Network Security

CSE 461

Ratul Mahajan

Where we are

- Security crosses all layers

Application
Transport
Network
Link
Physical
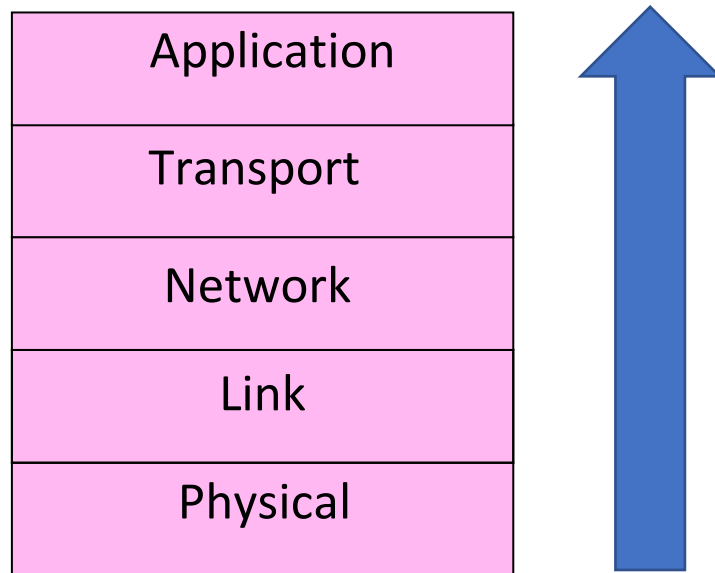
Security Threats

- "Security" is like "performance"
  - Means many things to many people
  - Must define the properties we want

- Key task is clearly stating the <u>threat model</u>
  - The dangers and attacker's abilities
 - Can't assess risk or solution effectiveness otherwise

Security Threats (2)

- Some example threats
  - It's not all about encrypting messages

Attacker	Ability	Threat
Eavesdropper	Intercept messages	Read contents of message
Observer	Inspect packet destinations	Collect conversations
Intruder	Compromised host	Tamper with contents of message
Impersonator	Remote social engineering	Trick party into giving information
Extortionist	Remote / botnet	Disrupt network services

Risk Management

- Security is hard as a negative goal
  - Try to ensure security and don't let anything bad happen!

- Only as secure as the weakest link
  - Could be design flaw or bug in code
  - But often the weak link is elsewhere …
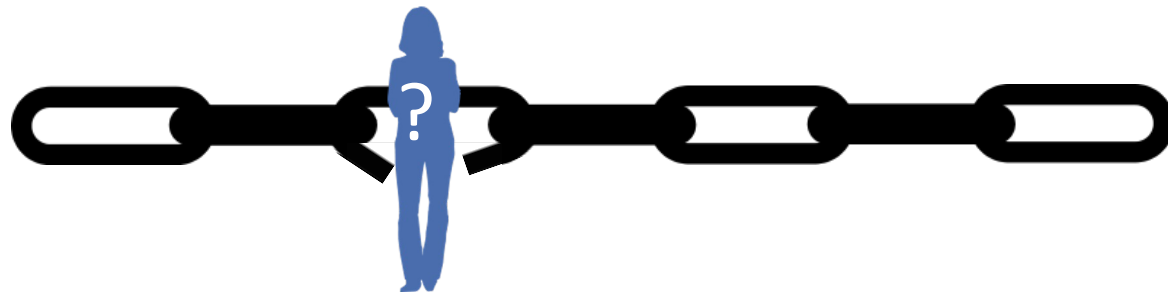
Risk Management (2)

802.11 security … early on, WEP:

- Cryptography was flawed; can run cracking software to read WiFi traffic

Today, WPA2/802.11i security:

- Computationally infeasible to break!

So that means 802.11 is "secure"?

Risk Management (3)

802.11 is more secure against eavesdropping but is not "secure". Many possible threats; we made just the first one harder!

Threat Model	Old WiFi (WEP)	New WiFi (WPA2)
Break encryption from outside	Very easy	Very difficult
Guess WiFi password	Often possible	Often possible
Get password from computer	May be possible	May be possible
Physically break into home	Difficult	Difficult

Cryptography

Cryptology

Rich history, especially spies / military

- From the Greek "hidden writing"
- Modern emphasis is on codes that are "computationally infeasible" to break

Cryptography: Focus is encrypting information

Cryptanalysis: Focus is how to break codes

Uses of Cryptography

Encrypting information is useful for more than deterring eavesdroppers (confidentiality)

- Prove message came from real sender (authentication)
- Prove remote party is who they say
- Prove message hasn't been altered (integrity)

Designing secure cryptographic scheme tricky!

- Use approved design (library) in approved way
- And even then ….. OpenSSL in 2014 (Heartbleed)

Internet Reality

Most protocols were developed before the Internet was mainstream
- It was a smaller, more trusted world
- So protocols lacked security …

We have strong security needs today
- Clients talk with unverified servers
- Servers talk with anonymous clients
- Security has been retrofitted
- This is far from ideal!

