University of Pittsburgh

Applied Cryptography and Network Security
CS 1653

Summer 2023

Sherif Khattab

ksm73@pitt.edu

(Slides are adapted from Prof. Adam Lee's CS1653 slides.)

Course Goals (and non-goals)

- Correctly **use** and do cool things with cryptosystems
  - Symmetric key
  - Public key
  - Threshold
- Design and implement functional and secure systems

Non-goals:

- design new cryptographic algorithms
- break into the [your favorite software system]

Topic Outline

- Introduction
- Cryptographic tools
- Applications of cryptography
- OS and Application Security
- Network Security
- Research Issues

Contact Info

- **Course website:** http://www.cs.pitt.edu/~skhattab/cs1653/
- **Instructor:** Sherif Khattab ksm73@pitt.edu
  - OH: TuTh 1-3pm (http://khattab.youcanbook.me)
  - 6307 Sennott Square and https://pitt.zoom.us/my/khattab
- **TA:** Pratik Musale
- **Communication preference**
  1. Piazza
  2. Email

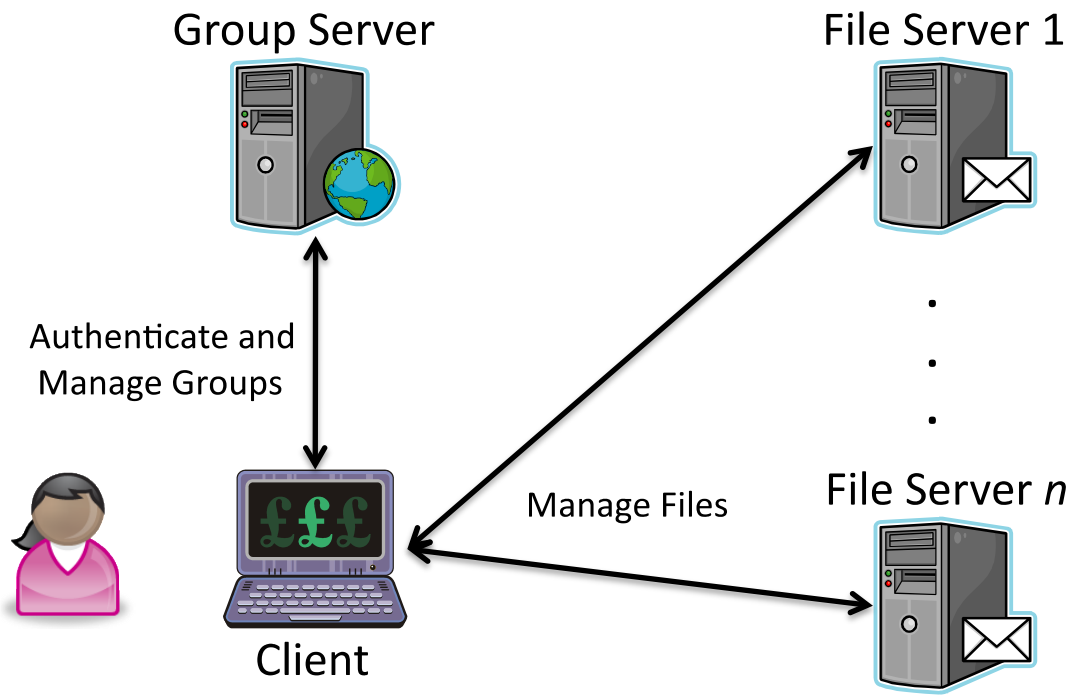
  Please expect a response within 72 hours.

