Kerberos
Design Goal

• Allow the sharing of a secret key between any two users who only share a password with a trusted server.
Needham-Schroeder protocol

Alice, Bob, $N_A$

$\{N_A, K_{AB}, Bob, \{K_{AB}, Alice\}\}_{K_{BS}}_{K_{AS}}$

$\{K_{AB}, Alice\}_{K_{BS}}$

$\{N_B\}_{K_{AB}}$

$\{N_B - 1\}_{K_{AB}}$
• Needham-Shroeder protocol is susceptible to a replay attack
• An adversary only has to record the message Alice sends to Bob to cause the same session key to be used again.
Kerberos 4
Obtaining session key and TGT

Alice                      Server (AS)

\[ \text{[AS_REQ]} \quad \text{Alice needs TGT} \]
\[ \text{[AS_REP]} \quad \{S_A || ID_{TGS} || TS_1 || \text{Lifetime}_1 || TGT\}_K^A \]

TGT is called a ticket-granting ticket
TGT = \{S_A || ID_A || AD_A || ID_{TGS} || TS_1 || \text{Lifetime}_1\}_K^{TGS}
Kerberos 4
Talking to remote node

Alice

Server (TGS)

[TGS_REQ]  Bob, TGT, \{ID_A \parallel AD_A \parallel TS_2\}_{SA}

[TGS_REP]  \{K_{AB} \parallel ID_B \parallel TS_3 \parallel Ticket\}_{SA}

Ticket = \{K_{AB} \parallel ID_A \parallel AD_A \parallel ID_B \parallel TS_3 \parallel Lifetime_3\}_{KB}
Kerberos 4
Talking to remote node

Alice          Bob

[AP_REQ] Ticket || \{timestamp\}_{K_{AB}}

[AP_REP] \{timestamp + 1\}_{K_{AB}}
Duplicate KDCs and Realms

- Multiple KDCs to avoid single point of failure and bottleneck
- Consistency maintained by having a single master and multiple ‘read-only slaves’
- KDCs are split in different realms
- All KDCs in a given realms have same master key
- KDCs of different realms have different master keys and different user database.
- A ‘name’ in Kerberos 4 consists of Name, Instance and Realm
Interrealm communication

• For Alice@Realm1 to communicate with Bob@Realm2, Realm2 has to be a ‘user’ in Realm1

Alice \hspace{1cm} KDC@R1 \hspace{1cm} KDC@R2

- [TGS_REQ] (Alice@R1,R2)
- [TGS_REP] (Session key to R2)

- [TGS_REQ] (Alice@R1, Bob@R2) in new session key
- [TGS_REP] (Session key to Bob@R2)
Privacy and Integrity in Kerberos 4

- Many Kerberos messages need to be private and correct (all messages containing keys for example).
- Most common method to ensure both privacy and integrity is to first encrypt then MAC, but this tends to require two passes on the message.
- To avoid this, the designers of Kerberos invented their own block cipher mode of operation: Plaintext (or Propagating) Cipher Block Chaining.

- It’s actually pretty easy to break integrity.
Privacy and Integrity in Kerberos 4

CBC
Privacy and Integrity in Kerberos 4

IV

$\begin{array}{c}
\text{DES} \\
c_1
\end{array}$

$\begin{array}{c}
\text{DES} \\
c_2
\end{array}$

$\begin{array}{c}
\text{DES} \\
c_3
\end{array}$

$\begin{array}{c}
\text{DES} \\
c_4
\end{array}$

PCBC
Integrity Only

• Use a modified Jueneman checksum appended at the end of the message

• Never broken, but never proven secure either.
Network Address in Ticket

- Prevent theft of authenticator and ticket within 5 minutes (and unencrypted session to Bob)
- Prevent delegation…but might be desirable
Why Bob name/instance

• If many services at the same location, immediately know which key to use
• No confusion if many services use the same key
Encrypted Stuff

