Lecture 5

Access Control

Dr. Charnon Pattiyanon
Assistant Director of IT, Instructor
Department of Artificial Intelligence and Computer Engineering
CMKL University

Today's Class Outline

- Upon successful of this lecture, you will know about:
  - An overview and concepts of **user authentication**.
  - An overview of the need of **access control** to files or objects.
  - Common **methodologies and tools** for access control.
  - Common **policy models** of access control.

What is Access Control?

- **ITU-T** (International Telecommuniction Union – Telecommunication Standardization Sector) **Recommendation X.800** defines **access control** as:

> "The prevention of ***unauthorized use of a resource***, including the prevention of use of a resource in an unauthorized manner"

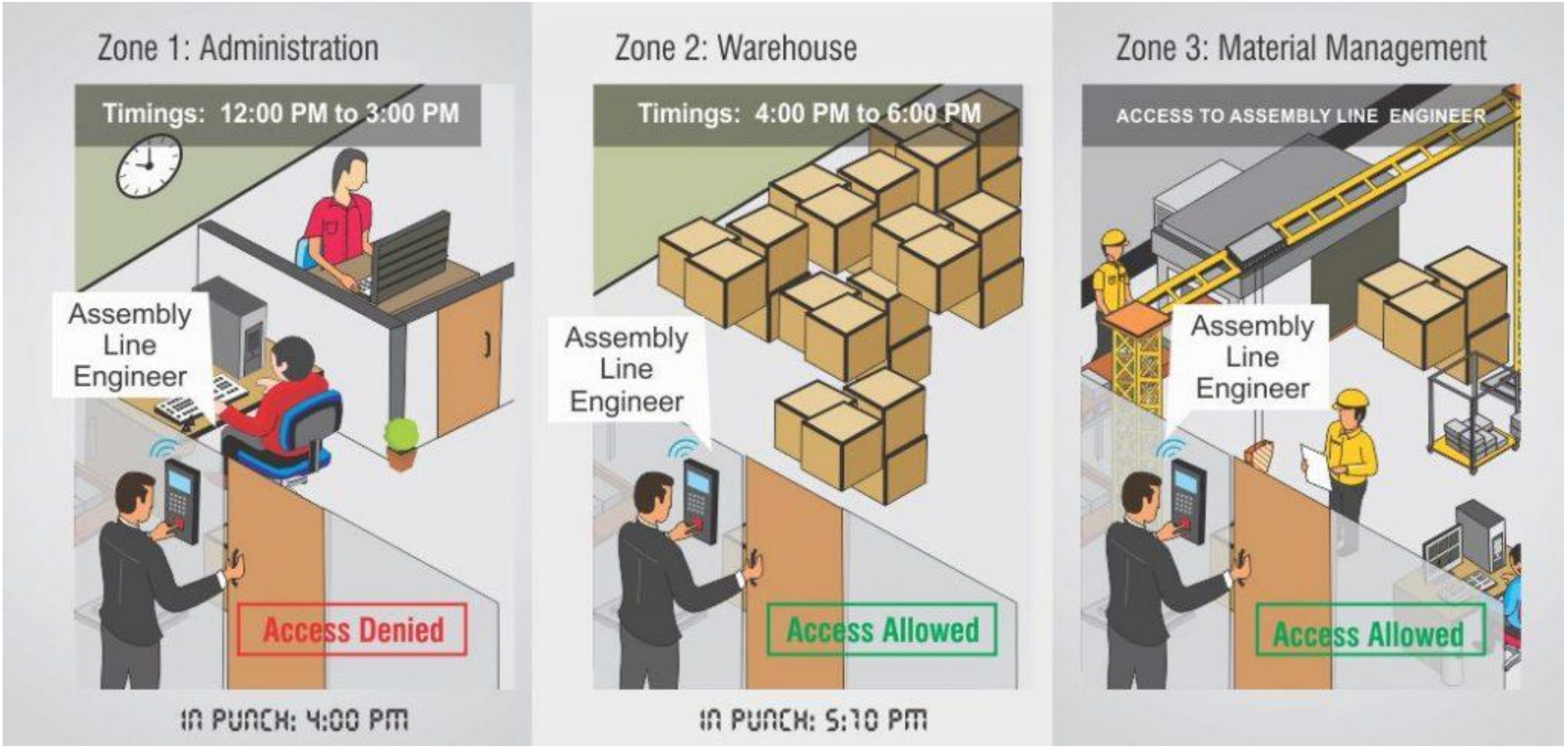
- **RFC 2828** defines **computer security** as:

> "Measures that implement and assure security services in a computer system, particularly those that assure ***access control services***."

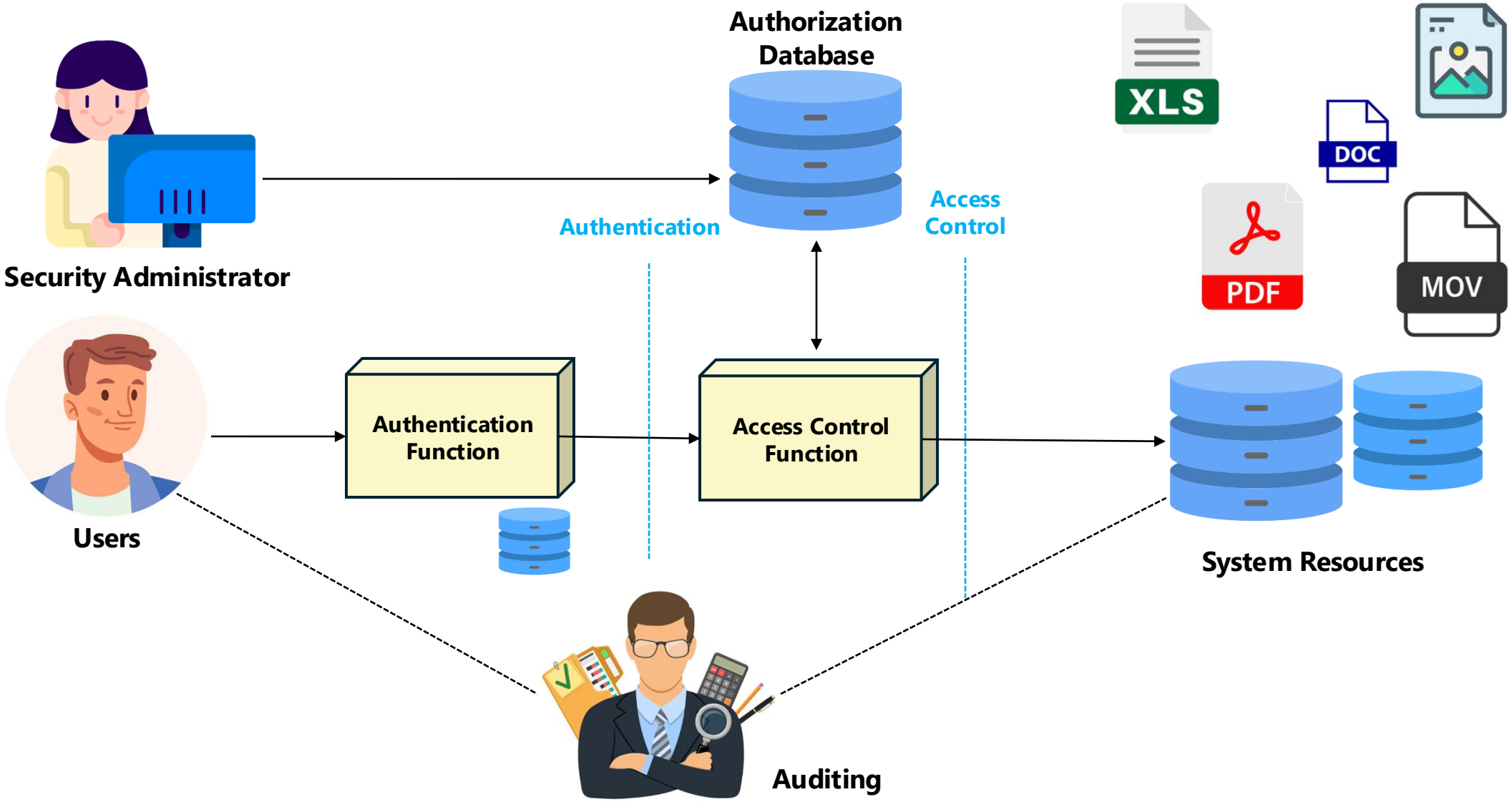
What is Access Control?

- **Social Network:** You can control who can access your personal information, e.g., you can access to all your personal profile, but your friends can access some parts of the profile.
- **Web Browsers:** Access only to a website (same origin policy).
- **Operating Systems:** One user cannot arbitrarily access/kill another user's files/processes.
- **Memory Protection:** Code in one region (e.g., Ring 3), cannot access the data in another more privileged region (e.g., Ring 0).
- **Firewalls:** If a packet matches with certain conditions, it will be dropped.

