NETWORK SECURITY PRACTICES – ATTACK AND DEFENSE

Access Control Models

From security policy to security model

- Security Policy
  - A statement that partitions the states of system into a set of *authorized* or *secure* states and a set of *unauthorized*, or *nonsecure* states.
  - E.g.
    - Access to NCTU's subscription of ACM Digital Library is restricted to 140.113.x.x subnet
    - An unprivileged user cannot change the clock time on a CS workstation
- Security Model
  - A formal description of a security policy
  - Irrelevant details in the corresponding policy are abstracted out
  - So the model can be proved (to be secure)
    - Model checking
    - But the actual system may still be insecure !

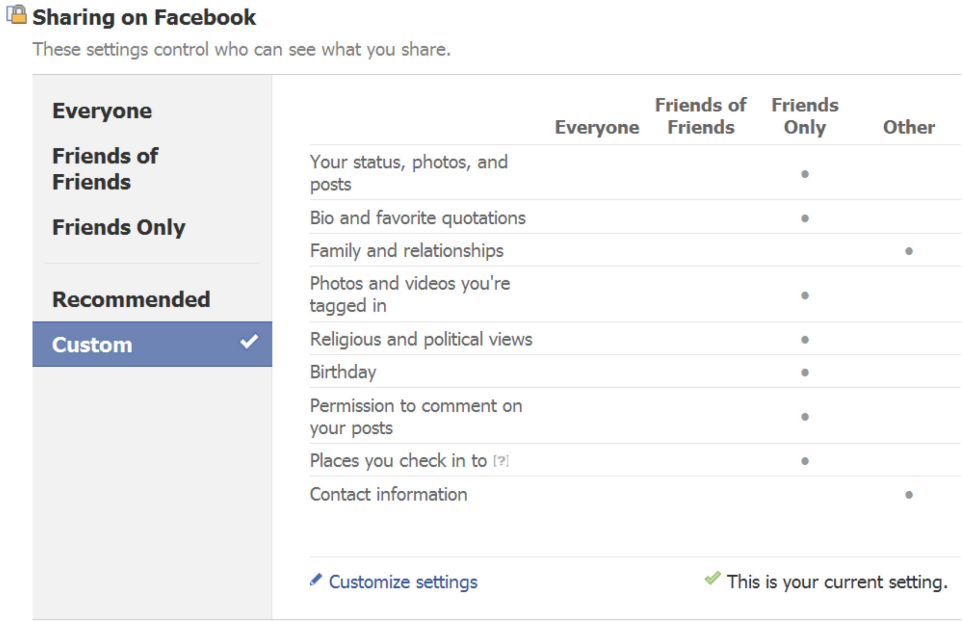
Security Models

- Confidentiality
  - Prevent the unauthorized disclosure of information
  - Multi-level security model (lattice)
  - Bell-LaPadula Model basis for many, or most, of these
- Integrity
  - Prevent unauthorized change to information
  - Biba
  - Clark-Wilson
- Hybrid of C & I
  - Chinese Wall
- Availability
  - Reliability Block Diagram
  - Fault Trees
  - Stochastic Petri Nets

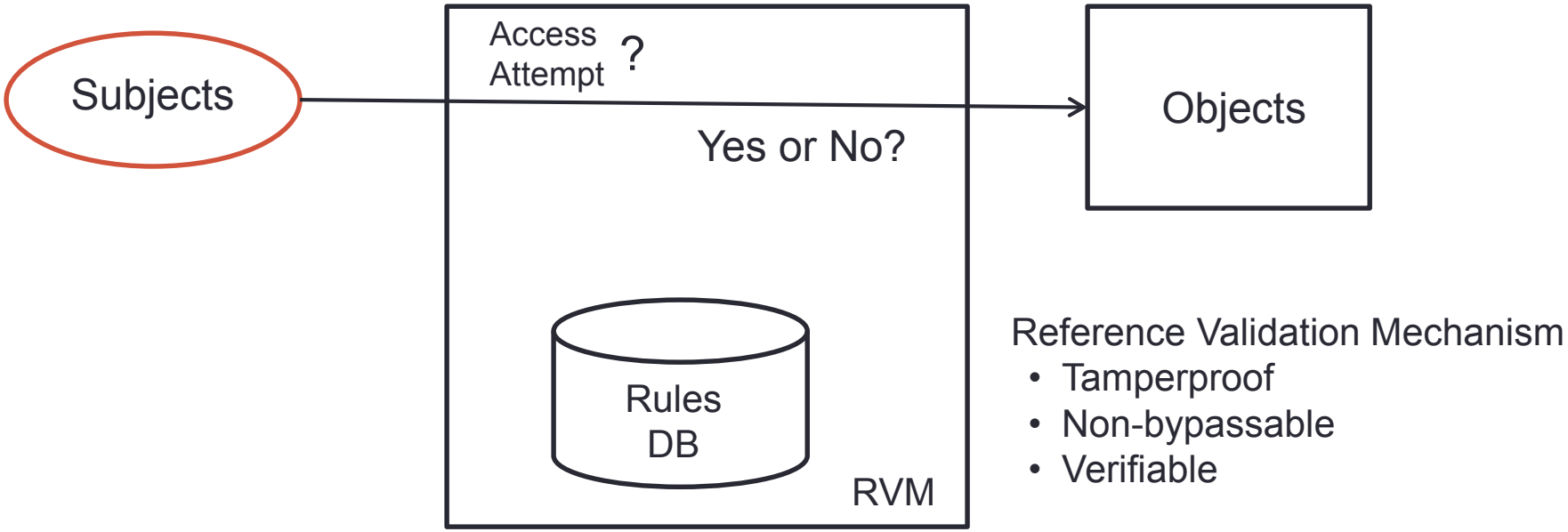
Access Control

Access Control

- Mechanisms enforcing access control
  - On UNIX
    - real / effective user ID, access mode, owner ID, group ID
    - POSIX ACLs
  - On Windows
    - SID, Group SID, DACL
  - SELinux, AppArmor
  - TrustedBSD (mac_biba, mac_mls…)
  - Movies Rating



