# Kerberos


**Alexandra Ellwood**
**MIT Kerberos for Macintosh Team**
**lxs@mit.edu**

# Agenda

- Overview of Kerberos
- The Kerberos protocol
- Kerberos for Macintosh 4.0
- Adding Kerberos to your application
- Demo
- Q&A

# What Kerberos Is

- A secure network authentication system
  - Authentication vs Authorization vs Encryption
- Proves who you are
- Uses a trusted third party model security model

# Who Uses Kerberos

- Government
- Large corporations
- Higher Education
- Every Windows 2000 installation

# Why Do They Use It?

- Mutual Authentication
- Single Sign on
- More secure than encrypted password exchanges
- Integrates well into centralized systems
- Multi-platform: Mac OS, Windows, most Unixes
- Source freely available

# A Brief History of Kerberos

- Invented in the 1980's at MIT as part of Project Athena
- Protocol lineage: v4, v5
- Implementation history on the Macintosh
  - KClient, Authman, MacLeland (v4)
  - MIT/Cygnus (v5)
  - Kerberos for Macintosh (v4 and v5)

# Kerberos Terms

- Realm
  - The administrative unit protected by a KDC
- Ticket
  - An authentication token
- Ticket-granting ticket (TGT)
  - An initial ticket that grants the user permission to get service tickets
- Service ticket
  - A ticket that authenticates the user to a particular service (e.g. ftp, mail, telnet)

# Kerberos Protocols

- Two versions of the protocol: v4 and v5
- We will discuss the Kerberos v4 protocol
- Kerberos v5 is conceptually similar to v4. . .

but too complicated to discuss in this session

# Kerberos v4 Protocol Notation

- $K_x$ = a secret key "x"
- $\{abc\}K_x$ = "abc" encrypted with key $K_x$

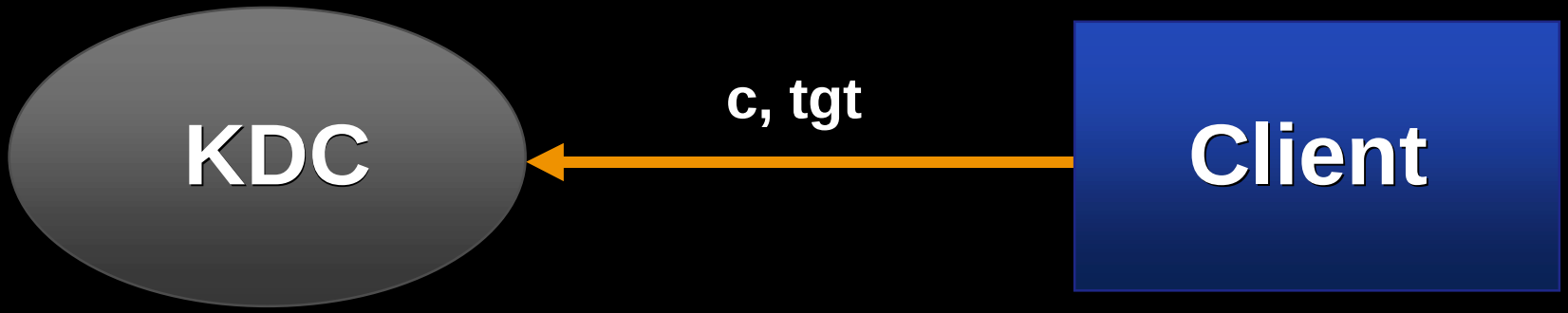
- c = the client name (the user's principal)
- $K_c$ = the client key (a one-way hash of the user's password)
- s, $K_s$ = the service name and key
- tgt, $K_{tgt}$ = the ticket granting service (TGS) and key
- $T_{c,s} = \{s, c, address, time, lifetime, K_{c,s}\}K_s$
- $K_{c,s}$ = the session key for service "s"

# Ticket Granting Ticket (TGT)

- Also called an “initial ticket” because it’s the first ticket a client gets
- Proves that the client is allowed to get tickets for other services
- Acts as a substitute for password
- Only valid for a limited period of time

# Getting a TGT

- Client sends a request to the KDC with the client name and the TGT service name




# Getting a TGT

- The KDC returns the TGT ($\{T_{c,tgt}\}K_{tgt}$) and session key ($K_{c,tgt}$) to the client encrypted with client's key

