COMP3052.SEC Computer Security

Session 05: Access Control

Acknowledgements

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, …

This Lecture

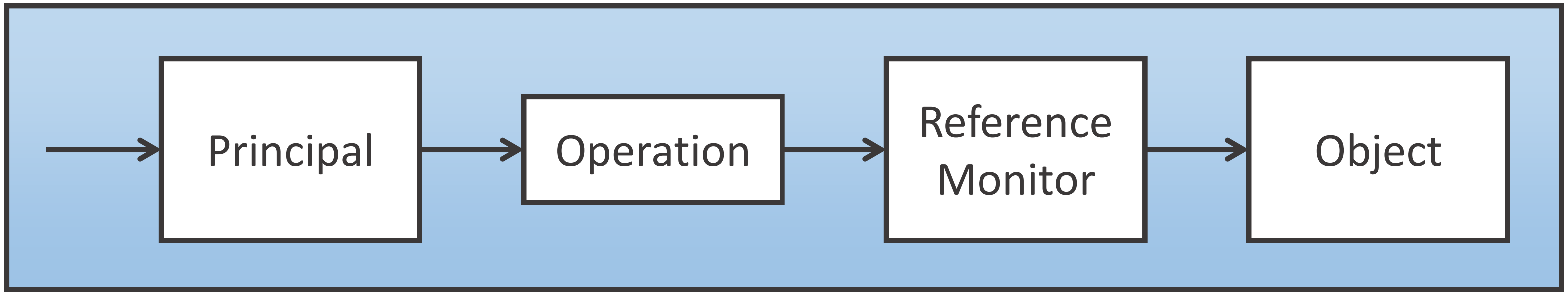
- Access Control fundamentals
- Principles, Subjects and Objects
- Access control structures
- Groups and Roles
- Evaluating access at runtime

Background

- Authentication lets us verify who we are to a system
- Assuming we’ve authenticated:
  - Some files are private, some are public
  - System files should be protected
  - We need to be able to access some applications
- We need a mechanism to enforce access control

Authentication & Authorisation

- Subject / Principal – an active entity
- Object – resource being accessed
- Access an operation
- Reference monitor – grants or denies access



Authentication & Authorisation

- Access control has two steps:
  - Authentication
    - Decide who has access to the system
  - Authorisation
    - Of those with access, who is authorised to do something to the resource (object)

Principal vs. Subject

- Principal

  "An entity that can be granted access to objects or can make statements affecting access control decisions"

  - E.g user identity in an OS
  - Used when discussing security policies

- Subject

  "An active entity within an IT system"

  - E.g. process running under a user identity
  - Used when discussing operational systems enforcing policies

Subject vs. Object

- Object – Files or resources
  - E.g. Memory, printers, directories
- Subject vs Object: Distinguish between the active and passive party in an access request
- Two options for focusing control:
  - What a subject is allowed to do
  - What may be done to an object