What is Access Control?



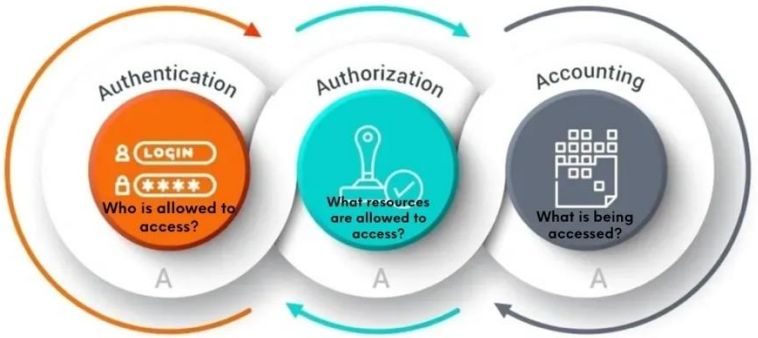
What is Access Control?



Section 5.1

User Identification and Authentication

User Authentication (AuthN)



AAA Security

- **Authentication (AuthN)** is the process of verifying the identity of users.
  - There are **two reasons** to do this:
    1. To make access control decisions. How far could this person access?
    2. To enable audit trails. Who have done the activity?
- **Authorization (AuthZ)** is the process of checking what the proven identity can do or access.
- **Accountability** is the process of tracking who do what in the scope.

Authentication Procedure

- The most common **authentication procedure** is as follows:

1. **An individual arrives at a checkpoint.**
   (login dialogues, doors, …)

2. **The individual claims an identity.**
   (Username, Smart Card, Tokens, …)

3. **The individual presents the item need to prove the identity.**
   (Password, PIN, Biometric, …)

"On the Internet, nobody knows you're a dog."

User Authentication Can Use

Something You Know
(Password, PIN, …)

Something You Have
(Keys, Badges, Tokens, Cards, …)

Something You Are
(Biometric, Retina Patterns, …)

Something You Do
(Handwriting, …)

Where You Are

Where you are?

- ❑ does not verify identity, ***unless only one person can enter that location.***
- ❑ could ***reduce*** the number of possible identities.
- ❑ but should rather be thought of as ***access restriction***, than an authentication mechanism

Something You Know: Password

- **Username + Password** is the standard first line of defense.
- The use of password is **widely-accepted** and **does not difficult to implement**.
- However, it may be **difficult to maintain and manage password securely**.
- Password is one of the **main and common target of attack**. In other words, attackers aim to obtain the valid password.

Something You Know: Password

- **Maintaining Passwords:**
  - **People cannot remember:**
    - Infrequently used items (e.g., can you remember where did you keep your sticky note you wrote 5 years ago?)
    - Frequently changed things (e.g., if you change where you place a jar every day, what is the current place?)
    - Many similar items (e.g., if you keep 100 keys in a bucket, where is the key for the storage room?
  - We cannot demand to forget something. **You forget them automatically!**
  - **Recall** your memory is ***harder*** than recognition of something.
  - Non-meaningful words are more difficult to remember.

Something You Know: Password

- **How attackers can obtain a valid password:**
  - Interception at the creation process.
  - Guess it.
  - Steal the note where you have written it down.
  - Watch you when you are entering it or use a key logger.
  - Eavesdrop on transmission
  - Find it in a memory buffer
  - Find it through a spoofing program, through phishing attacks, or more general social engineering approaches
  - Find it reused in another vulnerable program
  - Password recovery

Something You Know: Password

- **How attackers can obtain a valid password:**

Key Logger

Phishing Attack

RE: [Invoice] - Your Membership has been canceled due to payment failed. [#R3UJD9ID]

support@mailer.netflix.com <no-reply@talents-connect.fr>

Fri 9/18/2020 4:37 AM

To [redacted]

NETFLIX

Your Account Is On Hold!

Hi [redacted],

We've locked your account, as you asked. To continue using your account, you need to Update your Information.

If you did not ask to change your information we are here to help secure your account, just contact us.

–Your friends at Netflix

UPDATE INFORMATION >

Questions? Call 1-844-505-2993

100 Winchester Circle, Los Gatos, CA 95032, U.S.A.

Communication Settings | Terms of Use | Privacy | Help Center

This message was mailed to [kiran.94@live.com] by Netflix as part of your Netflix membership.

SRC: 12547_en_US

Something You Know: Password

- **How you can select a secure password:**
  - In general, when you want to protect a thing, you lock it with a key.
    - Houses, cars, and bicycle locks all have physical keys;
    - Protected files have encryption keys;
    - Bank cards have PIN numbers;
    - Your email have a password;
  - All these keys, both physical and electronic, have one thing in common:

    They open their respective locks just as effectively as in the hands of somebody else.
    - You can install advanced firewalls, secure email accounts, and encrypted disks, but if your password is weak, or if you allow it to fall into the wrong hands, they will not do you much good.

Something You Know: Password

- **How you can select a secure password:**

Your Password Should:

- ✓ Be long enough.
- ✓ Have enough variations to make guessing harder.
- ✓ Be easy to remember while it will not violate the points above.
- ✓ Be changed at a reasonable interval.

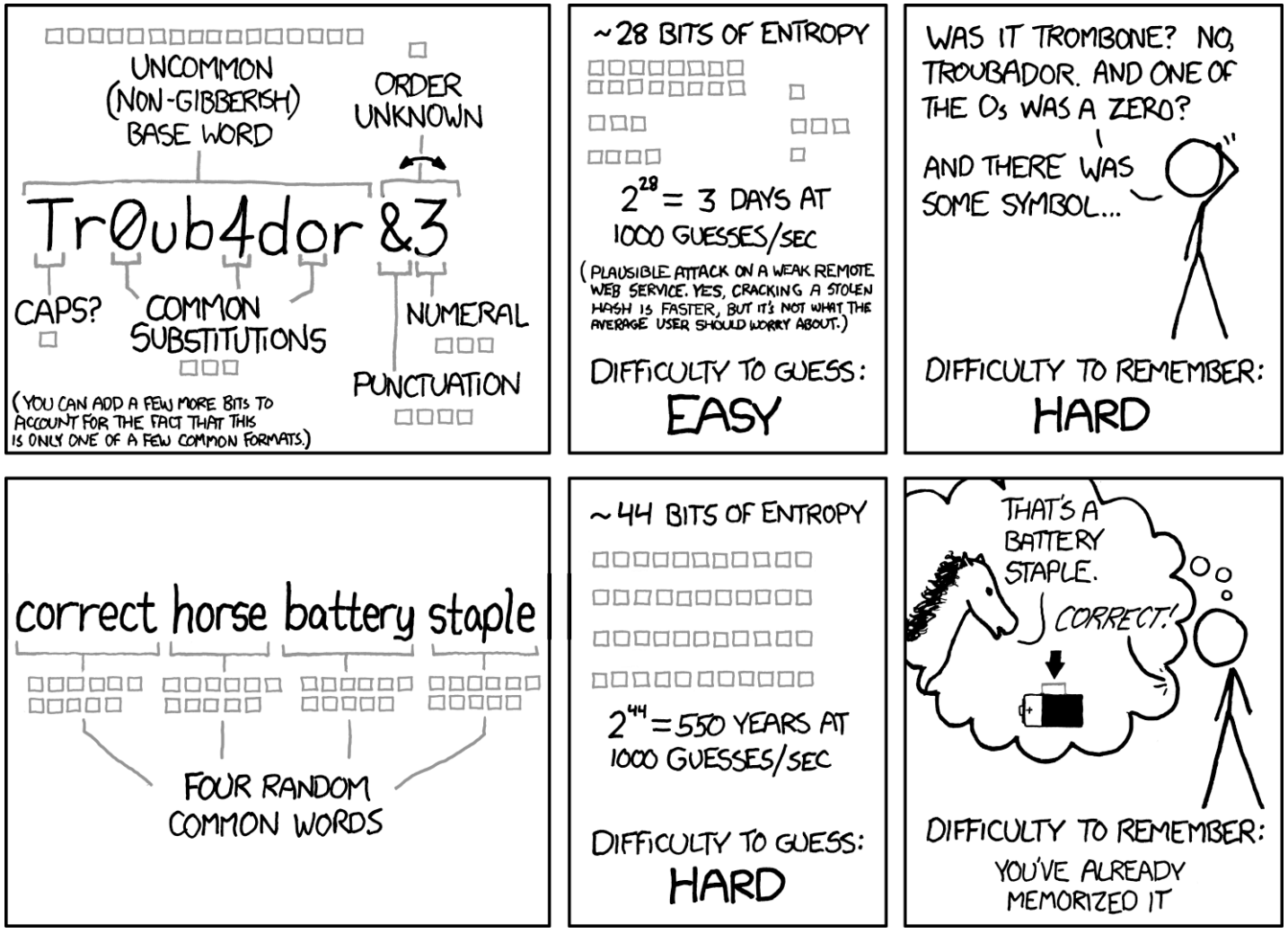
Choose a password you cannot remember, and you must not write it down!