# Getting a TGT

- The client uses its key (a one-way hash of the password) to extract the TGT ($\{T_{c,tgt}\}K_{tgt}$) and session key ($K_{c,tgt}$)

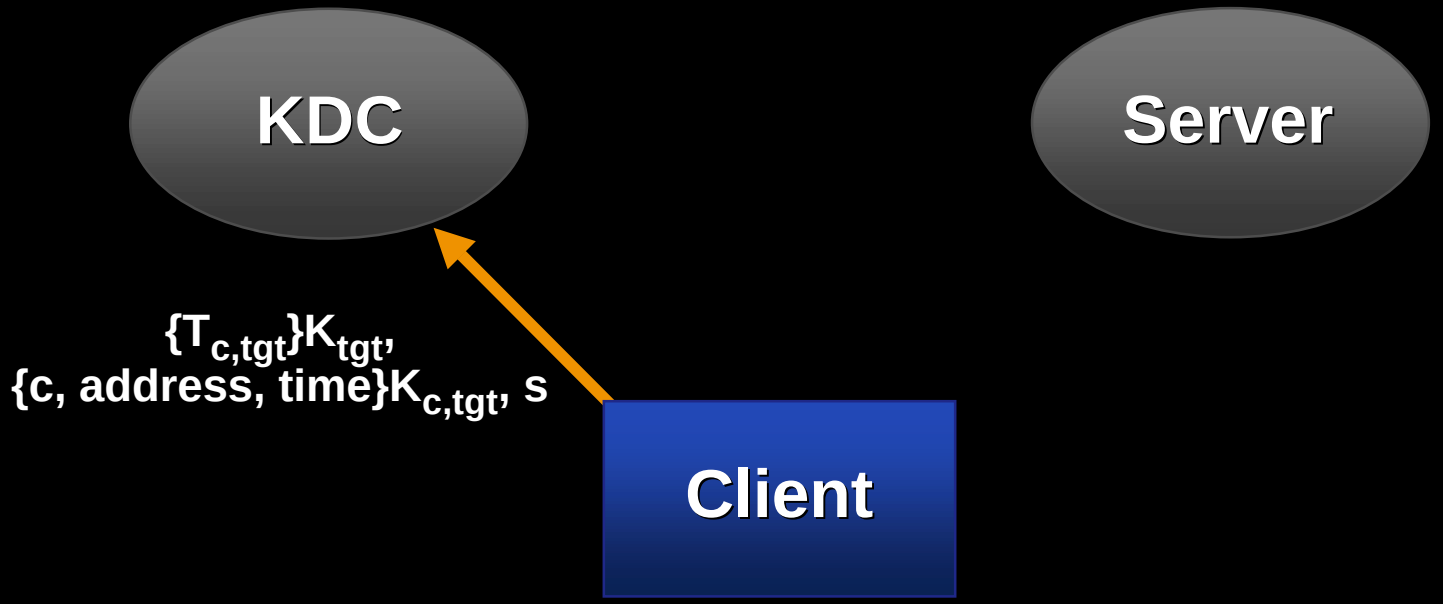


# Service Ticket

- A ticket used by the client to get access to particular service or server (e.g. ftp, mail, telnet)
- Contains a session key shared between the client and server, which they can use to encrypt data exchanges
- Protected from replay attacks by the authenticator: the client's name, local address and current time encrypted in the session key
- Only valid for a limited period of time