Grading

- **35%** on term project
- **18%** on higher graded exam
- **15%** on three programming assignments
- **12%** on lower graded exam
- **10%** on ten weekly homework assignments
- **10%** class participation: in-class quizzes using Tophat

The Project

Term project will take a security engineering approach to applying the material covered in class



The Project

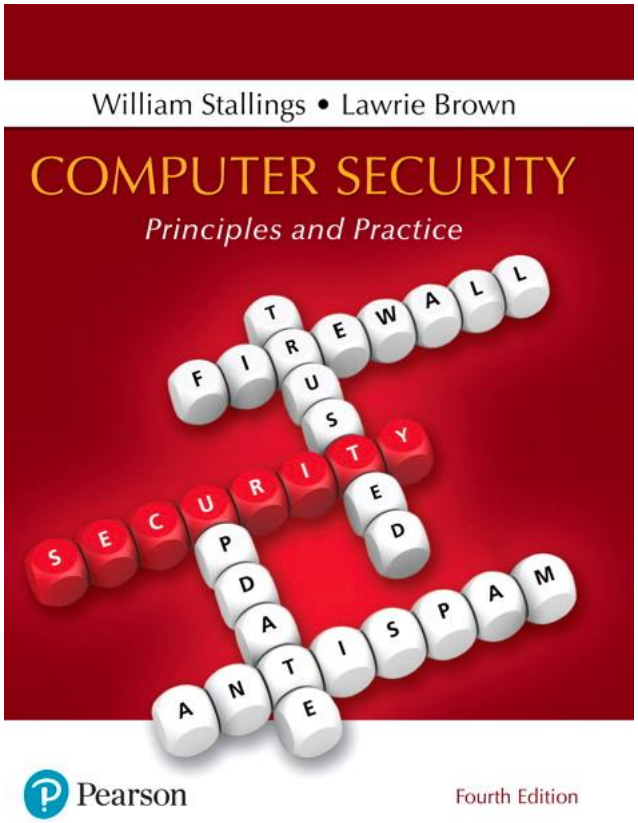
- Groups of 2-3 students
- Five phases (i.e., mini-projects) build upon one another
  - **Apply** the concepts that we cover in class
  - Test your ability to think about problems faced during the **development of "real" products**
  - Cultivate your ability to **work in groups**
- **Important notes:**
  - Choose your group carefully
    - You **cannot "quit"** your group part way through the semester
    - But you can (and must!) report on your division of labor
  - This project is progressive in nature
    - That is, later phases build on earlier phases
    - Working hard early will pay dividends later
    - What if your early phases don't work out so well?

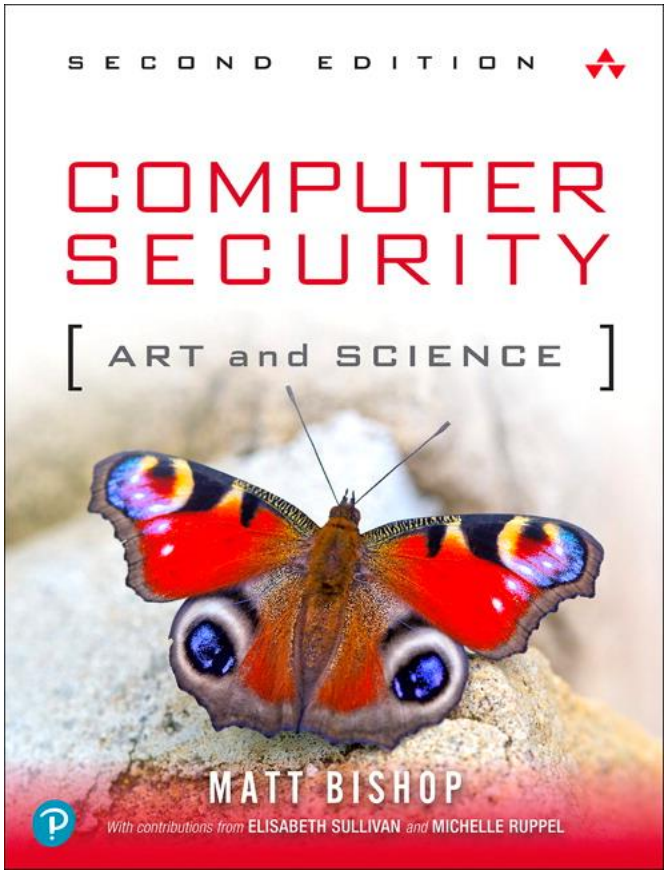
The Project

How will my final project grade be calculated?