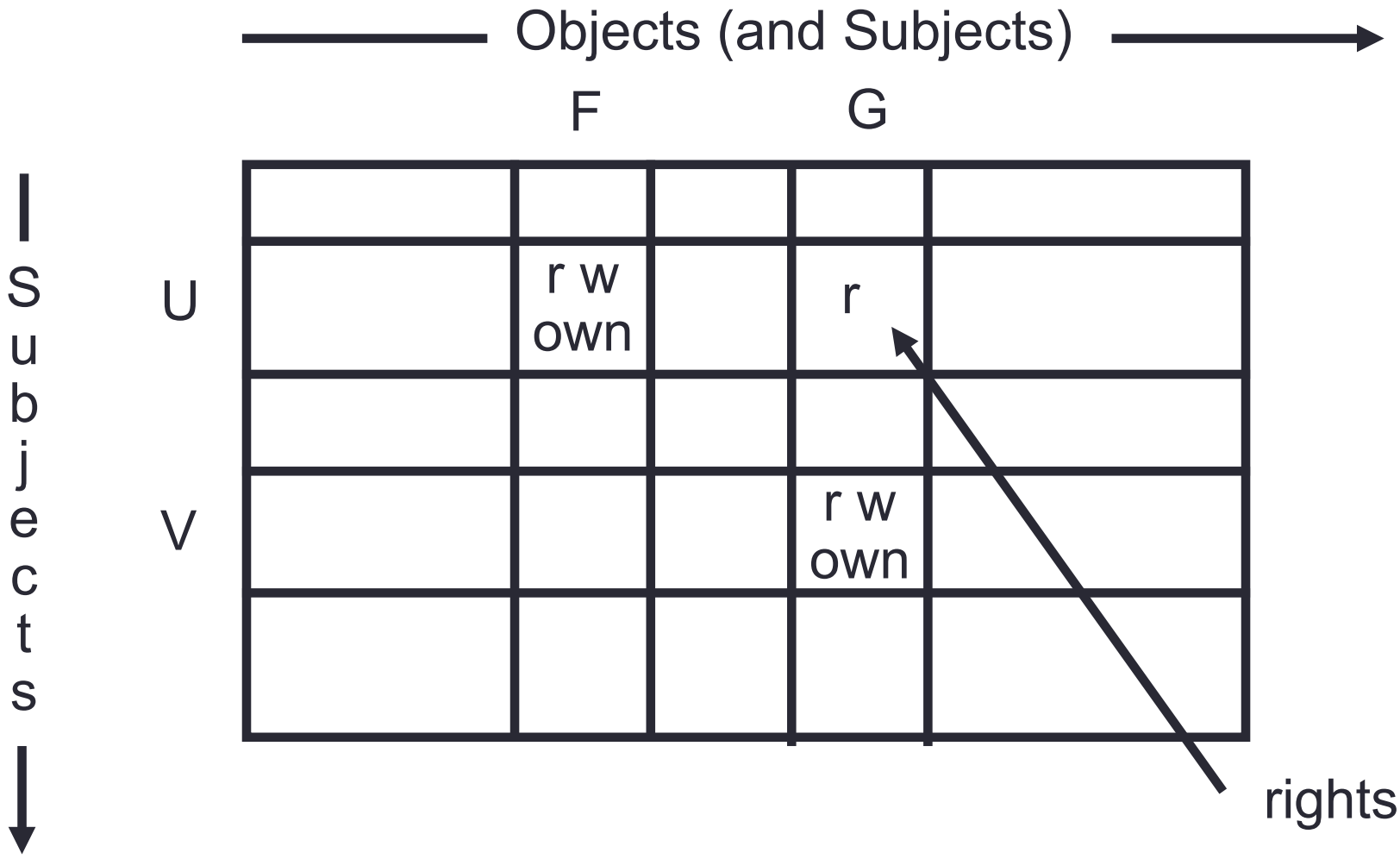
Why access control?



Reference Validation Mechanism
- Tamperproof
- Non-bypassable
- Verifiable

Anderson, James P. “Computer Security Technology Planning Study” (Oct. 1972)
Available at http://csrc.nist.gov/publications/history/ande72.pdf

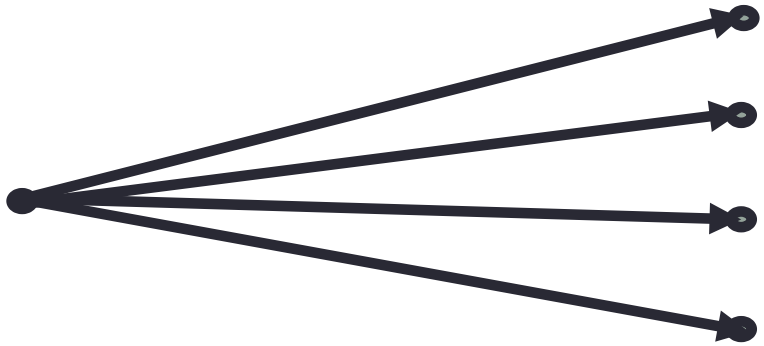
ACCESS MATRIX MODEL



ACCESS MATRIX MODEL

- Basic Abstractions
  - Subjects
  - Objects
  - Rights
- The rights in a cell specify the access of the subject (row) to the object (column)