Confidentiality: Goal, Threat Model

- Goal: Send a private message from Alice to Bob

- Threat: Eve will read the message
  - Eve is a passive adversary (observer)
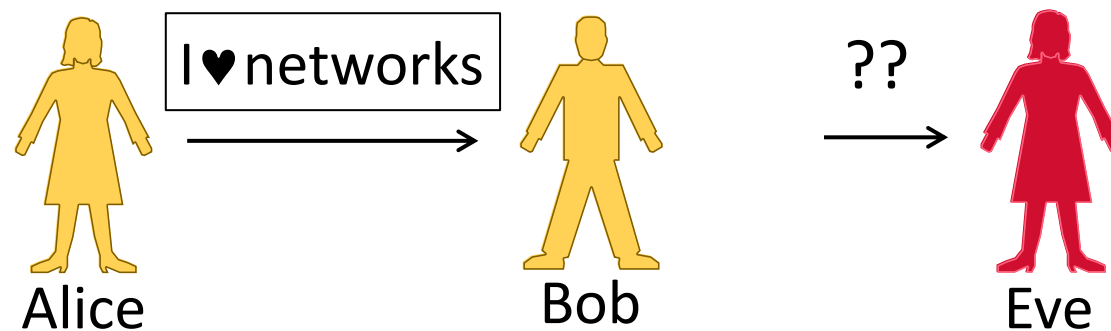
Encryption/Decryption Model

- Alice encrypts private message (<u>plaintext</u>) using key
- Eve sees <u>ciphertext</u> but not plaintext
- Bob decrypts using key to get the private message
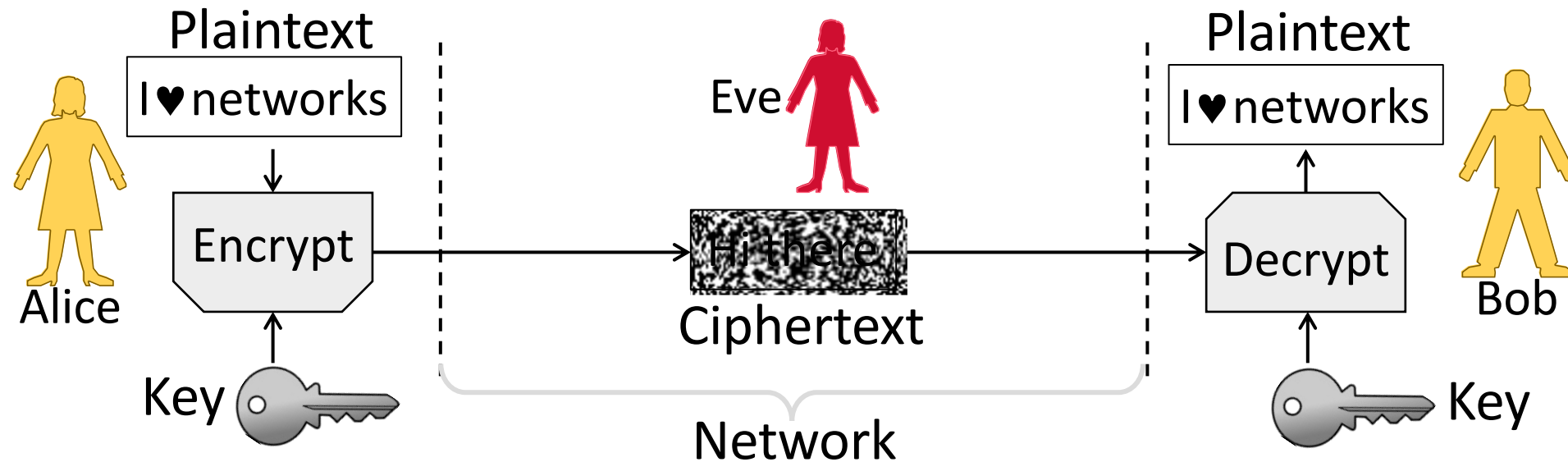
Encryption/Decryption (2)

- Encryption is a reversible mapping
  - Ciphertext is encrypted plaintext
- Assume attacker knows algorithm
  - Security does not rely on its secrecy
- Algorithm is parameterized by keys
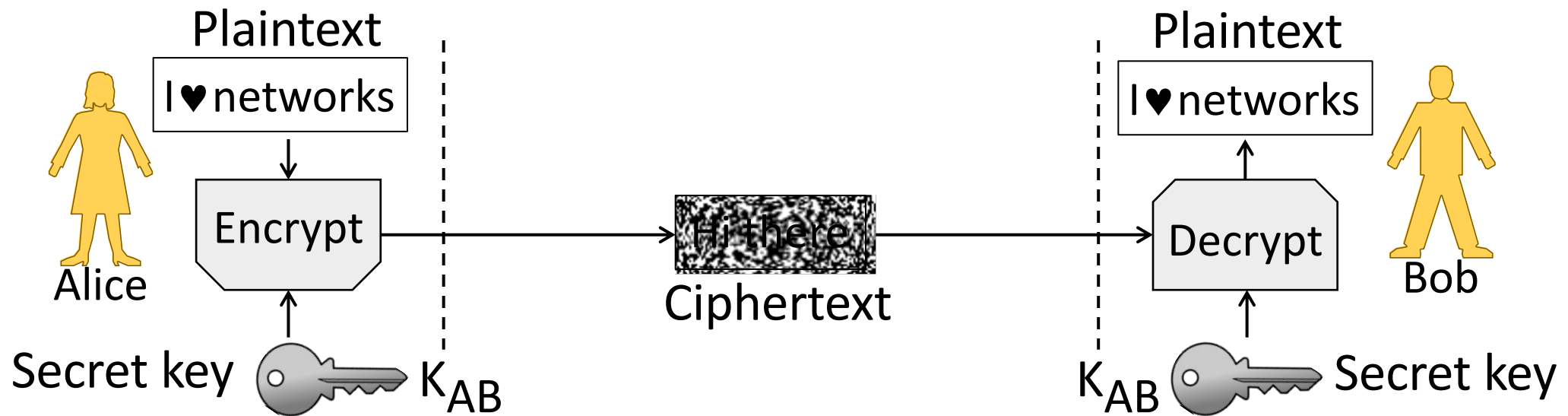 - Security does rely on key secrecy
  - Must be distributed (Achilles' heel)

Encryption/Decryption (3)

Two main kinds of encryption:

1. Symmetric key encryption », e.g., AES
   - Alice and Bob share secret key
   - Encryption is a bit mangling box

2. Public key encryption », e.g., RSA
   - Alice and Bob each have a key in two parts: a public part (widely known), and a private part (only owner knows)
   - Encryption is based on mathematics (e.g., RSA is based on difficulty of factoring)
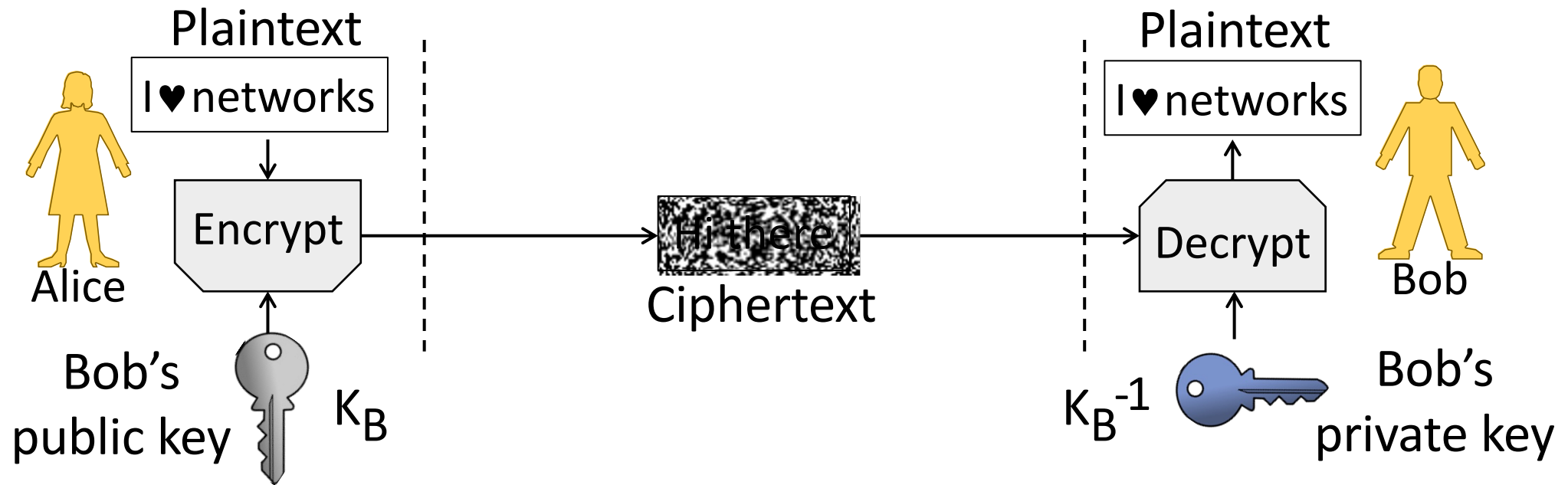
Symmetric (Secret Key) Encryption

- Alice and Bob have the same secret key, K_{AB}
  - Anyone with the secret key can encrypt/decrypt

Plaintext

I ♥ networks

Alice

Encrypt

Secret key K_{AB}

Hi there

Ciphertext

Plaintext
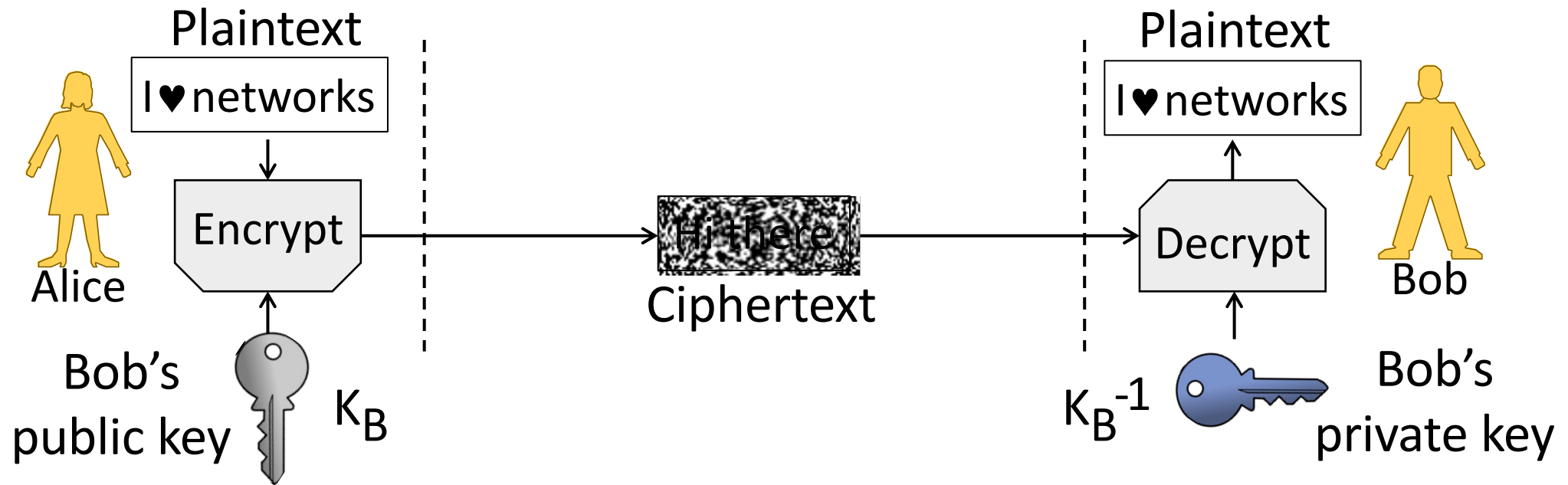
I ♥ networks

Bob

Decrypt

K_{AB} Secret key

Public Key (Asymmetric) Encryption

- Alice and Bob have public/private key pairs (K_B / K_B^{-1})
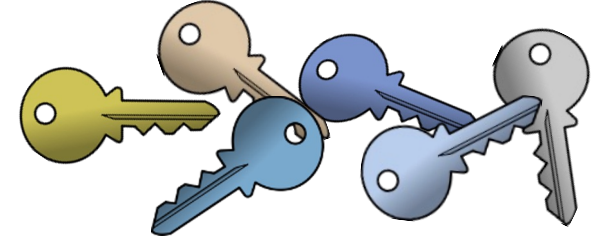 - Public keys are well-known, private keys are secret

