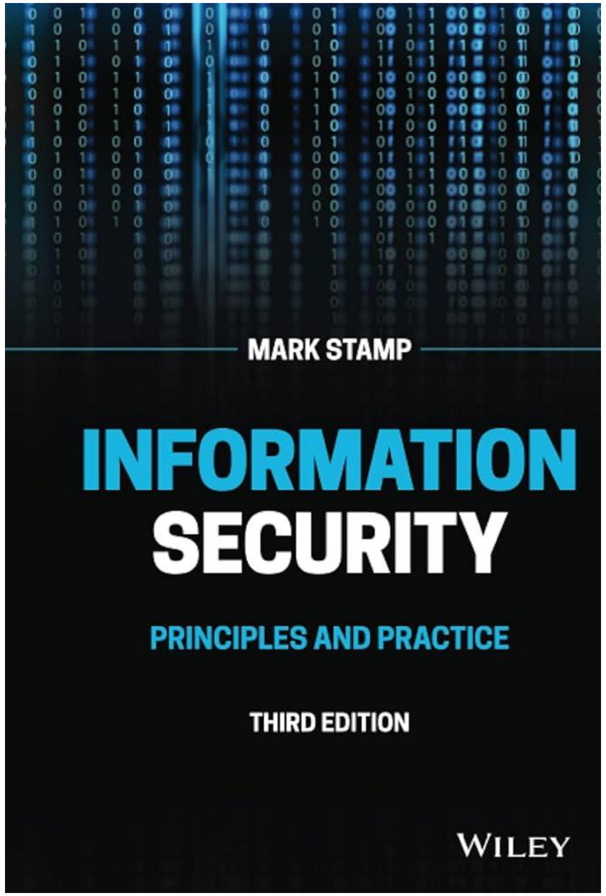
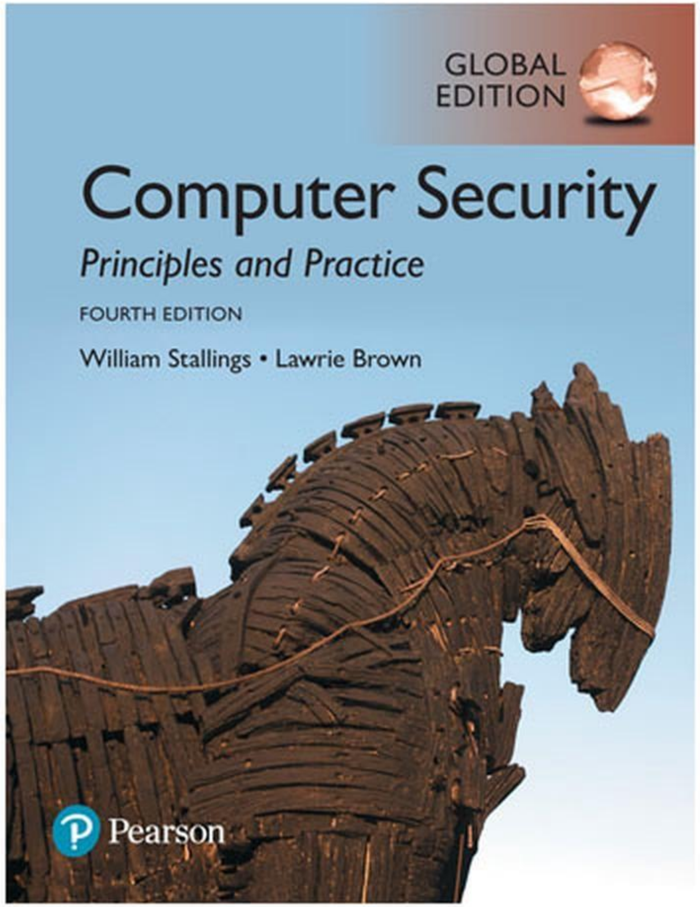
بسم الله الرحمن الرحیم

مبانی رایانش امن

جلسه ۲۱

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان





- فصل ۲۲ و ۲۳ استالینگ
- فصل ۱۰ استمپ

Single Sign-on

- A hassle to enter password(s) repeatedly
  - Alice would like to authenticate only once
  - "Credentials" stay with Alice wherever she goes
  - Subsequent authentications transparent to Alice
- Kerberos — a single sign-on protocol
- Single sign-on for the Internet?
  - Microsoft: **Passport**
  - Everybody else: **Liberty Alliance**
  - Security Assertion Markup Language (**SAML**)

