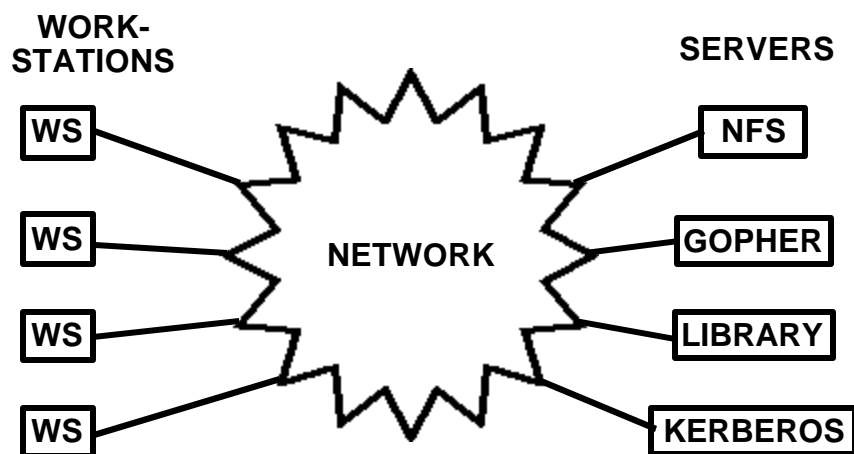


INFS 766
Internet Security Protocols

Lecture 9
Kerberos

Prof. Ravi Sandhu

SYSTEM MODEL



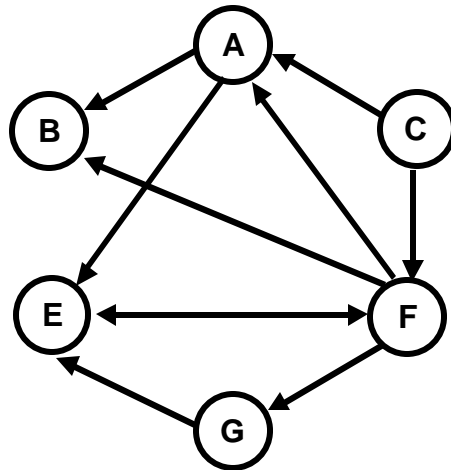
PHYSICAL SECURITY

- ❖ **CLIENT WORKSTATIONS**
 - None, so cannot be trusted
- ❖ **SERVERS**
 - Moderately secure rooms, with moderately diligent system administration
- ❖ **KERBEROS**
 - Highly secure room, with extremely diligent system administration

KERBEROS OBJECTIVES

- ❖ provide authentication between any pair of entities
- ❖ primarily used to authenticate user-at-workstation to server
- ❖ in general, can be used to authenticate two or more secure hosts to each other on an insecure network
- ❖ servers can build authorization and access control services on top of Kerberos

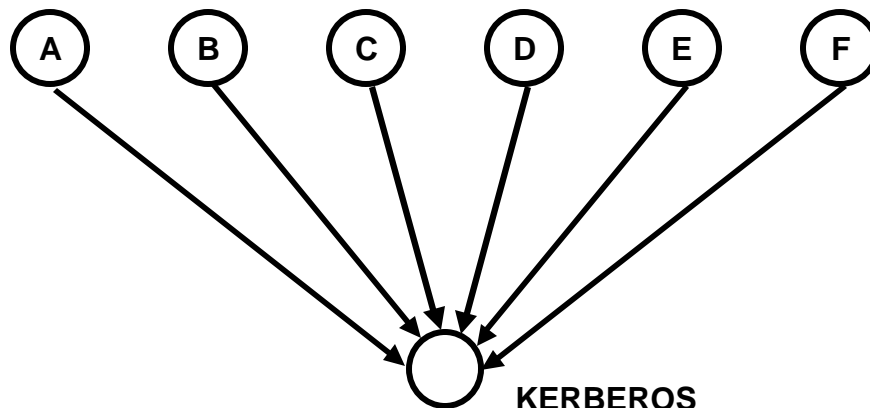
TRUST: BILATERAL RHOSTS MODEL



A → B

A trusts B
A will allow users
logged onto B to
log onto A without
a password

TRUST: CONSOLIDATED KERBEROS MODEL



TRUST: CONSOLIDATED KERBEROS MODEL

- ❖ **breaking into one host provides a cracker no advantage in breaking into other hosts**
- ❖ **authentication systems can be viewed as trust propagation systems**
 - **the Kerberos model is a centralized star model**
 - **the rhosts model is a tangled web model**

WHAT KERBEROS DOES NOT DO

- ❖ **makes no sense on an isolated system**
- ❖ **does not mean that host security can be allowed to slip**
- ❖ **does not protect against Trojan horses**
- ❖ **does not protect against viruses/worms**

KERBEROS DESIGN GOALS

- ❖ **IMPECCABILITY**
 - no cleartext passwords on the network
 - no client passwords on servers (server must store secret server key)
 - minimum exposure of client key on workstation (smartcard solution would eliminate this need)
- ❖ **CONTAINMENT**
 - compromise affects only one client (or server)
 - limited authentication lifetime (8 hours, 24 hours, more)
- ❖ **TRANSPARENCY**
 - password required only at login
 - minimum modification to existing applications