Public Key Encryption (2)

- Alice encrypts w/ Bob's pubkey K_B; anyone can send
- Bob decrypts w/ his private key K_B^{-1}; only he can

Key Distribution

- This is a big problem on a network!
  - Often want to talk to new parties
- Symmetric encryption problematic
  - Have to first set up shared secret
- Public key idea has own difficulties
  - Need trusted directory service
  - We'll look at <u>certificates</u> later

Symmetric vs. Public Key

- Have complementary properties
  - Want the best of both!

| Property	Symmetric	Public Key
Key Distribution	Hard – share secret per pair of users	Easier – publish public key per user
Runtime Performance	Fast – good for high data rate	Slow – few, small, messages

Winning Combination

- Alice uses public key encryption to send Bob a small private message
  - It's a key! (Say 256 bits.)
- Alice/Bob send messages with symmetric encryption
  - Using the key they now share
- The key is called a <u>session key</u>
  - Generated for short-term use

Message Authentication

Goal and Threat Model

- Goal is for Bob to verify the message is from Alice and unchanged
  - This is called integrity/authenticity
- Threat is Trudy will tamper with messages
  - Trudy is an active adversary (interferes)

| Alice | Trudy | Bob |
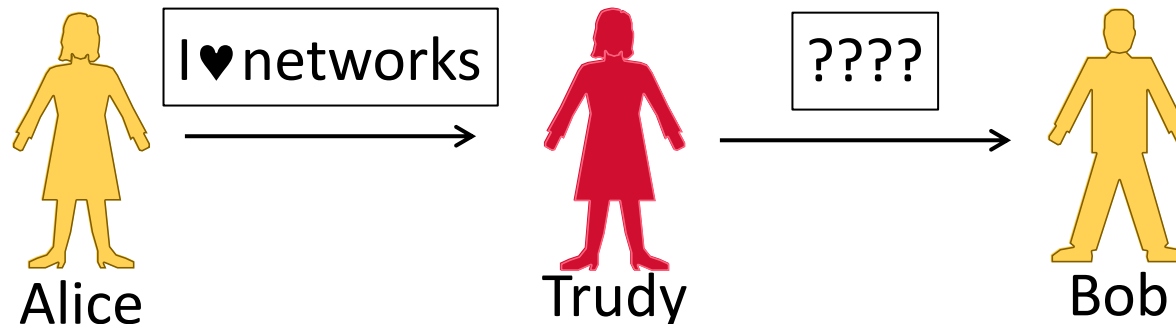
Wait a Minute!

- We're already encrypting messages to provide confidentiality

- Why isn't this enough?

Encryption Issues

- What will happen if Trudy flips some of Alice's message bits?
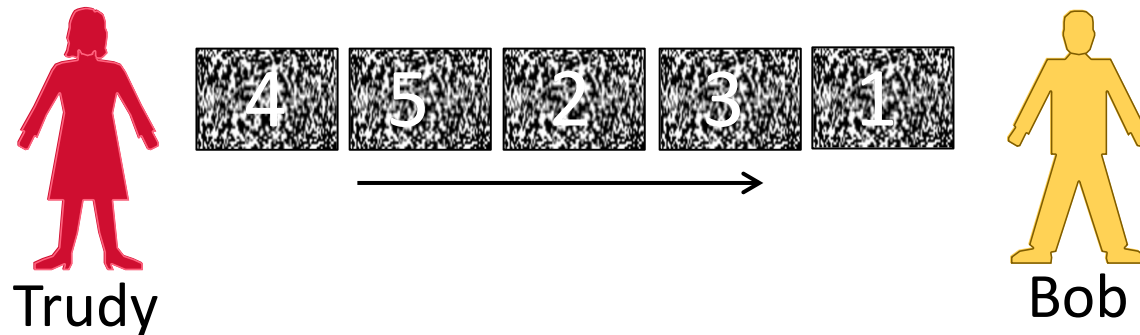 - Bob will decrypt it, and ...

Trudy

Bob
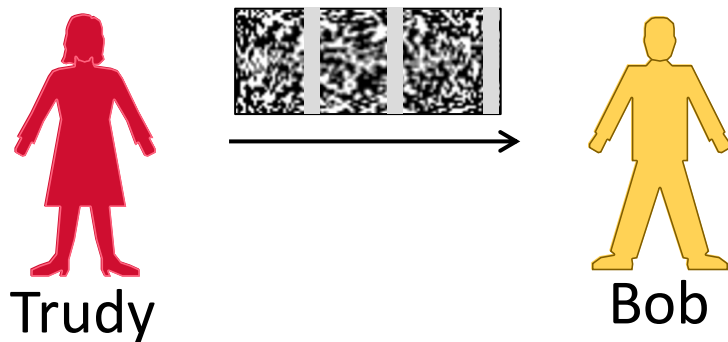
Encryption Issues (2)