Authorization

Authentication vs Authorization

- ❑ Authentication — Are you who you say you are?
  - o Restrictions on who (or what) can access system
- ❑ **Authorization** — Are you allowed to do that?
  - o Restrictions on actions of authenticated users
- ❑ Authorization is a form of **access control**
- ❑ But first, we look at system certification…

System Certification

- Government attempt to certify "security level" of products

Orange Book Outline

- Goals
  - Provide way to assess security products
  - Provide general guidance/philosophy on how to build more secure products
- Four ***divisions*** labeled D thru *A*
  - D is lowest, *A* is highest
- Divisions split into numbered ***classes***

D and C Divisions

- D — minimal protection
  - Losers that can't get into higher division
- C — discretionary protection, i.e., don't enforce security, just have means to detect breaches (audit)
  - C2 slightly stronger than C1 (both vague)

B Division

- B — mandatory protection
- B is a ***huge*** step up from C
  - C: break security, you might get caught
  - B: “mandatory”, so you can’t break it
- labeled security protection
  - All data labeled, which restricts what can be done with it
  - This access control cannot be violated

A Divisions

- A — verified protection
  - Like B3, but ***proved*** using formal methods
  - Such methods still (mostly) impractical

Authentication vs Authorization

- Authentication — Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** — Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- Classic view of authorization…
  - Access Control Lists (ACLs)
  - Capabilities (C-lists)

Access control

- کنترل دسترسی: جلوگیری از استفاده غیرمجاز از منابع

- از یک جنبه، کل بحث امنیت درگیر کنترل دسترسی است.

- ما به مفهوم دقیق تری به نام مدل کنترل دسترسی میپردازیم.

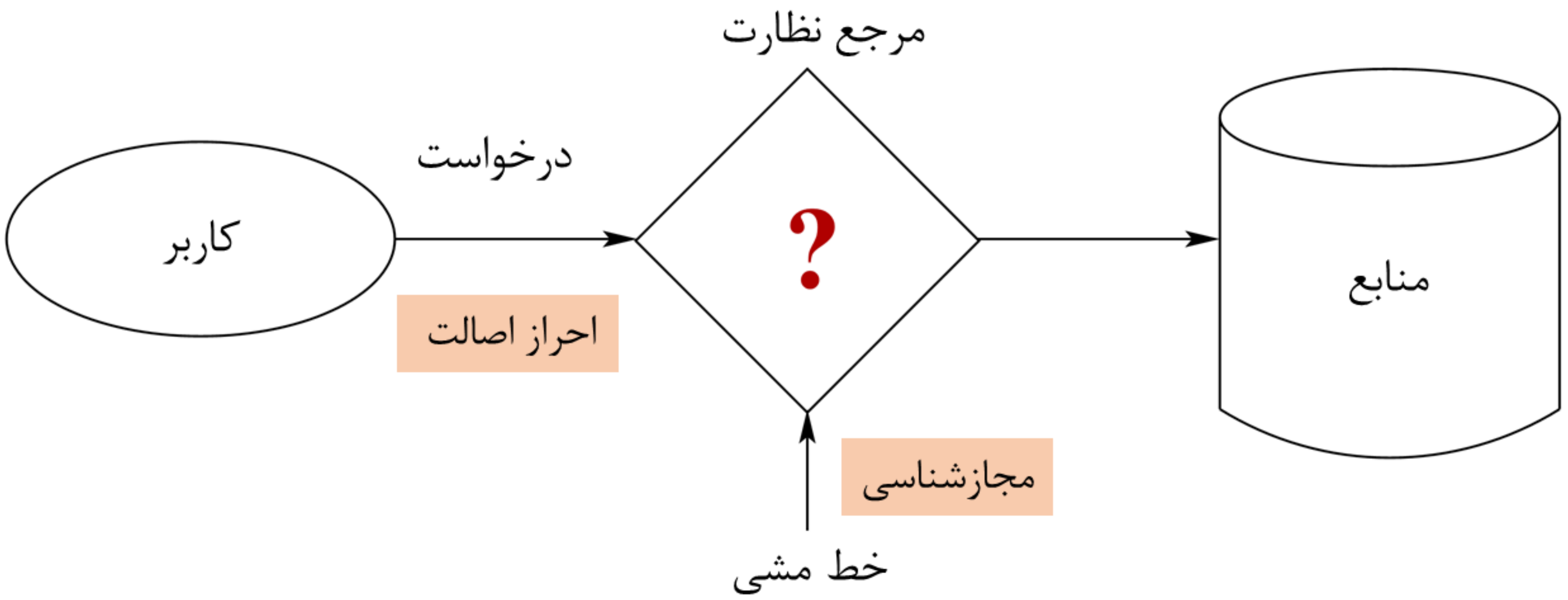
- مدل کنترل دسترسی بیان میکند که چه کسی به چه منابعی و چه نوع دسترسی داشته باشد(ممکن است چه زمانی را هم شامل شود).