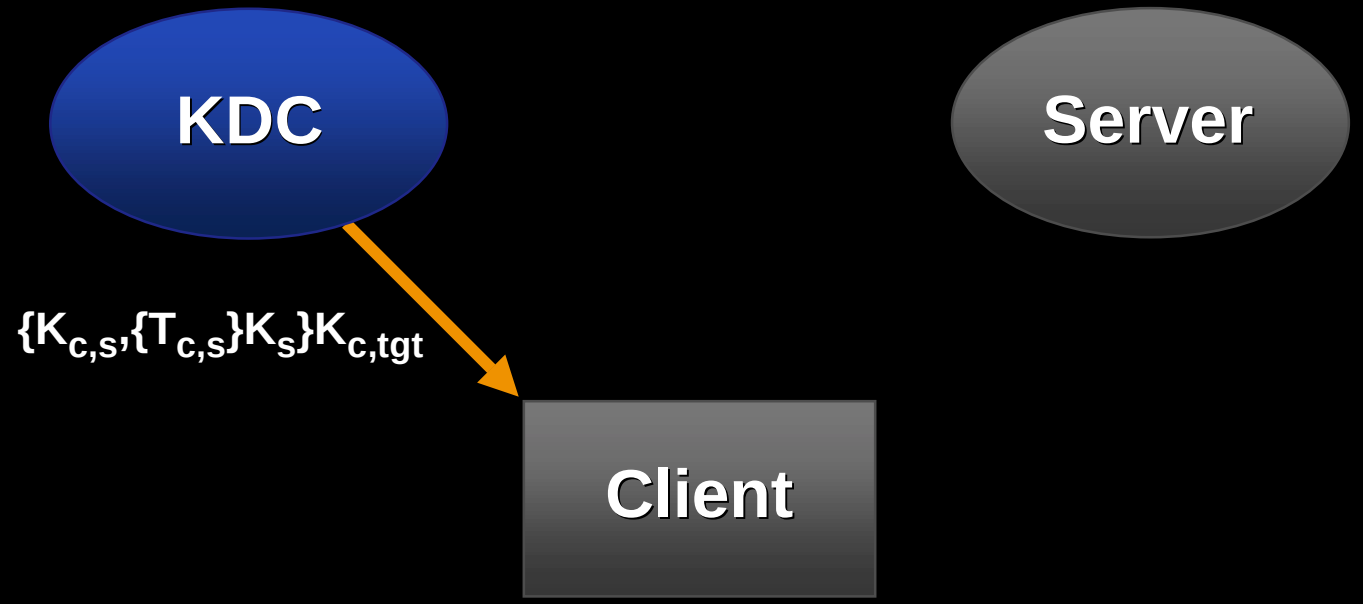
# Getting a Service Ticket

- The client sends its TGT ($\{T_{c,tgt}\}K_{tgt}$) the authenticator ($\{c, address, time\}K_{c,tgt}$) and the service name to the KDC