KERBEROS DESIGN DECISIONS

- ❖ **Uses timestamps to avoid replay.
Requires time synchronized within a small window (5 minutes)**
- ❖ **Uses DES-based symmetric key cryptography**
- ❖ **stateless**

KERBEROS VERSIONS

- ❖ We describe Kerberos version 4 as the base version
- ❖ Kerberos version 5 fixes many shortcomings of version 4, and is described here by explaining major differences with respect to version 4

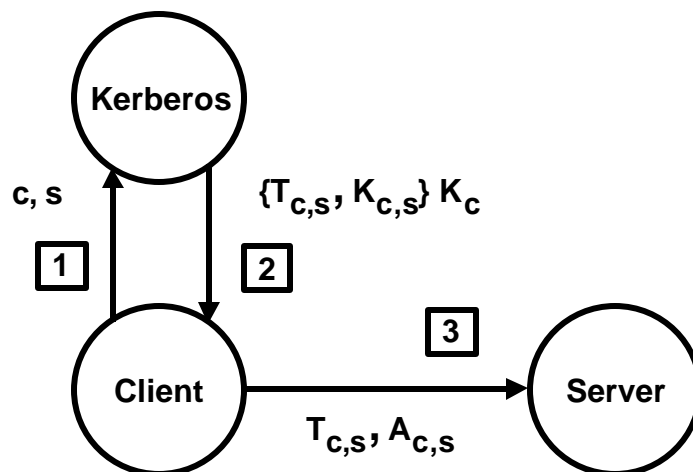
NOTATION

c	client principal
s	server principal
K_x	secret key of “x” (known to x and Kerberos)
$K_{c,s}$	session key for “c” and “s” (generated by Kerberos and distributed to c and s)
$\{P\}_{K_q}$	P encrypted with K_q
$T_{c,s}$	ticket for “c” to use “s”(given by Kerberos to c and verified by s)
$A_{c,s}$	authenticator for “c” to use “s” (generated by c and verified by s)

TICKETS AND AUTHENTICATORS

- ❖ $T_{c,s} = \{s, c, \text{addr}, \text{time}_o, \text{life}, K_{c,s}\}K_s$
- ❖ $A_{c,s} = \{c, \text{addr}, \text{time}_a\}K_{c,s}$
- ❖ **addr** is the IP address, adds little removed in version 5

SESSION KEY DISTRIBUTION



USER AUTHENTICATION

- ❖ for user to server authentication, client key is the user's password (converted to a DES key via a publicly known algorithm)

TRUST IN WORKSTATION

- ❖ untrusted client workstation has K_c
- ❖ is expected to delete it after decrypting message in step 2
- ❖ compromised workstation can compromise one user
- ❖ compromise does not propagate to other users

AUTHENTICATION FAILURES

- ❖ **Ticket decryption by server yields garbage**
- ❖ **Ticket timed out**
- ❖ **Wrong source IP address**
- ❖ **Replay attempt**

KERBEROS IMPERSONATION

- ❖ **active intruder on the network can cause denial of service by impersonation of Kerberos IP address**
- ❖ **network monitoring at multiple points can help detect such an attack by observing IP impersonation**

KERBEROS RELIABILITY

- ❖ availability enhanced by keeping slave Kerberos servers with replicas of the Kerberos database
- ❖ slave databases are read only
- ❖ simple propagation of updates from master to slaves

USE OF THE SESSION KEY

- ❖ Kerberos establishes a session key $K_{c,s}$
- ❖ session key can be used by the applications for
 - client to server authentication (no additional step required in the protocol)
 - mutual authentication (requires fourth message from server to client $\{f(A_{c,s})\}_{K_{c,s}}$, where f is some publicly known function)
 - message confidentiality using $K_{c,s}$
 - message integrity using $K_{c,s}$

TICKET-GRANTING SERVICE

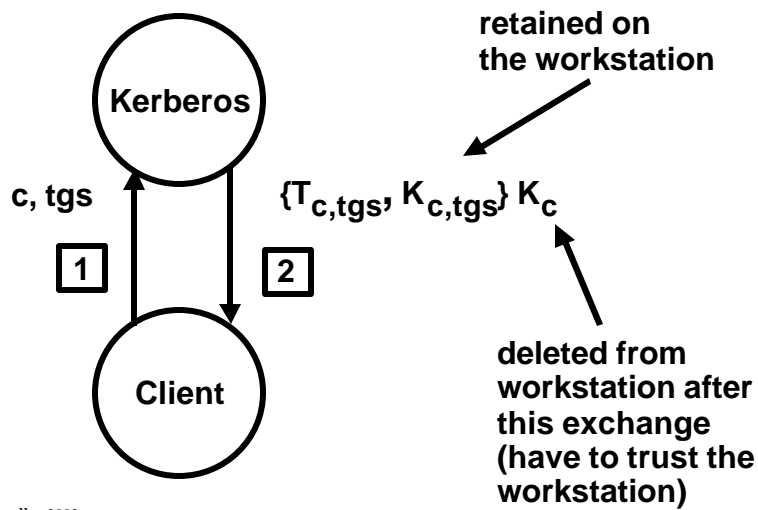
❖ Problem: Transparency

- user should provide password once upon initial login, and should not be asked for it on every service request
- workstation should not store the password, except for the brief initial login