- Kerberos version, message type (8)
- message type (6)
- length of encrypted stuff
- encrypted stuff
  - length of data, data
  - 5-millisecond timestamp
  - sender’s IP address
  - direction flag, timestamp

- Similar for integrity-checked data (with checksum at the end)
Attacks on Kerberos 4

• Offline dictionary attack simply by requesting a TGT
• Integrity not guaranteed for encrypted messages
• Many implementations have buffer overflow vulnerabilities
Other Limitations of Kerberos 4

• Name and Instance fields are limited to 40 bytes
• Lifetime of ticket is a 1 byte (intervals of 5 minutes) → maximum lifetime of ~21 hours
• Double encryption in AS_REP and TGT_REP
Kerberos 5

• Fixes some design flaws in Kerberos 4
• Adds new functionalities, allows greater flexibility

• spec now 133 pages long
Kerberos 5

- Use ANS.1 syntax
- Allows optional fields, fields of variable length
- Adds overhead (4 byte address in ver.4 becomes 11 bytes in ver.5)

- Name no longer limited to 40 characters, no longer need for Instance field.
Cleanups

• Ticket no longer doubly-encrypted
• More flexible cryptographic algorithms
• Fix integrity protection (not PCBC or Jueneman)
Preventing Dictionary Attack

Requires an authenticator to obtain a TGT

Alice  

Server (AS)

[AS_REQ] Alice needs TGT, \{timestamp\}_{K_A}

[AS REP] \{S_A\}_{K_A} \{“Alice”,S_A\}_{K_{TGS}}

Note that this is still vulnerable to eavesdropper (but this is unavoidable)
Preventing Dictionary Attack

- A user could still get a ticket encrypted with another user’s key by making a TGS_REQ on that user
- In Kerberos 5, KDC database contains flag to prevent issue of a ticket to users whose key is derived from a password
Delegation of Rights

• In Kerberos 4, Alice could not give her session key and TGT to Bob since TGT contains network address
  – Had to either give your password or your TGT or session key (problem: tickets contain IP addresses)

• In Kerberos 5, Alice can request a TGT with a network address other than her own

• TGTs can also be proxiable, ie can be used to obtain tickets for a different address (but the TGT can only be used by Alice)
Ticket Lifetime

• In Kerberos 4, 4 byte start time, 1 byte lifetime (~21 hours)
• In Kerberos 5, tickets have start time, end time, authtime and renew-till (each 17 bytes)
• Lifetime now just about unlimited, but time in seconds (added 5 byte for microseconds)
• Allow renewable tickets and postdated tickets.
Different Realm, Different Key

• In Kerberos 4, the permanent key depended on password only
  → same password in different realms meant
    same key

• In Kerberos 5, the realm is name is part of function for key derivation
Cryptographic Algorithms

• Integrity: Kerberos 5 abandoned the Jueneman checksum for better algorithms, but not yet ‘provably secure algorithms’
• rsa-md5-des (r)
• des-mac (r) des-mac-k (r)
• rsa-md4-des (o) rsa-md4-des-k (o)
Interrealm Authentication

- Authentication through a chain of KDC is allowed in Kerberos 5
- Added a TRANSITED field that lists all the KDCs in the chain
- Bob can decide on whether or not to accept a ticket according to that list
- KDCs often organized in a tree – realms share keys with parent and children
- Path through lowest common ancestor (finding path up to user)
Double TGT Authentication

• When Alice makes a TGS_REQ, she sends both her TGT and Bob’s TGT, then the ticket to Bob in TGS_REP is encrypted with Bob’s session key
  – To avoid use of permanent key
  – In case Bob ‘forgets’ his master key but remembers his session key

• Inspired by XWINDOWS.
Public Keys

• If users had public keys, could use them to authenticate themselves to server instead of password
• But if everyone has a public key with trusted certificate, what’s the use of Kerberos at all?
2005: Overhaul of Kerberos 5

• New crypto must be supported:
  – AES256-CTS-HMAC-SHA1-96
  – HMAC-SHA1-96-AES256