CSE 120: Principles of Operating Systems
Lecture 14: Security

Prof. Joseph Pasquale
University of California, San Diego

March 6, 2019

What is Computer Security?

- How to protect computer systems
  - System contents: data, software, hardware
  - System operation: performance, reliability
  - System services: what the user sees and expects
- From various threats
  - Theft
  - Damage
  - Disruption

Aspects of Security

- Confidentiality
  - disclose only to those authorized
- Integrity
  - only authorized changes
- Authenticity
  - is it really who/what it claims to be?
- Availability
  - can I access the service?

Security Threats

- Interception
  - eavesdropping
- Interruption
  - destroying, denial of service
- Modification
  - tampering with data or programs
- Fabrication
  - new data/programs, replaying messages

User Authentication

- Passwords are most common method
  - User and computer know secret
  - User proves knowledge of secret
  - Computer checks
- Encrypted passwords
  - Computer stores only encrypted passwords
  - User provides password
  - Computer encrypts, checks

Problem with Passwords

- Assume 100 possible characters for passwords

# chars	# passwords	100G/s	100T/s
6	100^6	10 sec	10 msec
7	100^7	17 min	1 sec
8	100^8	1.2 days	1.7 min
9	100^9	116 days	2.8 hr

- But most are uncommon, hard to remember
- Using dictionary words (~250,000): 2.5 µsec!

Challenge/Response Protocol

- Challenge/response, algorithmic passwords
  - User and system know secret algorithm
  - System challenges user's knowledge, user responds
- Example: say secret algorithm is $f(x) = x^2$
  - System challenges user: sends user a value, say 3
  - User computes f(3) = 9: sends system 9
  - System concludes user must know secret algorithm
  - Next time, system can provide different challenge
- Secret is never sent, only challenge/response

Trojan Horse

- Greeks invaded Troy in hollow wooden horse
- Program that contains hidden malicious code
  - User thinks program does something useful
  - In actuality, it (also) does something harmful
- Program runs as process in user's domain
  - Can do harm to user's environment
  - Can do harm under that user's name