Project # (contribution)	Description
1 (10%)	Requirements specification and team formation
2 (25%)	Core functionality
3 (25%)	Security Features I
4 (25%)	Security Features II
5 (15%)	Security Features III

Textbooks





William Stallings and Lawrie Brown, Computer Security: Principles and Practice (4th Edition), Pearson, ISBN-13: 9780134794105

Matt Bishop, Computer Security: Art and Science (2nd Edition), Addison-Wesley, ISBN-13:9780321712332

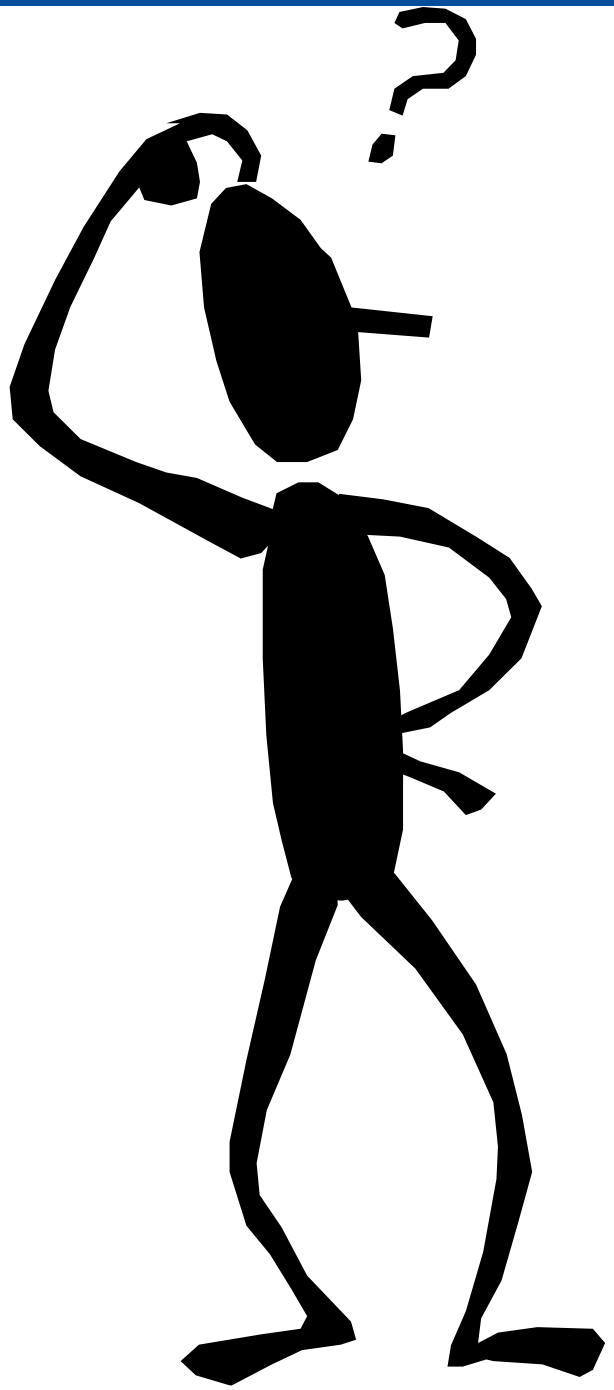
Canvas Walkthrough

- Lectures posted on Tophat and GitHub
  - Draft PDF slides available on Github before class
- Lecture recordings
  - under Panopto Video
- RedShelf Inclusive Access for the Textbooks
  - You can cancel and get a refund before Add/Drop
- Piazza for discussion and communication
- Gradescope for programming assigments
- Academic Integrity
- NameCoach

Expectations

- Your **continuous feedback** is important!
  - Anonymous Qualtrics survey
  - Midterm and Final OMET
- Your **engagement** is valued and expected with
  - classmates
  - teaching team
  - material

Why study computer security?



