# Access Control

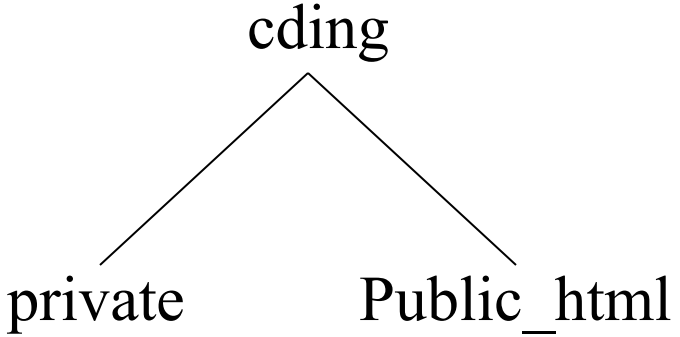
Cunsheng Ding
HKUST, Hong Kong, CHINA

# Agenda of this Lecture

- The basic concepts of access control, ACLs, capabilities, etc.
- Two approaches to access control
- Further reading

# An Example

- I, the owner of the home directory, have total control over all files in all directories and subdirectories.
- Everyone else can read all files in "Public_html", but should not do other operations on the files in this subdirectory.




**Question**: How do I do the access controls?

# Access Control

- **Computer security**: it deals with the prevention and detection of unauthorized actions.
- Computer systems control access to data and shared resources, like memory, printers, etc, primarily for reasons of **integrity**, not so much for **confidentiality**.
- Access control is at the core of computer security.

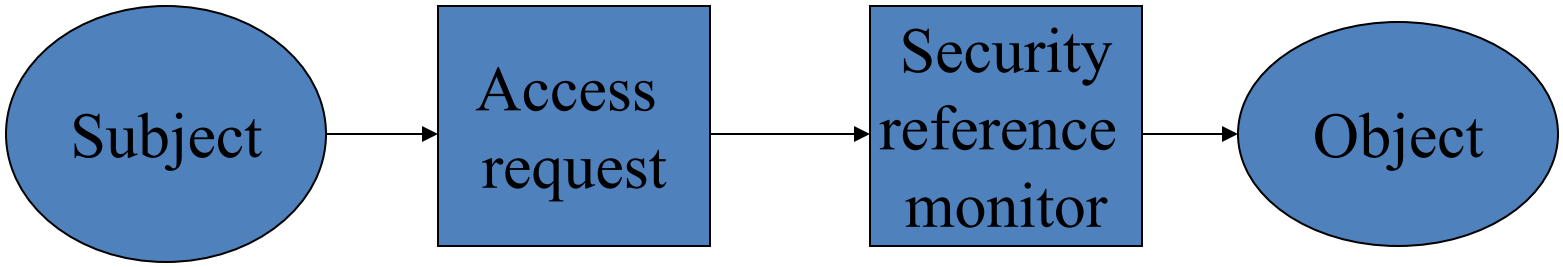
# Subjects and Objects

- Terminology for access control:
  - **subject**: active entity --- user or process
  - **object**: passive entity --- file or resource
  - **access operation**: read, write, ...
- Subjects and objects provide a different focus of control (first design principle)
  - What is the subject allowed to do? (1st approach)
  - What may be done with an object? (2nd approach)

# The Two Approaches in Practice

- Traditionally, multi-user operating systems manage files and resources, i.e. objects.
  - Access control takes the 2nd approach.
- Application-oriented IT systems, like database management systems, offer services directed to the end user and may well control the actions of subjects.
  - Access control takes the 1st approach.

# The Fundamental Model of Access Control




The security reference monitor will check the access control policy and will grant or reject the request.

Real World Examples ?

# Access Operations and Access Rights

# Access Operations

- **Access operations:** No uniform definition. They differ from system to system.
- **Examples:** basic memory access, method calls in an object-oriented system.
- We will look at a few typical sets of access operations. On the most elementary level, a subject may
  - ◊ **observe an object** or
  - ◊ **alter an object**.
- **Observe** and **Alter** are called **access modes**.