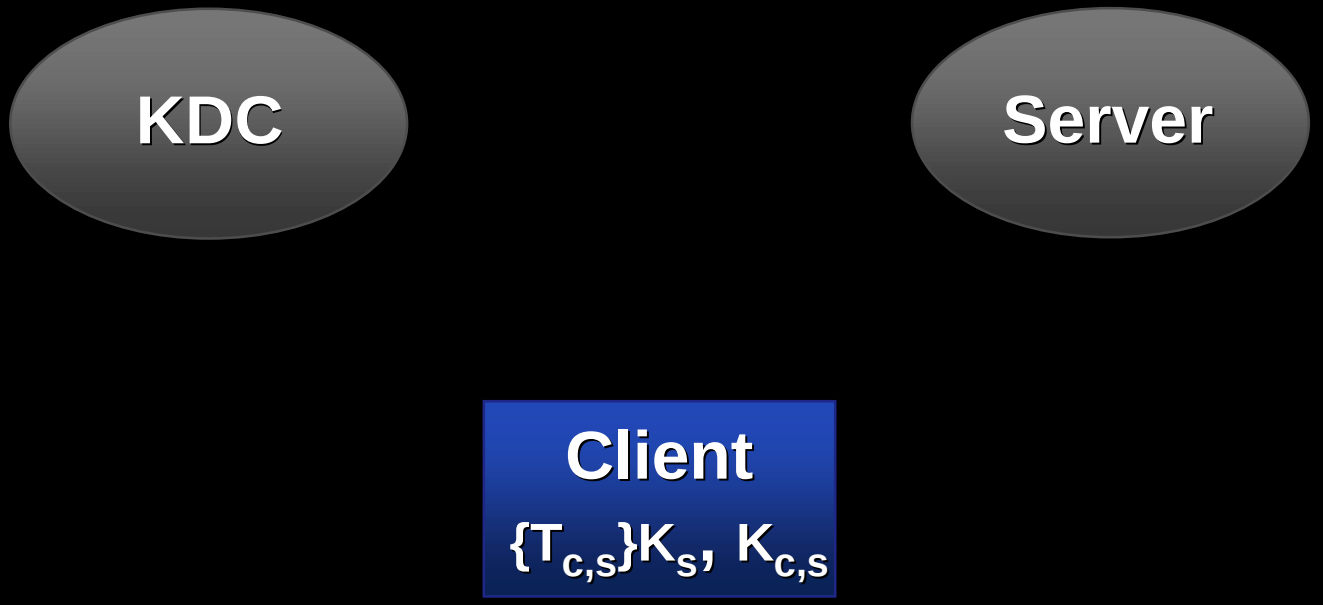
# Getting a Service Ticket

- KDC returns the service ticket ($\{T_{c,s}\}K_s$) and the service session key ($K_{c,s}$) encrypted in the client's TGS session key



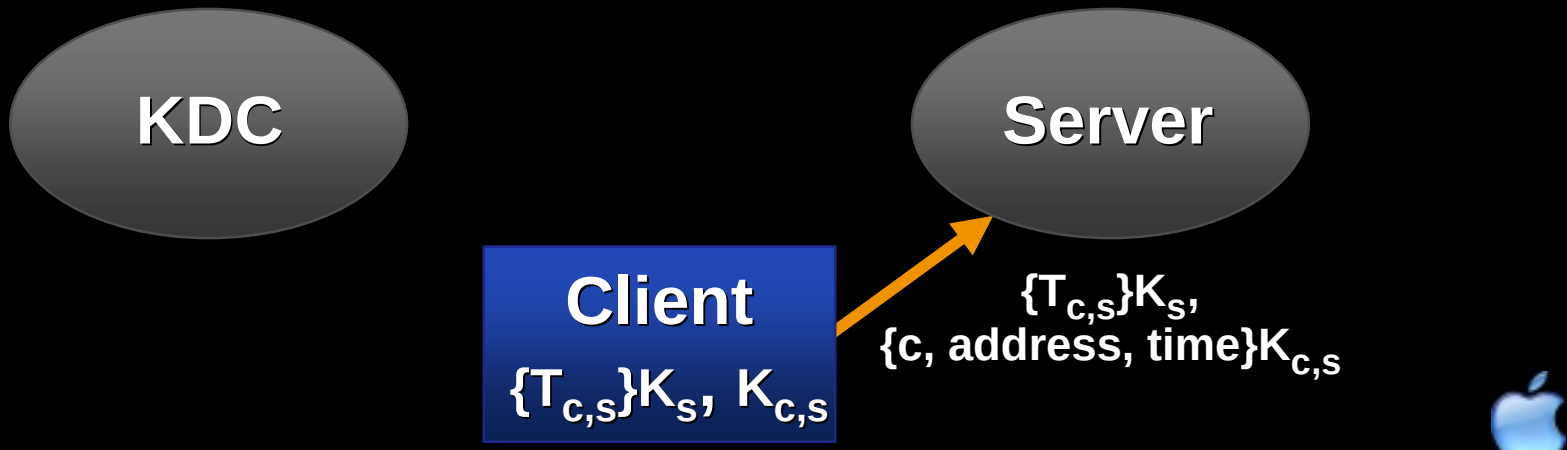
# Getting a Service Ticket

- The client uses the session key from its TGT to extract the service ticket ($\{T_{c,s}\}K_s$) and the service session key ($K_{c,s}$)



# Connecting to a Service: One-Way Authentication

- The client sends its service ticket ($\{T_{c,s}\}K_s$) and the authenticator ($\{c, address, time\}K_{c,tgt}$) to the server