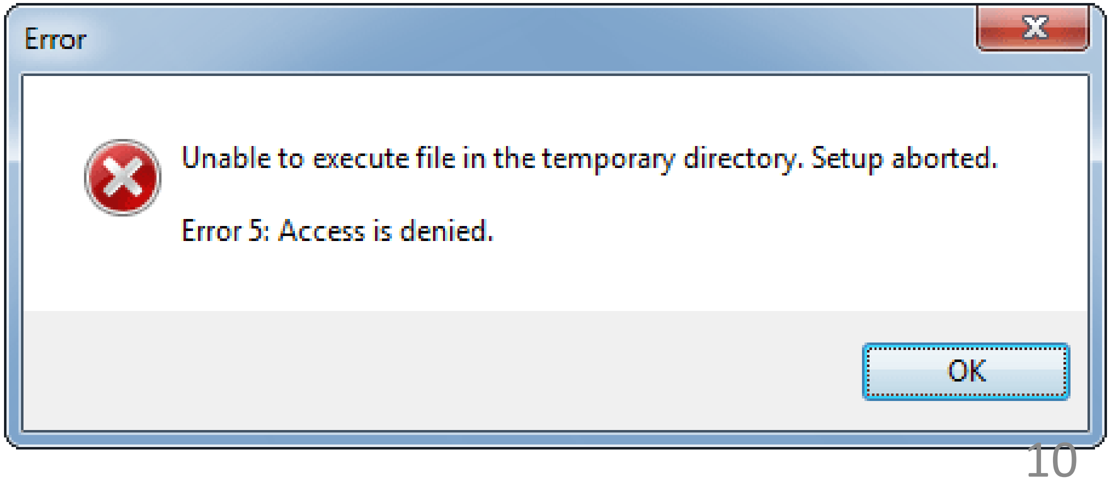
Access Operations

- From reading / writing to method calls
- Varies from system to system
- Sometimes similarly named operations in two systems will have different meanings

- Access modes
  - Observe – Subject may look at contents of an object
  - Alter – Subject may change the contents of an object
- Too abstract for practical use!

General Model

- We'll settle on some common access on files:
  - Read – Simply viewing (Confidentiality)
  - Write – Includes changing, appending, deleting (Integrity)
  - Execute – Can run the file without knowing its contents



Unable to execute file in the temporary directory. Setup aborted.

Error 5: Access is denied.

Ownership

- Who is in charge of setting security policies?
- Discretionary: Owner can be defined for each resource
  - Owner controls who gets access
- Mandatory: Could be a system-wide policy
- Most OSs support the concept of ownership

Access Control Structures

- Help express access control policy
- Often focused upon efficient lookup – resources are accessed a lot!

- Access Control Matrix
- Access Control Lists
- Capabilities

Access Control Matrix

- Access rights are defined individually for each combination of subject and object
  - Quite an abstract concept, but would allow for very fine grained control
  - Not practical, think of the memory required in scaling it up!

	budget.xlsx	game.exe	msexcel.exe
Alice	r,w,e	r,e	r,e
Bob	r		r,e
Claire	r		r,e
Dave		r,w,e	r,e

Access Control List

- Stored with an object itself, corresponding to a column of an ACM

	budget.xlsx	game.exe	msexcel.exe
Alice	r,w,e	r,e	r,e
Bob	r		r,e
Claire	r		r,e
Dave		r,w,e	r,e

budget.xlsx	Alice: r,w,e	Bob: r	Claire: r

Access Control List

- Better:
  - Much less memory intensive
  - If stored with a file is quick to access
- However:
  - Management of individual subjects is cumbersome
  - Obtaining an overview of permissions is challenging
  - Tedious to set this up properly for all subjects and objects

UNIX

- Unix simplifies the ACL structure to consider only the user, group and others
    - User is the current owner
    - Group is a named group entity
    - Everyone else
- Unix offers Read, Write and Execute access controls

Windows

- Windows predominantly uses ACLs, and has done since Windows NT
- Extends the usual read, write and execute with:
  - Take ownership
  - Change permissions
  - Delete
- A higher degree of control, with the associated complexity increase!

Capabilities

- Access rights are stored with a subject, not a resource
- Every subject is given a capability:

  “An unforgeable token specifying the subject’s access rights”

- Corresponds to a row in an access control matrix:

Alice	budget.xlsx: {rwx}	game.exe {r,e}	msexcel.exe {r,e}

Capabilities

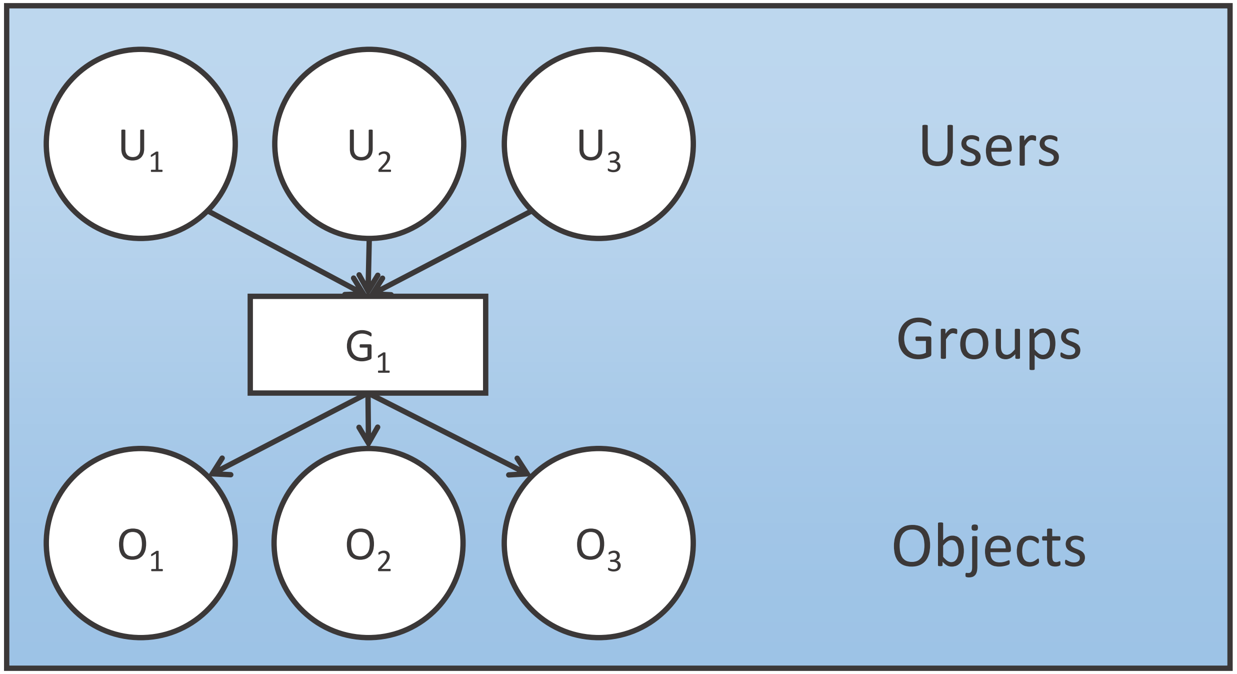
- Typically associated with discretionary access control
  - Subjects can pass on their capabilities
- Not widely used – e.g. exists in Linux but rare
- Difficult to get an overview of access rights on a file, and revoke them

Intermediate Controls

- Problems of complexity and scalability solved by indirection
  - Groups
  - Negative Permissions
  - Privileges
  - Role-based Access Control
  - Protection Rings