# Access Rights

- **Access rights** of the Bell-LaPadula Access control model:
- The four Bell LaPadula access rights:
  - ◊ execute
  - ◊ read
  - ◊ append, also called blind write
  - ◊ write
- Mapping between these access rights and access modes above.

| | execute | append | read | write |
|---|---|---|---|---|
| Observe | | | X | X |
| Alter | | X | | X |

# Rationale

- **A user has to open a file to get access**. Files can be opened for read access or for write access so that the O/S can avoid potential conflicts.
- **Write access usually includes read access**. A user editing a file should not be asked to open it twice. Hence, the write right includes Observe and Alter mode.
- **Few systems actually implement append**. altering an object without observing its content is rarely useful (exception: audit log).
- **A file can be used without being opened (read).** **Example:** use of a cryptographic key. This motivates the execute right, which includes neither Observe nor Alter mode.

# Unix

- **Access control policies** are expressed in terms of three operations:
  - ◊ **read**: read from a file
  - ◊ **write**: write to a file
  - ◊ **execute**: execute a file
- Applied to a **directory**, the access operations take this meaning:
  - ◊ **read**: list contents
  - ◊ **write**: create or rename files in the directory
  - ◊ **execute**: search the directory

These operations differ from the Bell-LaPadula model. E.G., Unix write access does **not** imply read access.

Creation and deletion of files are governed by access control to the directory.

# Windows NT

**Access operations** in the NTFS (New Technology File System) file system of Windows NT:

- ◊ read
- ◊ write
- ◊ execute
- ◊ delete
- ◊ change permission
- ◊ change ownership

•We no longer rely on operations on directories to handle deletion of files or change of access rights.
• Operations for modifying access rights are another ingredient you may want to use when setting security policies.

# Basic Problems in Access Control

- Who should be in charge of defining access control policies in your security system?
- How to express and capture your security policies with a data structure correctly?
- How to store your access control policies in your security system?
- How to retrieve security policies?
- How to make your access control system very efficient?

# Ownership for Manipulating Access Rights

# Ownership (1)

- **Security policies specify how subjects are allowed to access objects.**
- **Who is in charge of setting the policy?**
  - *The owner of a resource decrees who is allowed to have access. Such a policy may be called discretionary as access control is at the owner's discretion.*
  - *A system wide policy decrees who is allowed to have access. Such a policy may be called mandatory.*

# Ownership (2)

- Most operating systems support the concept of ownership of a resource and consider ownership when making access control decisions.
- Operations for manipulating access rights are grant and revoke.

# How to Capture and Implement Access Control Policies

- Access decision is based on a set of access control policies
- What data structure should be used to express the set of policies?
- How to make an access decision as quickly as possible?

# Access Control Structures

- **Several options for defining access control**:
  - *The access control structure should allow you to express the access control policy you want to enforce.*
  - *You should be able to check that your policy has been captured correctly.*
- Access rights can be defined individually for each combination of subject and object.
- For large numbers of subjects and objects, such structures are cumbersome to manage. Intermediate levels of control are preferable.

# Access Control Matrix

- **Notation:**
  - $S$ … set of subjects
  - $O$ … set of objects
  - $A$ … set of access operations
- **Access control matrix**: $M = (M_{so})_{s \in S, o \in O}$, $M_{so} \subseteq A$.
- The entry $M_{so}$ specifies the operations subject $s$ may perform on object $o$.

| | bill.doc | edit.exe | fun.com |
|---|---|---|---|
| Alice | - | {exec} | {exec,read} |
| Bob | {read,write} | {exec} | {exec,read,write} |

# Access Control Matrix ctd.

- The access control matrix is
  - an abstract concept
  - not very suitable for direct implementation
  - not very convenient for managing security

# Capabilities

- **Focus on the subject**
  - access rights are stored with the subject
  - capabilities ≡ rows of the access control matrix

| Alice | edit.exe: {exec} | fun.com: {exec,read} |
|---|---|---|

- Subjects may grant rights to other subjects. Subjects may grant the right to grant rights.
- Problems:
  - How to check who may access a specific object?
  - How to revoke a capability?
- Distributed system security has created renewed interest in capabilities.

# Access Control with Capability in HKUST

| Room 001 | Computer 111 | Printer 1234 | …… | TV 999 |
|---|---|---|---|---|
| Yes | No | Yes | ……. | No |

Capability for Cunsheng Ding in HKUST

# Access Control Lists (ACLs)

- **Focus on the object**
  - access rights are stored with the object
  - ACLs ≡ columns of the access control matrix

| fun.com | Alice: {exec} | Bill: {exec,read,write} |
|---|---|---|

- Access rights are defined for groups of users.
  - Unix: owner, group, others
- Problem: How to check access rights of a specific subject?
- ACLs are typical for certain secure operating systems.