Access control

- موجودیت های درگیر در فرآیند کنترل دسترسی
  - عامل (subject): هر کس که متقاضی دسترسی باشد (انسان، ماشین، ...) .
  - شی (object): هر چیزی که قرار است مورد دسترسی قرار گیرد (باید از آن محافظت کنیم).
  - عمل (action): عملی که توسط عامل بر روی شی انجام شود (خواندن، نوشتن، حذف، ...). این عمل توسط حق دسترسی تعیین میشود.
- یک قانون برای دسترسی میتواند به صورت زیر باشد:

عامل A دسترسی خواندن به شی B را دارد.

Access control



Lampson's Access Control Matrix

- **Subjects** (users) index the rows
- **Objects** (resources) index the columns

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Are You Allowed to Do That?

- **Access control matrix** has **all** relevant info
- Could be 100's of users, 10,000's of resources
  - Then matrix has 1,000,000's of entries
- How to manage such a large matrix?
- Note: We need to check this matrix before access to any resource by any user
- How to make this more efficient/practical?

Access Control Lists (ACLs)

- ACL: store access control matrix by **column**
- Example: ACL for **insurance data** is in **blue**

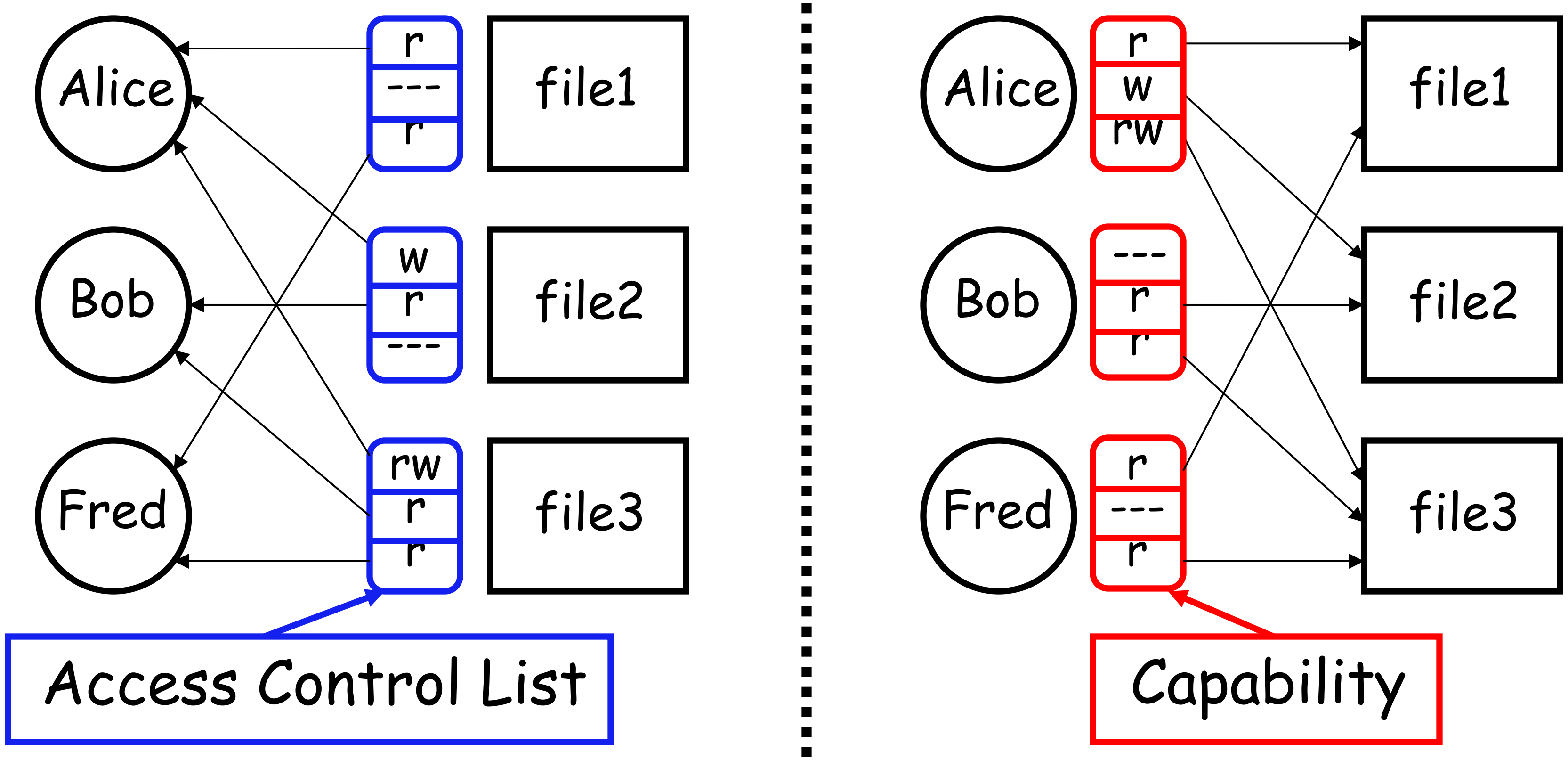
	OS	Accounting program	Accounting data	**Insurance data**	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Capabilities (or C-Lists)

