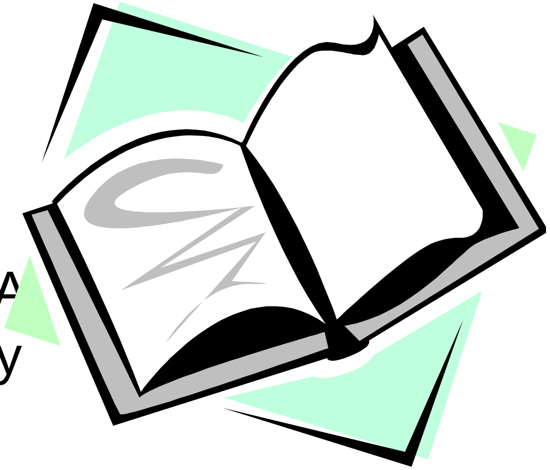
# Data Security and Privacy

Topic 6: Integrity Protection

# Related Readings for This Lecture

- Related Papers (Optional):
  - Kenneth J. Biba: "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.
  - David D. Clark and David R. Wilson. “A Comparison of Commercial and Military Computer Security Policies.” In IEEE SSP 1987.
  - David FC. Brewer and Michael J. Nash. “The Chinese Wall Security Policy.” in IEEE SSP 1989.



# Motivations

- BLP focuses on confidentiality
- In most systems, integrity is equally, if not more, important
- Data integrity vs. System integrity
  - Data integrity means that data cannot be changed without being detected.

# What is integrity in systems?

- Attempt 1: Critical data do not change.
- Attempt 2: Critical data changed only in “correct ways”
  - E.g., in DB, integrity constraints are used for consistency
- Attempt 3: Critical data changed only through certain “trusted programs”
- Attempt 4: Critical data changed only as intended by authorized users.

# Biba: Integrity Levels

- Each subject (process) has an integrity level
- Each object has an integrity level
- Integrity levels are totally ordered

- Integrity levels different from security levels in confidentiality protection
  - Highly sensitive data may have low integrity
  - What is an example of a piece of data that needs high integrity, but no confidentiality?

# Strict Integrity Policy (BLP reversed)

- Rules:
  - ■ s can read o iff $i(s) \leq i(o)$
    - no read down
    - stops indirect sabotage by contaminated data
  - ■ s can write to o iff $i(s) \geq i(o)$
    - no write up
    - stops directly malicious modification

- Fixed integrity levels
- No information path from low object/subject to high object/subject

# Subject Low-Water Policy

- Rules
  - s can always read o; after reading $i(s) \leftarrow \min[i(s), i(o)]$
  - s can write to o iff $i(s) \geq i(o)$
- Subject's integrity level decreases as reading lower integrity data
- No information path from low-object to high-object

# Object Low-Water Mark Policy

- Rules
  - s can read o; iff $i(s) \leq i(o)$
  - s can always write to o; after writing
    $i(o) \leftarrow \min[i(s), i(o)]$

- Object's integrity level decreases as it is contaminated by subjects

- In the end, objects that have high labels have not been contaminated

# Low-Water Mark Integrity Audit Policy

- Rules
  - s can always read o; after reading
    $i(s) \leftarrow \min[i(s), i(o)]$
  - s can always write to o; after writing
    $i(o) \leftarrow \min[i(s), i(o)]$
- Tracing, but not preventing contamination
- Similar to the notion of tainting in software security

# The Ring Policy

- Rules
  - Any subject can read any object
  - s can write to o iff $i(s) \geq i(o)$

- Integrity levels of subjects and objects are fixed.

- Intuitions:
  - subjects are trusted to process low-level inputs correctly

# Five Mandatory Policies in Biba

- Strict integrity policy
- Subject low-water mark policy
- Object low-water mark policy
- Low-water mark Integrity audit policy
- Ring policy

- In practice, one may be using one or more of these policies, possibly applying different policies to different subjects
  - E.g., subjects for which ring policy is applied are trusted to be able to correctly handle inputs;

# Object Integrity Levels

- The integrity level of an object may be based on
  - Quality of information  (levels may change)
    - Degree of trustworthiness
    - Contamination level:
  - Importance of the object  (levels do not change)
    - Degree of being trusted
    - Protection level: writing to the objects should be protected

- What should be the relationship between the two meanings, which level should be higher?