Your Password Should Not:

- ✓ Be anything you reveal outside the authentication.
- ✓ Be the same for two sites where one site is more sensitive and another is less trusted.
- ✓ Be stored in plaintext.
- ✓ Be sent in plaintext.
- ✓ Be connectable to your physical identity, such as your birthday.

Something You Know: Password

- **How you can select a secure password:**



Something You Know: PIN

- Financial PINs are often **four or six digits**.
- While some banks force their customers to set a PIN, some let the customers choose to set.
- Please ***avoid*** using 1111, 2222, 1234, 2345, 123456, 987654, 654321, …, or your birthdate or such information like that.
- Many systems allow ***three attempts*** before locking the card, giving a 0.06% of guessing it for four-digit PINs.
- PIN generation through the IBM 3624 standard uses the account number to generate the PIN (through encryption).
- Despite the encryption key being secret, the connection to the account number allows guessing the PIN on the average in **15 attempts** (Zielinski and Bond, 2002).

Something You Have

- Something you have:
  - Can be stolen;
  - Can be found by others, if it loses;
  - Can be copied, if attackerss know the correct properties;
    - Skimming,
    - Guessing the correct properties,
    - Radio eavasdropping on RFID,
    - Taking photos of the metal key.

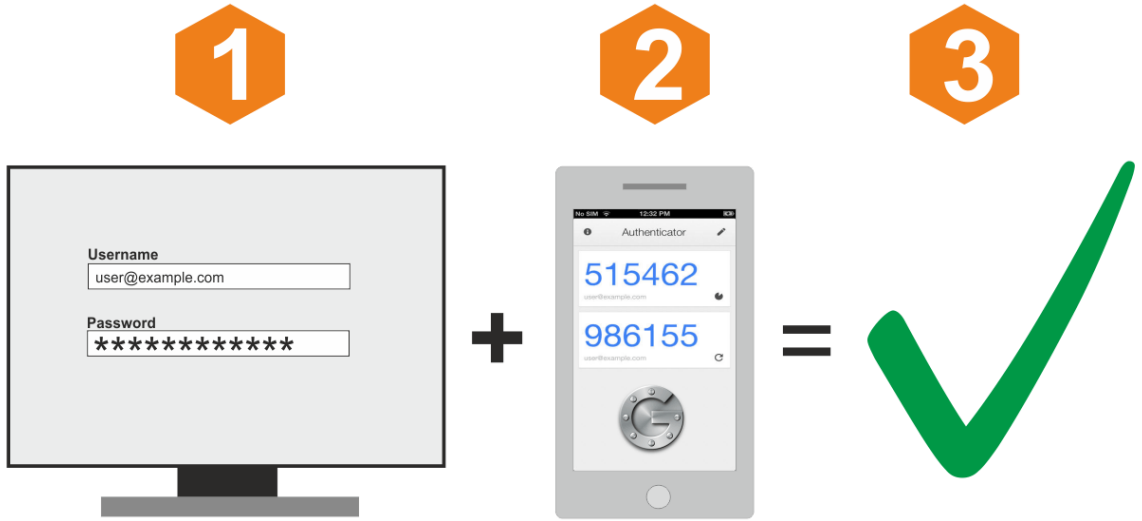
Something You Have

- **Secure Object:** Yubikey
  - Connects as a USB keyboard or via NFC
  - Issue one-time passwords
  - Contains secret AES key used to encrypt a counter
  - The AES key cannot be retrieved, so the key cannot be copied
  - Slightly better security than a physical, ordinary key

- **Other secure objects:**
  - Modern car keys, passports, credit cards, cell phone SIMs, mobile banking IDs, identity cards, smart cards, and bank identification device (“Bankdosa”)

Two-Factor Authentication (2FA)

- In the recent years, **most of secure applications** are moving to use **2FA.**
  - Such as onlione banking applications, ranked computer games, online Bitcoin wallets, hospital systems, etc.
- Two-factor authentication is often referred to as an authentication with **a password + a device**
- Two-factor authentication provides some protection against **phishing and simple password capture attacks**.
- But it is still vulnerable to **Man-in-the-Middle attacks**.



Section 5.2

Access Control Models and Policies

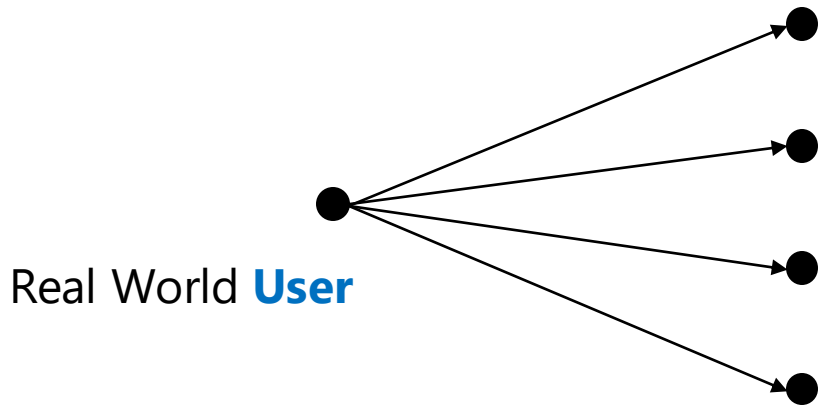
Access Control Vocabulary

- **Basic Abstraction:**
  - **Subjects**
  - **Objects**
  - Access **Rights** or Access **Permission**

- A **Subject** is an entity who wishes to access a certain **Object**, which is a resource (e.g., a file or a network packet).
- The different modes of access (e.g., ***reading***, ***writing***) are called access **rights** or access **permissions**.

Access Control Vocabulary

- **Users and Principals:**

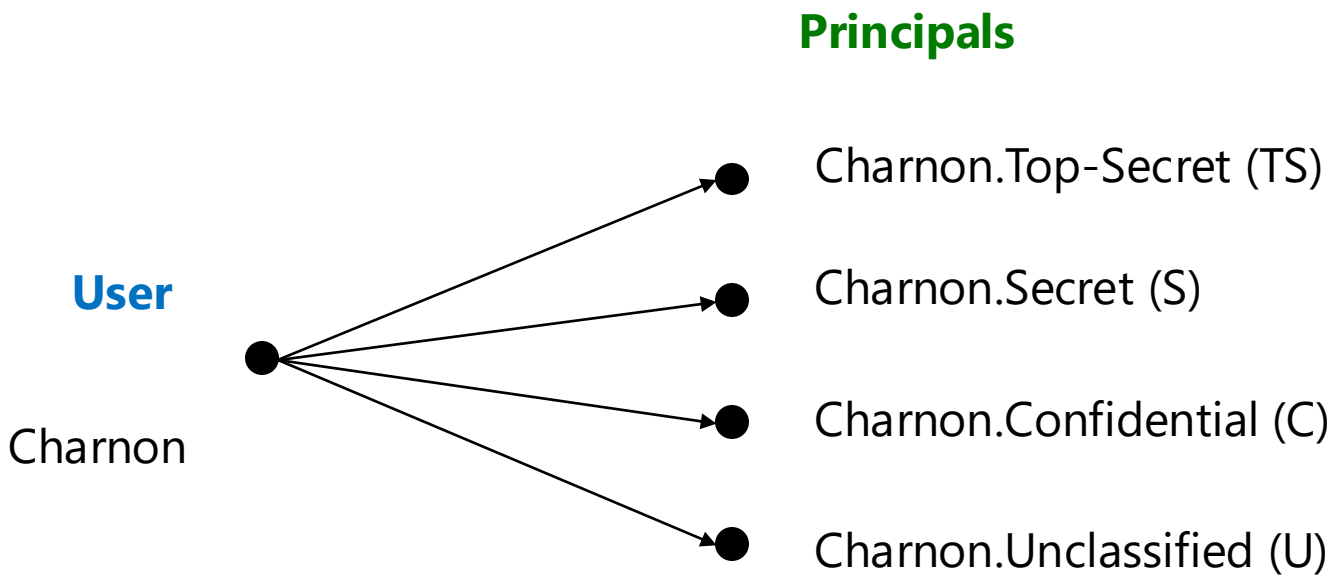


Principals – Unit of Access Control and Authorization.

- A **Principal** is a **User** authenticated at a context.