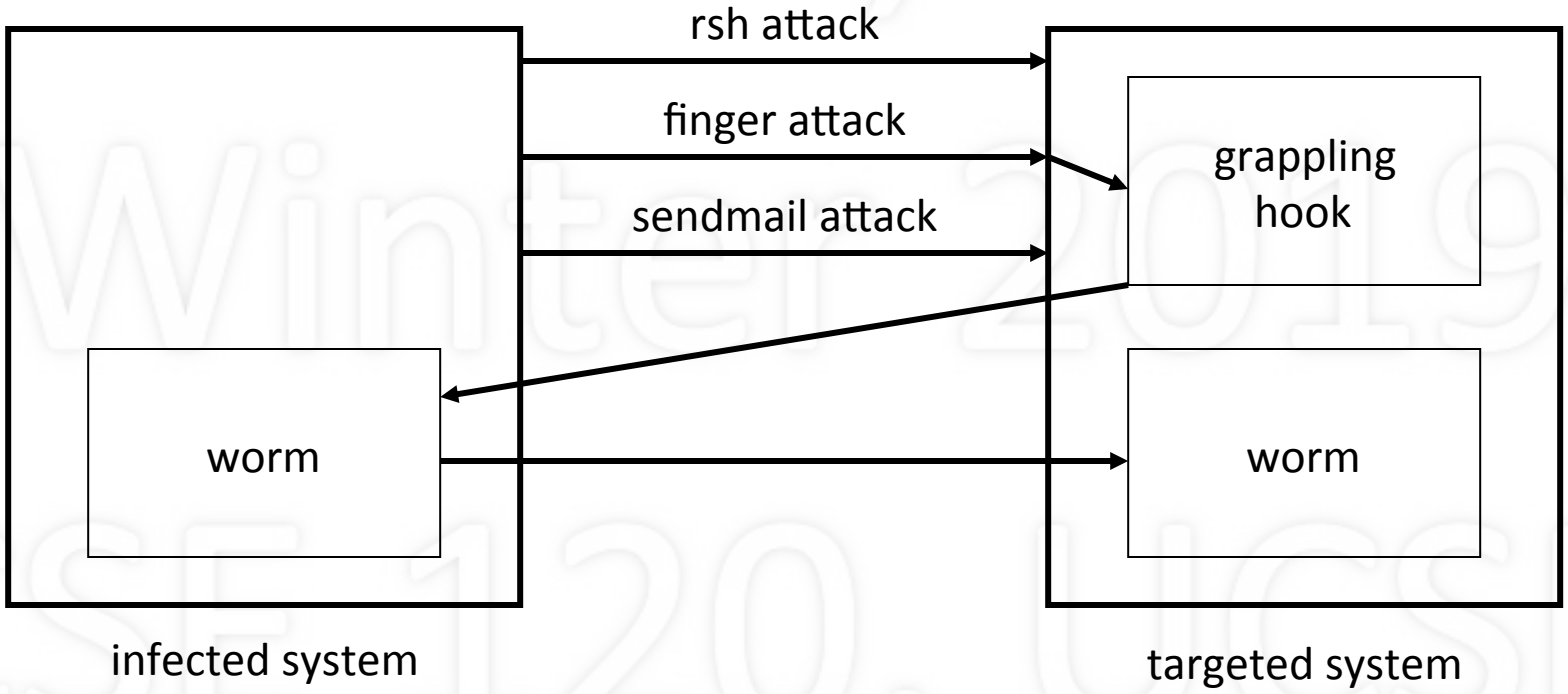
Trap Door

- Secret access point in program
- Designer develops program for someone else
- Once loaded in system, designer can access
- Consider if trap door is added by compiler
  - Compiler adds trap doors to programs
  - Designer of compiler can then access
  - Can't tell from program source code
  - Even if new compiler written, must be compiled!

Virus

- Code attached to legitimate program
- When program runs, the virus runs
  - causes damage
  - spreads, attaching itself to other programs
- Disinfectants
  - Check that programs look normal (not modified)
  - Check for known virus patterns in programs

Internet Worm



- Worm: program that copies itself over network
- Internet worm (Nov 2, 1988)