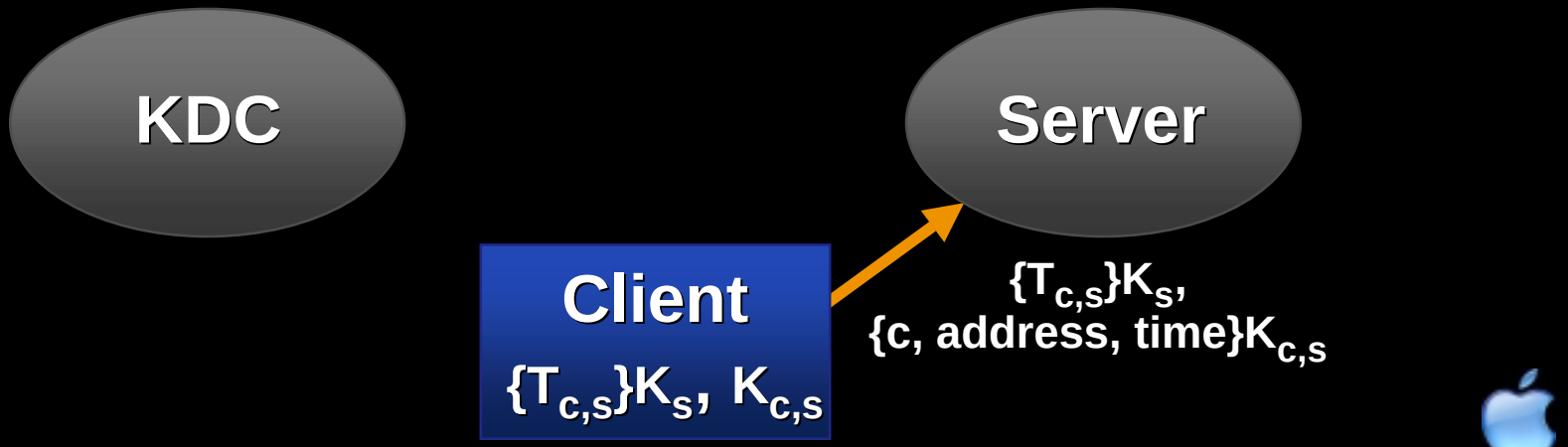


# Connecting to a Service: One-Way Authentication

- The server receives the service ticket $(\{T_{c,s}\}K_s)$ and authenticates the client
- The server may use the session key from the service ticket $(K_{c,s})$ to encrypt subsequent traffic

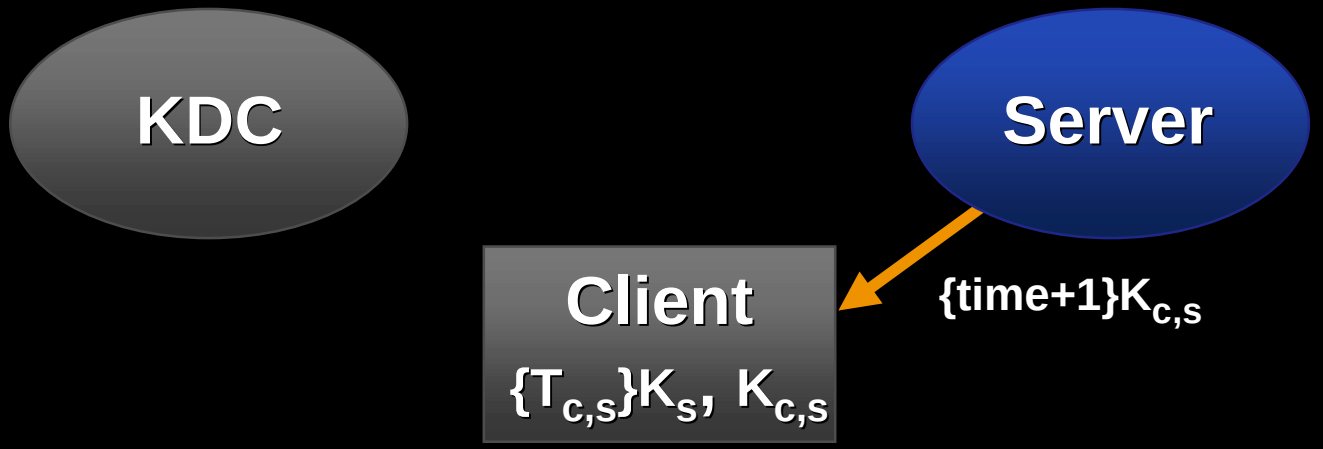
# Connecting to a Service: Mutual Authentication

- The client sends its service ticket ($\{T_{c,s}\}K_s$) and the authenticator ($\{c, address, time\}K_{c,tgt}$) to the server



# Connecting to a Service: Mutual Authentication

- The server returns the time from the authenticator, incremented by 1, encrypted with session key from service ticket ($\{time+1\}K_{c,s}$)
- The server may also use the session key ($K_{c,s}$) to encrypt subsequent traffic




