CS 114: Network Security

Spring 2023 - Wrap Up

Prof. Daniel Votipka

Tufts University

In today's lecture …

- Course Review
  - Part 1: Crypto
  - Part 2: Network protocols and how to secure them
  - **Part 3: Network defense**
  - **Part 4: Web security**
  - **Part 5: Human factors in security**
- Course evaluation

https://www.cs.tufts.edu/comp/114/2023S/gradecalculator.html

Part 1
Cryptobabble

Kerckhoffs' Principles

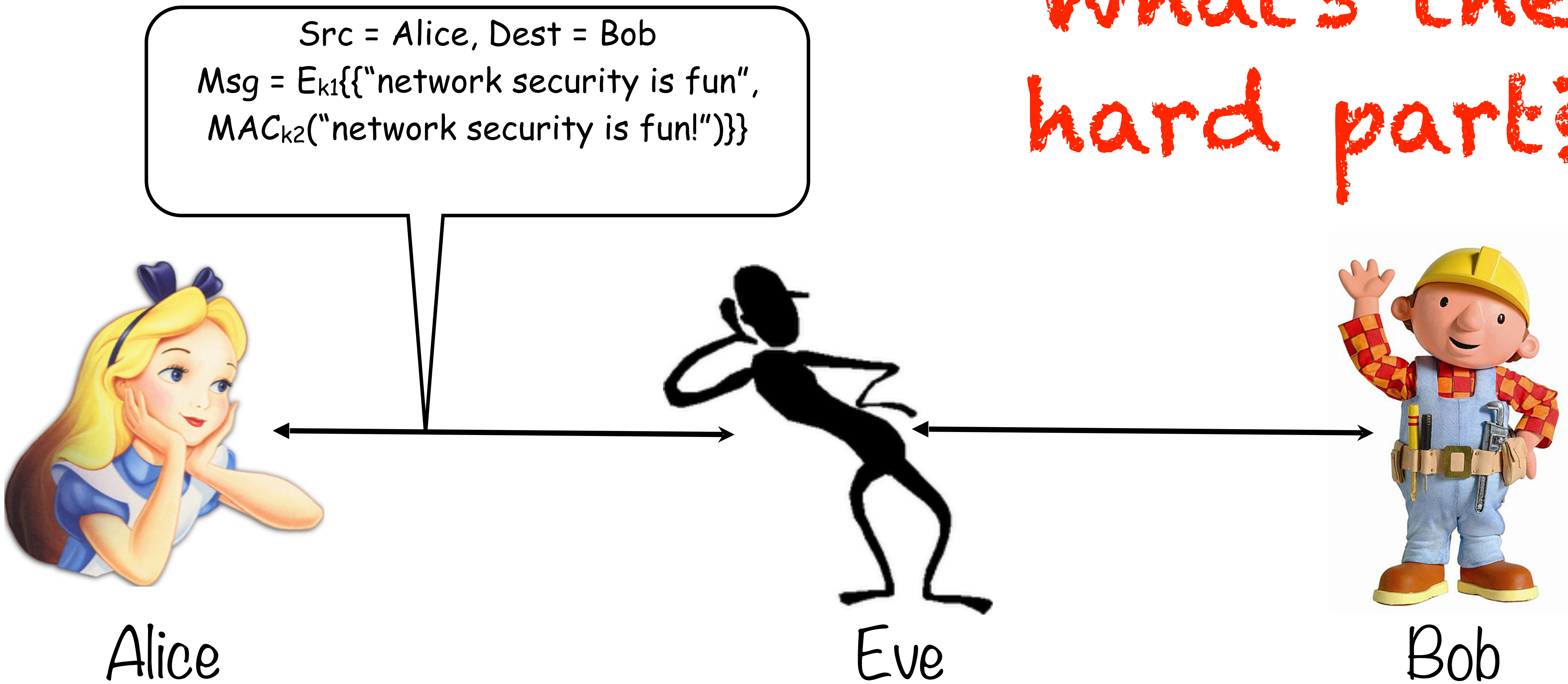
- Modern cryptosystems use a key to control encryption and decryption
- Ciphertext should be undecipherable without the correct key
- Encryption key may be different from decryption key.
- Kerckhoffs' principles [1883]:
  - Assume Eve knows cipher algorithm
  - Security should rely on choice of key
  - If Eve discovers the key, a new key can be chosen

Crypto: Secret Key

- **C**onfidentiality
  - Block ciphers & stream ciphers
    - avoiding misuse of stream ciphers
    - block cipher modes
- **I**ntegrity
- **A**uthenticity
  - Hashes
  - Message Authentication Codes (MACs)

Encryption and Message Authenticity

What's the hard part?

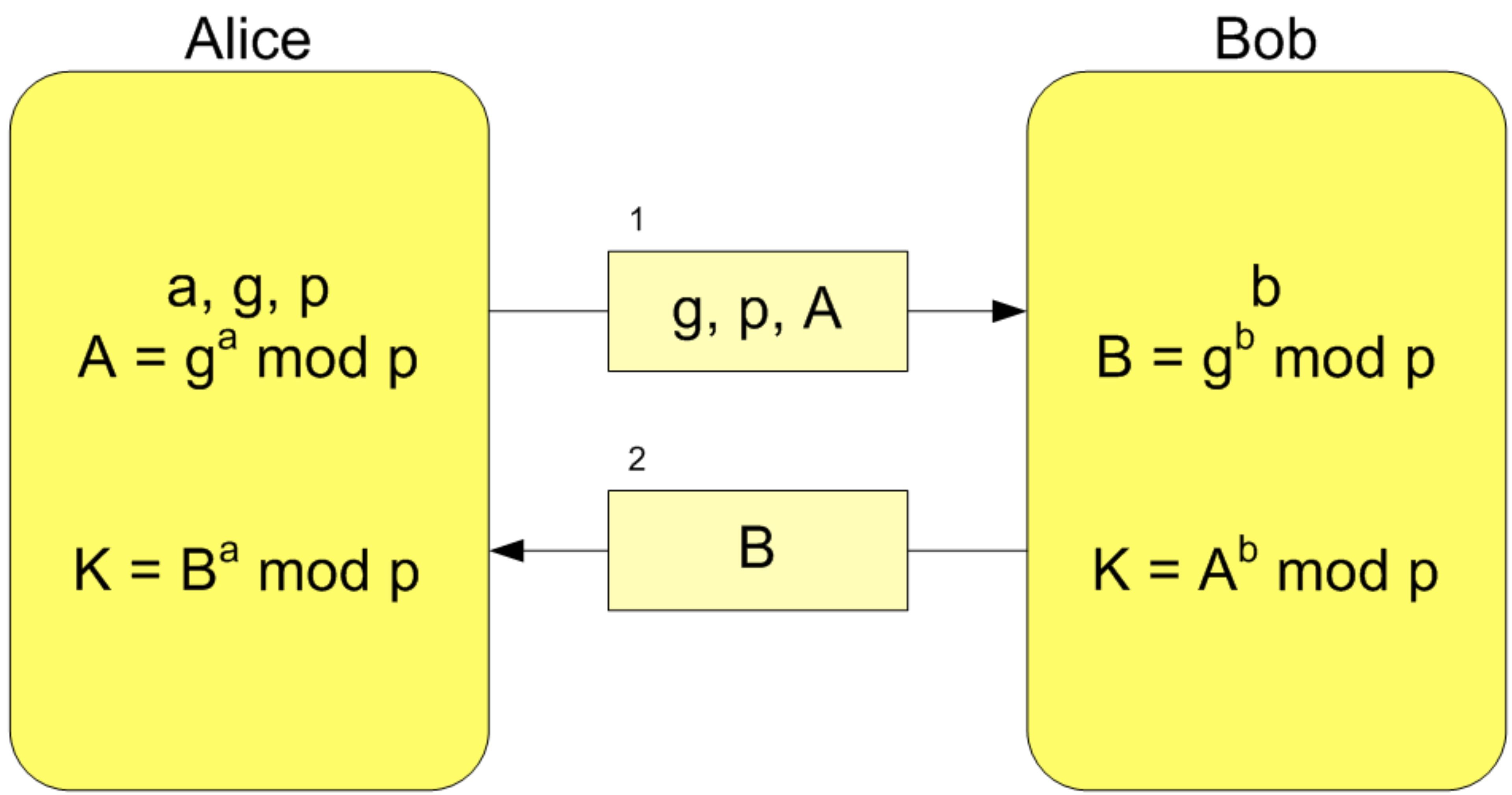


Without knowing *k1*, Eve can’t read Alice’s message.