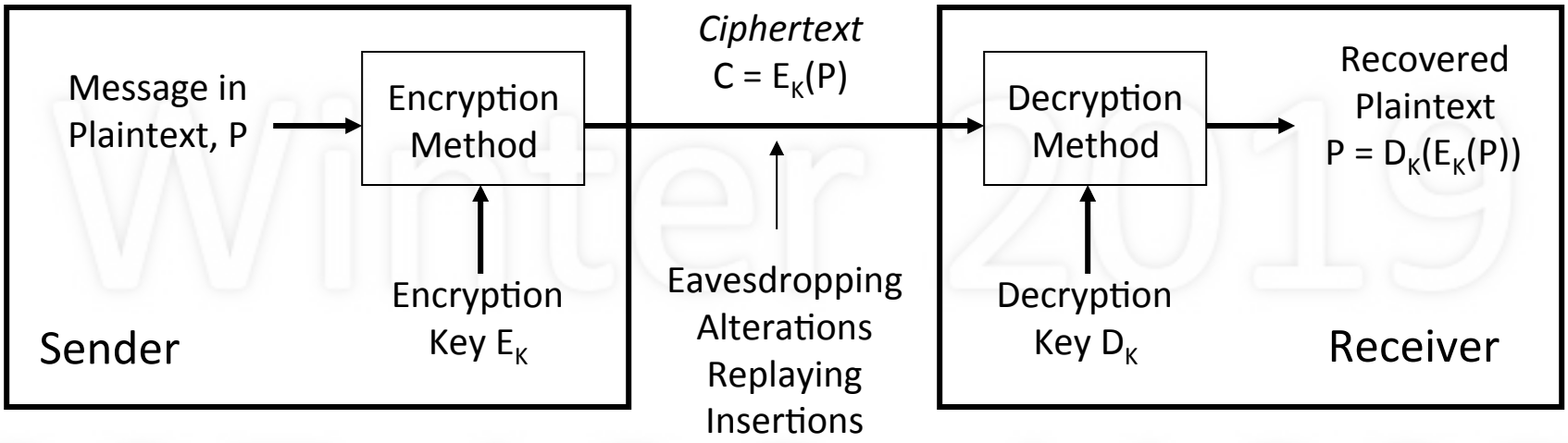
Denial of Service

- Preventing others from using system
  - by using lots of resources
  - by bombarding with network requests or traffic
- Example
  - Repeatedly request TCP connection
  - Don't answer responses; system times them out
  - Eventually, no TCP ports left available

Intrusion Detection

- Detecting if there is an intruder, or an attack
- Signature-based
  - Look for specific patterns of attack behavior
  - Example: repeated login attempts
- Anomaly-based
  - Look for unusual behavior
  - Example: unusual command/system call patterns
- Solution: create audit trail (log), then analyze

Cryptography



- Encoding messages to
  - limit who can view the original message
  - determine who sent a message

Secret Key Encryption

- Secret key (symmetric)
  - Same key K is used to encrypt and decrypt
  - Sender encrypts $E_K(P)$, Receiver decrypts $D_K(E_K(P))$
- DES: Data Encryption Standard (1977)
  - Weak due to 56-bit keys
- AES: Advanced Encryption Standard (2001)
  - 128, 192, 256-bit keys

Public Key Encryption

- Public key (asymmetric)
  - Different keys to encrypt and decrypt
  - Each user has two keys: one public, one private
- If A wants to send to B
  - A encrypts using B's public key
  - B decrypts using its private key
- RSA (Rivest, Shamir, and Adelman)