# Kerberos for Macintosh 4.0

**Miro Jurišić**
**MIT Kerberos for Macintosh Team**
**meeroh@mit.edu**

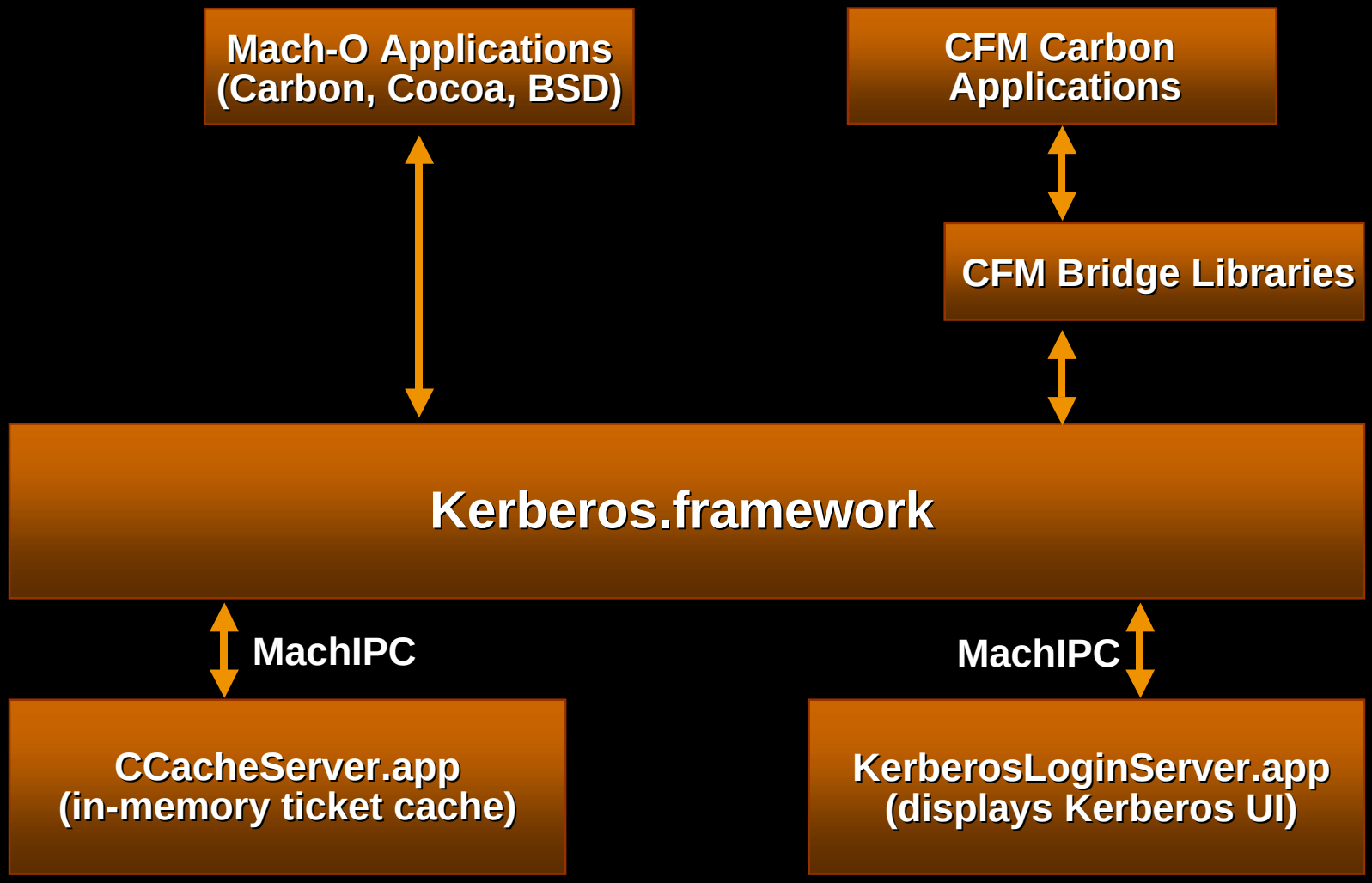
# Kerberos for Macintosh 4.0

- Supports both Mac OS X and Mac OS 9
- Provides Kerberos libraries on Mac OS X to
  - Carbon
  - Cocoa
  - Command Line
- Classic Mac OS support includes
  - Mac OS 8.1 and later
  - CarbonLib 1.0.4 and later