- What will happen if Trudy flips some of Alice's message bits?
  - Bob will receive an altered message
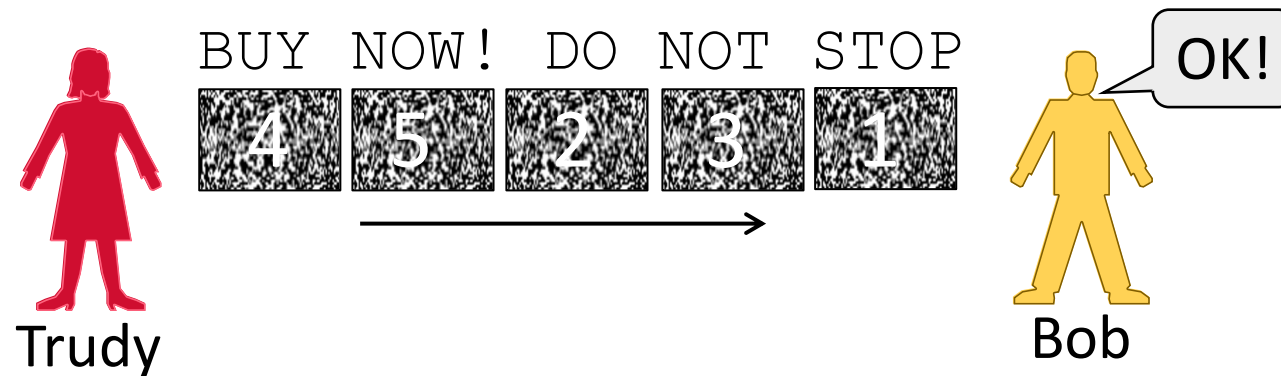
yuiE#E3@

Um??

Trudy

Bob
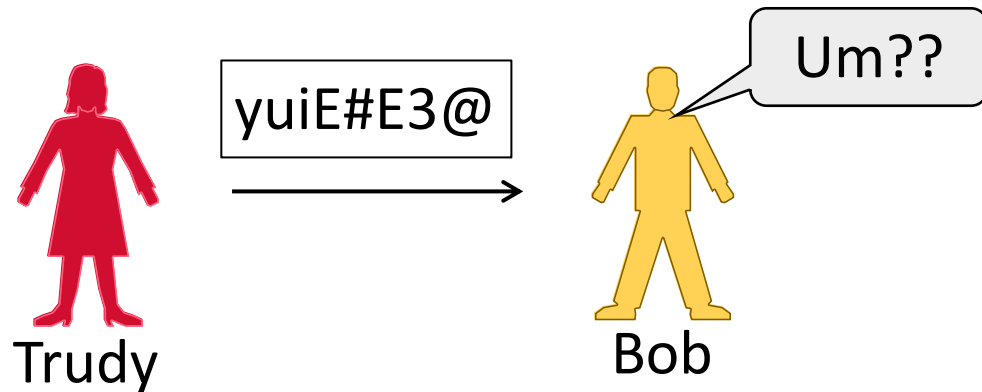
Encryption Issues (3)

- Typically encrypt blocks of data

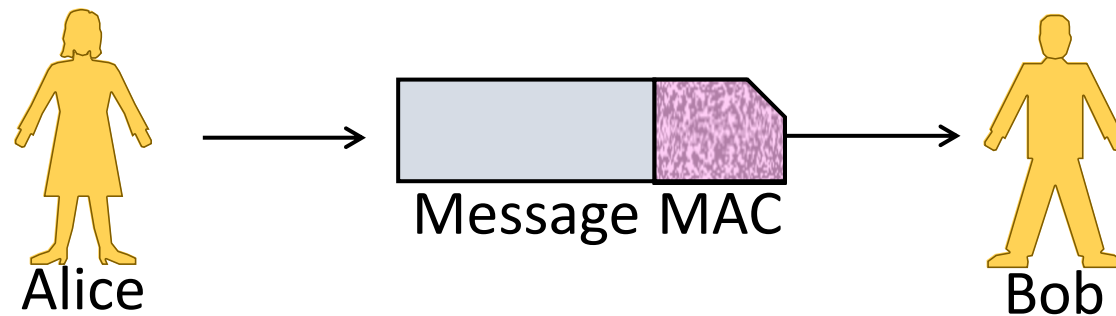- What if Trudy reorders message?
  - Bob will decrypt, and …

Encryption Issues (4)

- ## What if Trudy reorders message?
 - ### Bob will receive altered message

MAC (Message Authentication Code)

- MAC is a small token to validate the integrity/authenticity of a message
  - Send the MAC along with message
  - Validate MAC, process the message
  - Example: HMAC scheme
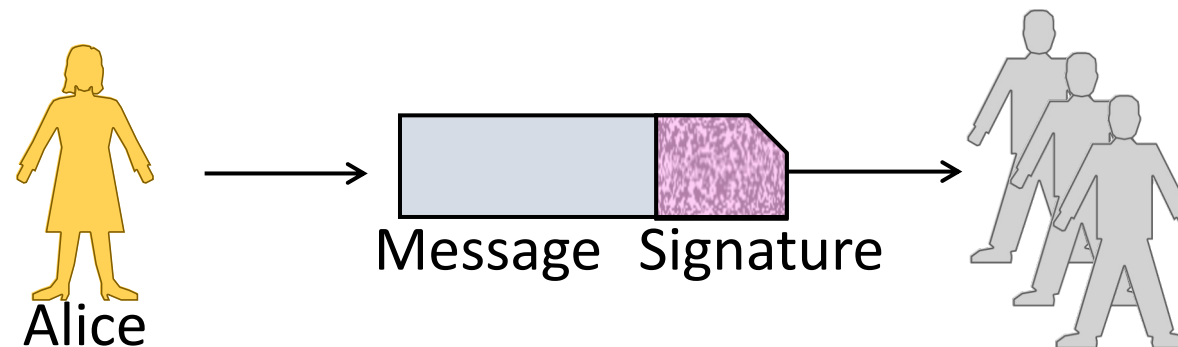
Alice

Message MAC

Bob

MAC (2)