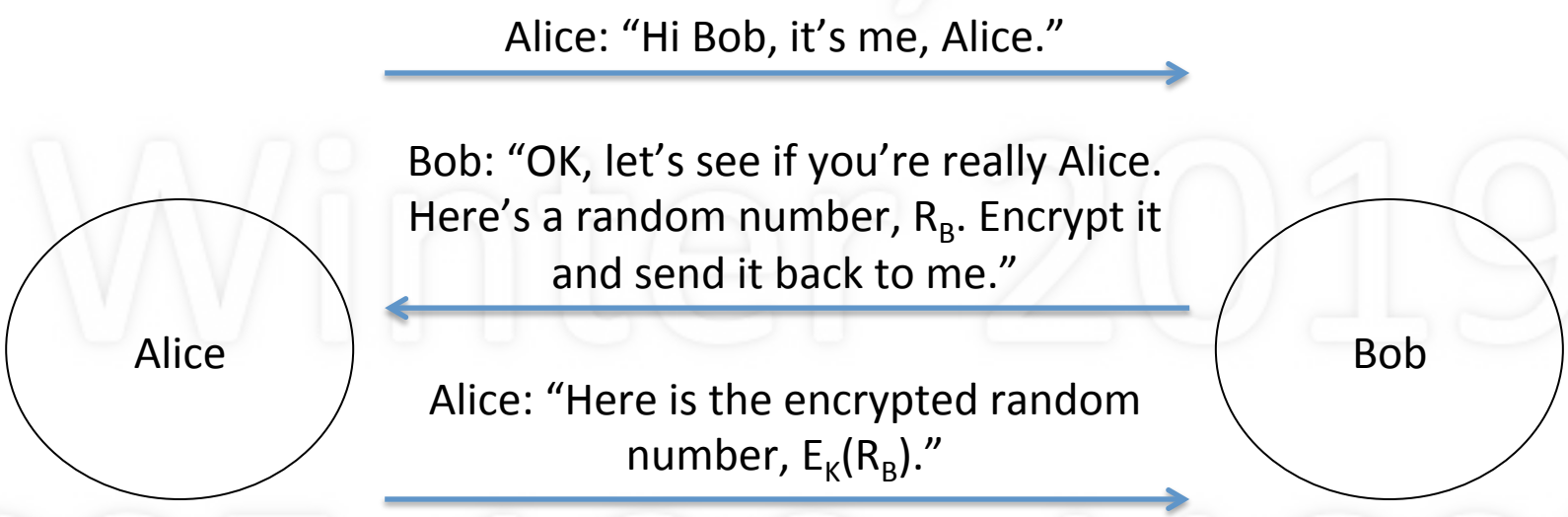
Public Key vs. Secret Key

- Secret key
  - Operates fast
  - Difficult to distribute keys
- Public key
  - Time-consuming operation
  - Convenient for key distribution
- Combination
  - Public key to start session: exchange secret key
  - During session, use secret key

Authentication Using Secret Key

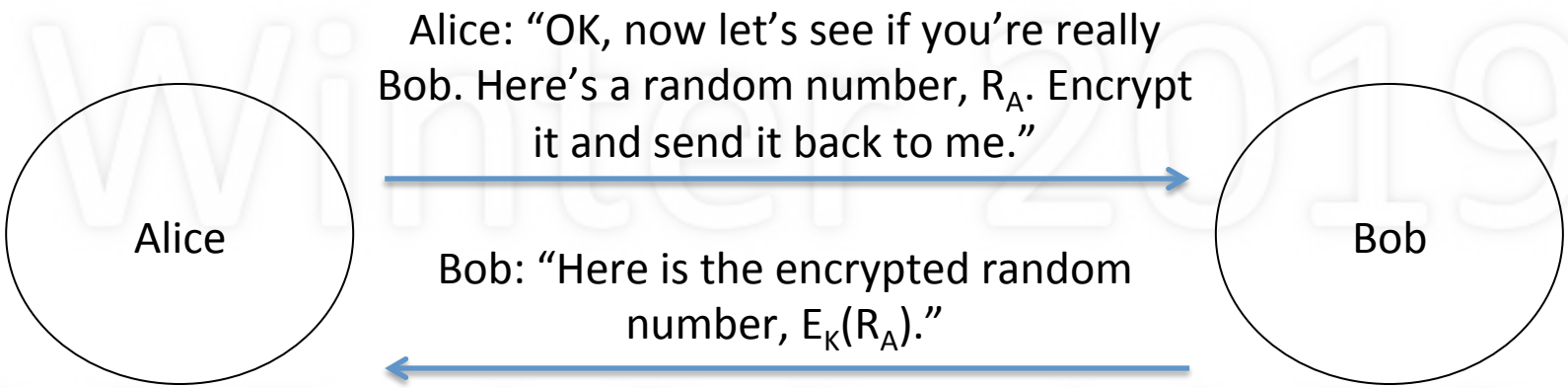
- Alice sends Bob identity: "Hi Bob, it's me, Alice"
- Bob sends challenge to Alice: random number R_B
- Alice encrypts R_B using K: $E_K(R_B)$; sends to Bob
- Bob decrypts $E_K(R_B)$ using K; must be Alice
- Alice sends challenge to Bob: random number R_A
- Bob encrypts R_A using K: $E_K(R_A)$; sends to Alice
- Alice decrypts $E_K(R_A)$ using K; must be Bob

Bob Authenticates Alice



Bob decrypts $E_K(R_B)$ and gets R_B. Since this is what he sent to Alice, and only he and Alice know the key K, only Alice could have sent R_B. And since R_B was never and will never be sent again, it could not be a replay of a previous message. Bob now knows he’s talking to Alice. But Alice doesn’t know if she’s really been talking to Bob.

