COMP7120/8120 Cryptography and Data Security

Kerberos
Goals of Kerberos

1. User ↔ server **mutual** authentication
2. Users should only need to **authenticate once** to obtain services from **multiple servers**
3. Should **scale** to large numbers of users and servers
   - makes use of a **Key Distribution Center** so servers don’t need to store information about users
Some Properties

- Kerberos uses **only secret key** (symmetric) encryption
  - originally, only DES, but now 3DES and AES as well

- A **stateless** protocol
  - KDCs do not need to remember what messages have previously been generated or exchanged
  - the **state** of the protocol negotiation is contained in the message contents
Example Scenario

- Alice wants to make use of services from X, contacts the KDC to authenticate, gets ticket to present to X

- Bob wants to make use of services from X and Y, contacts the KDC, gets tickets to present to X and Y
The KDC

- Infrastructure needed (KDC components)
  1. the **database** of user information (IDs, password hash, shared secret key, etc.)
  2. an authentication server (**AS**)
  3. a ticket-granting server (**TGS**)
- The KDC of course is critical and should be carefully guarded
Secrets Managed by the KDC

- A *personal* key used for encrypting/decrypting the database
- A *master* (semi-permanent) *shared key* for each user
- a master shared key for each server
Basics of the Kerberos v4 Standard
Protocol Sketch (Common Case)

Alice

#1 Login + Password

#4 Request service from V

Alice’s Workstation

#2 Alice wants to authenticate

#3 Here’s Alice’s TGT

TGT (ticket-granting ticket)

#5 Alice wants service from V

#6 Here is key + ticket to use

#7 Here is Alice’s ticket for service + key to use

#8 Alice’s request for service is granted, using key supplied

KDC

Server V
Alice types in her user ID and password in unencrypted form into her workstation.
Msg#2: Request for Authentication

- Workstation sends a message to KDC with Alice’s ID (in unencrypted form)
- Many of these messages contain timestamps, for a) liveness, and b) anti-replay

![Diagram showing the sequence of events]

1. Alice logs in with her password.
2. Workstation (W) requests authentication from KDC.
3. KDC responds with Alice's TGT (Ticket Granting Ticket).
4. Workstation requests service from V.
5. KDC grants service from V to Alice.
6. KDC sends key + ticket to use to V.
7. V sends Alice's ticket for service + key to use.
8. Alice's request for service is granted, using key supplied.
Msg#3: Authentication Success

#3. KDC → W: $K_{A,KDC}(ID_A | TS_3 | Lifetime_3 | K_{A-S} | ID_{KDC} | TGT)$

- KDC sends Alice’s workstation a session key and a TGT (ticket-granting ticket)
  - encrypted with the master key shared between Alice and the KDC
- $K_{A,KDC}$ is derived from Alice’s password, used to decrypt session key $K_{A-S}$

![Diagram showing the authentication process]

- #1 Login + Password
- #2 Alice wants to authenticate
- #3 Here’s Alice’s TGT
- #5 Alice wants service from V
- #6 Here is key + ticket to use
- #7 Here is Alice’s ticket for service + key to use
- #8 Alice’s request for service is granted, using key supplied
Msg#3: ... (cont’d)

- The TGT is what allows the KDC to be **stateless**
  - means simpler, more robust KDC design
  - allows replicated KDCs (see later)

- The TGT contains
  - the session key to be used henceforth
  - the user ID (Alice)
  - the **valid lifetime** for the TGT
Msg#4: Alice Requests Service V

- Alice enters a request in the workstation to access the service provided by V

Alice → W: ReqServ(V)

1. Alice logs in with her credentials (Login + Password)
2. Alice wants to authenticate
3. Here’s Alice’s TGT
4. Alice requests service from V
5. Alice wants service from V
6. Here is key + ticket to use
7. Here is Alice’s ticket for service + key to use
8. Alice’s request for service is granted, using key supplied
Msg#5: Workstation Requests Service V

- Workstation sends to the KDC...
  - the TGT previously granted, the server she wishes to request service from
  - an authenticator for this message

Alice → KDC:

#5 W→KDC: TGT | authenticator₅ | TS₅ | Lifetime₅ | IDᵥ

Alice’s Workstation

- #1 Login + Password
- #2 Alice wants to authenticate
- #3 Here’s Alice’s TGT
- #4 Request service from V
- #5 Alice wants service from V
- #6 Here is key + ticket to use
- #7 Here is Alice’s ticket for service + key to use
- #8 Alice’s request for service is granted, using key supplied

Server V

KDC
Msg#5... (cont’d)

- The authenticator is an encrypted timestamp
  - why needed?
  - (reminder: timestamps requires user and KDC clocks to be loosely synchronized)
Msg#6: KDC Generates Ticket

#6 KDC $\rightarrow$ W: $K_{AS}(ID_A|TS_6|\text{Lifetime}_6|K_{AV}|ID_V|TKT_V)$

- KDC decrypts the TGT and...
  - checks that lifetime has not expired; gets the shared key $K_{AS}$
- KDC sends back to workstation
  - identity of the server; a shared key ($K_{AV}$) for Alice and the server; a ticket (TKT) for Alice to present to V

Alice's Workstation

Alice

#1 Login + Password

#4 Request service from V

#2 Alice wants to authenticate

#3 Here's Alice's TGT

#5 Alice wants service from V

#6 Here is key + ticket to use

#7 Here is Alice's ticket for service + key to use

#8 Alice's request for service is granted, using key supplied

KDC

Server V
Msg#6... (cont’d)

\[ K_{V-KDC}(ID_A \mid Addr_A \mid K_{A-V} \mid Lifetime_{TKT} \mid TS_{TKT} \mid ID_V) \]

- The ticket contains
  - ID of the initiating user
  - shared key \( K_{A-V} \)
  - lifetime of the ticket
Msg#7: Workstation Contacts Server

#7 W→V: \( ID_V \mid TKT_V \mid \text{authenticator}_7 \)

- **Message contains**
  - ticket (from the KDC); authenticator

![Diagram of message flow](image)

- #1 Login + Password
- #2 Alice wants to authenticate
- #3 Here’s Alice’s TGT
- #4 Request service from V
- #5 Alice wants service from V
- #6 Here is key + ticket to use
- #7 Here is Alice’s ticket for service + key to use
- #8 Alice’s request for service is granted, using key supplied
Msg#7... (cont’d)

\[ K_A - V (ID_A \mid \text{Chksum}_{\text{auth7}} \mid \text{TS}_{\text{auth7}}) \]

- Authenticator is valid for 5 minutes
  - loose synchronization required
  - replay attack possible for short period if server does not store previous authenticators
Msg#8: Server Authenticates to Alice

#8 V→W: $K_{A-V}(\text{Chksum}_{auth7} + 1)$

- Reply to Alice’s workstation contains
  - checksum sent by Alice, incremented by 1

Alice → #1 Login + Password

Alice’s Workstation → #2 Alice wants to authenticate

Alice’s Workstation → #3 Here’s Alice’s TGT

Alice’s Workstation → #4 Request service from V

Alice’s Workstation → #5 Alice wants service from V

Alice’s Workstation → #6 Here is key + ticket to use

Alice’s Workstation → #7 Here is Alice’s ticket for service + key to use

Alice’s Workstation → #8 Alice’s request for service is granted, using key supplied

KDC → #5 Alice wants service from V

KDC → #6 Here is key + ticket to use

Server V → #8 Alice’s request for service is granted, using key supplied
Done!

1. Alice has authenticated to KDC (which is trusted by server)
2. Server has authenticated to Alice
3. A session key has been negotiated, for encryption, message authentication, or both.