- Sorta symmetric encryption operation – key shared
  - Lets Bob validate unaltered message came from Alice
 - Doesn't let Bob convince Charlie that Alice sent the message
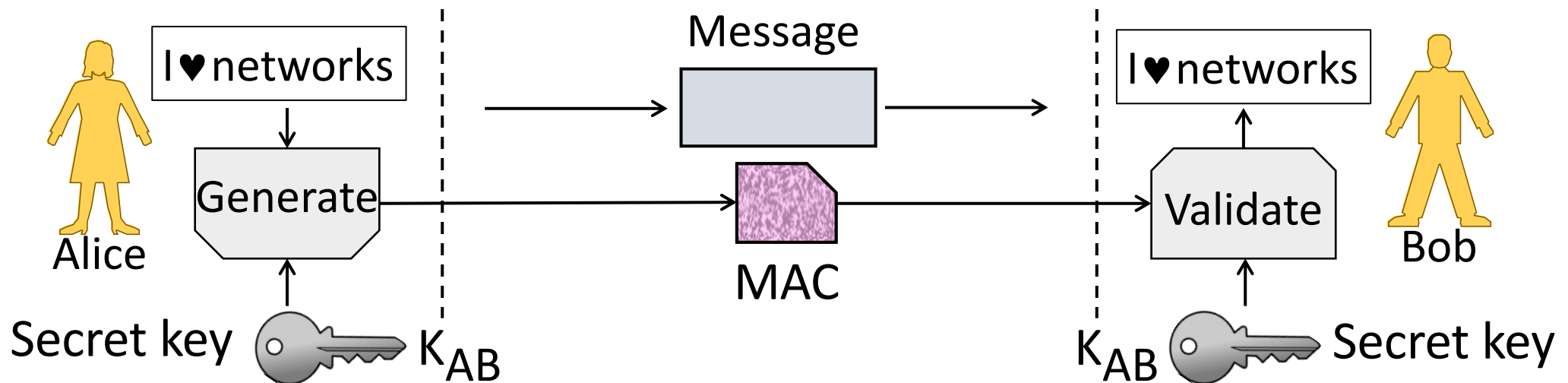
Digital Signature

- Signature validates the integrity/authenticity of message
  - Send it along with the message
  - Lets all parties validate
  - Example: RSA signatures

Alice

Message   Signature

Digital Signature (2)

- Kind of public key operation – pub/priv key parts
  - Alice signs w/ private key, K_A^{-1}, Bob verifies w/ public key, K_A
 - Does let Bob convince Charlie that Alice sent the message
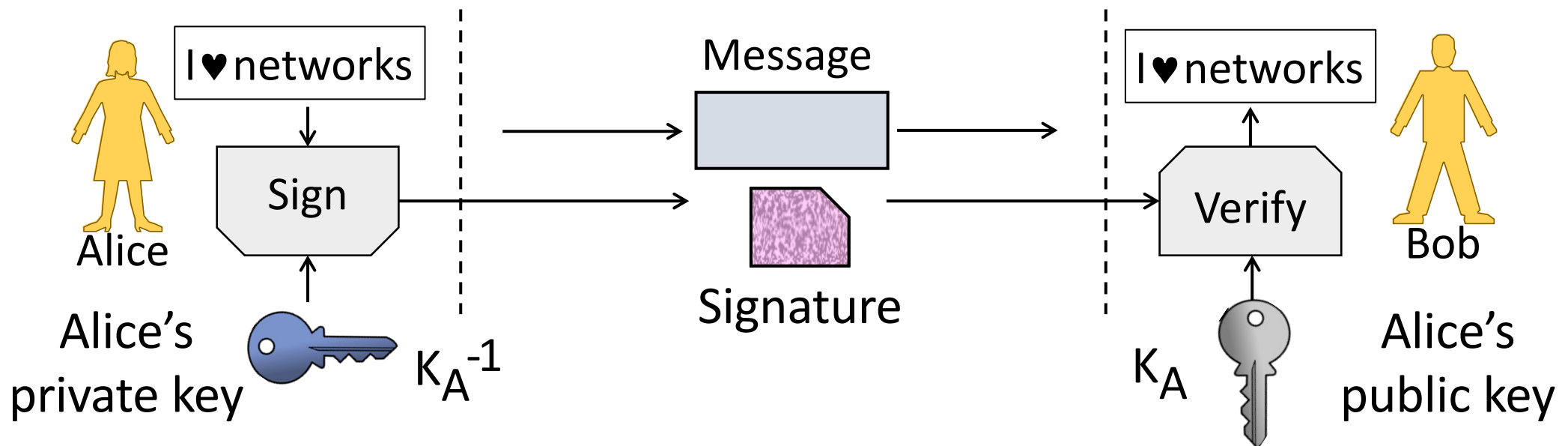
Speeding up Signatures

- Same tension as for confidentiality:
  - Public key has keying advantages
  - But it has slow performance!
- Use a technique to speed it up
  - <u>Message digest</u> stands for message
 - Sign the digest instead of full message

Message Digest or Cryptographic Hash

- Digest/Hash is a secure checksum
  - Deterministically mangles bits to pseudo-random output (like CRC)
  - Can't find messages with same hash
  - Acts as a fixed-length descriptor of message – very useful!