Access Control Vocabulary

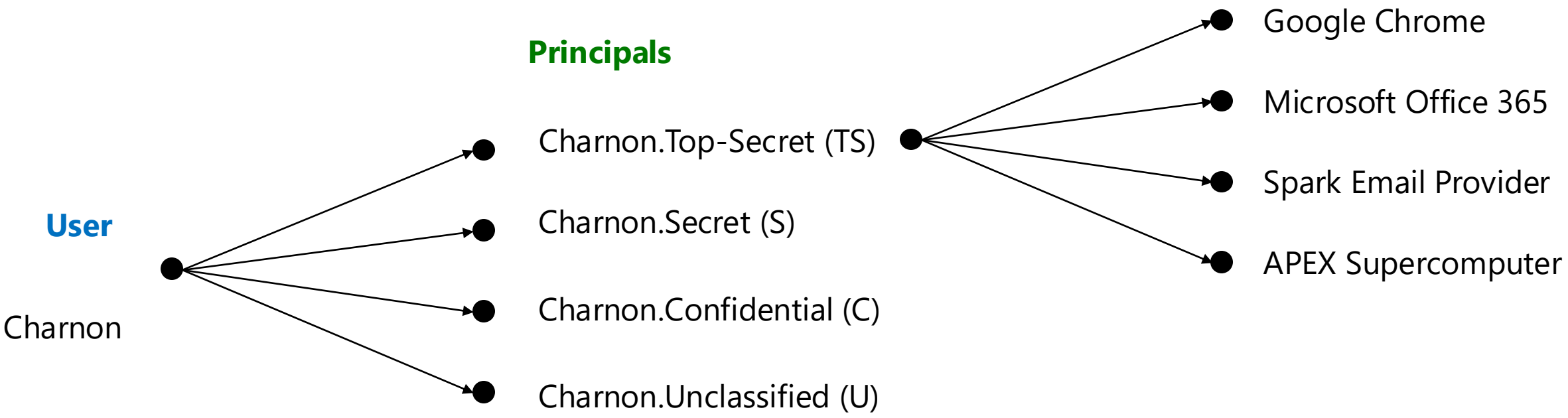
- **Users and Principals:**



- **Example:** A **User** generates multiple **API keys**.

Access Control Vocabulary

- **Users and Principals:**



- A **Subject** is a program executing on behalf of the **Principal**.

Access Control Vocabulary

- The relation between **Users** and **Principals** is One-to-Many (1-to-M):
  - The relation allows accountability of users' actions.
  - This relation provides least privileges required for doing a task.
  - For example, we use API keys instead of the root password.

- For simplicity, **Principals** and **Subjects** can be treated as identical concepts and can be used interchageably.

Access Control Vocabulary

- **Object**:
  - An **Object** is anything on which a **Subject** can perform operations (mediated by **Rights**).
  - **Objects** are usually persistent, for example:
    - File
    - Directory (or Folder)
    - Memory Segment
  - However, **Subjects** (e.g., processes) can also be **Objects**, with operations
    - Kill
    - Suspend
    - Resume

Section 5.3

Access Control Tools and Methodologies

Access Control Tools and Methods

- **Access Control Matrix**
  - An access control matrix is a matrix ($M_{S,O}$) whose rows are subjects and columns are object.

    An element $m_{s_i,O_j} \in M_{S,O} \subseteq P$ is the set of permissions that a subject $s_i \in S$ is authorized to access an object $o_j \in O$.
  - Disadvantages: In a large system, the matrix will be enormous in size and mostly sparse.

	O_1	O_2	O_3	O_4	O_5	O_6
S_1	Y	Y	Y	Y	Y	N
S_2	N	N	Y	N	Y	N
S_3	N	N	N	N	N	Y

Access Control Tools and Methods

- **Access Control Matrix**
  - An access control matrix is a matrix ($M_{S,O}$) whose rows are subjects and columns are objects.
  - **Let's try this together:**
    - Suppose the private key file for S_1 is object O_1, i.e., only S_1 can read,
    - Suppose the public key file for S_1 is object O_2, i.e., all subjects can read, and only S_1 can write.
    - Suppose all subjects can read and write object O_3.
    - ***What is the access control matrix?***

	O_1	O_2	O_3
S_1	?	?	?
S_2	?	?	?
S_3	?	?	?

Access Control Tools and Methods

- **Access Control Matrix**
  - An access control matrix is a matrix ($M_{S,O}$) whose rows are subjects and columns are objects.
  - **Let's try this together:**
    - Suppose the private key file for S_1 is object O_1, i.e., only S_1 can read,
    - Suppose the public key file for S_1 is object O_2, i.e., all subjects can read, and only S_1 can write.
    - Suppose all subjects can read and write object O_3.
    - ***What is the access control matrix?***

	O_1	O_2	O_3
S_1	R	RW	RW
S_2	-	R	RW
S_3	-	R	RW

Access Control Tools and Methods

- **Access Control Matrix:**
  - **Secrecy:** Does this protection state ensure secrecy of S_1's private key in O_1?
  - **Integrity:** Does this access control matrix ensure the integrity of S_1's public key in O_2?
  - **Trust Processes:** Does it matter if we don't trust some of S_1's processes?

	O_1	O_2	O_3
S_1	R	RW	RW
S_2	-	R	RW
S_3	-	R	RW

	O_1	O_2	O_3
S_1	R	RW	-
S_2	-	R	RW
S_3	-	R	RW

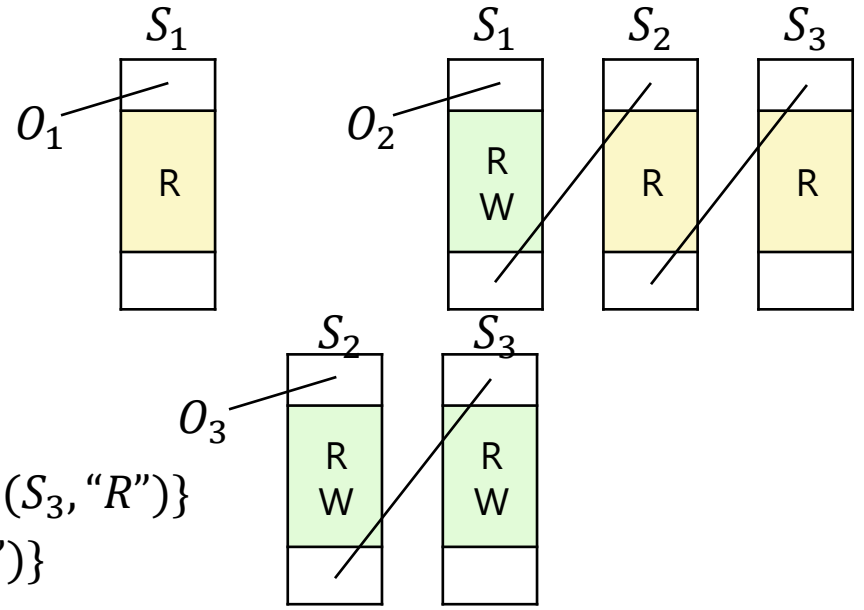
- Non-malicious process should not leak the private key by writing it to O_3.
- A potentially malicious process may contain a **Trojan horse** that can write the private key to O_3.

- **Least Privilege** – Limit permissions to those required only.
- For instance, restrict privilege of the subject S_1 to prevent leaks.

Access Control Tools and Methods

- **Access Control List**
  - An access control list is a set $\{A_o | o \in O\}$, i.e., one element of **object**. The elements of the list are the pair (s, p) of **subject** s who has **permission** p to that object.

	O_1	O_2	O_3
S_1	R	RW	-
S_2	-	R	RW
S_3	-	R	RW



$$A_{O_1} = \{(S_1, "R")\}$$
$$A_{O_2} = \{(S_1, "RW"), (S_2, "R"), (S_3, "R")\}$$
$$A_{O_3} = \{(S_2, "RW"), (S_3, "RW")\}$$

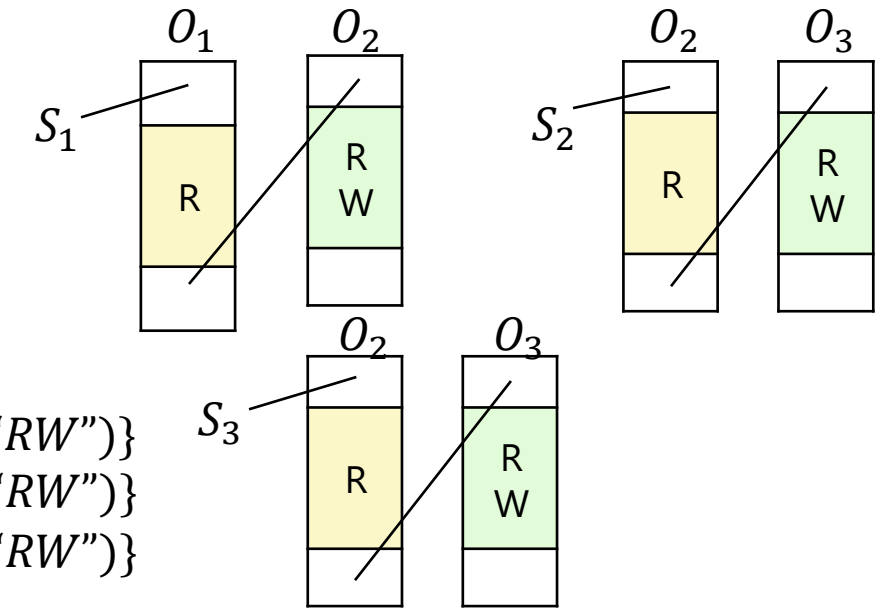
Access Control Tools and Methods

- **Access Control List**
  - **Advantages:**
    - Easy to determine who can access a given object.
    - Easy to revoke all access to an object. Delete one list and revoke access for all subjects.
  - **Disadvantages:**
    - Difficult to know the access rights of a given subject.
    - Difficult to revoke a user's right on all objects.