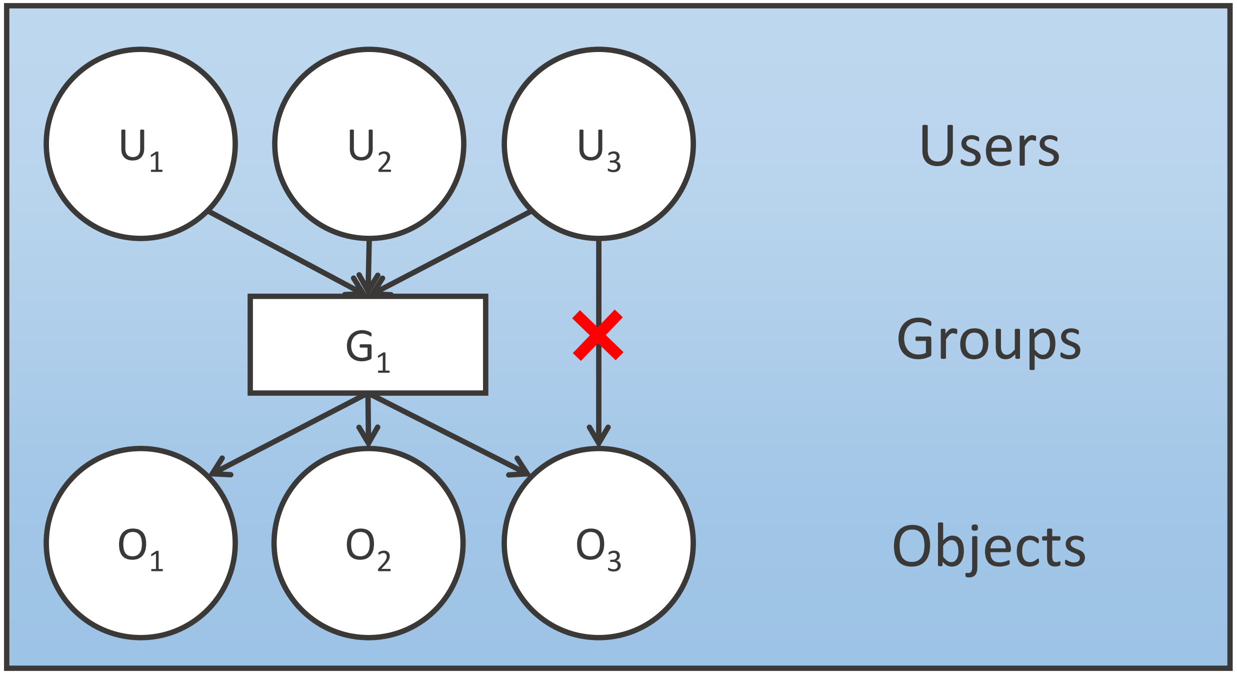
Groups

- Users with similar access rights can be collected into groups
- Groups are given permissions to access objects



Negative Permissions

- An operation that a user cannot perform
- Policy conflict – resolved by the reference monitor



Alternatives

- Identity Based Access Control (IBAC)
  - The standard approach we’ve been discussing
    - e.g. ACLs
  - Scales better than a matrix, but not to enterprise level
- Role-based Access Control (RBAC)
  - Access is based on a role, e.g. accountants should access certain financial files
  - Easier to scale and use, but nothing is perfect!

Role-based Access Control

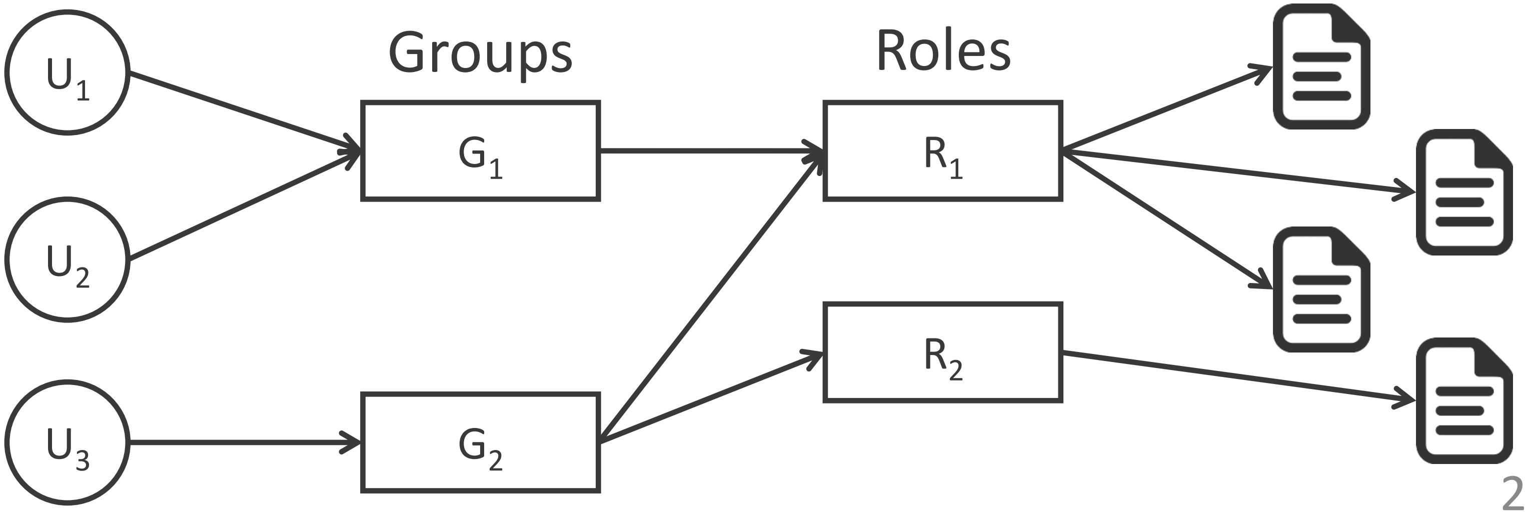
- A role – Collection of application specific operations or resource access
- Subjects derive access rights from the role they perform
- RBAC focuses on users and the jobs they perform
- Much more applicable to large networks and organisations