I might be a tiny bit sick of networks… Input

e.g., SHA1
(160 bits)   Hash function   →   Output
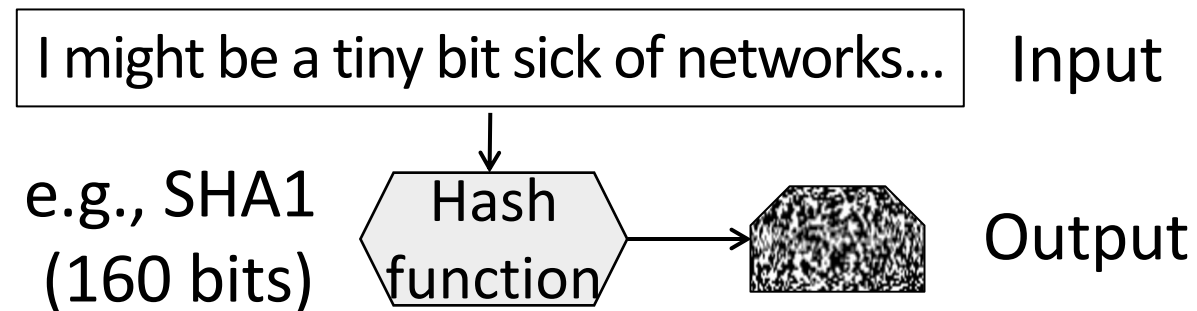
Speeding up Signatures (2)

- Conceptually similar except sign the hash of message
  - Hash is fast to compute, so it speeds up overall operation
 - Hash stands for msg as can't find another w/ same hash
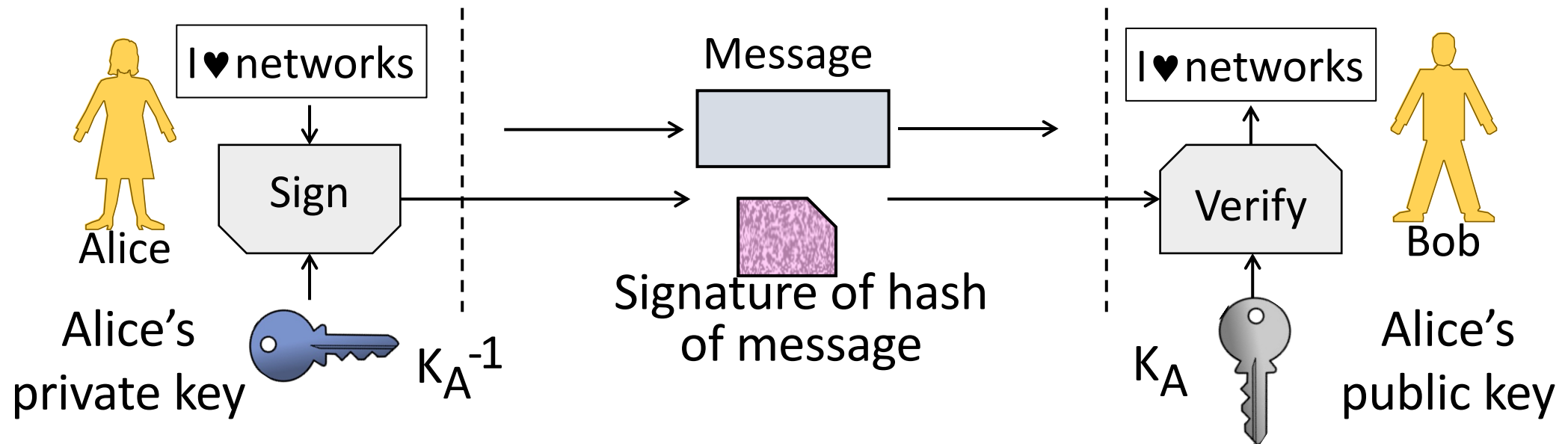
Preventing Replays

- We normally want more than confidentiality, integrity, and authenticity for secure messages!
  - Want to be sure message is fresh
- Need to distinguish message from <u>replays</u>
  - Repeat of older message
 - Acting on it again may cause trouble

Preventing Replays (2)