- Store access control matrix by **row**
- Example: Capability for **Alice** is in **red**

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

ACLs vs Capabilities



- ❑ Note that arrows point in opposite directions…
- ❑ With ACLs, still need to associate users to files

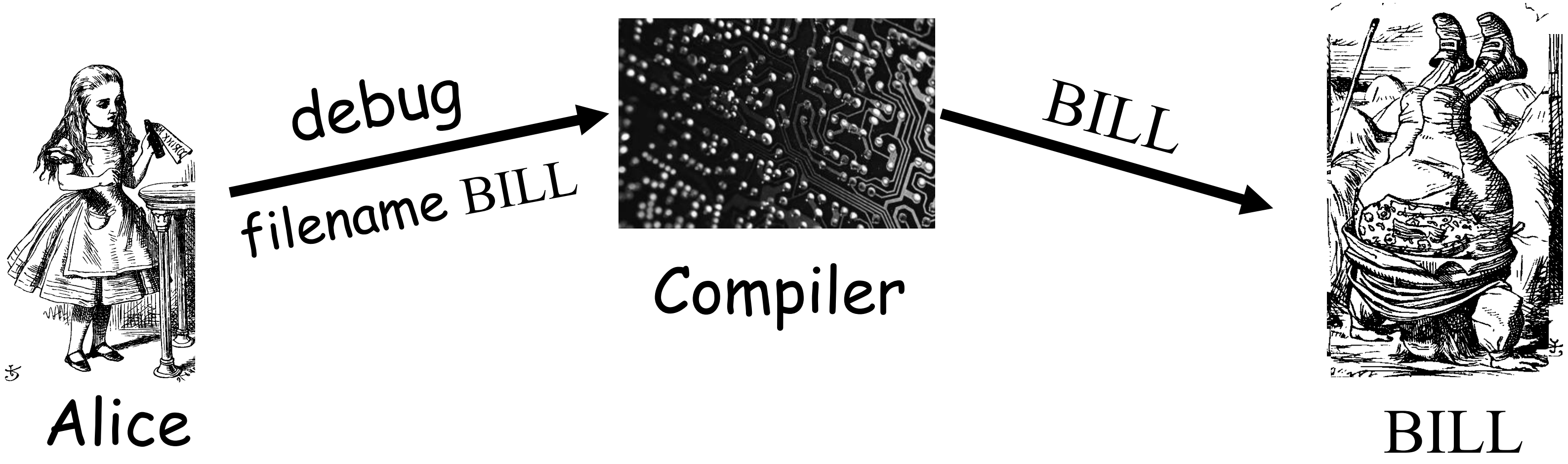
Confused Deputy

- Two resources
  - Compiler and BILL file (billing info)
- Compiler can write file BILL
- Alice can invoke compiler with a debug filename
- Alice not allowed to write to BILL

- Access control matrix

	Compiler	BILL
Alice	x	—
Compiler	rx	rw

ACL's and Confused Deputy



debug

filename BILL

Compiler

BILL

Alice

BILL

- ❑ Compiler is **deputy** acting on behalf of Alice
- ❑ Compiler is **confused**
  - o Alice is not allowed to write BILL
- ❑ Compiler has confused its rights with Alice's

Confused Deputy

- Compiler acting for Alice is confused
- There has been a separation of **authority** from the **purpose** for which it is used
- With ACLs, more difficult to prevent this
- With Capabilities, easier to prevent problem
  - Must maintain association between authority and intended purpose
- Capabilities — easy to **delegate** authority

ACLs vs Capabilities

- ACLs
  - Good when users manage their own files
  - Protection is data-oriented
  - Easy to change rights to a resource
- Capabilities
  - Easy to delegate — avoid the confused deputy
  - Easy to add/delete users
  - More difficult to implement
  - The "Zen of information security"
- Capabilities loved by academics
  - Capability Myths Demolished