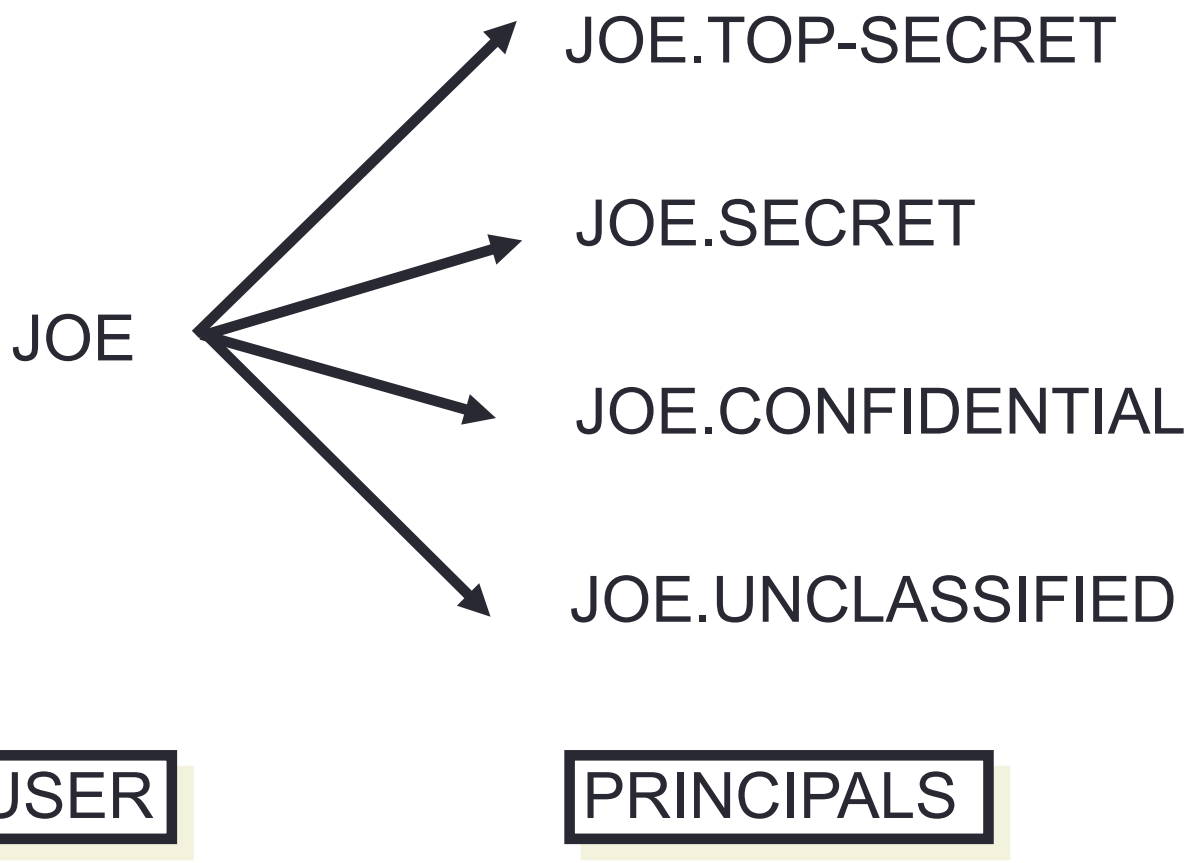
USERS AND PRINCIPALS



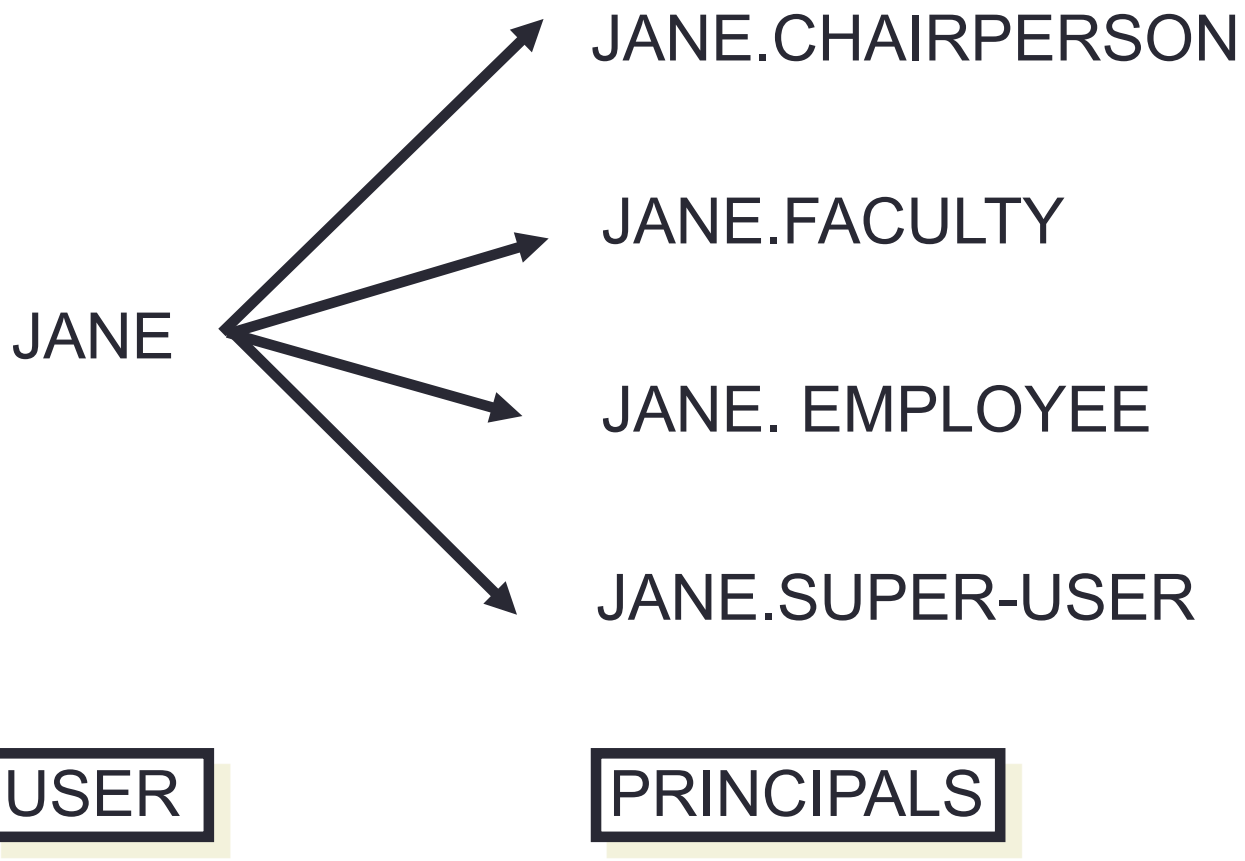
USERS	PRINCIPALS
Real World User	Unit of Access Control and Authorization