Access Control Tools and Methods

- **Capability List**
  - A capability list is a set $\{A_s \mid s \in S\}$, i.e., one element of **subject**. A capability can be thought of as a pair (o, p) where o is the name of an **object** and p is a set of **permissions**.

	O_1	O_2	O_3
S_1	R	RW	-
S_2	-	R	RW
S_3	-	R	RW



$$A_{S_1} = \{(O_1, "R"), (O_2, "RW")\}$$
$$A_{S_2} = \{(O_2, "R"), (O_3, "RW")\}$$
$$A_{S_3} = \{(O_2, "R"), (O_3, "RW")\}$$

Access Control Tools and Methods

- **Capability List**
  - **Advantages:**
    - Easy to know the access right of a given subject.
    - Easy to revoke a user access right on all objects. Delete one list can revoke access to all objects for a given subject.
  - **Disadvantages:**
    - Difficult to know who can access a given object.
    - Difficult to revoke all acces right to an object.

Access Control Tools and Methods

- An example of real-world access control tools: **UNIX File Access Control**.
  - Unique user identification number (User ID)
  - Member of a primary group identified by a group ID belongs to a specific group.
  - There are 12 protection bits, specifying read, write, and execute permission for the owner of the file, member of the group, and all other users.

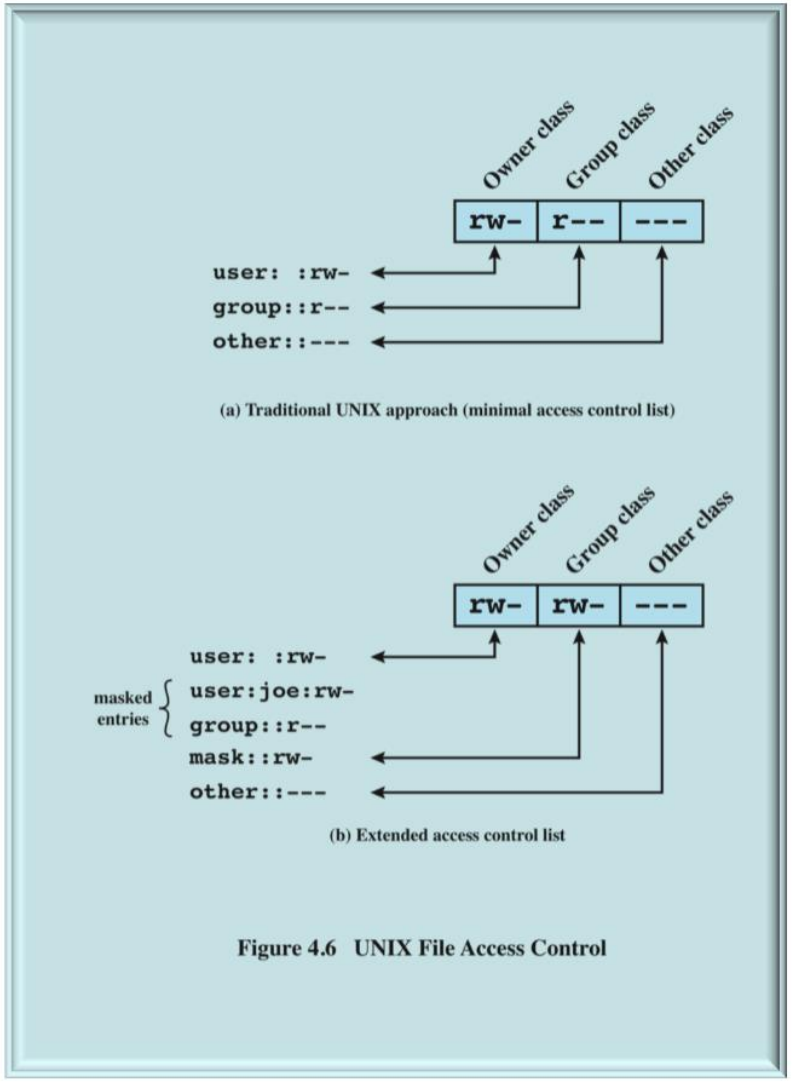
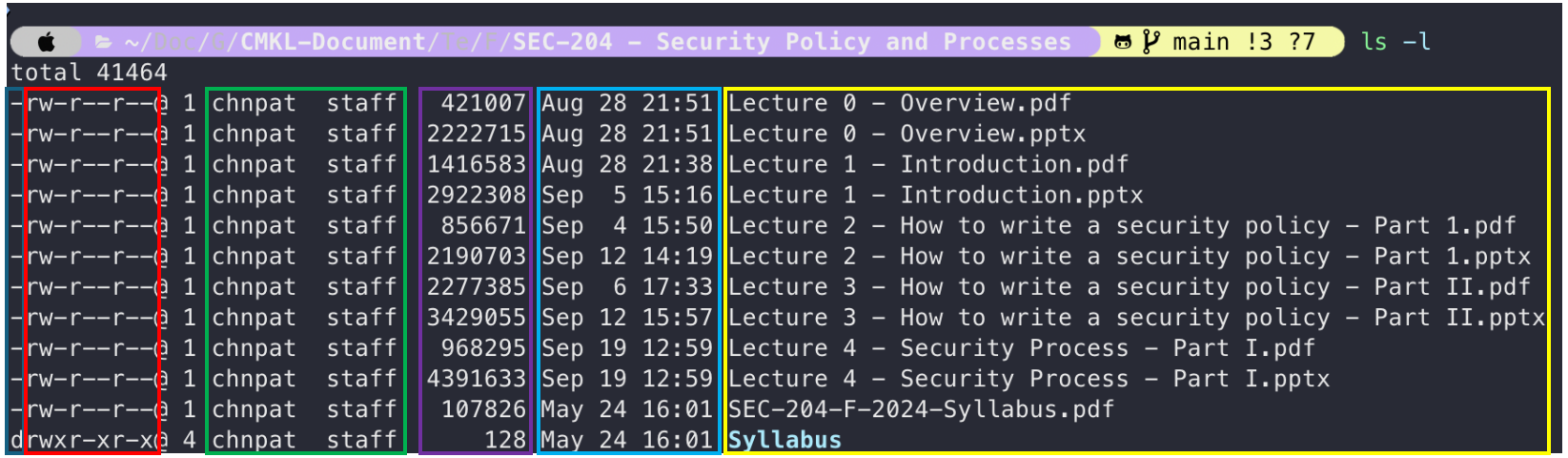


Figure 4.6 UNIX File Access Control

Access Control Tools and Methods

- An example of real-world access control tools: **UNIX File Access Control**.



File Type
D = directory
- = file

Permission List
R = Read
W = Write
X = Execute
--- | --- | ---
Owner | Group | Others

Owner/Group

Last Edited Date and Time

Filename

File size in bytes

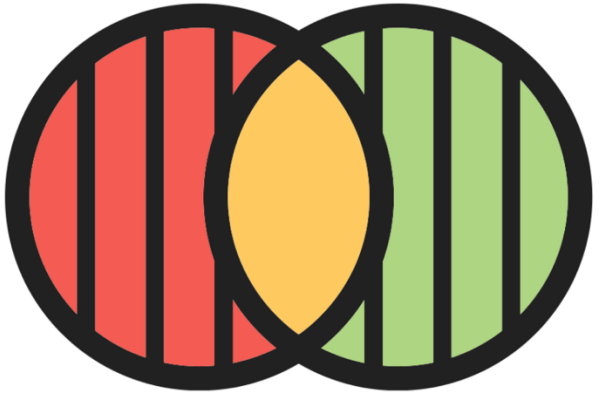
Section 5.4

Access Control Policy Models

Access Control Policy Models

- Access control policy models define how access control policies are configured.