The good ol' days

Security was much less complicated years ago…

Early desktops had

- A single user
- A single address space
- No permissions
- No network
- Limited data-processing abilities

Leading threat: Computer viruses

Boot-sector viruses

Executable viruses

Both threats could be controlled reasonably well by using anti-virus software, and basic “software hygiene”

The times, they are a-changin'

The Internet

Changes introduced

- Constant data exchange
  - Email, web, etc.
- Machines no longer isolated
- Looooooong uptimes

Results

- Faster virus propagation
  - Email: Days to weeks
  - Active worms: Minutes
- Active attacks against end-user hosts

The times, they are a-changin'

Software/Data Complexity

Changes introduced

Data formats more complex

Software functionality bloating

Boundary between data and executable content is blurred

Results

"Data hygiene" not as easy

Plethora of new vulnerabilities

Vulnerabilities more easily exploited

The times, they are a-changin'

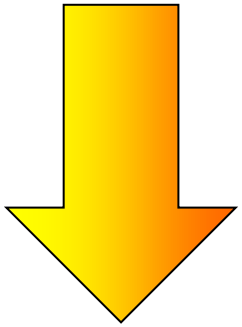
Attacker Motivation

Old days

- Attacks on end hosts not really valuable
- Get "fame" for viruses, etc.

Today

- Everything is online!
- Attackers can benefit financially from viruses, worms, and compromised hosts



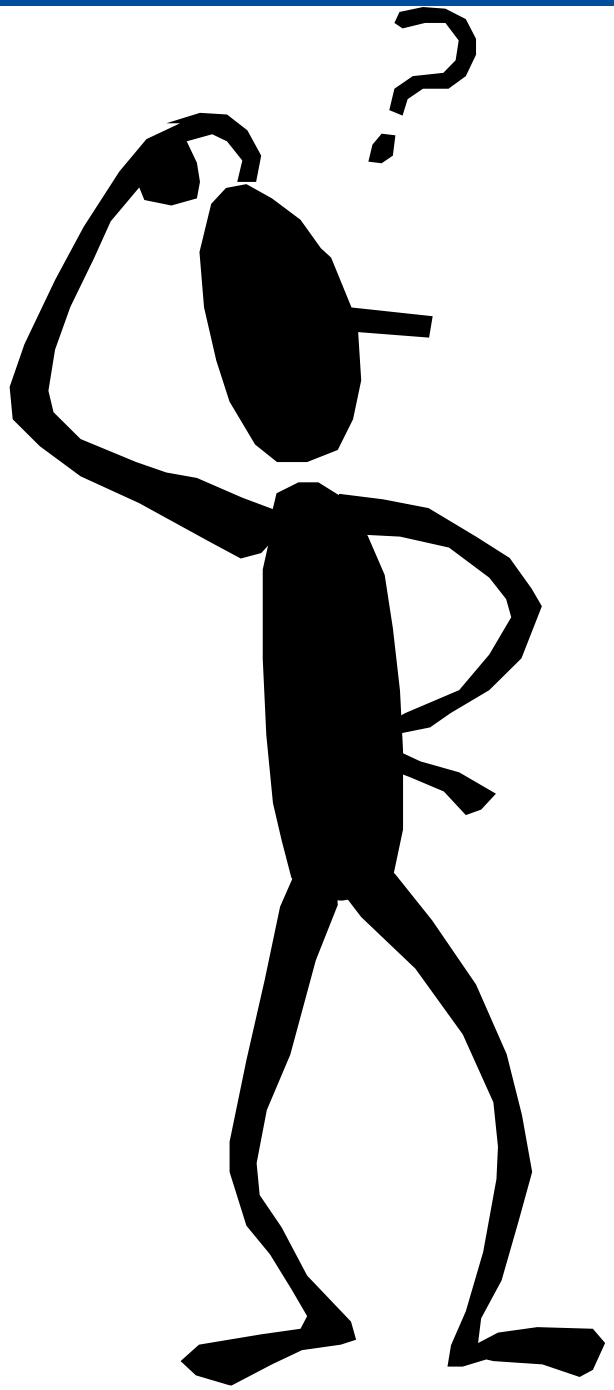
Results

- Spam and botnets
- Organized online crime

Studying security is more important now than ever before

Security is a challenging problem

What is computer security?



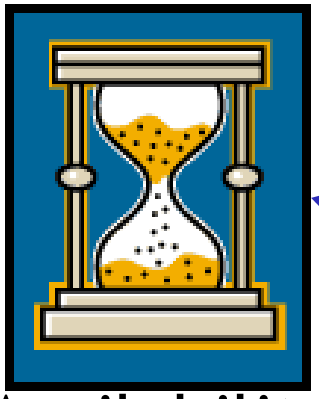
Computer security is typically defined with respect to three types of properties

How do I ensure that my secrets remain secret?