# Why Not Just Unix Kerberos?

- Does not include APIs that existing Mac applications depend on
- Does not provide functionality for the level of user interaction Mac users demand
- Does not support all runtime environments of Mac OS X

# KfM 4.0 Architecture



# Kerberos v4

- KClient 3.0
  - Compatibility libraries
  - New version 3 APIs
- Cygnus Kerberos v4 implementation
  - The final Cygnus release (version 96q4)
  - Modifications to support KerberosLoginLibrary
- Applications can use either API for v4 calls
  - KClient 3.0 recommended
  - Historic v4 API only for porting existing Kerberos code

# Kerberos v5

- MIT Kerberos v5 1.2.2
  - Includes 3DES
  - Improved hardware preauthentication support
  - Numerous buffer overrun problems fixed
  - Partial IPv6 support
- GSS API
  - Supports RFC #1509 - “Generic Security Service API : C-bindings” & RFC #1964 - “The Kerberos Version 5 GSS-API Mechanism”

# Credentials Cache

- Bottom layer—where the tickets/credentials are stored
- Stored in memory as opposed to on disk
- Provides API for accessing the credentials used by KClient, v4, v5 libraries

# Kerberos Login Library

- Simplifies the process of getting Kerberos TGTs
  - Allows you to get or destroy tickets with a single call
  - Transparently handles v4 and v5 initial tickets
- Also provides API for adjusting Kerberos Login Dialog options
- Used by programs that need to perform actions on TGTs, such as administrative applications

# Work in Progress

- Directory Services plug-in
  - Apple and MIT are working to define standard specification
- Login Authenticator
- Shared tickets between Classic and Mac OS X
- More comprehensive integration on X

# How Can I Help?

- Download the KfM 4.0 public betas and report problems
- Port existing “Kerberized” applications
- Add Kerberos support to your product

# How to Kerberize an Application

- Find the protocol specification (RFCs, etc.)
- Determine which API to use (KClient, Kerberos v5, GSS API, etc.)
- Connect the API to the protocol
  - The data returned by the API usually needs to be massaged into a form required by the protocol

# Kerberizing FTP

- Protocol: RFC 2228, RFC 959
- Supports Kerberos v4 and GSS
  - Use KClient for Kerberos v4
  - Use GSS API for Kerberos v5

# Kerberizing FTP: Kerberos v4

- Session management
  - KClientNewClientSession
- Authentication
  - KClientGetAuthenticatorForService
  - KClientVerifyProtectedServiceReply
- Encryption & decryption
  - KClientEncrypt
  - KClientDecrypt
- Connecting the API to the protocol
  - Base 64 encoding

# Kerberizing FTP: GSS

- Session management & authentication
  - gss_init_sec_context
- Encryption & decryption
  - gss_wrap
  - gss_unwrap
- Connecting the API to the protocol
  - Base 64 encoding

# How Do I Get Started?

- Download developer release of Kerberos for Macintosh 4.0 from MIT
- Current release includes nearly all needed functionality for Mac OS X application developers
- Use on-line resources – web documentation, mailing lists and newsgroups for assistance

# Demo


## Kerberos for Macintosh 4.0

# For More Information

- MIT Kerberos for Macintosh
  - **http://web.mit.edu/macdev/www/kerberos.html**
- Kerberos Papers & Documentation
  - **http://web.mit.edu/kerberos/www/papers.html**
- Internet RFCs
  - **http://www.ietf.org/rfc.html**
- Kerberos Development List
  - **krbdev-request@mit.edu**
- Usenet
  - **comp.protocols.kerberos**

# Who to Contact

**Kerberos for Macintosh Team**
Massachusetts Institute of Technology
**macdev@mit.edu**

**Kerberos Development Team**
Massachusetts Institute of Technology
**krbdev@mit.edu**

# Q&A


**Miro Jurišić and Alexis Ellwood**
MIT Kerberos for Macintosh Team
**macdev@mit.edu**


Wendy's Was Dangerously Close?
We Want Discount Computers?
Wireless Workstations Desire Current?
Windows Won't Do Crap?
Wild Wacky Dangerous Code?

We Will Demand Caffeine!