Similarly, Alice Authenticates Bob

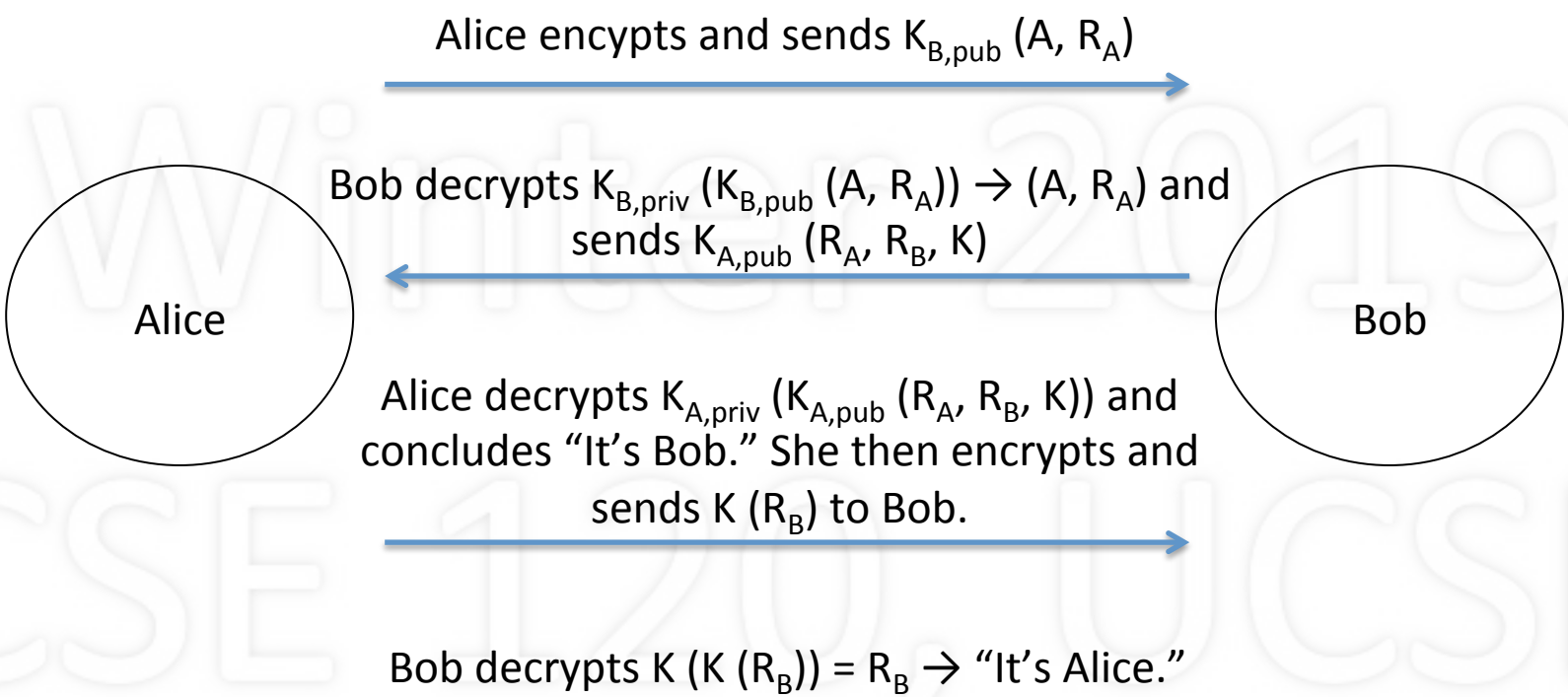


Alice decrypts $E_K(R_A)$ and gets R_A. Since this is what she sent to Bob, and only she and Bob know the key K, only Bob could have sent R_A. And since R_A was never and will never be sent again, it could not be a replay of a previous message. Alice now knows she's been talking to Bob.