Confidentiality

Can I trust the services that I use?

Integrity

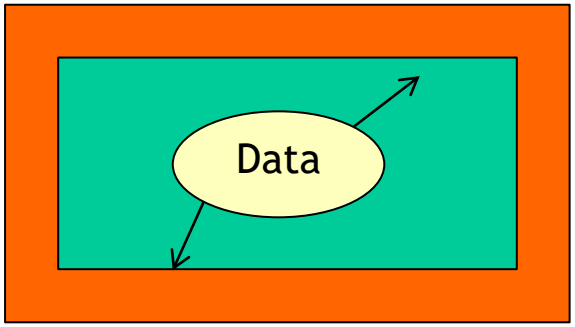


Availability

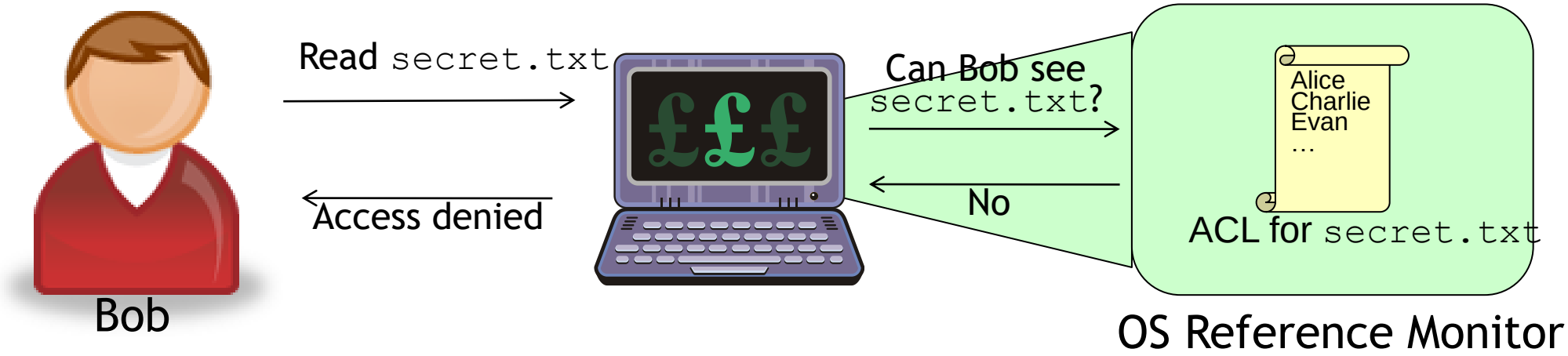
Am I able to do what I need to do?

Confidentiality

- The need to conceal information or resources
- The need for confidentiality arises in both military and civilian information systems
  - Military: Classified information processing and intelligence
  - Civilian: Proprietary data, sensitive records, etc.
- There are many flavors of confidentiality. For example:
  - Data: "No one should know my bank account balance"
  - Existence: "No one should know that I shop at XYZ.com"
  - Configuration: "No one should know what software I run"
- The precise notion of confidentiality used in a system depends on the environment in which the system will be used



Access control systems are designed to preserve data confidentiality



Note: We must trust the OS kernel in order for the reference monitor approach to work.