  - **Discretionary Access Control (DAC)**
  - **Mandatory Access Control (MAC)**
  - **Role-Based Access Control (RBAC)**



Access Control Policy Models

- **Discretionary Access Control (DAC)**
  - **Definition:** *An individual user can set access control mechanisms to allow or deny access to an object he/she owns.*
  - Relies on the **object owner** to control access.
  - DAC is widely implemented in **operating systems**, such as Windows, UNIX, etc.
  - Its **flexibility** is the key strength of DAC, so users can dynamically control access to their own objects.
- **Formal Definition of DAC**
  - Let S be the set of all subjects, O be the set of all objects, and P be the set of all permission. The description of access control can be given by a set $A \subseteq S \times O \times P$.
  - When a new permission is **granted**, a new triplet is added to A.

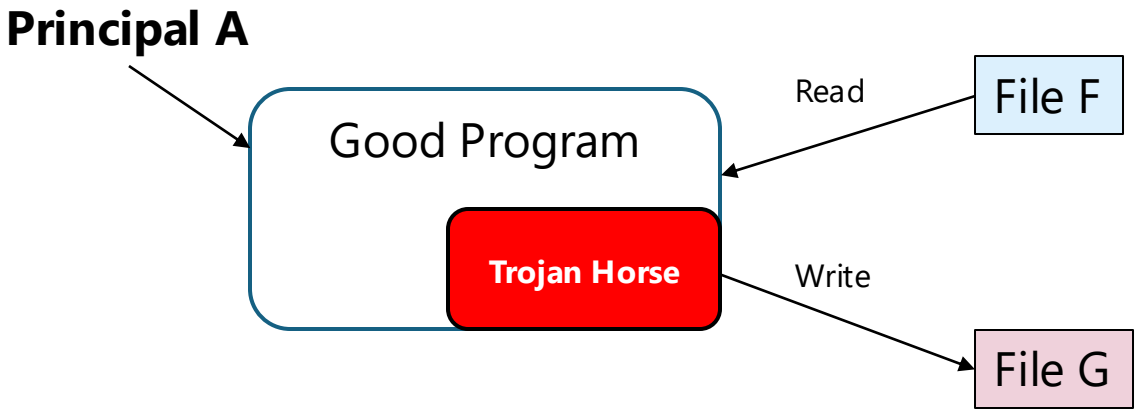
$$A = \{(S_1, O_1, \{\text{“Read”}, \text{”Write”}\})\} \xrightarrow{G(S_1, O_2, \{\text{“Read”}\}))} A = \{(S_1, O_1, \{\text{“Read”}, \text{”Write”}\}), (S_1, O_2, \{\text{“Read”}\})\}$$

  - When a permission is removed or **revoked**, a corresponding triplet is deleted from A.

$$A = \{(S_1, O_1, \{\text{“Read”}, \text{”Write”}\}), (S_1, O_2, \{\text{“Read”}\})\} \xrightarrow{R(S_1, O_2, \{\text{“Read”}\}))} A = \{(S_1, O_1, \{\text{“Read”}, \text{”Write”}\})\}$$

Access Control Policy Models

- **Discretionary Access Control (DAC)** – ***Problems***
  - The philosophy behind the DAC policy model is that subjects can determine who has access to their objects.
    - There is a difference, though, between trusting a person, and trusting a program.
    - When you allow a program to access your objects, do you believe that the program will not be used maliciously?
  - The copies of files are not controlled.
  - The Trojan Horse attack since 1970.



Access Control List

File F

Principal A: Read
Principal A: Write

File G

Principal B: Read
Principal A: Write

Principal B cannot read file F, but the Trojan Horse can read file F and write to file G.

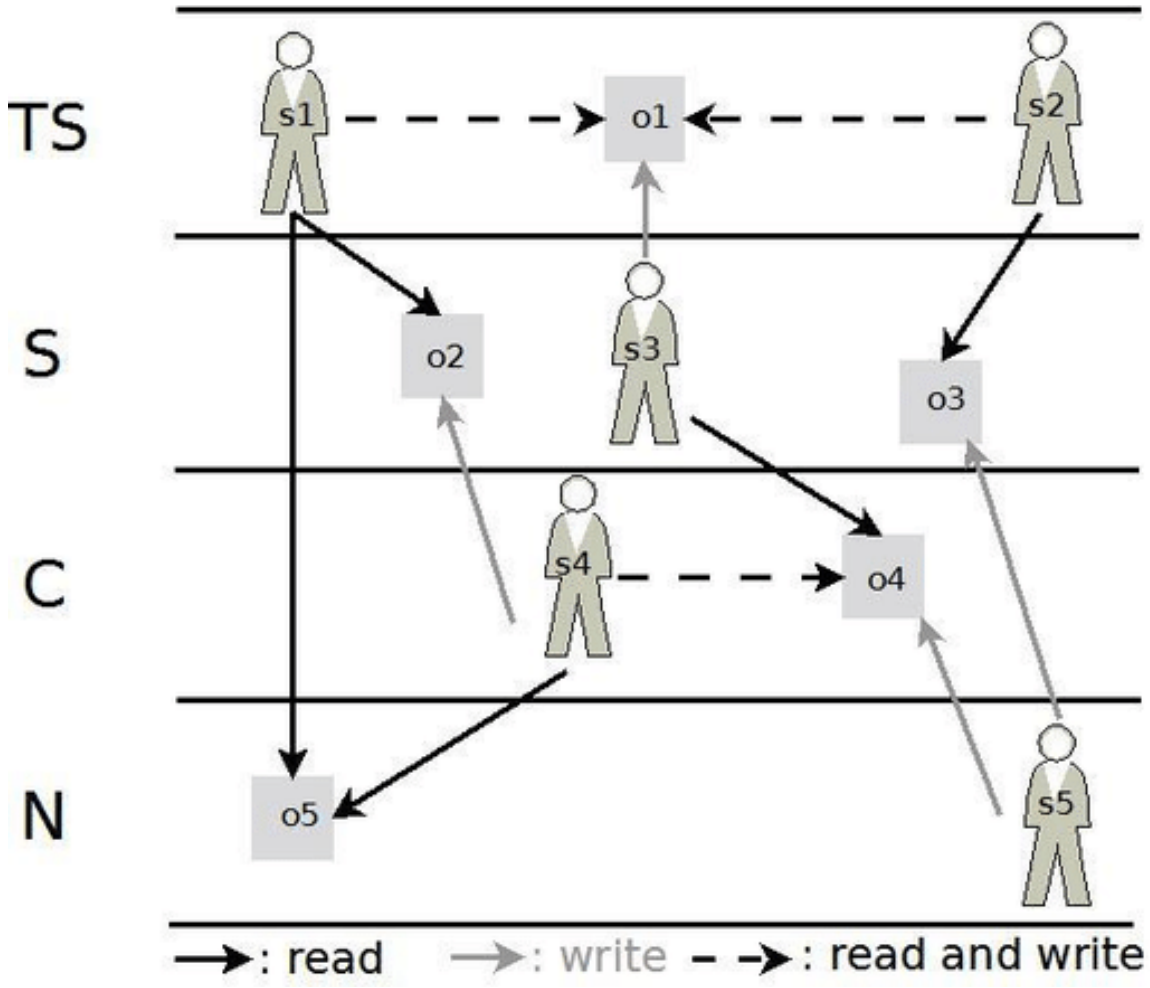
This means that computers with only DAC cannot be trusted to process information classified at different levels.

Access Control Policy Models

- **Mandatory Access Control (MAC)**
  - **Definition:** *A system-wide policy decrees who is allowed to have accesss and individual user cannot alter that access policy.*
  - Relies on the **system** to control access.
  - **Example:** The law allows a court to access driving records without the owners' permission.
  - Traditional MAC mechanisms have been tighly coupled to **a few security models**.
  - Recently, systems supporting flexible security models start to appear (e.g., SELinux, Trusted Solaris, TrustedBSD, etc.)