# Trusted vs. Trustworthy

- A component of a system is trusted means that
  - the security of the system depends on it
  - failure of component can break the security policy
  - determined by its role in the system

- A component is trustworthy means that
  - the component deserves to be trusted
  - e.g., it is implemented correctly
  - determined by intrinsic properties of the component

# Integrity vs. Confidentiality

| Confidentiality | Integrity |
|---|---|
| Control reading<br>preserved if confidential info is not read | Control writing<br>preserved if important obj is not changed |
| For subjects who need to read, control writing after reading is sufficient, no need to trust them | For subjects who need to write, has to trust them, control reading before writing is not sufficient |

Integrity requires trust in subjects!

# Analogy

- Confidentiality violation: leak a secret
  - CAN be prevented even if I tell the secret to a person I do not trust, so long as I can lock the person up AFTERWARDS to prevent further leakage
    - The person cannot leak confidential info w/o talking
- Integrity violation: follow a wrong instruction
  - CANNOT be prevented if I follow instruction from an person I do not trust even if I lock the person up BEFOREHAND to prevent the person from receiving any malicious instruction
    - The person can invent malicious instruction without outside input

# Key Difference between Confidentiality and Integrity

- For confidentiality, controlling reading & writing is sufficient
  - theoretically, no subject needs to be trusted for confidentiality; however, one does need trusted subjects in BLP to make system realistic

- For integrity, controlling reading and writing is insufficient
  - one has to trust all subjects who can write to critical data

# Impacts of The Need to Trust Subjects

- Trusting only a small security kernel is no longer possible

- No need to worry about covert channels for integrity protection

- How to establish trust in subjects becomes a challenge.

# Application of Integrity Protection

- Mandatory Integrity Control in Windows (since Vista)
  - Uses four integrity levels: Low, Medium, High, and System
  - Each process is assigned a level, which limit resources it can access
  - Processes started by normal users have Medium
  - Elevated processes have High
    - Through the User Account Control feature
  - Some processes run as Low, such as IE in protected mode
  - Reading and writing do not change the integrity level
    - Ring policy.

# The Clark-Wilson Model

- David D. Clark and David R. Wilson. “A Comparison of Commercial and Military Computer Security Policies.” In IEEE SSP 1987.
- Military policies focus on preventing disclosure
- In commercial environment, integrity is paramount
  - no user of the system, even if authorized, may be permitted to modify data items in such a way that assets or accounting records of the company are lost or corrupted

# Two High-level Mechanisms for Enforcing Data Integrity

- Well-formed transaction
  - a user should not manipulate data arbitrarily, but only in constrained ways that preserve or ensure data integrity
    - e.g., use an append-only log to record all transactions
    - e.g., double-entry bookkeeping
    - e.g., passwd

Data can be manipulated only through trusted code!

# Two High-level Mechanisms for Enforcing Data Integrity

- Separation of duty
  - ensure external consistency: data objects correspond to the real world objects
  - separating all operations into several subparts and requiring that each subpart be executed by a different person
  - e.g., the two-man rule

# Implementing the Two High-level Mechanisms

- Mechanisms are needed to ensure
  - control access to data: a data item can be manipulated only by a specific set of programs
  - program certification: programs must be inspected for proper construction, controls must be provided on the ability to install and modify these programs
  - control access to programs: each user must be permitted to use only certain sets of programs
  - control administration: assignment of people to programs must be controlled and inspected

# The Clarke-Wilson Model for Integrity

- Unconstrained Data Items (UDIs)
  - data with low integrity
- Constrained Data Items (CDIs)
  - data items within the system to which the integrity model must apply
- Integrity Verification Procedures (IVPs)
  - confirm that all of the CDIs in the system conform to the integrity specification
- Transformation Procedures (TPs)
  - well-formed transactions

# Differences from MAC/BLP

- A data item is not associated with a particular security level, but rather with a set of TPs

- A user is not given read/write access to data items, but rather permissions to execute certain programs