Role-based Access Control

- Layers (between subjects and objects)
  - Roles – collection of procedures assigned to users
  - Procedures – high level access control methods
  - Data types – each object of certain data type

Roles vs. Groups

- Sound the same, but are subtly different
  - Groups are collections of users
  - Roles are collections of permissions
- Most operating systems are user / group based, so role-based access can be provided using nested groups



Evaluating Security Policies

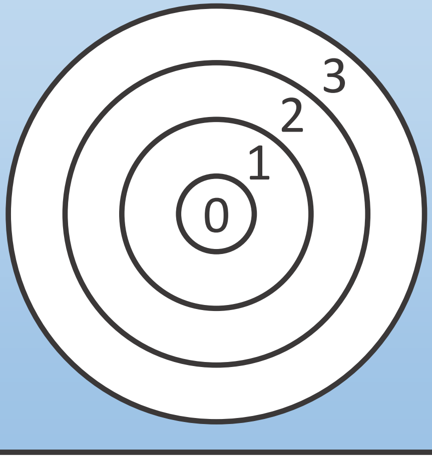
- At a basic level: quality check against Access Control Entry
- More complicated:
  - Protection Rings
  - Partial Orderings
  - Lattices

Privileges

- A collection of rights to execute certain operations
- Come pre-defined with an OS
- An intermediate layer between subjects and operations
- Usually associated with operating system functions
  - Installing software, network access etc.

Protection Rings

- Hardware-based access control
- Each subject and object assigned a number depending on importance
- Decisions are made by comparing subject’s to object’s numbers