Authentication Using Public Key

- Alice
  - Sends $K_{B,pub}$ (A, R_A) to Bob (uses Bob's public key)
- Bob
  - Decrypts: $K_{B,priv}$ ($K_{B,pub}$ (A, R_A)) → (A, R_A)
  - Encrypts and sends $K_{A,pub}$ (R_A, R_B, K) to Alice
- Alice
  - Decrypts: $K_{A,priv}$ ($K_{A,pub}$ (R_A, R_B, K)) → "it's Bob"
  - Encrypts and sends K (R_B) to Bob
- Bob
  - Decrypts K (K (R_B)) = R_B → "it's Alice"

Authentication Using Public Key

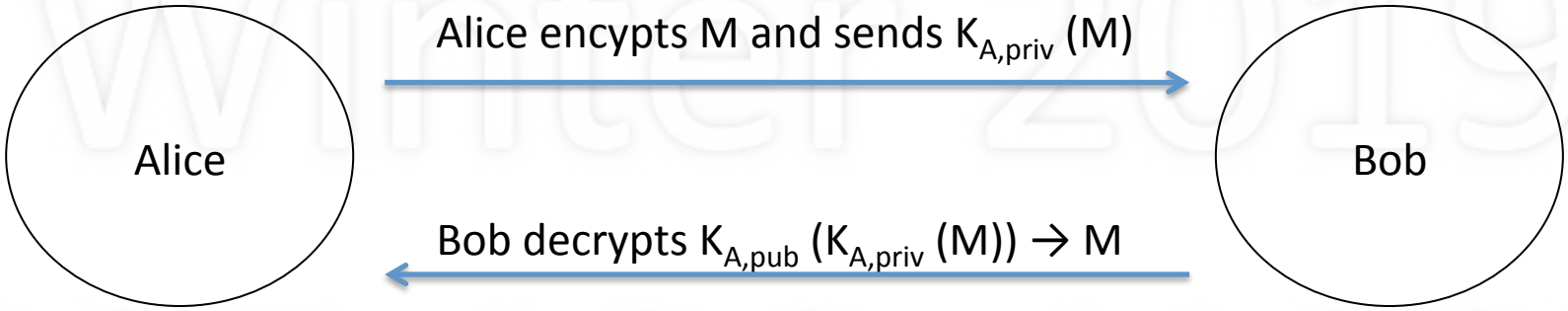


Bob decrypts K (K (R_B)) = R_B → "It's Alice."