the system authenticates the user in context of a particular principal

USERS AND PRINCIPALS



USERS AND PRINCIPALS



USERS AND PRINCIPALS

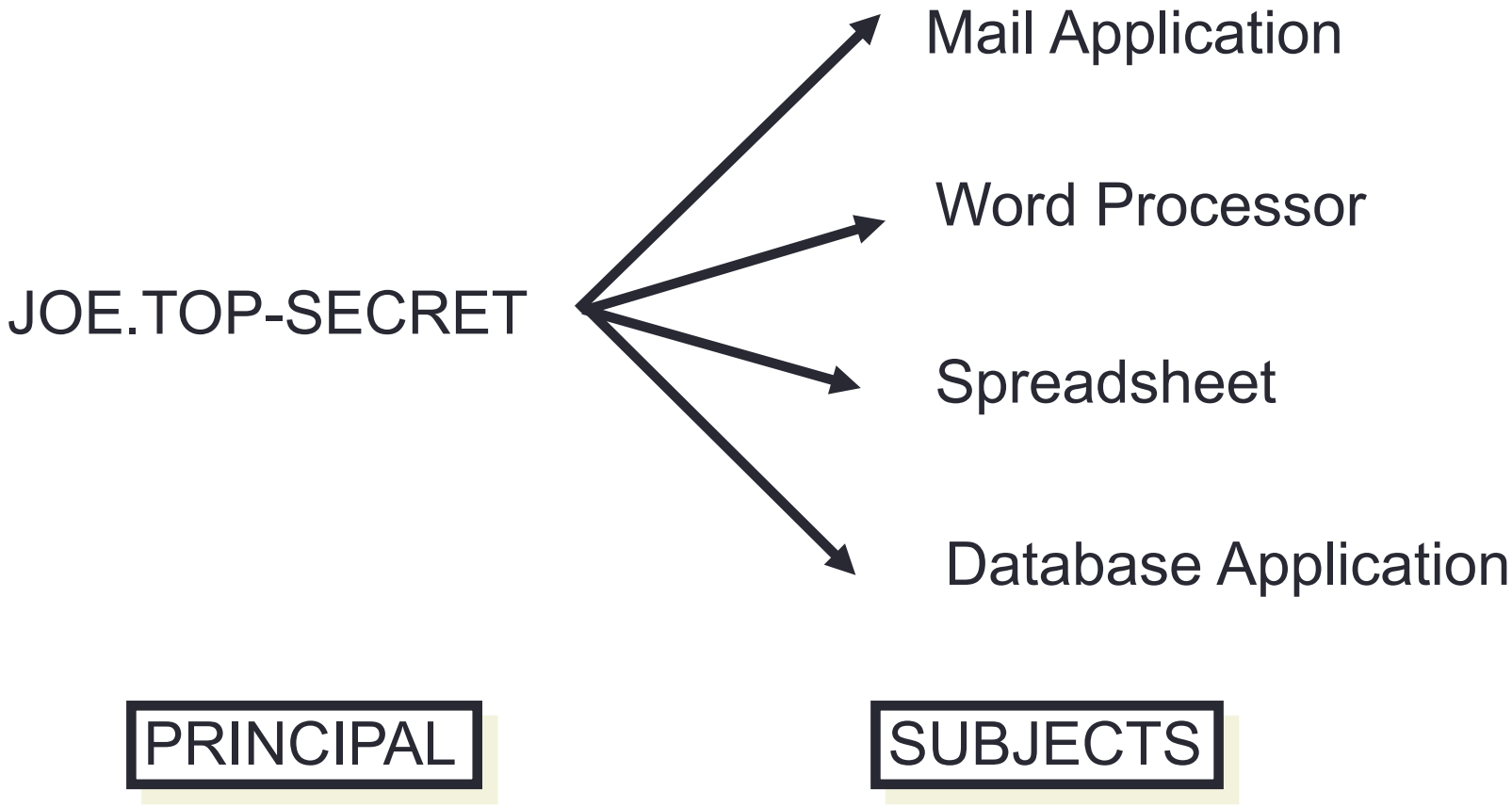
- There should be a one-to-many mapping from users to principals
  - a user may have many principals, but
  - each principal is associated with an unique user
- This ensures accountability of a user's actions