Without knowing *k2*, Eve can’t compute a valid MAC for her forged message.

Diffie-Hellman (DH) Key Agreement

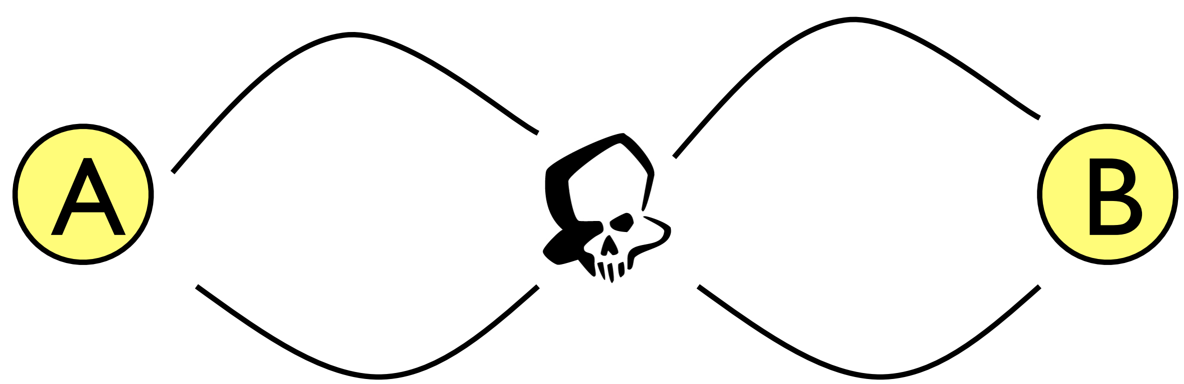
- Proposed by Whitfield Diffie and Martin Hellman in 1976
- g=base, p=prime, a=Alice's secret, b=Bob's secret
- Eve cannot compute K without knowing either a or b (neither of which is transmitted), even if she (passively) intercepts all communication!



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

Attacks on Diffie-Hellman

- Subject to **Man-in-the-Middle** (MitM) attack
  - You really don't know anything about who you have exchanged keys with

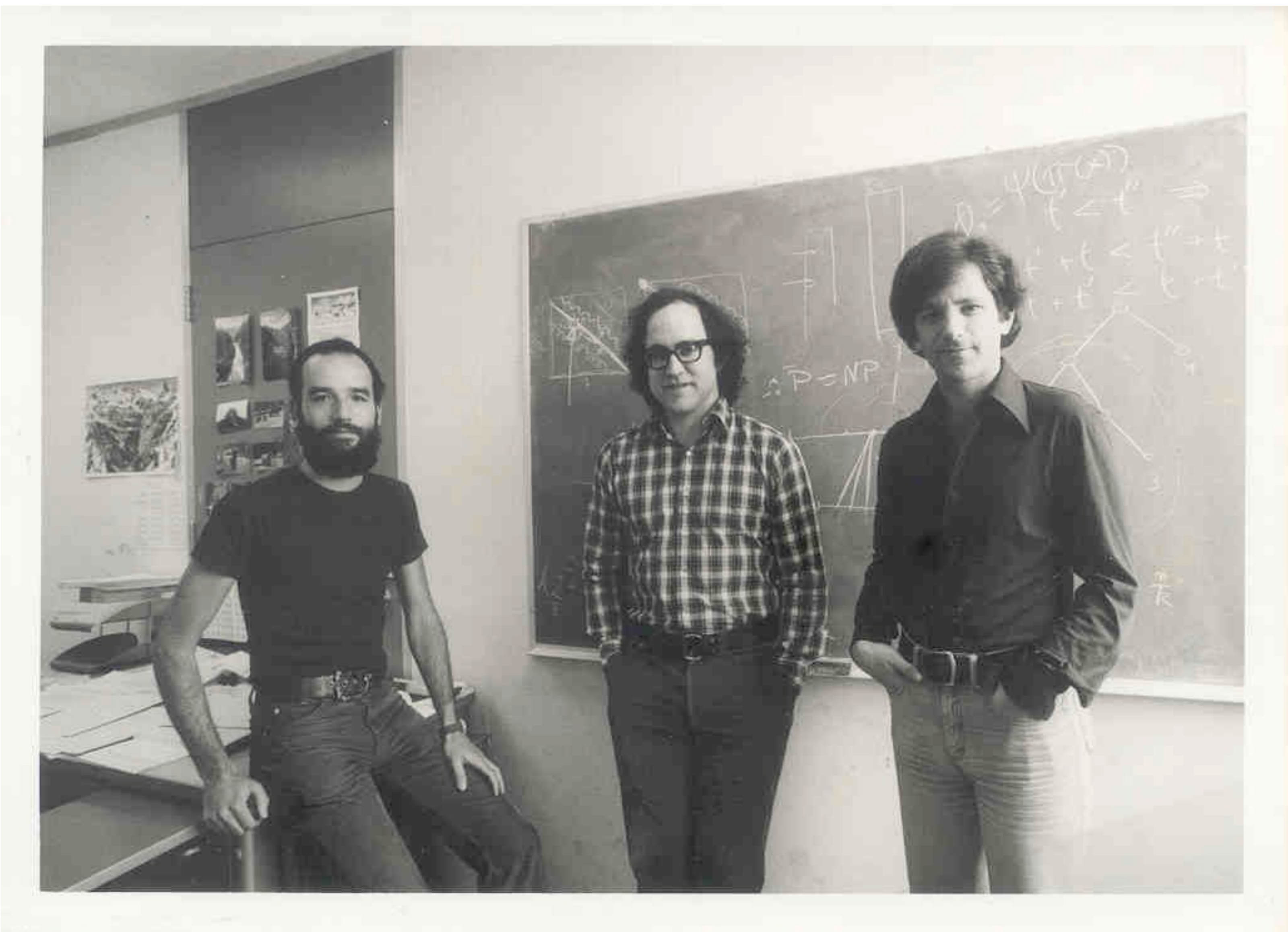


  - Alice and Bob think they are talking directly to each other, but Mallory is actually performing two separate exchanges

Public Key Crypto
(10,000 ft view)

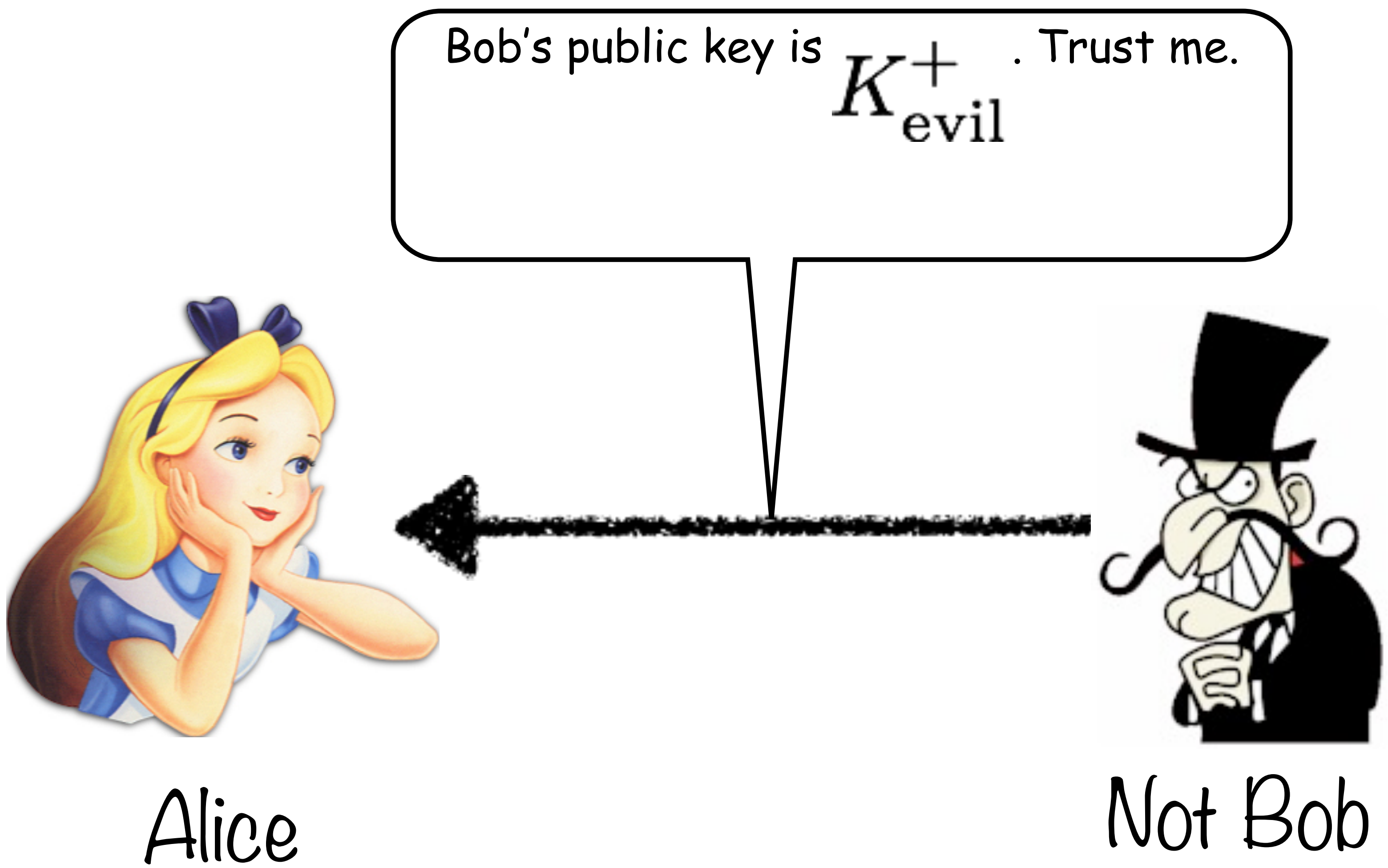
- **Separate** keys for encryption and decryption
  - Public key: anyone can know this
  - Private key: kept confidential
- Anyone can encrypt a message to you using your public key
- The private key (kept confidential) is required to decrypt the communication
- Alice and Bob no longer have to have *a priori* shared a secret key
- Relies on **trapdoor functions**: functions that are easy to compute in one direction but very difficult to reverse without knowing a key (the "trapdoor")