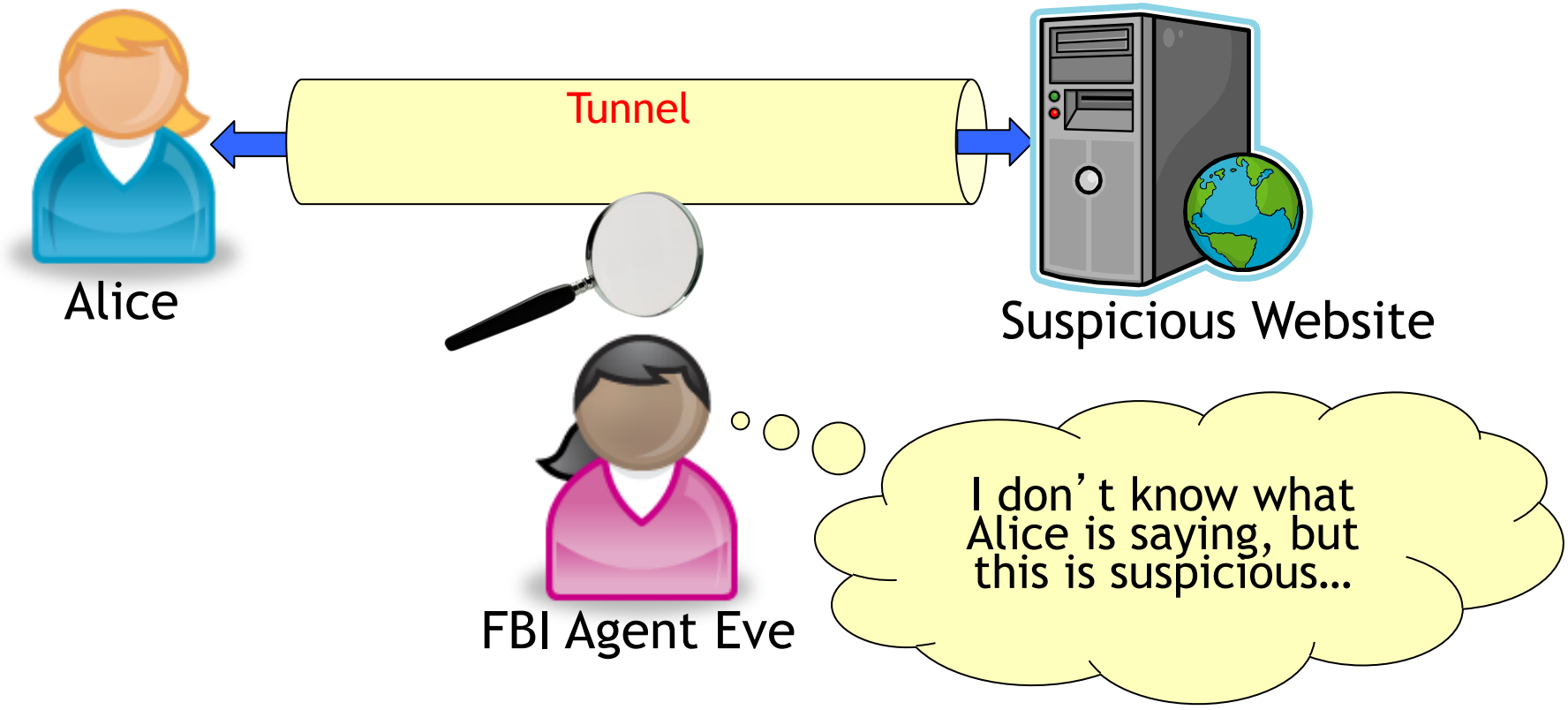
Can you think of a solution that avoids this trust?

Cryptographic protection!

Does cryptography guarantee confidentiality?

While cryptography does provide data confidentiality, it does not always provide existence confidentiality.

Example: SSL



Integrity refers to the trustworthiness of information or resources

Systems are typically concerned with two types of integrity

Data integrity: "Is my bank account balance correct"

Origin integrity: "Was this application really written by Microsoft?"

Integrity mechanisms can either prevent or detect violations

Pro: Keep system consistent

Con: Difficult to design!

"Easy" to keep bad guys out

But how do we identify malicious insiders?

Pro: Easier to design in some cases

- File checksums
- Transaction consistency checks
- Etc.

Con: Causes of violations often remain unknown…

Availability refers to the ability to use information or resources

Systems are usually designed with the expectation of certain patterns of usage

- Electricity demand is higher during the work day
- Roads are more utilized during commute hours
- Web sites experience peak traffic during certain hours
- Etc.

By violating these assumptions, the security properties of the system can be altered!

- ***Example:*** Poorly-protected backup servers

Interesting: Detecting malicious availability violations is non-trivial

- e.g., Botnet-based DDoS attack or “Slashdotted” article?

Discussion

Which of the CIA properties do you feel is most important? Why? Are there other properties that are as important or more important than these?