In other words, shared accounts (principals) are bad for accountability

PRINCIPALS AND SUBJECTS

- A subject is a program (application) executing on behalf of a principal
- A principal may at any time be idle, or have one or more subjects executing on its behalf

PRINCIPALS AND SUBJECTS



PRINCIPALS AND SUBJECTS

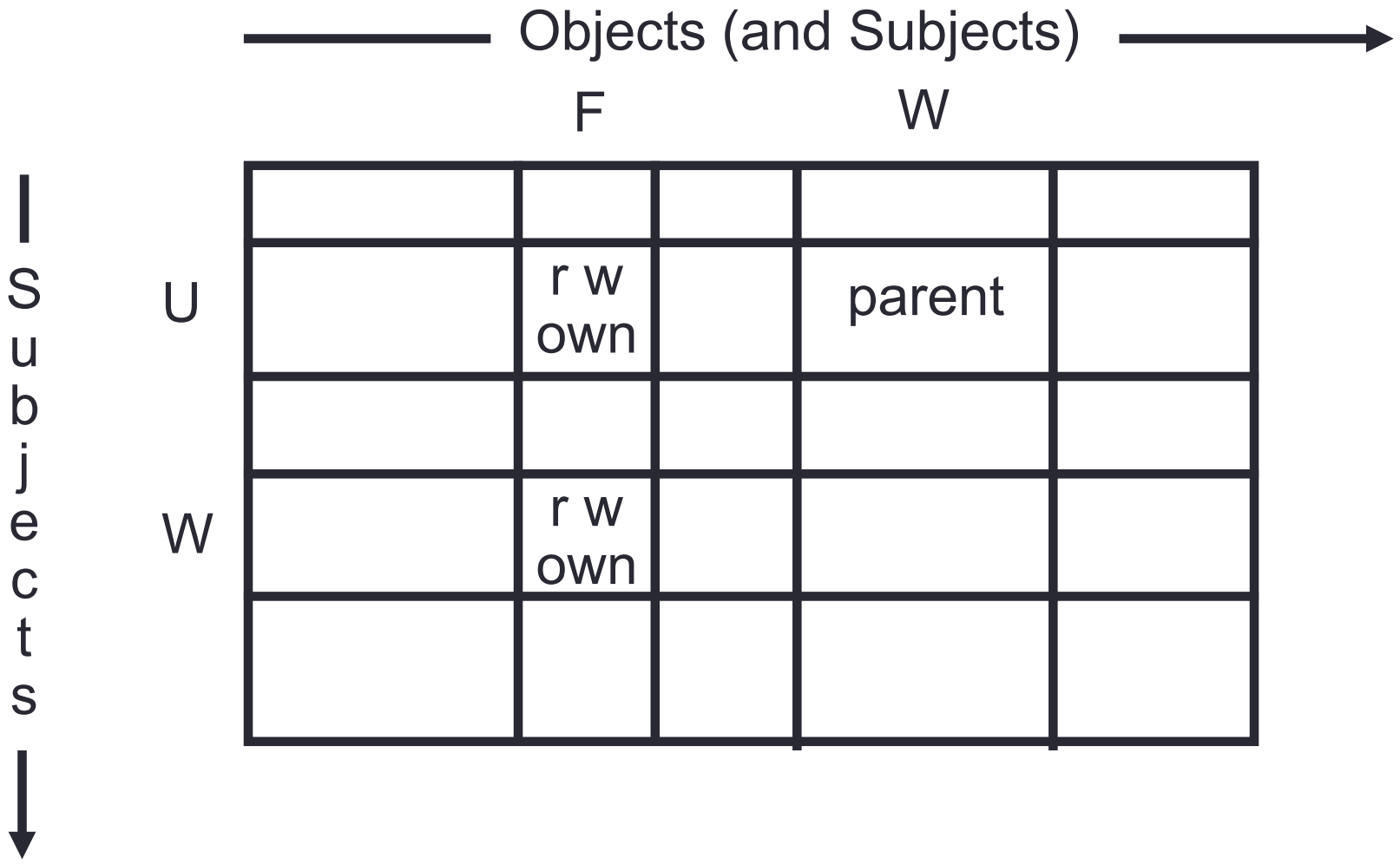
- Usually (but not always)
  - each subject is associated with a unique principal
  - all subjects of a principal have identical rights (equal to the rights of the invoking principal)
- This case can be modeled by a one-to-one mapping between subjects and principals

For simplicity, a principal and subject can be treated as identical concepts. On the other hand, a user should always be viewed as multiple principals

OBJECTS

- An object is anything on which a subject can perform operations (mediated by rights)
- Usually objects are passive, for example:
  - File
  - Directory (or Folder)
  - Memory segment
- But, subjects can also be objects, with operations
  - kill
  - suspend
  - resume