RSA
(Rivest, Shamir, Adelman)



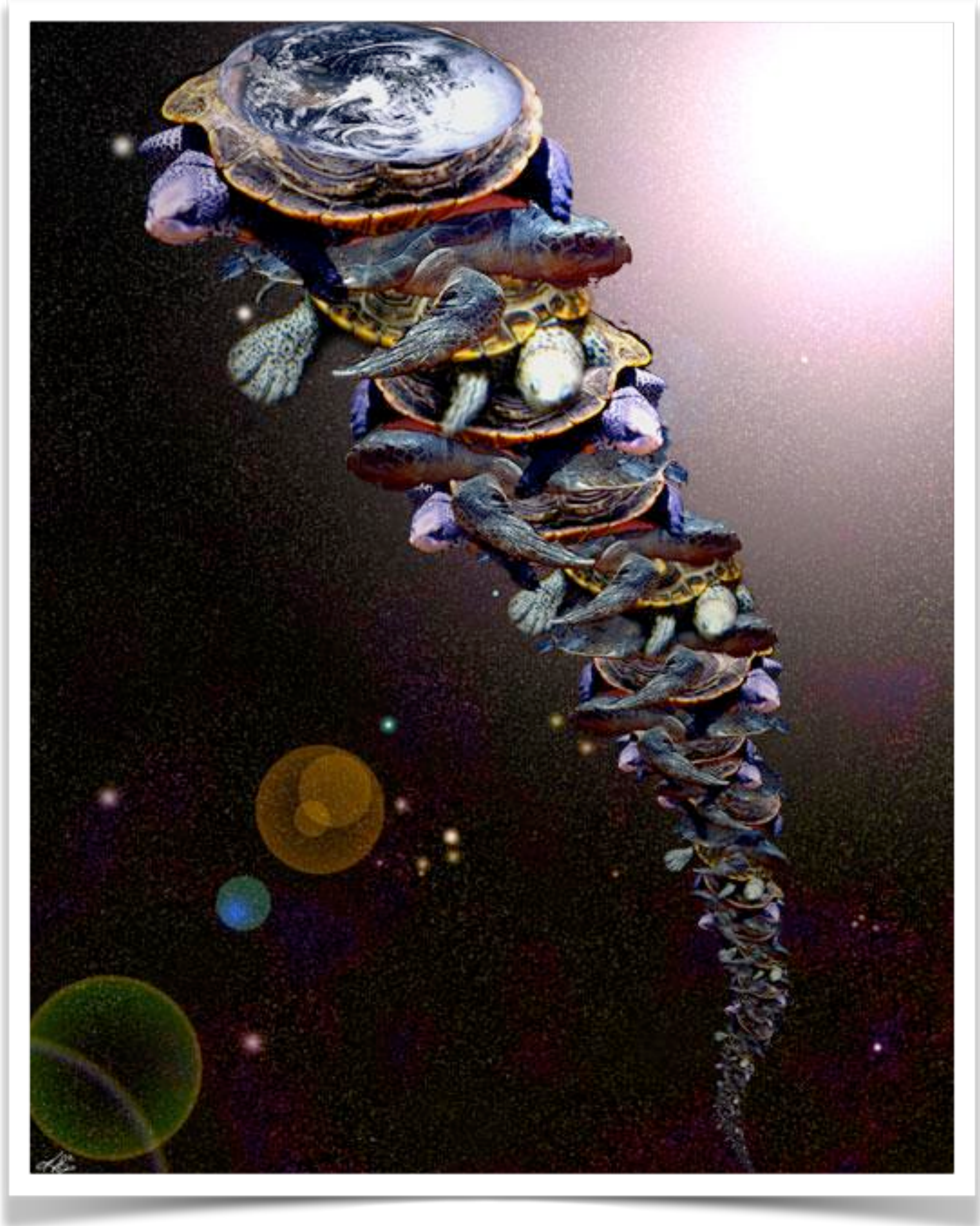
"A method for obtaining Digital Signatures and Public Key Cryptosystems“, Communications of the ACM, Feb. 1978.

But how do we *verify* we're using the correct public key?



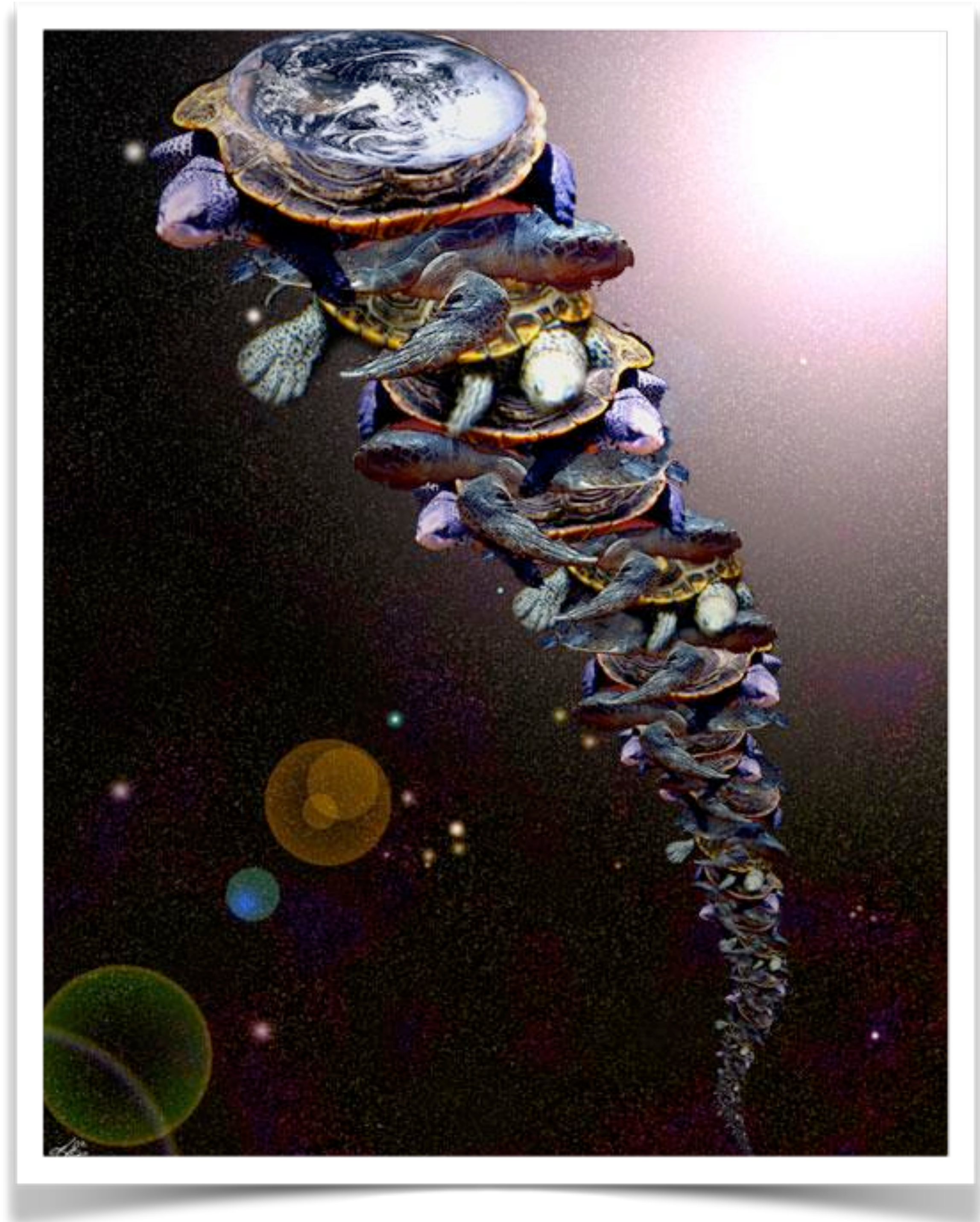
Short answer:
We can't.