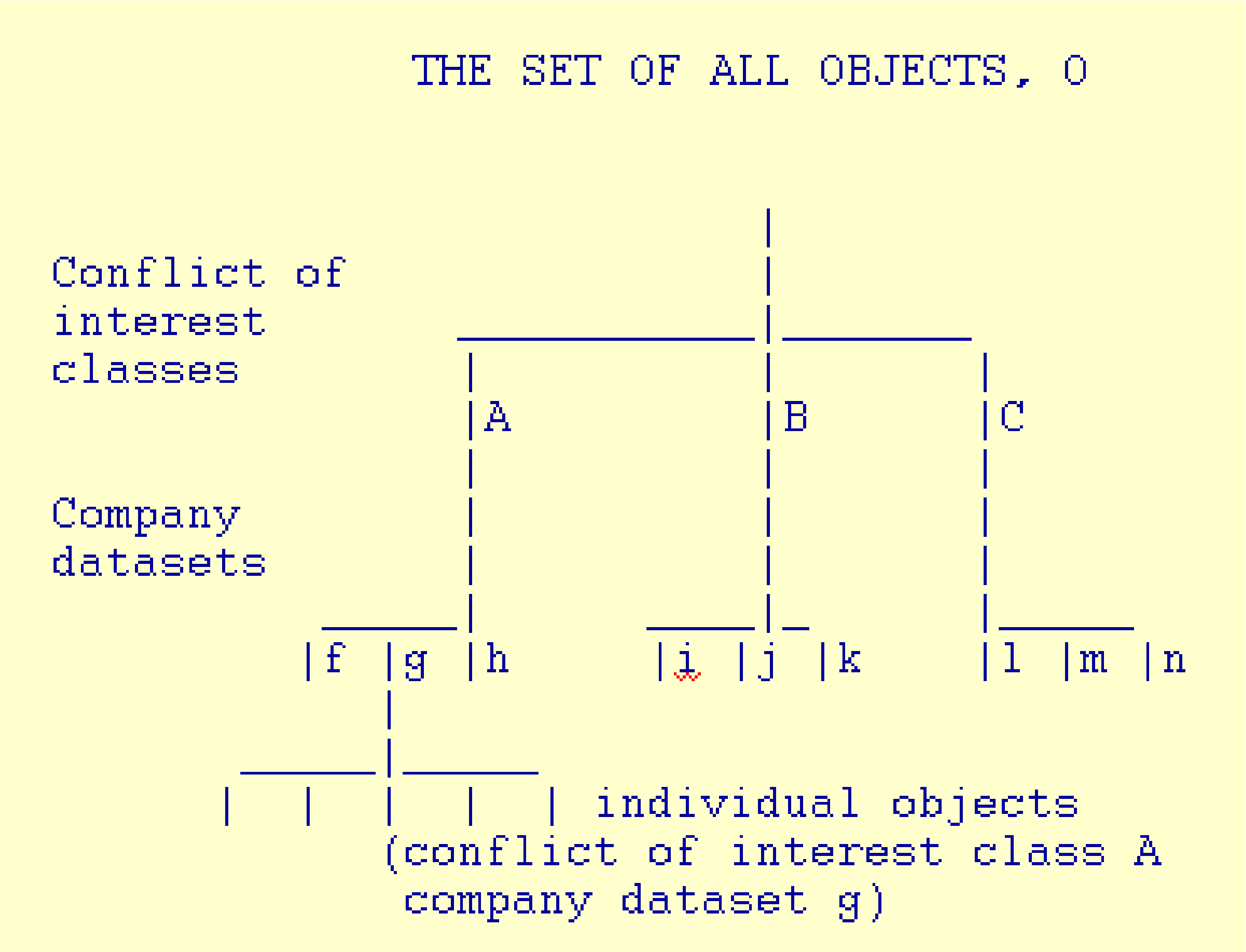
# Comparison with Biba

- Biba lacks the procedures and requirements on identifying subjects as trusted

- Clark-Wilson focuses on how to ensure that programs can be trusted

# The Chinese Wall Security Policy

- Goal: Avoid Conflict of Interest
- Data are stored in a hierarchical arranged system
  - the lowest level consists of individual data items
  - the intermediate level group data items into company data sets
  - the highest level group company datasets whose corporation are in competition



```
             THE SET OF ALL OBJECTS, O


                          |
Conflict of               |
interest       ___________|_______
classes        |          |       |
               |A         |B      |C
               |          |       |
Company        |          |       |
datasets       |          |       |
          _____|      ____|_      |_____
         |f |g |h     |i |j |k    |l |m |n
            |
       _____|_____
      |  |  |  |  | individual objects
            (conflict of interest class A
             company dataset g)
```

# Simple Security Rule in Chinese Wall Policy

- Access is only granted if the object requested:
  - is in the same company dataset as an object already accessed by that subject, i.e., within the Wall,

  or

  - belongs to an entirely different conflict of interest class.

# Coming Attractions …

- Applying Integrity Protection