EurOpen, Spring 1991

The Evolution of the *Kerberos* Authentication Service

John T. Kohl

Project Athena
Digital Equipment Corporation
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

jtkohl@mit.edu
jtkohl@irc.cbm.dec.com
info-kerberos@mit.edu

Overview of presentation

- The problems Kerberos addresses
- Protocol version 4 review
- Version 5 changes and motivations
- Implementation, applications, migration, future

What Kerberos addresses

- A *Network Authentication* service
- designed for ubiquitous "open" network
- Kerberos applications want to:
    - authenticate client and server
    - establish shared secret key
    - use secret key for protection

Definitions & Terminology

Terms:

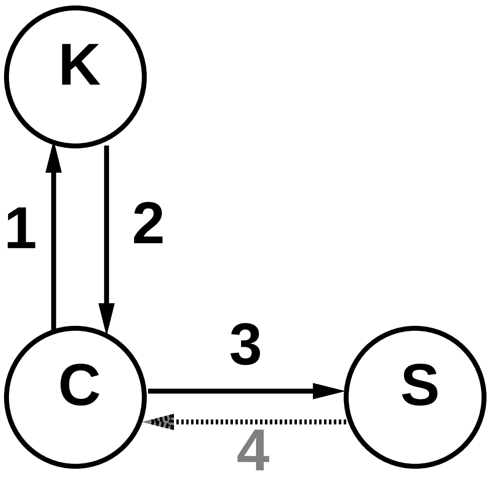
- authentication
- client
- server
- private key --> K_X
- session key --> $K_{c,s}$
- encryption --> $\{abc\}K_X$
- ticket to server --> $T_{c,s}$
- authenticator --> $A_{c,s}$

The Model

Needham and Schroeder, 1978 (with enhancements):

- **Trusted Third Party Network Authentication Service**
  - **basis of trust**
  - **keeps keys secret**
- **Key Distribution Center**
- **Private Key Encryption (Two parties share a secret)**
- **synchronized system clocks**

Normal message exchange (v4)



1. c,s
2. rep = $\{T_{c,s}, \mathbf{K_{c,s}}\}\ K_c$;
   [cache $\mathbf{T_{c,s}}$, $\mathbf{K_{c,s}}$]
3. $\mathbf{T_{c,s}}$, $\mathbf{A_{c,s}}$
4. [optional] $\{f(\textbf{data})\}\ \mathbf{K_{c,s}}$

$\mathbf{T_{c,s}}$ = { s, c, *addr*, time, life, $\mathbf{K_{c,s}}$ } $\mathbf{K_s}$
$\mathbf{A_{c,s}}$ = { c, *addr*, time } $\mathbf{K_{c,s}}$

Unsuitable/undesirable features

Unsuitable:

- Internet address, and only one
- requires US Data Encryption Standard (DES)
- non-standard cryptographic checksum

Undesirable:

- Byte order dependence
- Double ticket encryption

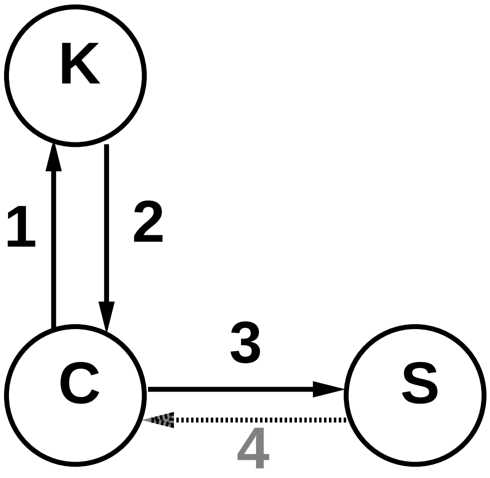
Desired new features

- user-to-user support (no K_S)
- longer-lived tickets
- true replay detection
- password-to-key parameterization
- per-association keys
- more flexible names
- authorization data restrictions in ticket

Result: new protocol

- ASN.1 encodings
  - solves byte order
  - sufficient change to redo all messages
- flags in each ticket
- typed data in protocol
- pre-authentication data fields

Normal message exchange (v5)



1. c,s
2. rep = $T_{c,s}$, {s, $K_{c,s}$} K_c;
   [cache $T_{c,s}$, $K_{c,s}$]
3. $T_{c,s}$, $A_{c,s}$
4. [optional] {*f(*data*)*} $K_{c,s}$

$T_{c,s}$ = s, { c, *addr(s), times*, $K_{c,s}$ } K_s
$A_{c,s}$ = { c, time, *key* } $K_{c,s}$

Implementation notes

- completely new implementation
- Kerberos v5 library
    - replaceable parts
    - os-dependent pieces isolated
- DES library
- KDC database library
- v4/v5 glue libraries
- 700 files, 80,000 lines of C code

Applications in MIT distribution

- telnet
- Post Office (movemail client, popper daemon)
- Sample programs
    - TCP-based
    - UDP-based
    - user-to-user code

Migrating applications

- GSSAPI
- v4/v5 link-level glue library
- KDC v4 protocol handling
- both v4 & v5 in same program
- separate programs

Future areas

implementation:

- C++ would be nice
- Is OS support done right?
- remote administration
- validation suite

Protocol/interoperability:

- validation suite
- public key support
- hand-held authentication devices
- Official protocol registration

EurOpen, Spring 1991

The Evolution of the *Kerberos* Authentication Service

John T. Kohl

Project Athena
Digital Equipment Corporation
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

jtkohl@mit.edu
jtkohl@irc.cbm.dec.com
info-kerberos@mit.edu