It's turtles all the way down.



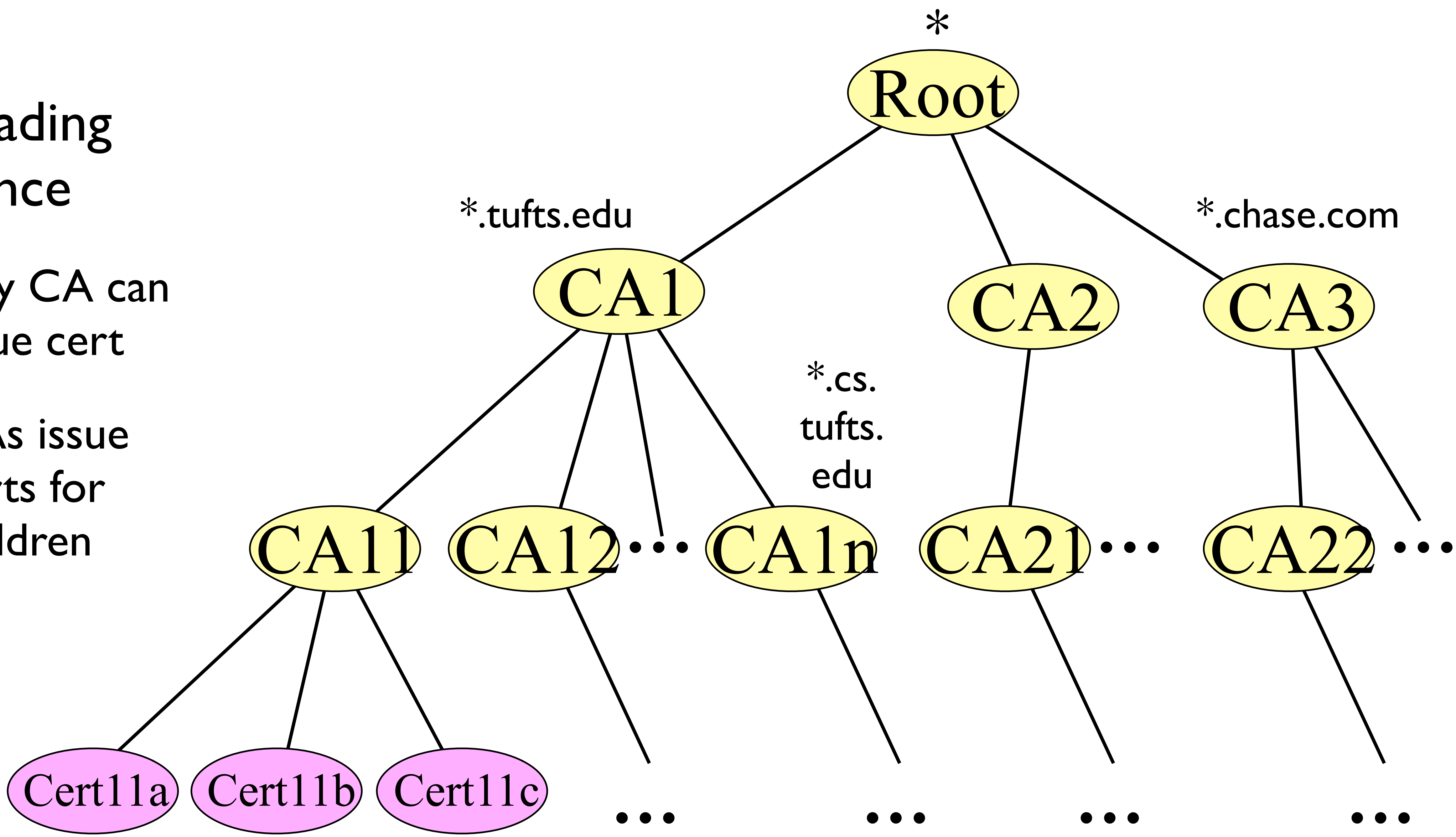
Solving the Turtles Problem (sorta)

- We need a **trust anchor**
  - there must be someone with authority
  - requires *a priori* trust
- Solution: form a trust hierarchy
  - "I believe **X** because..."
  - "**Y** vouches for **X** and..."
  - "**Z** vouches for **Y** and..."
  - "I <u>implicitly</u> trust **Z**."



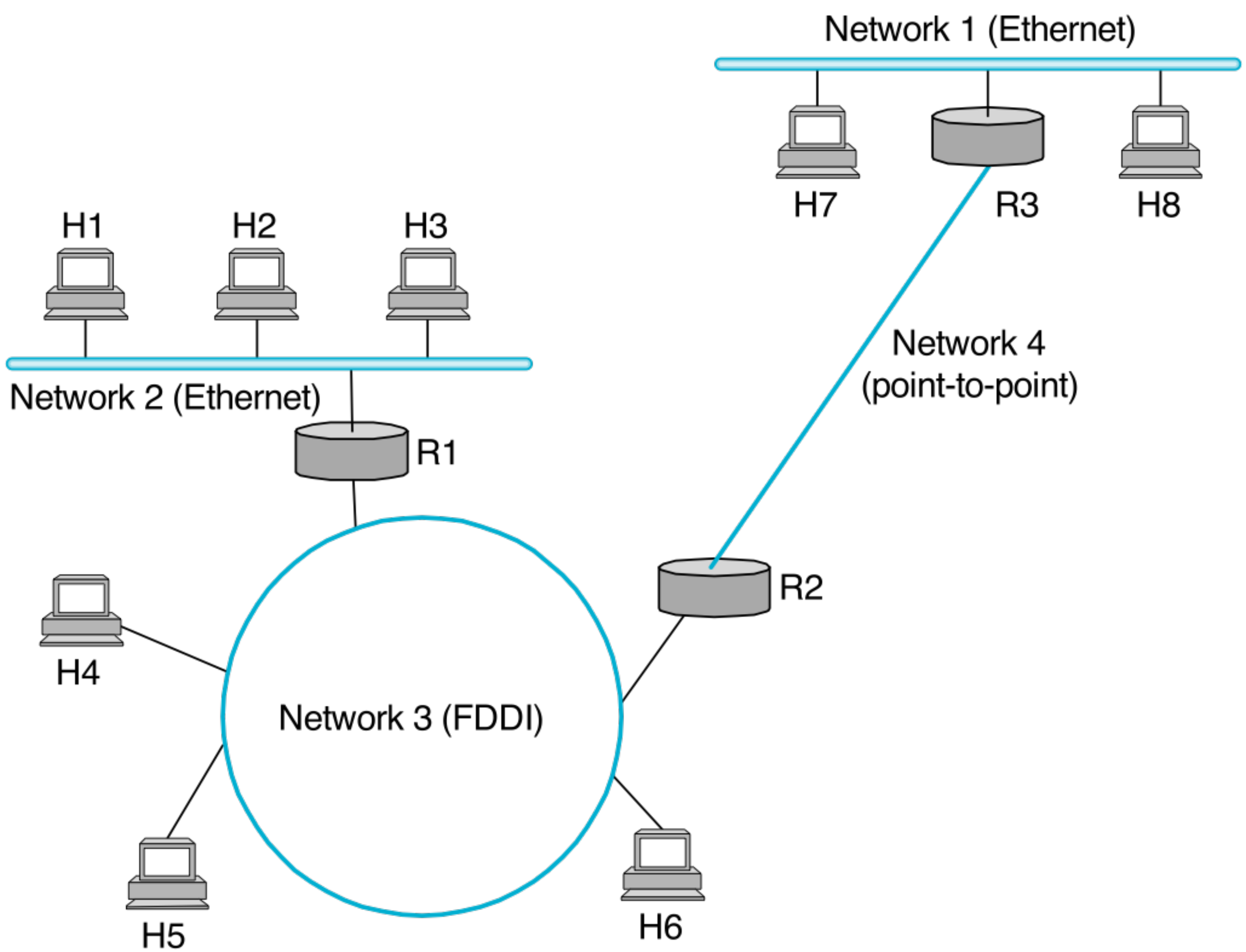
Public Key Infrastructure

- Rooted tree of CAs
- Cascading issuance
  - Any CA can issue cert
  - CAs issue certs for children



Part II
What is this network thingy that we're trying to protect?