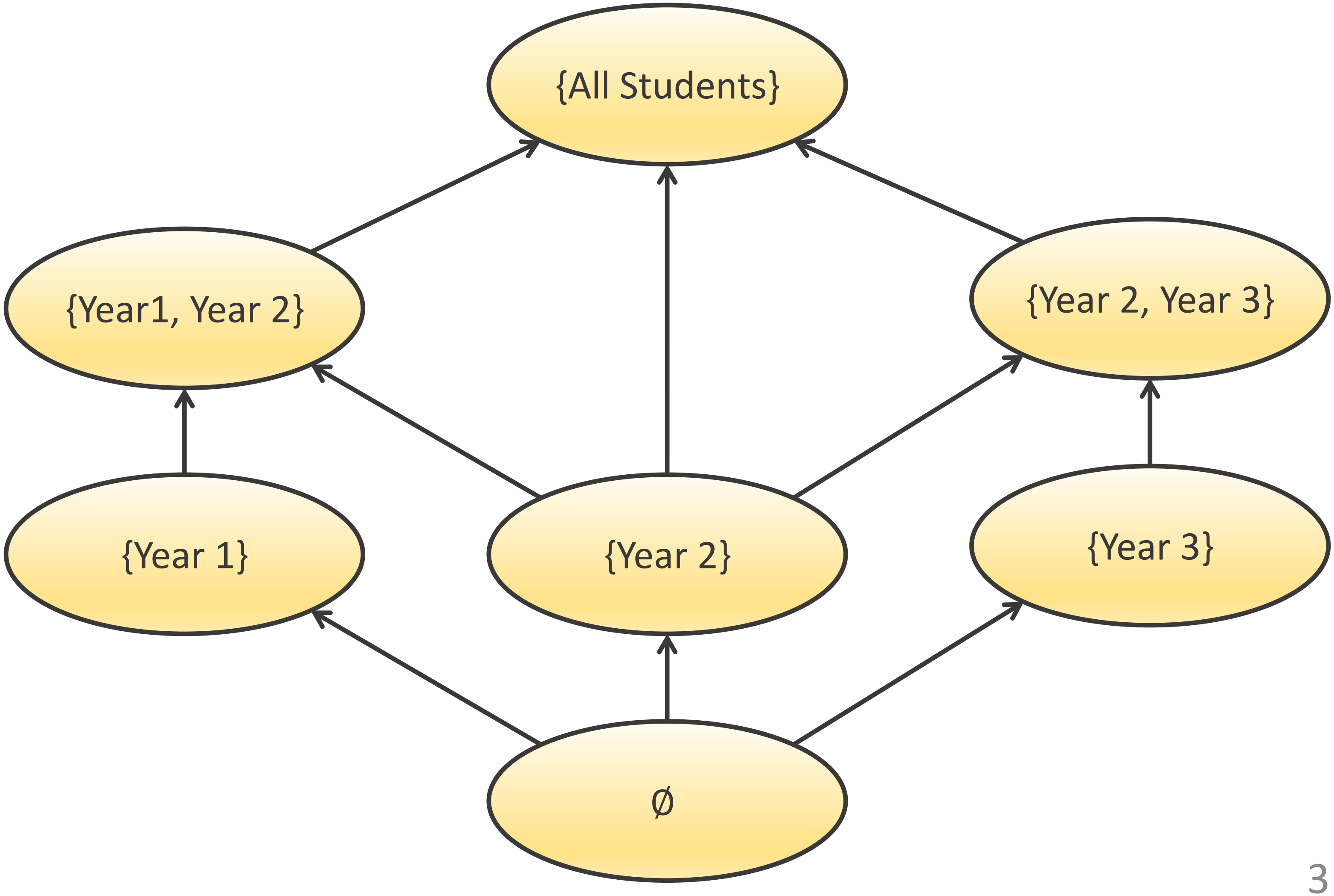


0 – OS Kernel
1 – Operating System
2 – Utilities
3 – User Processes

Partial Ordering

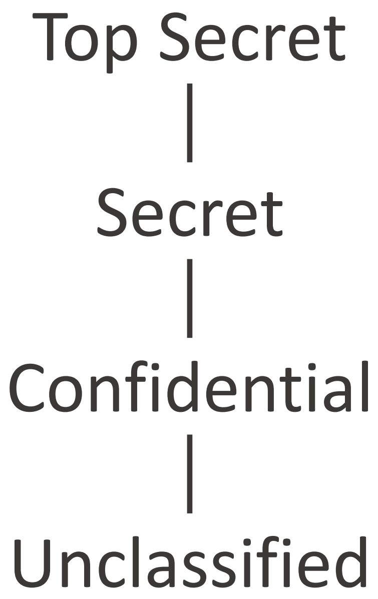
- Imagine groups containing students in year 1, year 2 etc.
- Partial orderings (≤) provide relations between subsets of groups
- For example, {Year1} ≤ {Year1, Year2}
- A security policy might grant access to an object if the subject label is ≤ object label

Partial Ordering



Multi-Level Security

- Often military applications will define access based on security levels
- Easy to implement, but we can't express complex security policies



