Access Control

Model: access matrix
rows represent principals
columns represent objects

Implementation:

1. using access control lists ACLs
   keep non-null entries
   of column j with object j

2. using capabilities
   keep non-null entries
   of row i with principal i

ACL entry = principal name + access rights

capability = object name + access rights

Assume managers for the various types of object

On an access request, the manager must check
that the requesting principal
has the appropriate right to access the object

- checks ACL list entry

- principal passes capability with request

ACLs versus capabilities

ACLs subtle expression of policy
may be on an individual basis

slow to check (don't scale)
- if expressive
- if many principals and groups

awkward to delegate rights
e.g. to a printer

easy to revoke
- change ACL entries
may not have instant effect

capabilities

quick to check (scale)

problems/issues - see below

Capability-based access control - issues

1. protection of capabilities

   must prevent construction, tampering, theft

2. propagation

   can principals pass on copies?
   must they ask the object manager?
   if so - how is this enforced?

3. delegation

   example of propagation
   often with restricted rights
   for a limited time

4. (selective) revocation

   if the access control policy changes and
   certain principals should lose their rights
   their capabilities should be revoked
   without affecting others.

Capabilities in centralised systems

Several capability architectures were
designed and built, e.g. Plessey PP250, CAP

Capabilities can be protected
by hardware and/or the OS

* held in protected OS space per process

* held in typed capability segments in user space
  - operations insert, delete, use as argument.

Capabilities in distributed systems

* can’t be protected by hardware/OS

* have to be transferred across networks

Capabilities in distributed systems

Must be protected by encryption

The object manager keeps a SECRET (random number) with each object.

A capability is constructed using:

check digits = f(SECRET, object-name, rights)

object name	rights	check digits

When a capability is presented as an argument with an operation invocation, the manager checks that:

f(SECRET, object-name, rights)= check digits

If not, the invocation is rejected.

1. protection

protects against tampering - adding rights
NOT against theft - eavesdropping on the network

2. propagation

still no control over transfer
to other principals

3. delegation

object manager must be asked to generate a capability with reduced rights to pass to another principal for delegation

works indefinitely - duration not controlled

4. revocation

- create a hit list of principals
  adds to overhead on all accesses

- change the SECRET
  not selective - a new SECRET can be used
  to generate new capabilities
  old ones will no longer work
  principals with old capabilities will have
  to ask for new ones to be constructed
  principals rejected by a new policy
  are refused access.

Principal-specific capabilities
(Li Gong, Cambridge 1989)

Include the name of the principal as a field
in the capability
- for generation and checking

1. protection - as before

2. propagation
   a capability for the new principal
   must be created by the object manager

3. delegation
   a capability for the delegated principal
   must be created by the object manager

4. revocation - can be selective
   e.g. keep a hot list of principals
   to be rejected by new policy

Dynamically maintained capabilities

OASIS (Hayton, Cambridge 1996)

services issue role membership certificates
RMCs (capabilities)

e.g. logged-in user, staff-member, student,
candidate, examiner, chief-examiner

possession of an RMC from one service
may be one condition for acquiring
an RMC from another service

The services involved maintain a data structure
that corresponds to a dynamic proof tree

If a condition becomes false, this is
signalled immediately to any service which
has issued a capability based on that condition.
The proof tree has collapsed
- the capability is revoked.

Distributed systems failures?
Use a heartbeat protocol, tuneable pairwise
between services.
Services, according to their policy, can
suspend any active principals affected.