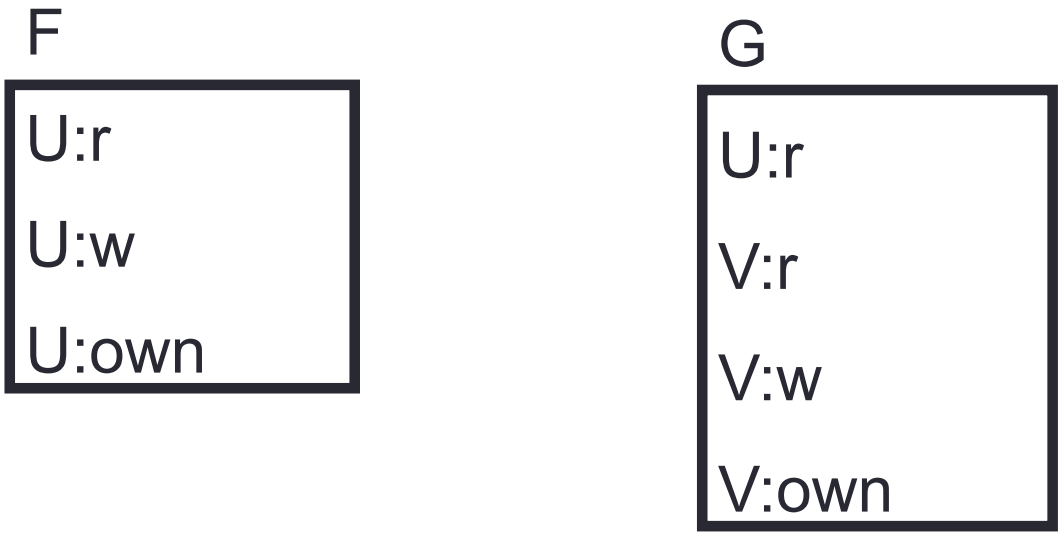
ACCESS MATRIX MODEL



IMPLEMENTATION

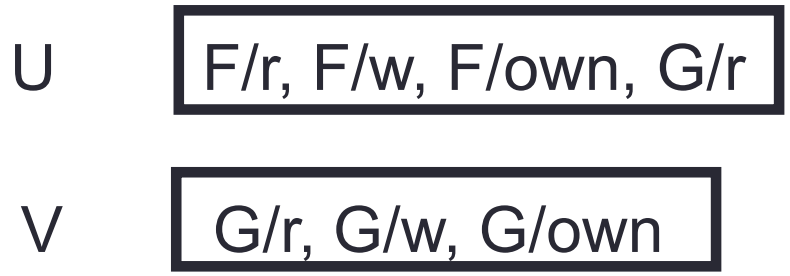
- Access Control Lists
- Capabilities
- Relations

ACCESS CONTROL LISTS (ACLs)



each column of the access matrix is stored with the object corresponding to that column

CAPABILITY LISTS



each row of the access matrix is stored with the subject corresponding to that row

ACCESS CONTROL TRIPLES

Subject	Access	Object
U	r	F
U	w	F
U	own	F
U	r	G
V	r	G
V	w	G
V	own	G

commonly used in relational database management systems

ACL'S VS CAPABILITIES

- ACL's require authentication of subjects
- Capabilities do not require authentication of subjects, but do require unforgeability and control of propagation of capabilities

ACL'S vs. CAPABILITIES

ACCESS REVIEW

- ACL's provide for superior access review on a per-object basis
- Capabilities provide for superior access review on a per-subject basis

REVOCATION

- ACL's provide for superior revocation facilities on a per-object basis
- Capabilities provide for superior revocation facilities on a per-subject basis

ACL'S vs. CAPABILITIES

- The per-object basis usually wins out so most Operating Systems protect files by means of ACL's
- Many Operating Systems use an abbreviated form of ACL's with just three entries
  - owner
  - group
  - other

ACL'S vs. CAPABILITIES

LEAST PRIVILEGE

- Capabilities provide for finer grained least privilege control with respect to subjects, especially dynamic short-lived subjects created for specific tasks

CONTENT-based CONTROLS

- Access permission determined by object's content
- content based controls such as
  - you can only see salaries less than 50K, or
  - you can only see salaries of employees who report to you
- are beyond the scope of Operating Systems and are provided by Database Management Systems

CONTEXT-based CONTROLS

- context dependent controls such as
  - you cannot access classified information via a remote login
  - salary information can be updated only at year end
  - the company's earnings report is confidential until announced at the stockholders meeting
- can be partially provided by the Operating System and partially by the Database Management System
- more sophisticated context dependent controls such as based on past history of accesses definitely require Database support

ATTRIBUTE-based CONTROLS

- Access permission determined by the attributes of a user
- Examples
  - You need to be over 18 years old to purchase alcohols
  - You need to be on campus to use NCTU's subscription of ACM Digital Library

DISCRETIONARY vs. MANDATORY

- Discretionary access controls (DAC) allow access rights to be propagated from one subject to another

  Possession of an access right by a subject is sufficient to allow access to the object
- Mandatory access controls (MAC) restrict the access of subjects to objects on the basis of security labels

INHERENT WEAKNESS OF DAC

- Unrestricted DAC allows information from an object which can be read to any other object which can be written by a subject
- Suppose our users are trusted not to do this deliberately. It is still possible for Trojan Horses to copy information from one object to another.