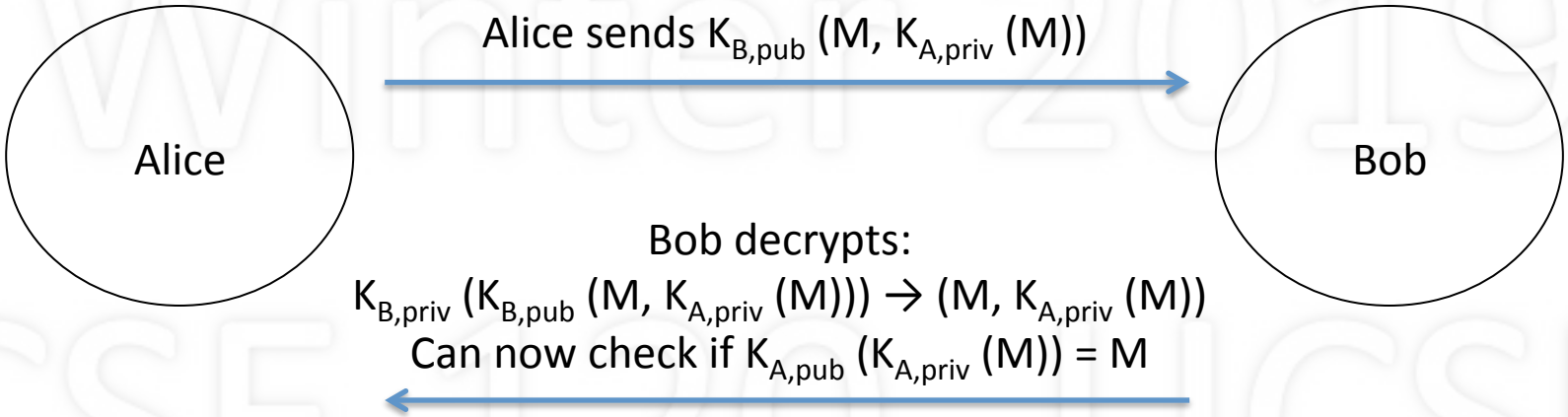
Digital Signatures

- If Alice wants to digitally sign message to Bob
  - Encrypt M using $K_{A,priv}$ and send $K_{A,priv}$ (M) to Bob
- When Bob receives, decrypts using $K_{A,pub}$
  - Can decrypt only if from Alice
- To sign and keep private
  - Alice sends $K_{B,pub}$ (M, $K_{A,priv}$ (M)) to Bob
  - Only Bob can decrypt: $K_{B,priv}$ ($K_{B,pub}$ (M, $K_{A,priv}$ (M)))
  - Decrypts using $K_{A,pub}$ proving Alice signed it

Digital Signatures



Digital Signatures



Summary

- Security
- Authentication
  - Passwords, Challenge-Response
- Attacks
  - Trojan Horse, Trap Door, Virus, Internet Worm, Denial-of-Service, Intrusion Detection
- Cryptography
  - Private vs. Public

Textbook

- Chapter 16 (on Security)
  - Lecture-related: 16.1-16.2, 16.4-16.6, 16.8
  - Recommended: 16.3, 16.7

Review & Research

- What is computer security?
- What distinguishes (what is different about) the three categories of what must be protected?**
- What distinguishes the three categories of threats?**
- Do all threats apply to all categories of what must be protected, why or why not?***

R&R

- What are the various aspects of security?
- For each aspect, can you provide an example? *
- What are the various security threats?
- For each threat, can you provide an example? *
- How do the threats on slide 4 correspond to the threats on slide 2?**

R&R

- What is user authentication?*
- What is key idea behind passwords?
- Why should passwords be encrypted?*
- By making a password one character longer, how much more time is needed to guess it on a modern computer?**
- What is the key user problem with passwords?

R&R

- What is a challenge/response protocol?*
- How does a challenge/response protocol differ from using passwords?**
- What is the key idea behind a challenge/response protocol?*

R&R

- What is a Trojan Horse?*
- From an operating systems point of view, what aspect of system operation does a trojan horse exploit?**
- Can you think of a general approach in which a trojan horse can be prevented from doing hard (or limited in what it can do)?***

R&R

- What is a Trap Door?*
- From an operating systems point of view, what aspect of system operation does a trap door exploit?**
- In what way is a Trap Door different from a Trojan Horse (or are they the same)?**
- Why is it hard to detect trap door that are added by the compiler?***

R&R

- What is a virus?*
- From an operating systems point of view, what aspect of system operation does a virus exploit?**
- In what way is a virus different from a Trojan Horse or Trap Door?**
- What are approaches to detecting viruses?***

R&R

- In the Internet Worm attach of 1988, what was it, and how did it work?**
- What was the key vulnerability that the Internet Worm exploited?**
- What is an approach to guard against these style of attack?***

R&R

- What is Denial of Service?
- What is an approach to dealing with Denial of Service attacks?***
- What is Intrusion Detection?
- What's the difference between signature-based and anomaly-based detection?**
- What is the value of an audit log?**

R&R

- What is cryptography?
- What is a Secret Key encryption, and how does it work?*
- Why is DES considered “weak”?*
- How is AES different than DES, and why is it stronger?*

R&R

- What is a Public Key encryption, and how does it work?*
- Why does Public key encryption use two keys? **
- What are the differences between Secret Key and Public Key encryption?* What are their pros/cons?**
- How can they be used in combination?**

R&R

- What is the protocol for authentication using Secret Key?*
- What is the purpose of using random numbers in this authentication protocol?***
- What is the protocol for authentication using Public Key?*
- What is the purpose of using random numbers in this authentication protocol?***

R&R

- What is a “digital signature”?
- What type of encryption is used for implementing a digital signature, and how does the protocol work?**
- What key aspect of the protocol allows us to prove the validity of the digital signature?***
- How might a digital signature be invalidated? ***