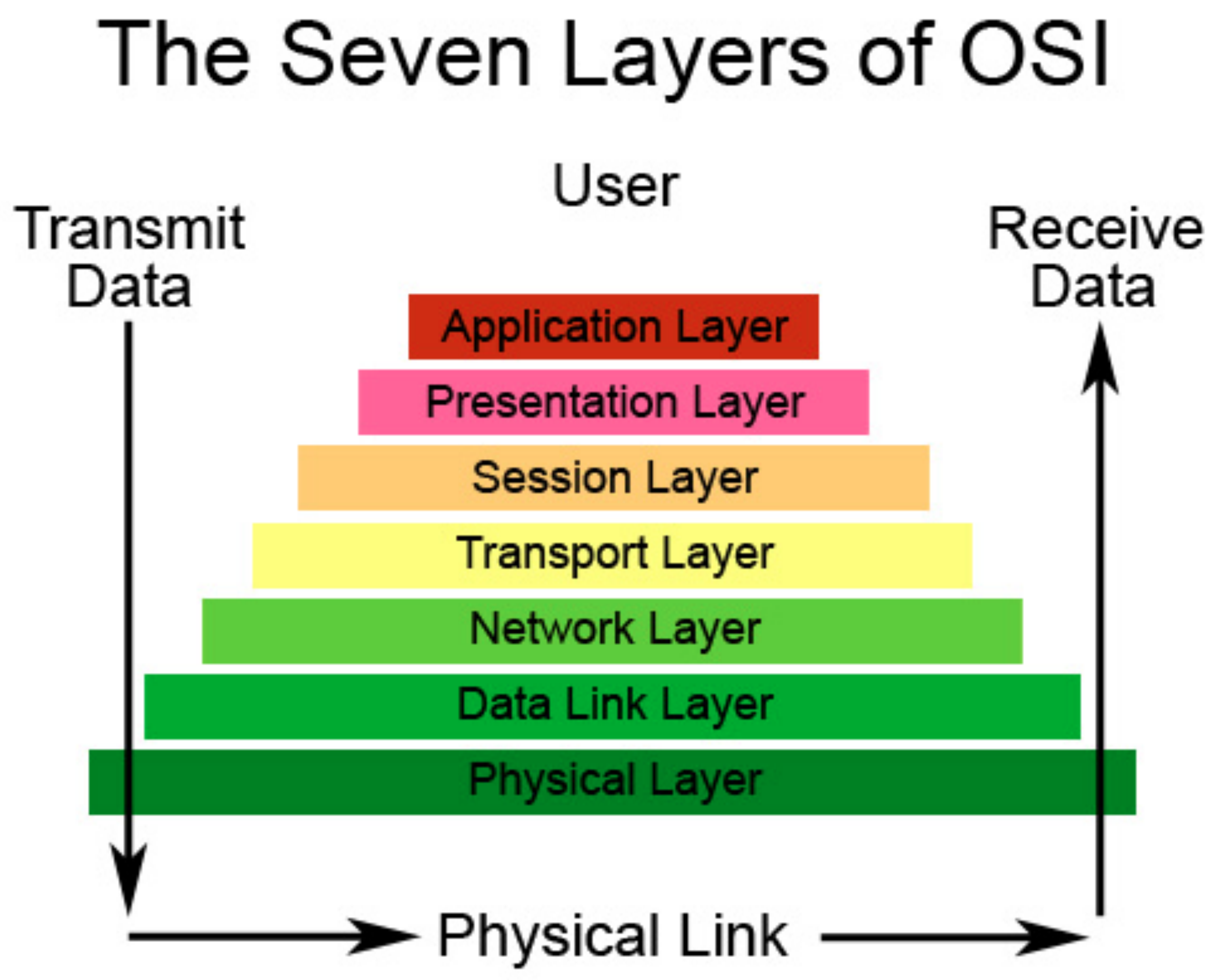
An Inter-network



Hn = host, Rn = router/gateway

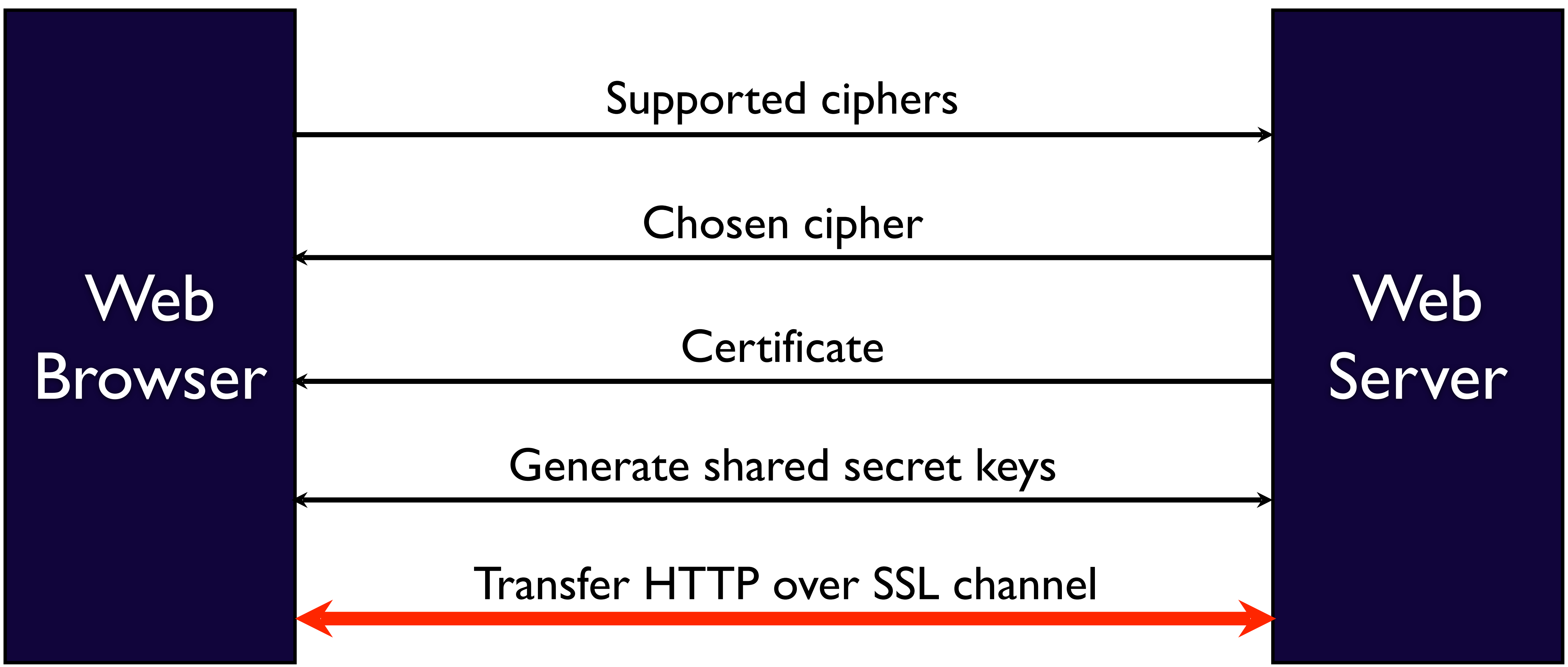
Layering

- System is broken into a vertical hierarchy of logically distinct entities (layers)
- Service provided by one layer is based solely on the service provided by layer below
- ISO: International Standards Organization / OSI: Open System Interface