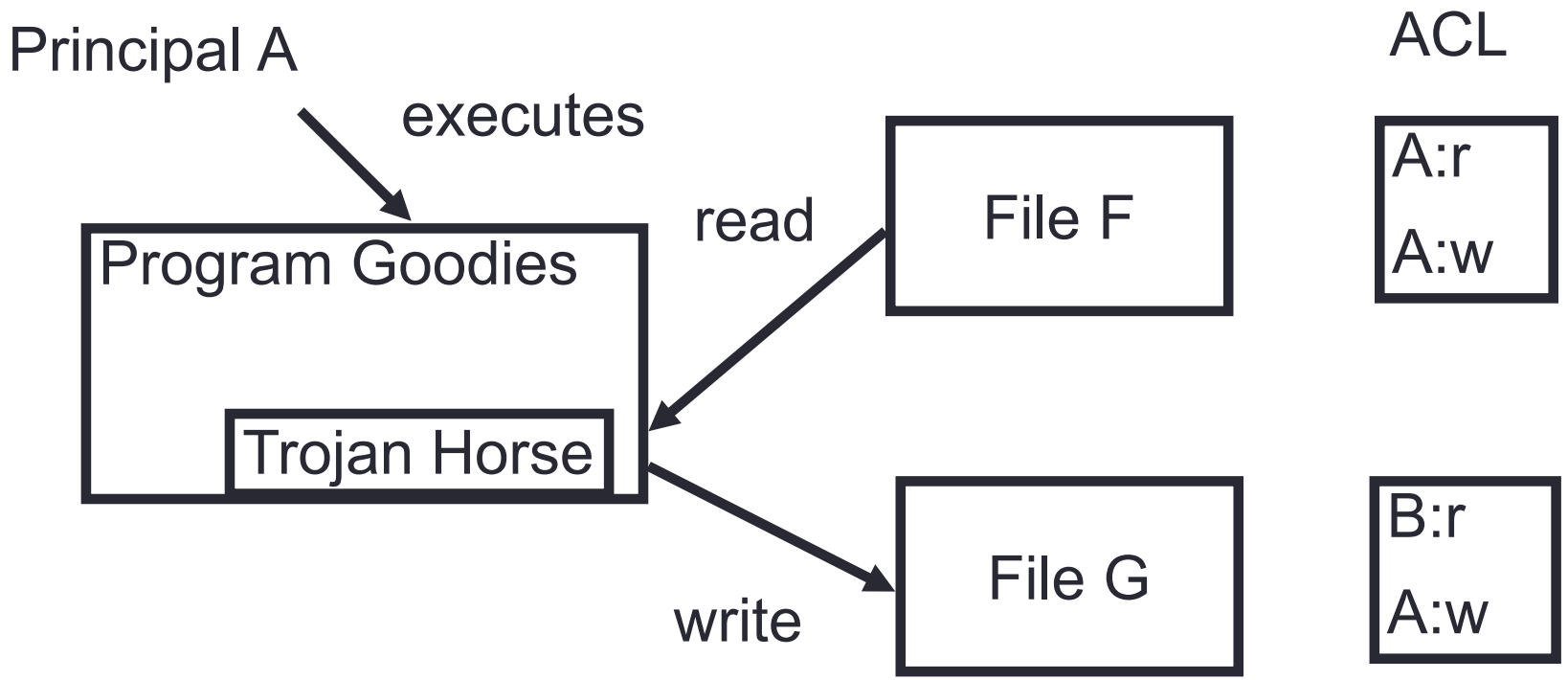
TROJAN HORSES

- A Trojan Horse is rogue software installed, perhaps unwittingly, by duly authorized users
- A Trojan Horse does what a user expects it to do, but in addition exploits the user's legitimate privileges to cause a security breach