Access Control Policy Models

- **Mandatory Access Control (MAC)** – **Bell LaPadula (BLP) Model**

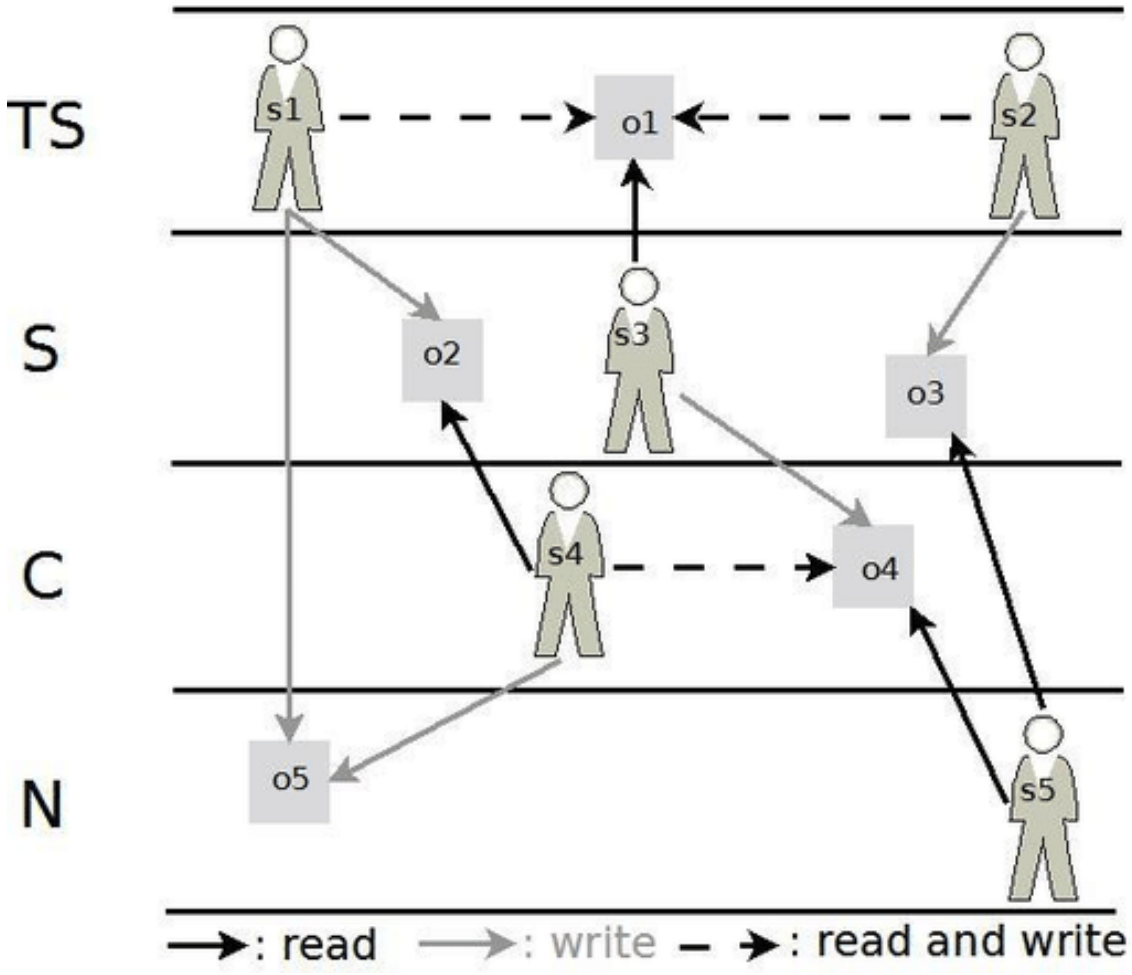


	o1	o2	o3	o4	o5
s1	read write	read			read
s2	read write		read		
s3	write			read	
s4		write		read write	read
s5			write	write	

Confidentiality
Rule 1: No Read Up
Rule 2: No Write Down

Access Control Policy Models

- **Mandatory Access Control (MAC)** – **Biba Model**

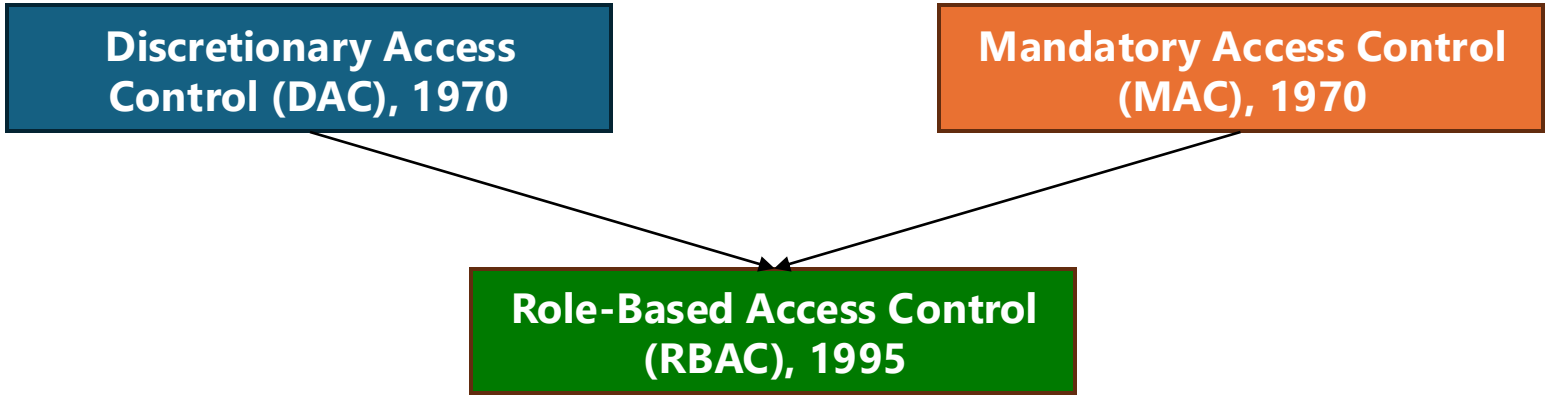


	o1	o2	o3	o4	o5
s1	read write	write			write
s2	read write		write		
s3	read			write	
s4		read		read write	write
s5			read	read	

Integrity
Rule 1: No Write Up
Rule 2: No Read Down

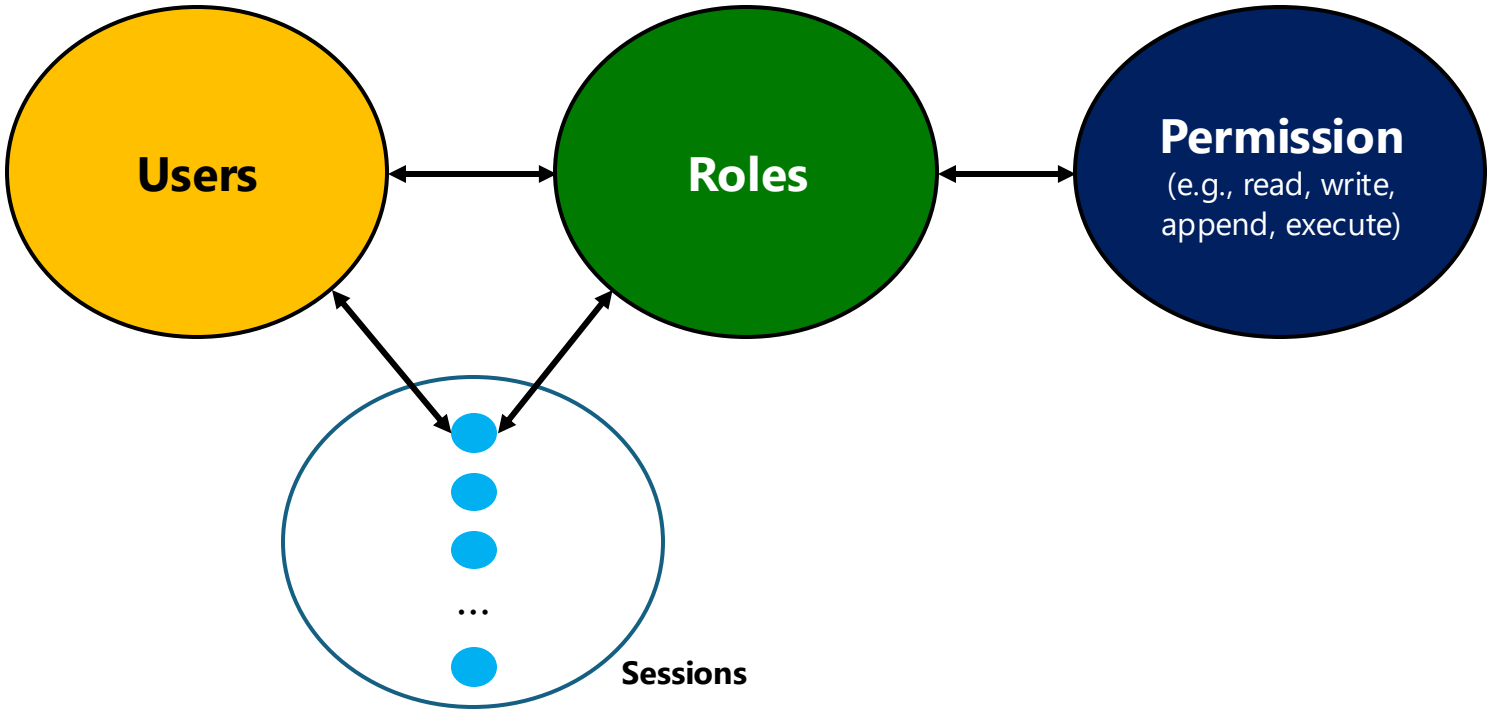
Access Control Policy Models

- **Role-Based Access Control (RBAC)**
  - In real-world situations, security policies and access control policies are dynamic.
  - For example, an employee has been promoted at his job. His permission must be changed according to the new role, i.e., adding or deleting permission.
  - RBAC has been introducued to provide the dynamic and flexible control of access.