SSL/TLS: Protecting the Internet by adding an "s" at the end of protocol names

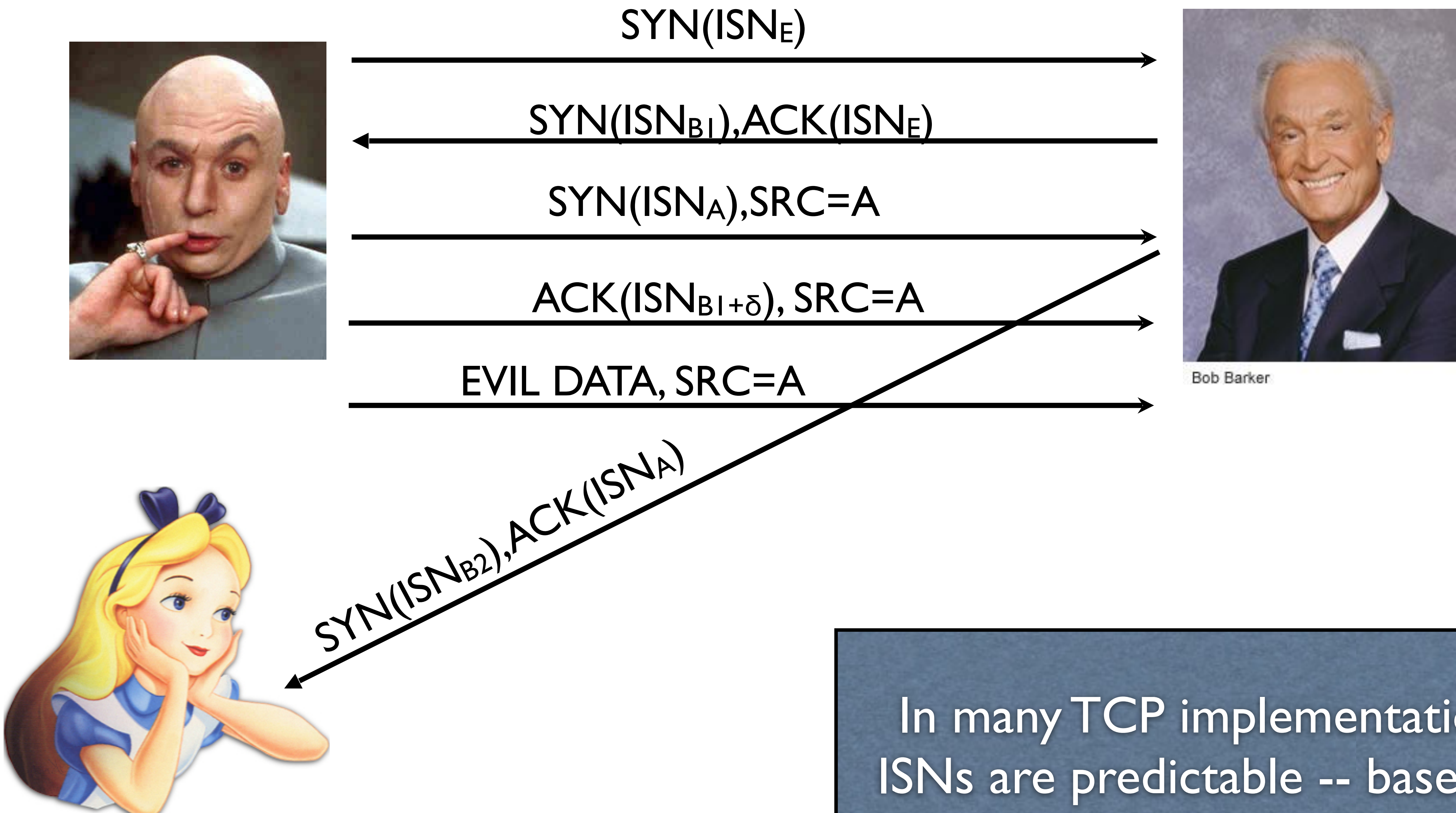
- HTTPS, FTPS, IMAPS, POP3S
- Add golden lock symbol



Unfortunately,

network + SSL ≠ secure network

Problems in TCP; e.g., Sequence Numbers



In many TCP implementations, ISNs are predictable -- based on time (e.g,. ++ each 1/128 sec)

ARP Spoofing

- Each ARP response overwrites the previous entry in ARP table -- last response wins!
- Attack: Forge ARP response
- Effects:
  - Man-in-the-Middle
  - Denial-of-service
- Also called **ARP Poisoning** or **ARP Flooding**

DNS Cache Poisoning

- All DNS requests have a unique query ID
- The nameserver/resolver uses this information to match up requests and responses -- this is useful since DNS uses UDP
- If an adversary can guess the query ID, then it can forge the responses and pollute the DNS cache
  - 16-bit query IDs (only 2^{16}=65536 possible query IDs)
  - Some servers increment IDs (or use some other predictable algo)
  - gethostbyname returns as soon as it gets a response, so first one in wins!!!
- Note: If you can observe the traffic going to a name server, you can pretty much arbitrarily own the Internet for the clients it serves

Routing Security

- Bad guys/gals/Internet-enabled toaster ovens play games with routing protocols.
- Implications for diverted traffic:
  - Enemy can see the traffic.
  - Enemy can easily modify the traffic.
  - Enemy can drop the traffic.

- Routing security in a nutshell: Cryptography can mitigate effects, but not stop them.

Wireless → Security--

- Packet sniffers
- Session hijacking
  - sniffing
  - mimic access point
- Jamming
  - effectively, DoS
- Interloping
  - access controls?
  - “hey, thanks for letting me use your unprotected wireless access point!”

Wireless Security

Let's sprinkle on some of that crypto stuff

But let's do it incorrectly (see WEP)

Virtual Private Networks (VPNs)

- Provides secure access to private network over public links
  - Often, goal is to provide access to corporate network (intranet) from outside (Internet)
  - Or, logically join physically separated networks
- Achieves some combination of:
  - Confidentiality
  - Integrity
  - Mutual authentication

Internet Anonymity 101: 10,000ft view

- Forward anonymous traffic at the application-layer via **network overlay**
  - Permits application-layer routing protocols
  - Overlay nodes act as intermediaries between sender and receiver
  - Packets transmitted using existing Internet infrastructure (no AS/ISP cooperation necessary)
- Use cryptography to prevent eavesdroppers from learning IDs of sender and/or receiver

Part III
Network Defenses

Worms and infection

- **The effectiveness of a worm is determined by how good it is at identifying vulnerable machines**
- Multi-vector worms use lots of ways to infect: e.g., network, email, drive by downloads, etc.

Botnets

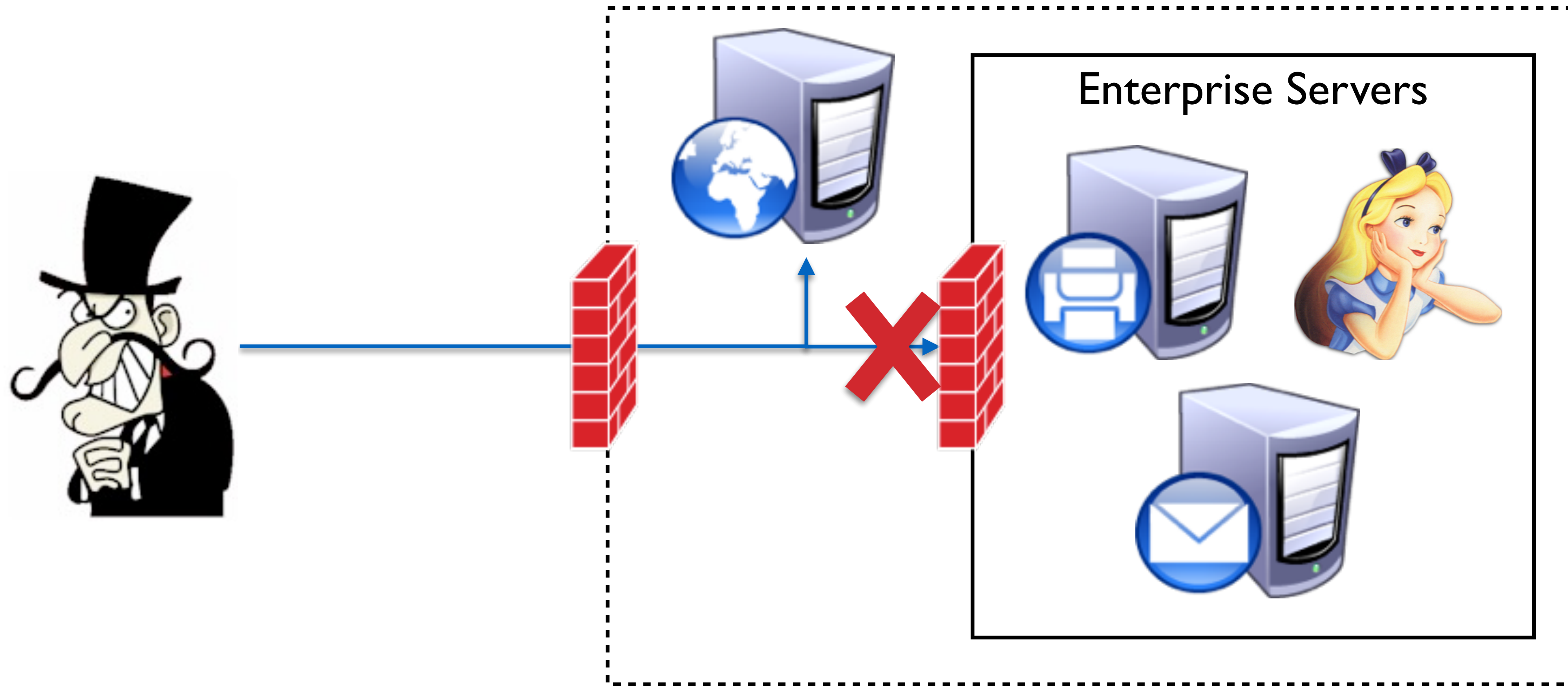
Distributed Denial-of-service (DDoS)