# Access Control with ACL in HKUST

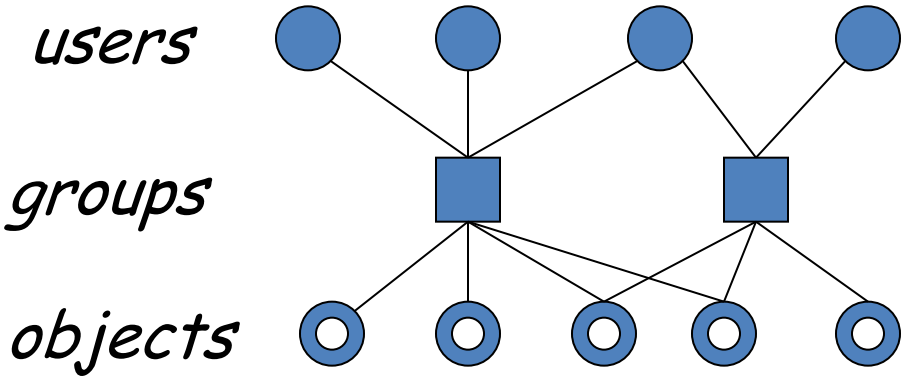
| | |
|---|---|
| Cunsheng Ding | Yes |
| John Wong | No |
| Paul Wu | Yes |
| ……. | ……. |
| Alice Fu | No |

ACL for Color Printer 111111

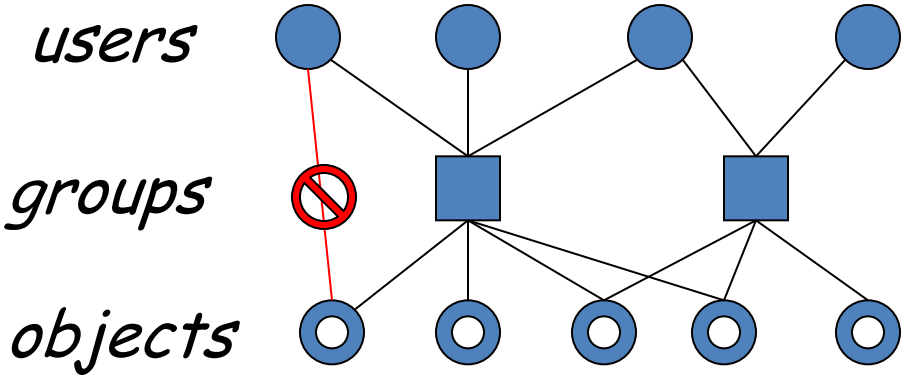
# How to Make Access Control Efficiently?

# Intermediate Controls - Groups

- Groups are an intermediate layer between users & objects.




- To deal with special cases, negative permissions withdraw rights



# Access Control in UST

This is used to illustrate the access control idea and model of Windows NT security.

# The Access Token: ID Card

- Name: Cunsheng Ding
- ID No. 008672
- AGroup: Academic staff
- DGroup: Computer Science
- SGroup: School of Engineering
- ATitle: Associate Professor
- MTitle: None

Photo

# Access Control in UST using ACLs

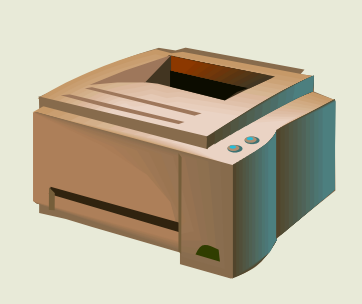
Read: all member

Borrow: academic and stud.

UST Libarary

Badminton: staffs, stud.
Basketball: acad. staffs, and their family memb.

UST Sport Facility



CSD color printer

Academic staffs
Postgradautes

Any other object in UST

**Access Control List**
Who can access?
What are you allowed to do?

# Access Control on Personal Information at HKUST

- **Question**: The Human Resource Office of the HKUST has personal data of each (teaching and administrative) staff, which contains salary information. What kind of access control policy would you suggest for UST? In other words, should your access control system focus on subject or object?

# Further reading

- Denning, D.E., "Cryptography and Security", Addison-Wesley, 1982
- Lampson, B., "Protection", *ACM Operating Systems Reviews*, vol. 8, 1974