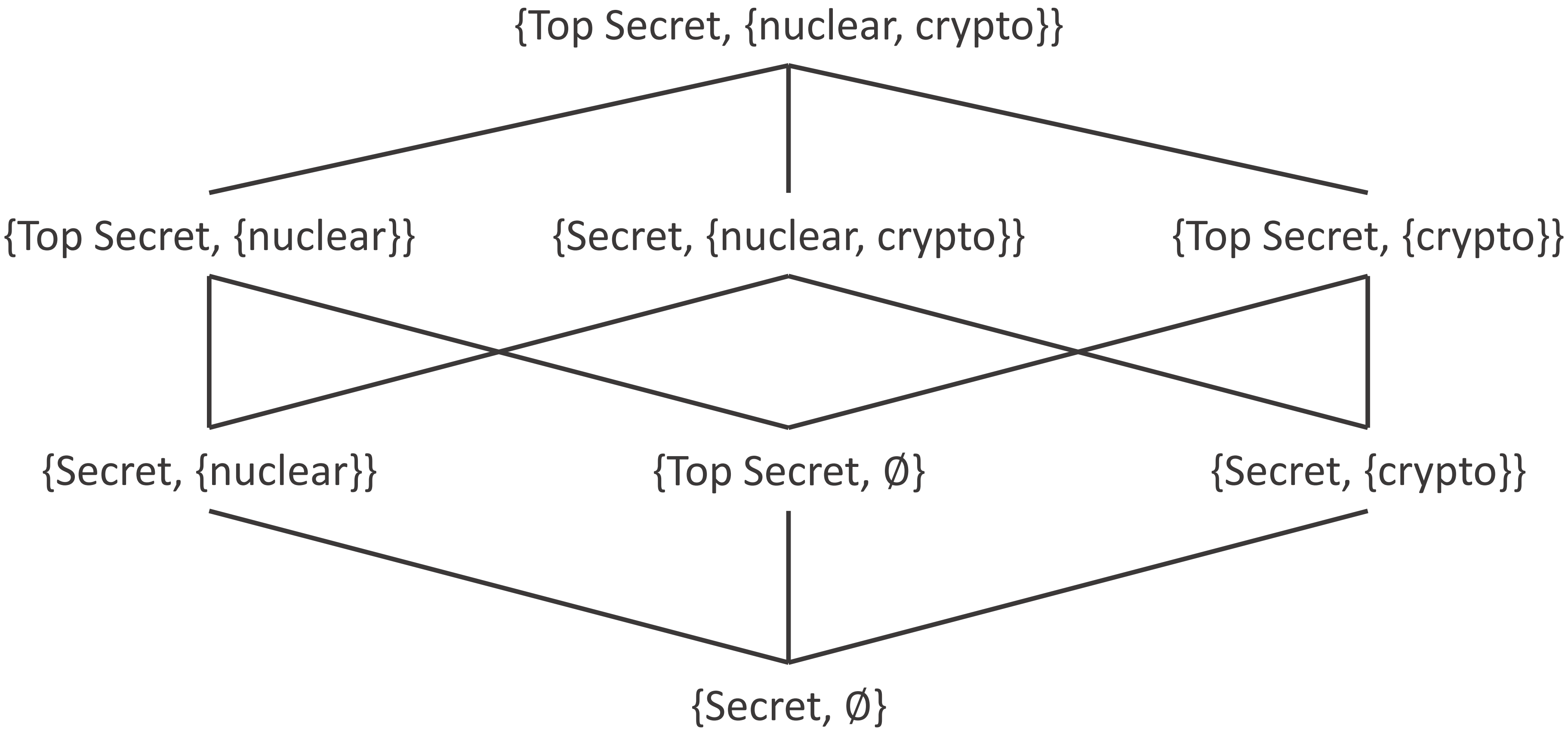
Top Secret

Secret

Confidential

Unclassified

Lattices


{Top Secret, {nuclear, crypto}}
{Top Secret, {nuclear}}
{Secret, {nuclear, crypto}}
{Top Secret, {crypto}}
{Secret, {nuclear}}
{Top Secret, Ø}
{Secret, {crypto}}
{Secret, Ø}

Summary

- Access Control
- Its structures
  - Access Control Matrices, Lists
  - Capabilities
- How we manage it
  - Groups, Role-based
  - Partial ordering and lattices

Anderson Chapter 4

Gollmann Chapter 5

↑

Further reading for interested people!