- DDoS: Network oriented attacks aimed at preventing access to network, host or service
  - Saturate the target's network with traffic
  - Consume all network resources (e.g., SYN flooding)
  - Overload a service with requests
    - Use "expensive" requests (e.g., "sign this data")
  - Can be extremely costly
- Result: service/host/network is unavailable
- Criminals sometimes use DDoS for racketeering
- Note: IP addresses of perpetrators are often hidden (spoofed)

Firewalls

- Filtering traffic based on **policy**
  - Policy determines what is acceptable traffic
  - Access control over traffic
  - Accept or deny
- May perform other duties
  - Logging (forensics, SLA)
  - Flagging (intrusion detection)
  - QoS (differentiated services)

DMZ (De-militarized Zone)



Intrusion Detection Systems

- Authorized eavesdropper that listens in on network traffic
- Makes determination whether traffic contains malware
  - usually compares payload to virus/worm signatures
  - usually looks at only incoming traffic
- If malware is detected, IDS somehow raises an alert
- Intrusion detection is a **classification problem**

Base Rate Fallacy

- Occurs when we assess P(X|Y) without considering prior probability of X and the total probability of Y
- Example:
  - *Base rate* of malware is 1 packet in 10,000
  - Intrusion detection system is 99% accurate
    - 1% false positive rate (benign marked as malicious 1% of the time)
    - 1% false negative rate (malicious marked as benign 1% of the time)
  - Packet X is marked by the NIDS as malware. What is the probability that packet X actually is malware?

Base Rate Fallacy

- 1% false positive rate (benign marked as malicious 1% of the time); TPR=99%
- 1% false negative rate (malicious marked as benign 1% of the time)
- *Base rate* of malware is 1 packet in 10,000
- Find Pr(IsMalware|MarkedAsMalware)
- Pr(Is|Marked) = Pr(Marked|Is)Pr(Is) / Pr(Marked) — By Bayes' Rule
  - Pr(Marked|Is)Pr(Is) = 0.99*1/10,000 = .000099
  - Pr(Marked) = Pr(Marked|Is)Pr(Is) + Pr(Marked|IsNot)Pr(IsNot)
    - Pr(Marked) = (.99*1/10,000) + (0.01*9,999/10,000) = .010098
- Pr(Is|Marked) = .000099/.010098 ~=0.98%

Honeypots

- **Honeypot:** a controlled environment constructed to trick malware into thinking it is running in an unprotected system
  - *"A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource."* -- Lance Spitzer
  - collection of decoy services (fake mail, web, ftp, etc.)
  - decoys often mimic behavior of unpatched and vulnerable services

Examining Malware

- **Trace system calls:**
  - most OSes support method to trace sequence of system calls
    - e.g., ptrace, strace, etc.
  - all "interesting" behavior (e.g., networking, file I/O, etc.) must go through system calls
  - capturing sequence of system calls (plus their arguments) reveals useful info about malware's behavior

Tracing System Calls

```
root@ubuntu:~# strace -o out.txt ./trace-me
What just happened??
```

```
mkdir("/tmp/.tomato", 0700)                    = 0
brk(NULL)                                      = 0x55eb8155e000
brk(0x55eb8157f000)                            = 0x55eb8157f000
openat(AT_FDCWD, "/tmp/.tomato/answer.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
write(3, "I Was created!!!!", 17)              = 17
close(3)                                       = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
write(1, "What just happened??\n", 21What just happened??
) = 21
exit_group(0)                                  = ?
+++ exited with 0 +++
```

https://malware.news/t/elf-malware-analysis-101-part-3-advanced-analysis/46838

These defenses aren’t a panacea

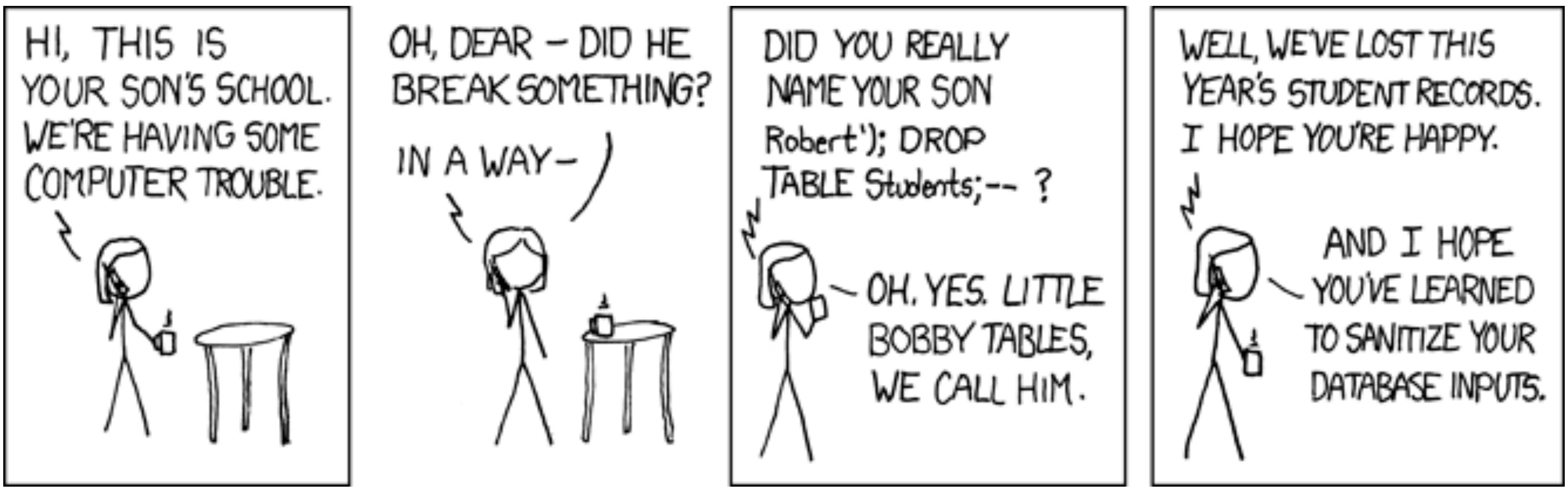
- Firewalls depend on accurate policies
- VERY difficult to get both good recall and precision in NIDSes
  - Malware comes in small packages
  - Looking for one packet in a million (billion? trillion?)
- Honeypots help us better understand malware, but often don’t protect us from new (unseen) malware

Part IV
Web Security

Adding State to the Web with Cookies

- Cookies were designed to offload server state to browsers
  - Not initially part of web tools (Netscape)
  - Allows users to have cohesive experience
  - E.g., flow from page to page
- Someone made a design choice
  - Use cookies to *authenticate* and *authorize* users (e.g., Amazon.com shopping cart, WSJ.com)
  - Web security depends on how we handle and protect our cookies

Little Bobby Tables



Preventing Web Attacks

- Broad Approaches
  - Validate input (also called **input sanitization**)
  - Limit program functionality
    - Don't leave open ended-functionality
  - Execute with limited privileges
    - Don't run web server as root
    - Apply policy of ***least privilege***
  - Input tracking, e.g., taint tracking
  - Source code analysis, e.g., c-cured

Drive-by-Downloads

- **Drive-by-downloads**: bypasses NAT, firewalls, proxies, etc. to attack victim machine
    - usually causes victim browser to open 0-by-0 pixel iFRAME pointing to site that installs malware using JavaScript loader
    - uses plugin vulnerabilities to infect machine
- "All Your iFRAMES Point to Us" -- study of drive-by-downloads by Google and Johns Hopkins
    - 1.3% of Google's search results contain malicious URL