Access Control Policy Models

- **Role-Based Access Control (RBAC)**
  - RBAC can be configured to do MAC, i.e., roles of the entire system.
  - RBAC can be configured to do DAC, i.e., roles based on the identities.
  - RBAC changes the underlying **subject-object** model.
    - Access policies are relations between **roles**, **objects**, and **permissions**.
  - Subjects are now assigned to roles – **Role Assignments**
  - Roles can be structured in **a hierarchy**.



Access Control Policy Models

- **Role-Based Access Control (RBAC)** - Role as a policy
  - A role brings together:
    - A collection of users
    - A collection of permissions
  - These collections will be changed over time.
  - A user can be a member of many roles.
  - Each role can have many users as members.

	R_1	R_2	R_3
S_1	x	-	-
S_2	-	x	x
S_3	-	-	x

	O_1	O_2	O_3
R_1	R	RW	-
R_2	-	R	RW
R_3	-	R	RW

Access Control Policy Models

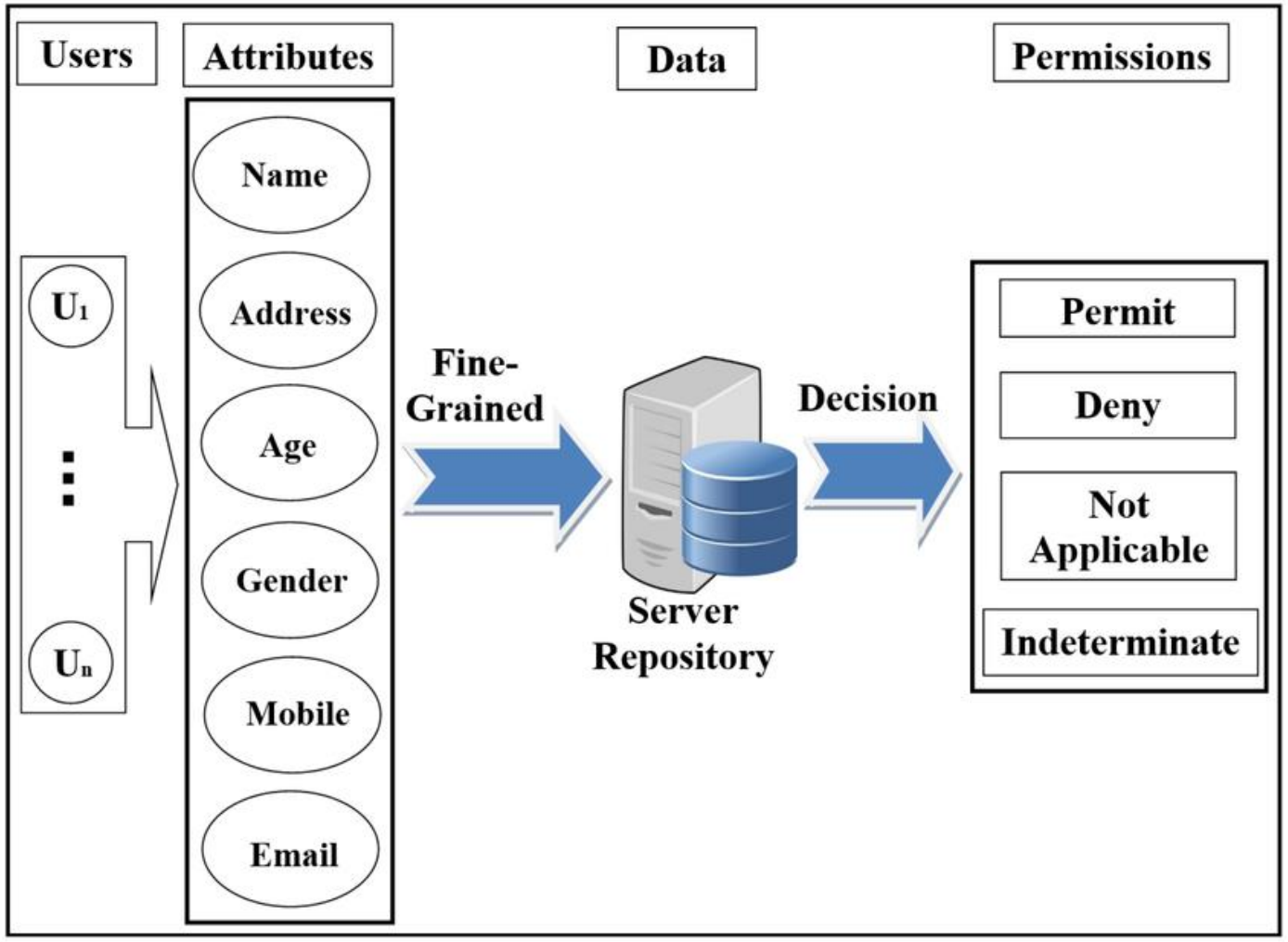
- **Role-Based Access Control (RBAC)** – RBAC Shortcomings
  - Role granularity is not adequate leading to **role explosion**. For example, if there are hundreds of roles defined, how to assign roles to a new employee properly.
  - Role design and engineering are difficult and expensive. **Any redundant role?**
  - Assignment of users/permissions to roles is **crumbersome**. How many roles should a user have?
  - **Adjustment** bvased on local/global situational factors is difficult. For example, the product line engineering role is temporarily prohibited to access the storage room because of the fraud case investigation.

Section 5.5

Future Access Control

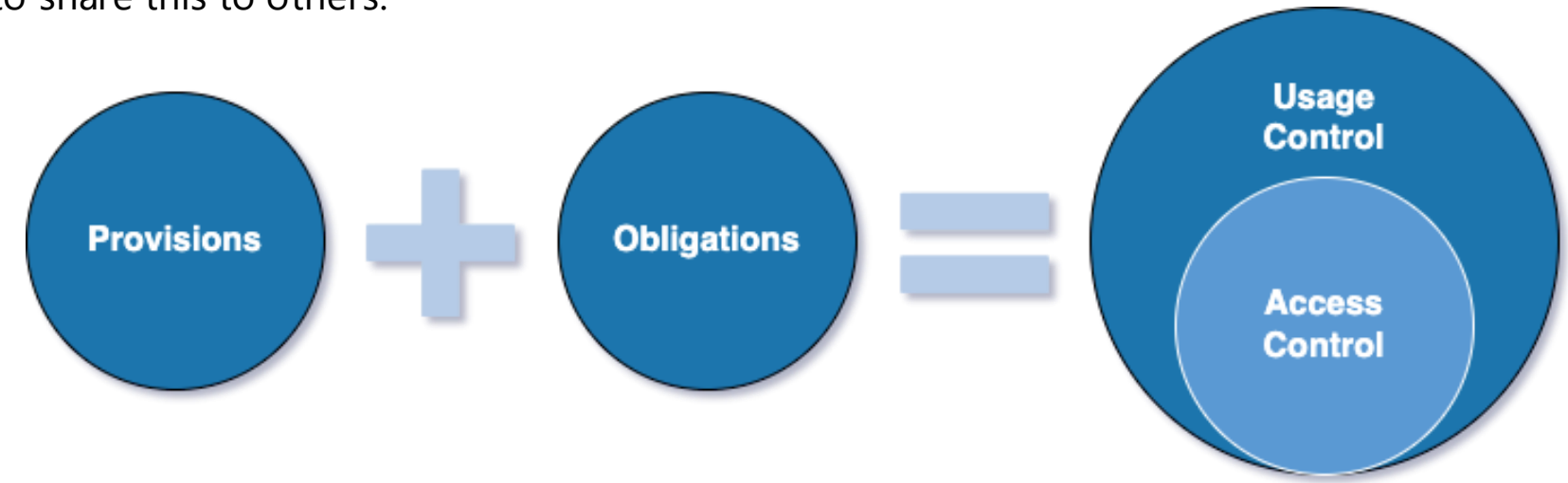
Future Access Control Policy Models

- **Attribute-Based Access Control (ABAC)**
  - **An attribute** is a name-value pair:
    - Attributes are possibly chained and dependent.
    - Values can be complex data structures.
  - An attribute is associated with users, subjects, objects, and contexts.



Future Access Control Policy Models

- **Usage Control (UCON)**
  - Unified framework for access control, trust management, and digital rights management.
  - Data usage control basically works by attaching data usage policy information to data being exchanged and continuously controlling the way data is **processed**, **aggregated**, or **forwarded** to other endpoints.
  - For example, a manager is obligated to receive a monthly report for item procurement, but the manager is not obligated to share this to others.



Today's Class Outline

- Upon successful of this lecture, you will know about:
  - An overview and concepts of **user authentication**.
  - An overview of the need of **access control** to files or objects.
  - Common **methodologies and tools** for access control.
  - Common **policy models** of access control.

End of the Lecture

Please don’t hesitate to raise your hand and ask questions if you're curious about anything!