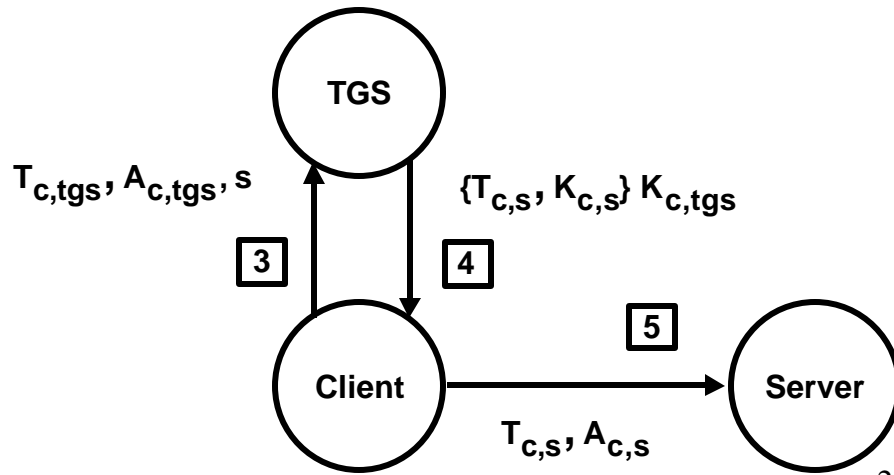
❖ Solution: Ticket-Granting Service (TGS)

- store session key on workstation in lieu of password
- TGS runs on same host as Kerberos (needs access to K_c and K_s keys)

TICKET-GRANTING SERVICE



TICKET-GRANTING SERVICE



TICKET LIFETIME

- ❖ **Life time is minimum of:**
 - requested life time
 - max lifetime for requesting principal
 - max lifetime for requesting service
 - max lifetime of ticket granting ticket
- ❖ **Max lifetime is 21.5 hours**

NAMING

- ❖ **Users and servers have same name format:**
 - name.instance@realm
- ❖ **Example:**
 - sandhu@isse.gmu.edu
 - sandhu.root@isse.gmu.edu
 - rcmd.ipc4@isse.gmu.edu
 - rcmd.csis@isse.gmu.edu
- ❖ **Mapping of Kerberos authentication names to local system names is left up to service provider**

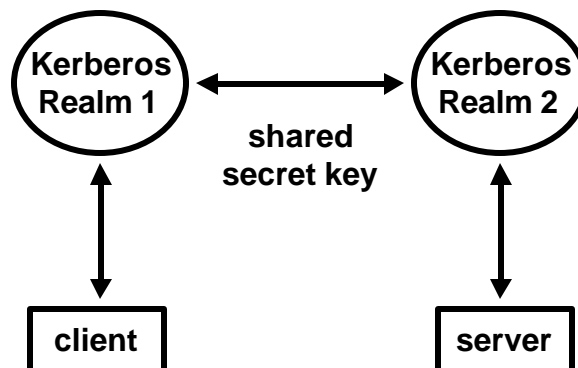
KERBEROS V5 ENHANCEMENTS

- ❖ **Naming**
 - Kerberos V5 supports V4 names, but also provides for other naming structures such as X.500 and DCE
- ❖ **Timestamps**
 - V4 timestamps are Unix timestamps (seconds since 1/1/1970). V5 timestamps are in OSI ASN.1 format.
- ❖ **Ticket lifetime**
 - V4 tickets valid from time of issue to expiry time, and limited to 21.5 hours.
 - V5 tickets have start and end timestamps. Maximum lifetime can be set by realm.

KERBEROS V5 ENHANCEMENTS

- ❖ Kerberos V5 tickets are renewable, so service can be maintained beyond maximum ticket lifetime.
- ❖ Ticket can be renewed until min of:
 - requested end time
 - start time + requesting principal's max renewable lifetime
 - start time + requested server's max renewable lifetime
 - start time + max renewable lifetime of realm

KERBEROS INTER-REALM AUTHENTICATION



KERBEROS INTER-REALM AUTHENTICATION

- ❖ **Kerberos V4 limits inter-realm interaction to realms which have established a shared secret key**
- ❖ **Kerberos V5 allows longer paths**
- ❖ **For scalability one may need public-key technology for inter-realm interaction**

KERBEROS DICTIONARY ATTACK

- ❖ **First two messages reveal known-plaintext for dictionary attack**
- ❖ **first message can be sent by anyone**
- ❖ **Kerberos v5 has pre-authentication option to prevent this attack**