Authorization

Authorization: Two Meanings

- Determining permission
  - Is principal P permitted to perform action A on object U?
- Adding permission
  - P is now permitted to perform action A on object U
- In this course, we use the first sense

Access Control

- Who is permitted to perform which actions on what objects?
- Access Control Matrix (ACM)
  - Columns indexed by principal
  - Rows indexed by objects
  - Elements are arrays of permissions indexed by action
- In practice, ACMs are abstract objects

Instantiations of ACMs

- Access Control Lists (ACLs)
  - For each object, list principals and actions permitted on that object
  - Corresponds to rows of ACM
  - Example: Kerberos admin system

Instantiations of ACMs

- Capabilities
  - For each principal, list objects and actions permitted for that principal
  - Corresponds to columns of ACM
  - Example: Kerberos restricted proxies
- The Unix file system is an example of…?

Problems

- Permissions may need to be determined dynamically
  - Time
  - System load
  - Relationship with other objects
  - Security status of host

Problems

- Distributed nature of systems may aggravate this
  - ACLs need to be replicated or centralized
  - Capabilities don't, but they're harder to revoke
- Approaches
  - GAA
  - Agent-based authorization

Agent-Based Authorization

- When object created on a host H, agent Q created along with it
- Agents distributed to clients
  - Either directly, or through agent server
- Client on host G instantiates agent for principal P, submits it to H as Q/P@G

Agent-Based Authorization

- Relieves scaling issues with ACLs
- Q is typically mobile code and data
  - Needs to be integrity-protected
  - May be confidentiality-protected
  - Agent environment on H must be trusted

Revocation in Agent-Based Systems

- Timeout-based
- Harder for malicious agents
  - Hosts must send RCLs to other hosts and/or principals
  - Must maintain their own RCL to restrict or deny incoming agents