CSci555: Advanced Operating Systems
Lecture 6 – September 30, 2005
Security Architecture

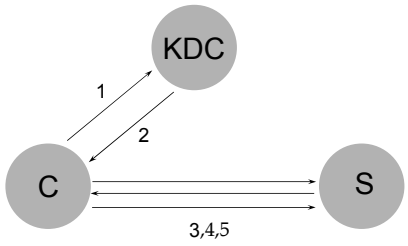
Dr. Dongho Kim
Dr. Ryutov
University of Southern California
Information Sciences Institute

Key distribution

- **Conventional cryptography**
  - **Single key shared by both parties**
- **Public Key cryptography**
  - **Public key published to world**
  - **Private key known only by owner**
- **Third party certifies or distributes keys**
  - **Certification infrastructure**
  - **Authentication**

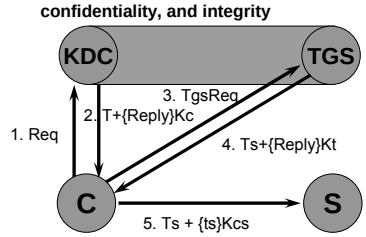
Authentication w/ Conventional Crypto

- **Kerberos or Needham Schroeder**



Kerberos

Third-party authentication service
- **Distributes session keys for authentication, confidentiality, and integrity**



Public Key Cryptography (revisited)

- **Key Distribution**
  - **Confidentiality not needed for public key**
  - **Solves n^2 problem**
- **Performance**
  - **Slower than conventional cryptography**
  - **Implementations use for key distribution, then use conventional crypto for data encryption**
- **Trusted third party still needed**
  - **To certify public key**
  - **To manage revocation**
  - **In some cases, third party may be off-line**

Certificate-Based Authentication

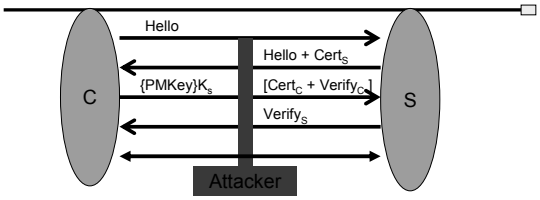
Certification authorities issue signed certificates
- **Banks, companies, & organizations like Verisign act as CA's**
- **Certificates bind a public key to the name of a user**
- **Public key of CA certified by higher-level CA's**
- **Root CA public keys configured in browsers & other software**
- **Certificates provide key distribution**

Certificate-Based Authentication (2)

Authentication steps

- **Verifier provides nonce, or a timestamp is used instead.**
- **Principal selects session key and sends it to verifier with nonce, encrypted with principal's private key and verifier's public key, and possibly with principal's certificate**
- **Verifier checks signature on nonce, and validates certificate.**

Secure Sockets Layer (and TLS)



Encryption support provided between
Browser and web server - below HTTP layer
Client checks server certificate
Works as long as client starts with the correct URL
Key distribution supported through cert steps
Authentication provided by verify steps

Trust models for certification

- **X.509 Hierarchical**
  - **Single root (original plan)**
  - **Multi-root (better accepted)**
  - **SET has banks as CA's and common SET root**
- **PGP Model**
  - **"Friends and Family approach" - S. Kent**
- **Other representations for certifications**
- **No certificates at all**
  - **Out of band key distribution**
  - **SSH**

Global Authentication Service

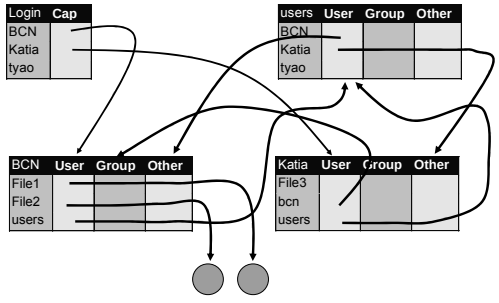
- **Pair-wise trust in hierarchy**
  - **Name is derived from path followed**
  - **Shortcuts allowed, but changes name**
  - Exposure of path is important for security
- **Compared to Kerberos**
  - **Transited field in Kerberos - doesn't change name**
- **Compared with X.509**
  - **X.509 has single path from root**
  - **X.509 is for public key systems**
- **Compared with PGP**
  - **PGP evaluates path at end, but may have name conflicts**

Capability Based Systems - Amoeba

"Authentication not an end in itself"

- **Theft of capabilities an issue**
  - **Claims about no direct access to network**
  - **Replay an issue**
- **Modification of capabilities a problem**
  - **One way functions provide a good solution**
- **Where to store capabilities for convenience**
  - **In the user-level naming system/directory**
  - **3 columns**
- **Where is authentication in Amoeba**
  - **To obtain initial capability**

Capability Directories in Amoeba



Security Architectures

- **DSSA**
  - **Delegation is the important issue**
    - **Workstation can act as user**
    - **Software can act as workstation - if given key**
    - **Software can act as developer - if checksum validated**
  - **Complete chain needed to assume authority**
  - **Roles provide limits on authority - new sub-principal**
- **Proxies - Also based on delegation**
  - **Limits on authority explicitly embedded in proxy**
  - **Works well with access control lists**

Distributed Authorization

- **It must be possible to maintain authorization information separate from the end servers**
  - **Less duplication of authorization database**
  - **Less need for specific prior arrangement**
  - **Simplified management**
- **Based on restricted proxies which support**
  - **Authorization servers**
  - **Group Servers**
  - **Capabilities**
  - **Delegation**

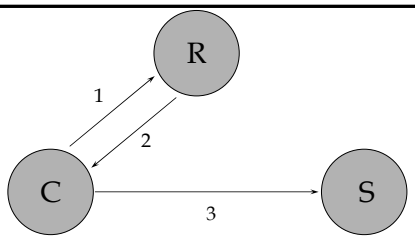
Proxies

- **A proxy allows a second principal to operate with the rights and privileges of the principal that issued the proxy**
  - **Existing authentication credentials**
  - **Too much privilege and too easily propagated**
- **Restricted Proxies**
  - **By placing conditions on the use of proxies, they form the basis of a flexible authorization mechanism**

Restricted Proxies

- **Two Kinds of proxies**
  - **Proxy key needed to exercise bearer proxy**
  - **Restrictions limit use of a delegate proxy**
- **Restrictions limit authorized operations**
  - **Individual objects**
  - **Additional conditions**

Authorization and Group Services



1. Authenticated authorization request (operation X)
2. [operation X only]R, {Kproxy} Ksession
3. [operation X only]R, authentication using Kproxy

Central Authorization

- **Authorization server uses extended ACLs**
  - **Conditions are not evaluated, but instead attached to credentials**
- **Groups implemented by auth server**
  - **Server grants right to assert group membership**
- **Application servers configured to use authorization server**
  - **Minimal local ACL**
  - **Can use multiple Authorization servers**

Applied Security

- **Electronic commerce**
  - **SSL Applies authentication and encryption**
  - **NetCheque applies proxies**
  - **SET applies certification**
  - **End system security a major issue**
- **What we have today**
  - **Firewalls**
  - **Web passwords, encryption, certificates**
  - **Windows 2000 uses Kerberos**

Trust Negotiation

- Problem: Identity is not relevant
- Solution: Access control decisions are based on attributes of both the client and server (mutual trust)
  - Client attributes: citizenship, security clearance, job classification, etc.
  - Server attributes: privacy policy satisfaction, result of recent security audit, etc.
- Credentials and Policies may contain sensitive information and should be treated as protected resources
- Trust Negotiation: The process of establishing trust between strangers in open systems based on the attributes of the participants