TROJAN HORSE EXAMPLE

Principal B cannot read file F

TROJAN HORSE EXAMPLE



Principal B can read contents of file F copied to file G

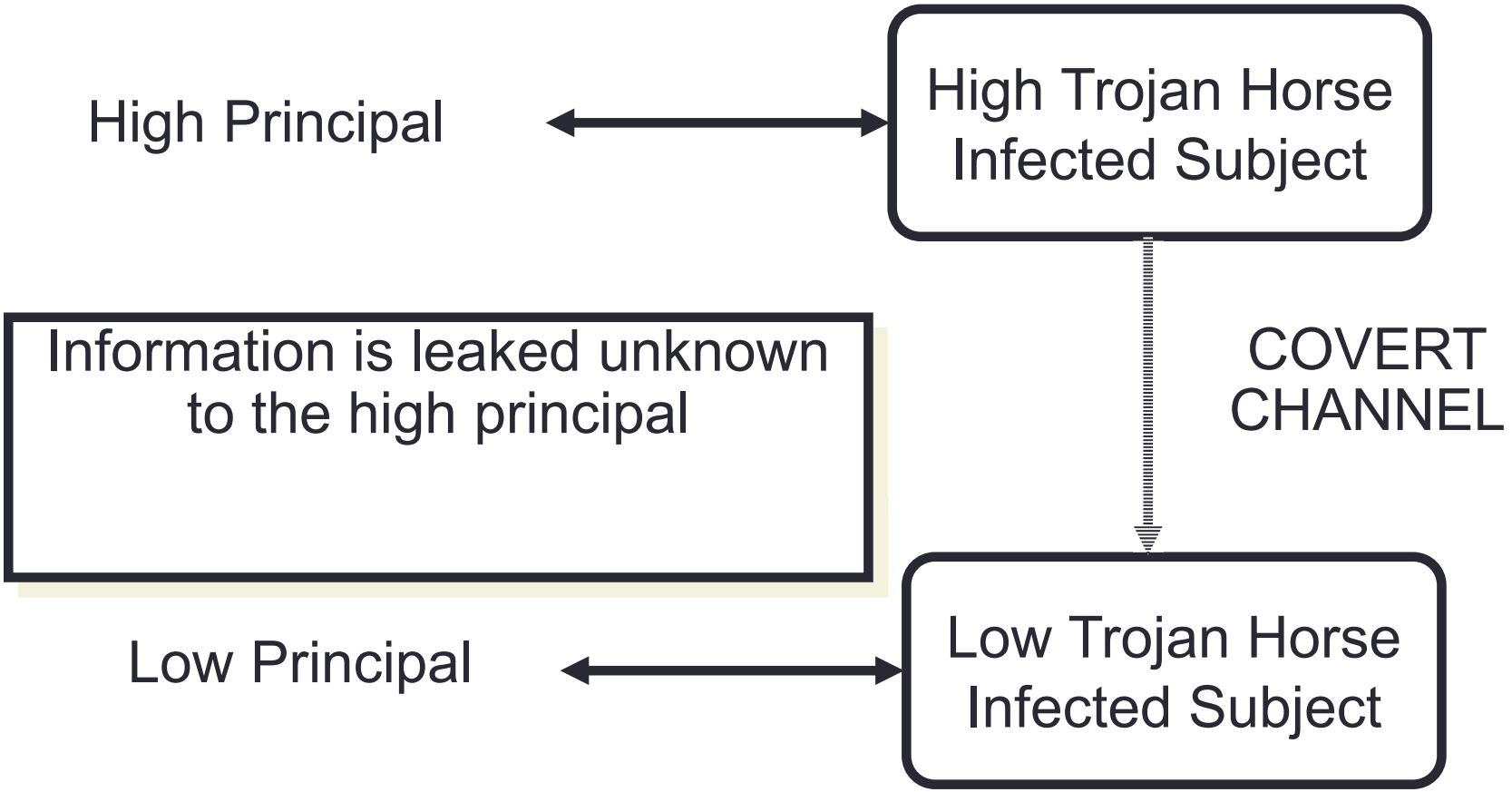
TROJAN HORSES

- Trojan Horses are the most insidious threat
- Viruses and logic bombs are examples of Trojan Horses
- It is possible to embed Trojan Horses in hardware and firmware
- It is possible to embed Trojan Horses in critical system software such as compilers and Database Management Systems

COVERT CHANNELS

- A covert channel is a communication channel based on the use of system resources not normally intended for communication between the subjects (processes) in the system

COVERT CHANNELS



COVERT CHANNELS

- The concern is with subjects not users
  - users are trusted (must be trusted) not to disclose secret information outside of the computer system
  - subjects are not trusted because they may have Trojan Horses embedded in the code they execute
- *-property prevents overt leakage of information and does not address the covert channel problem

RESOURCE EXHAUSTION CHANNEL

Given 5MB pool of dynamically allocated memory

HIGH PROCESS

bit = 1 $\Rightarrow$ request 5MB of memory

bit = 0 $\Rightarrow$ request 0MB of memory

LOW PROCESS

request 5MB of memory

if allocated then bit = 0 otherwise bit = 1

LOAD SENSING CHANNEL

HIGH PROCESS

bit = 1 $\Rightarrow$ enter computation intensive loop

bit = 0 $\Rightarrow$ go to sleep

LOW PROCESS

perform a task with known computational requirements

if completed quickly then bit = 0 otherwise bit = 1

COPING WITH COVERT CHANNELS

- identification
  - close the channel or slow it down
  - detect attempts to use the channel
  - tolerate its existence

COVERT CHANNELS AND THE ORANGE BOOK

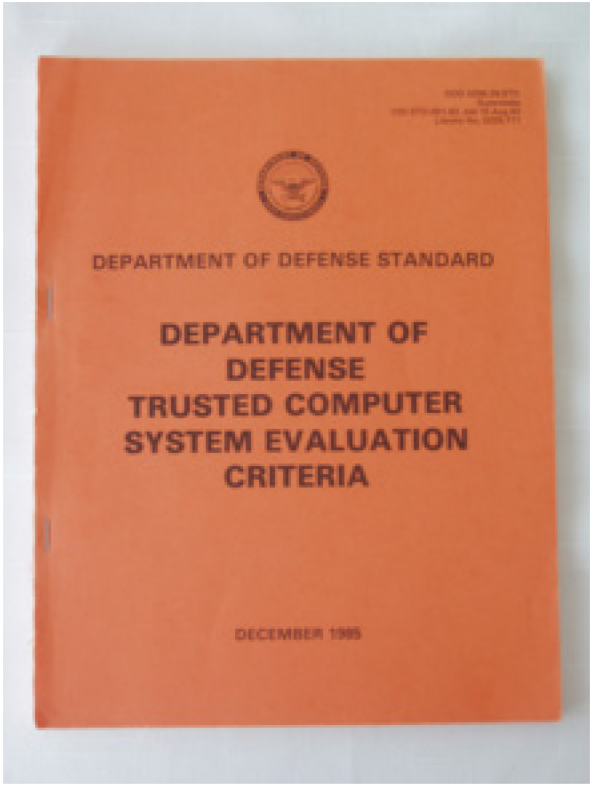
C2 No labels

B1 Labels with Bell-LaPadula controls, but no need to address covert channels

B2 Must address storage channels (such as resource exhaustion channel)

B3 Must also address timing channels (such as load sensing channel)

A1 Must use formal techniques (where available)



COVERT CHANNELS AND THE ORANGE BOOK

- XTS-400
  - Developed by BAE Systems
  - B3 level security
  - x86 Hardware
  - Linux-like STOP (Secure Trusted Operating System) OS
  - Mandatory Sensitivity Policy based on Bell-La Padula model
  - Mandatory Integrity Policy based on Biba model

BEYOND MAC DAC

- DAC and MAC are extreme points of a continuum of access controls
- There are legitimate policies that fall in between, for example:
  - Document release: a document cannot be released by a scientist without first obtaining approvals from a patent-officer and a security-officer
  - Originator control: information in an object should not be propagated without permission of the owner of the object

BEYOND MAC DAC

- There are security models which transcend the black and white MAC-DAC distinction, notably:
  - The HRU model

    Harrison, M.H., Ruzzo, W.L. and Ullman, J.D. “Protection in Operating Systems.” Communications of ACM, 19(8):461-471 (1976).
  - The TAM model

    Sandhu, R.S. “The Typed Access Matrix Model.“” Proceeding IEEE Symposium on Security and Privacy, Oakland, CA, May 4-6, 1992, pages 122-136.
  - The RBAC model

    Ravi Sandhu, Edward Coyne, Hal Feinstein and Charles Youman, “Role-Based Access Control Models.” IEEE Computer, Volume 29, Number 2, February 1996.