- Replay attack:
  - Trudy records Alice's messages to Bob
  - Trudy later replays them (unread) to Bob
    - She pretends to be Alice
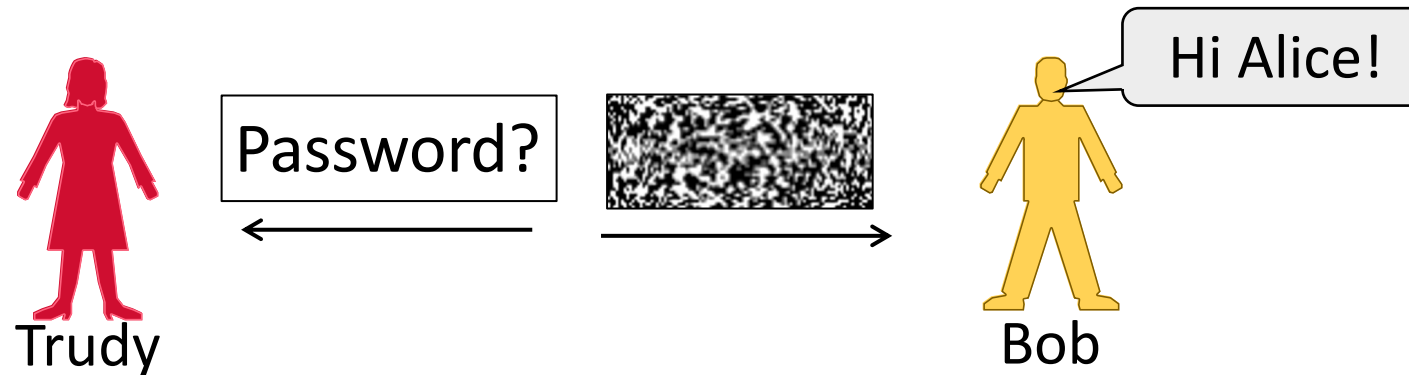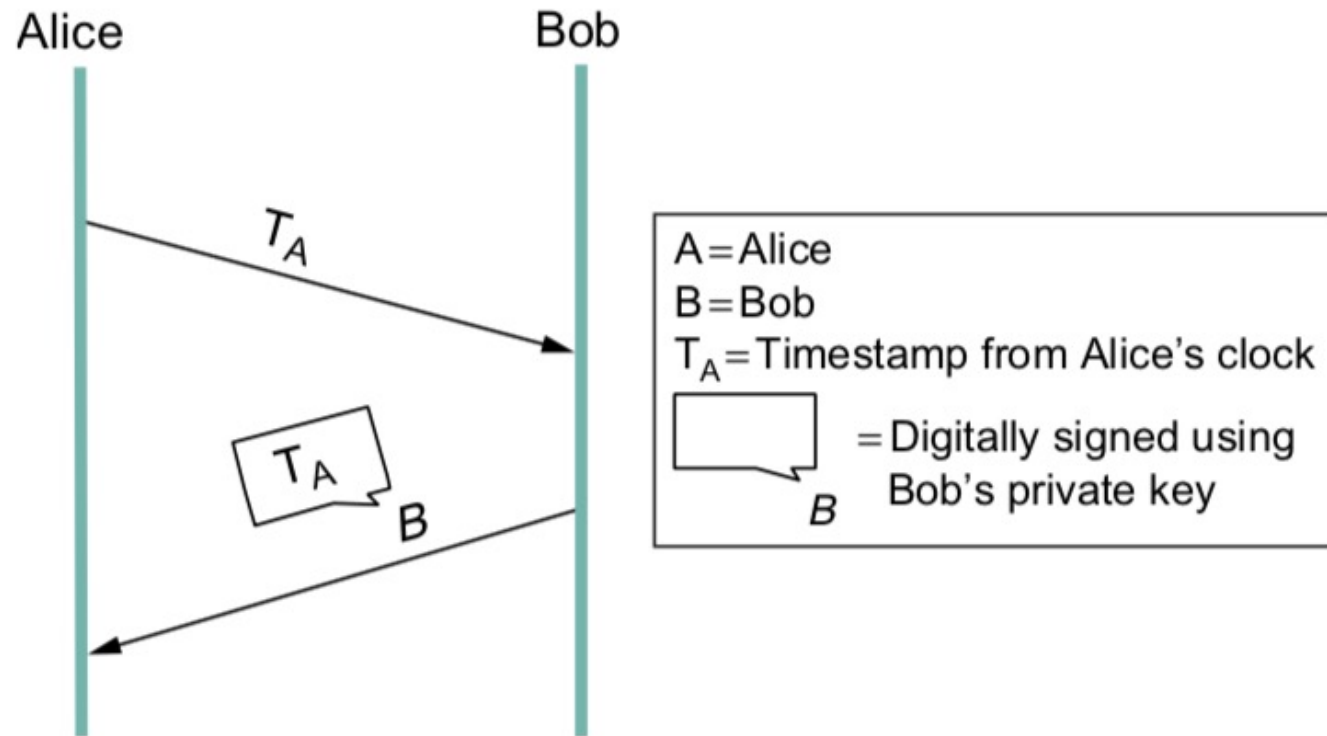
Preventing Replays (3)

- To prevent replays, include a proof of freshness in the messages
  - Use a timestamp, or <u>nonce</u>

Freshness

Tue 10:03:57: "sell stocks"

Alice

Message MAC

Confidentiality Authenticity/Integrity

OK Alice!

Bob
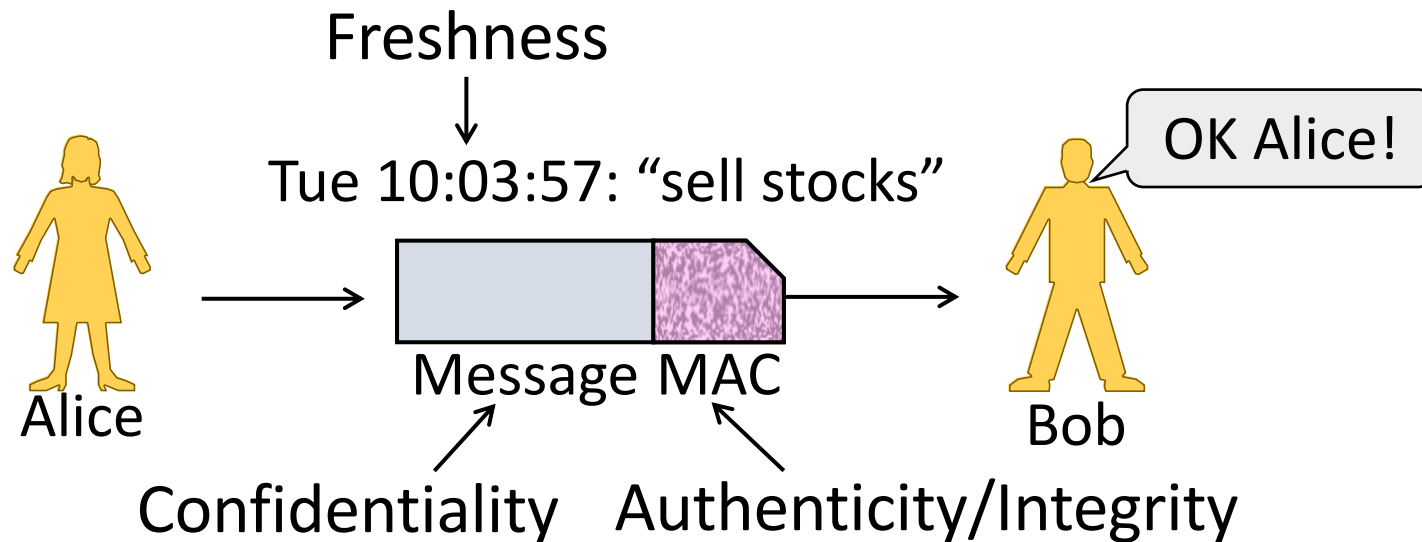
Using Timestamps

Takeaway

- Cryptographic designs can give us integrity, authenticity and freshness as well as confidentiality.
- Real protocol designs combine the properties in different ways
  - We'll see some examples
  - Note many pitfalls in how to combine, as well as in the primitives themselves