s}\}K_s$
- Authenticator: $A_{c,s} = \{c, t, key\}K_{c,s}$

Kerberos Protocol

1 Client to Kerberos:

c, tgs

2 Kerberos to Client:

$\{K_{c,tgs}\}K_c, \{T_{c,tgs}\}K_{tgs}$

3 Client to TGS:

$\{A_{c,s}\}K_{c,tgs}, \{T_{c,tgs}\}K_{tgs}$

4 TGS to client:

$\{K_{c,s}\}K_{c,tgs}, \{T_{c,s}\}K_s$

5 Client to Server:

$\{A_{c,s}\}K_{c,s}, \{T_{c,s}\}K_s$

Kerberos Protocol

1 Client to Kerberos:

c, tgs

2 Kerberos to Client:

$\{K_{c,tgs}\}K_c, \{tgs, \{c, a, v, K_{c,tgs}\}K_{tgs}\}K_{tgs}$

3 Client to TGS:

$\{c, t, key\}K_{c,tgs}, \{tgs, \{c, a, v, K_{c,tgs}\}K_{tgs}\}K_{tgs}$

4 TGS to client:

$\{K_{c,s}\}K_{c,tgs}, \{s, \{c, a, v, K_{c,s}\}K_s\}K_s$

5 Client to Server:

$\{c, t', key'\}K_{c,s}, \{s, \{c, a, v, K_{c,s}\}K_s\}K_s$