(Slightly) more secure web browsing via sandboxing

- **Sandboxing**
  - isolate pages in separate “spaces”
  - prevent pages from reading/writing data from other pages
  - goal is to isolate the effects of malicious websites

Part V
Humans and Computers and Security

Key Challenges

- Security is a secondary task
- Security concepts are hard
- Human capabilities are limited
- Misaligned priorities
- Active adversaries
  - Unlike traditional UX

Spam vs. Phishing vs. Spear Phishing

Spam

- Unsolicited email
- Low effort
- Not very effective

Phishing

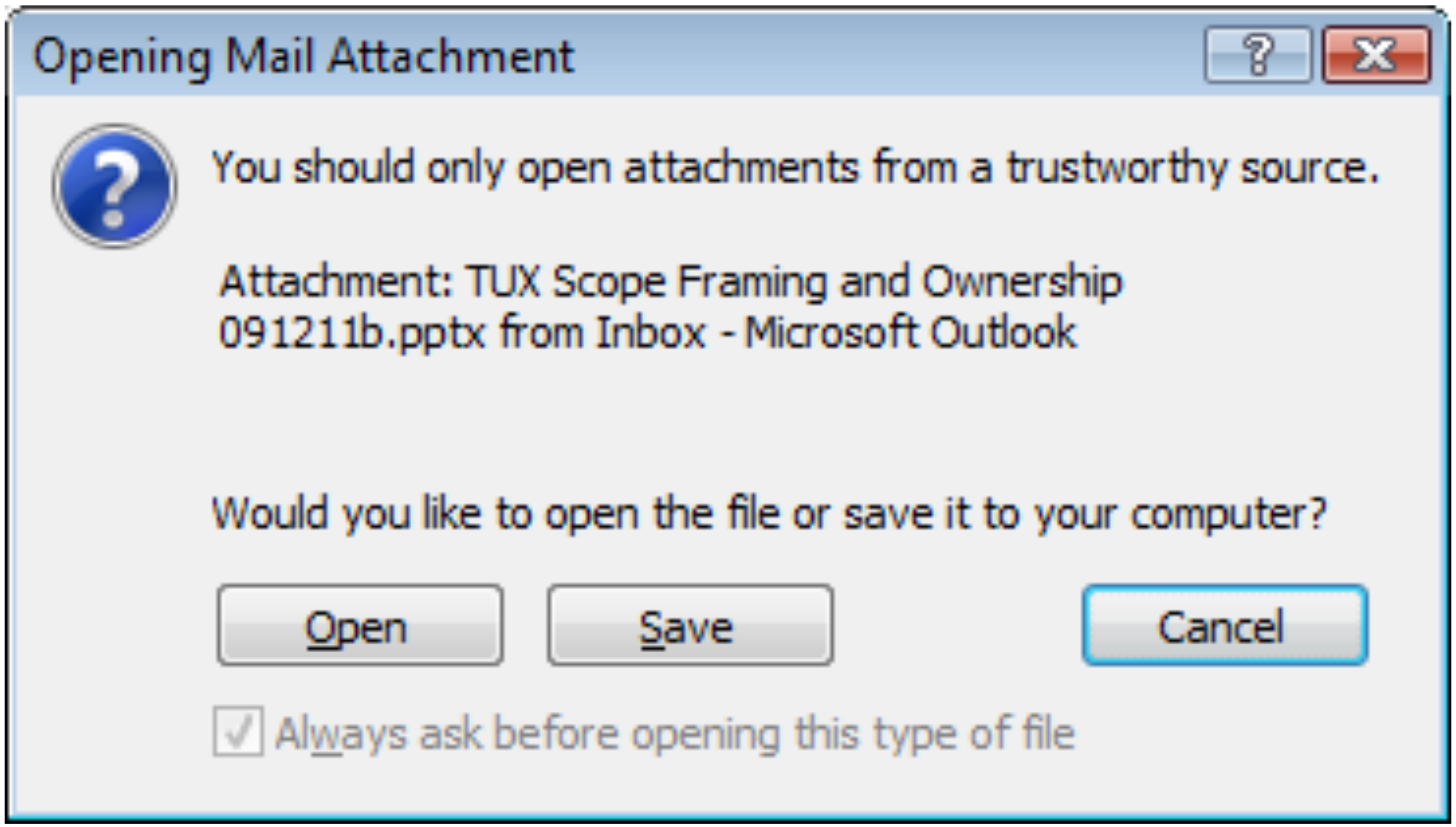
- Mimics a **trusted authority**
- Higher effort
- More effective than spam

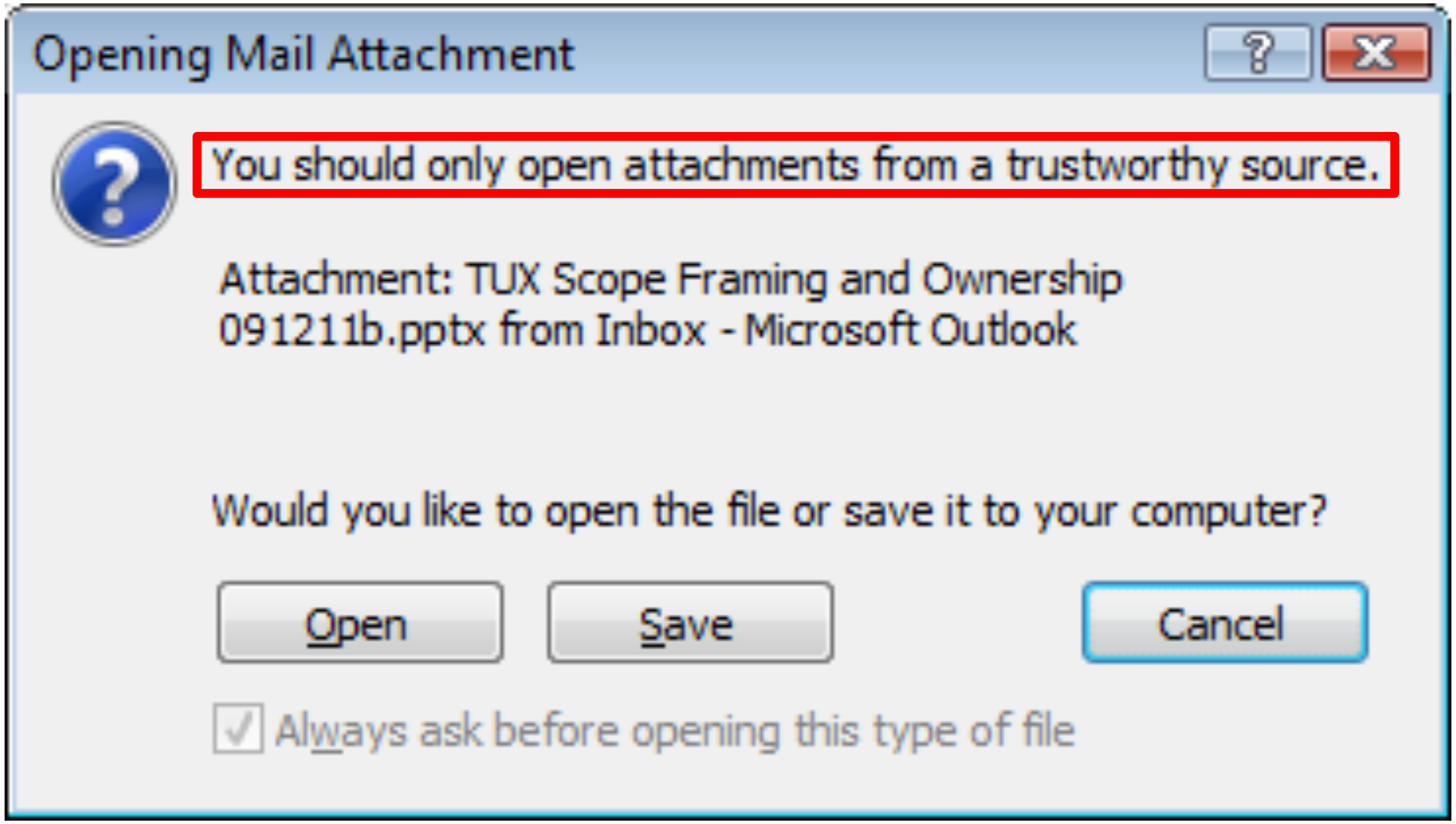
Spear Phishing

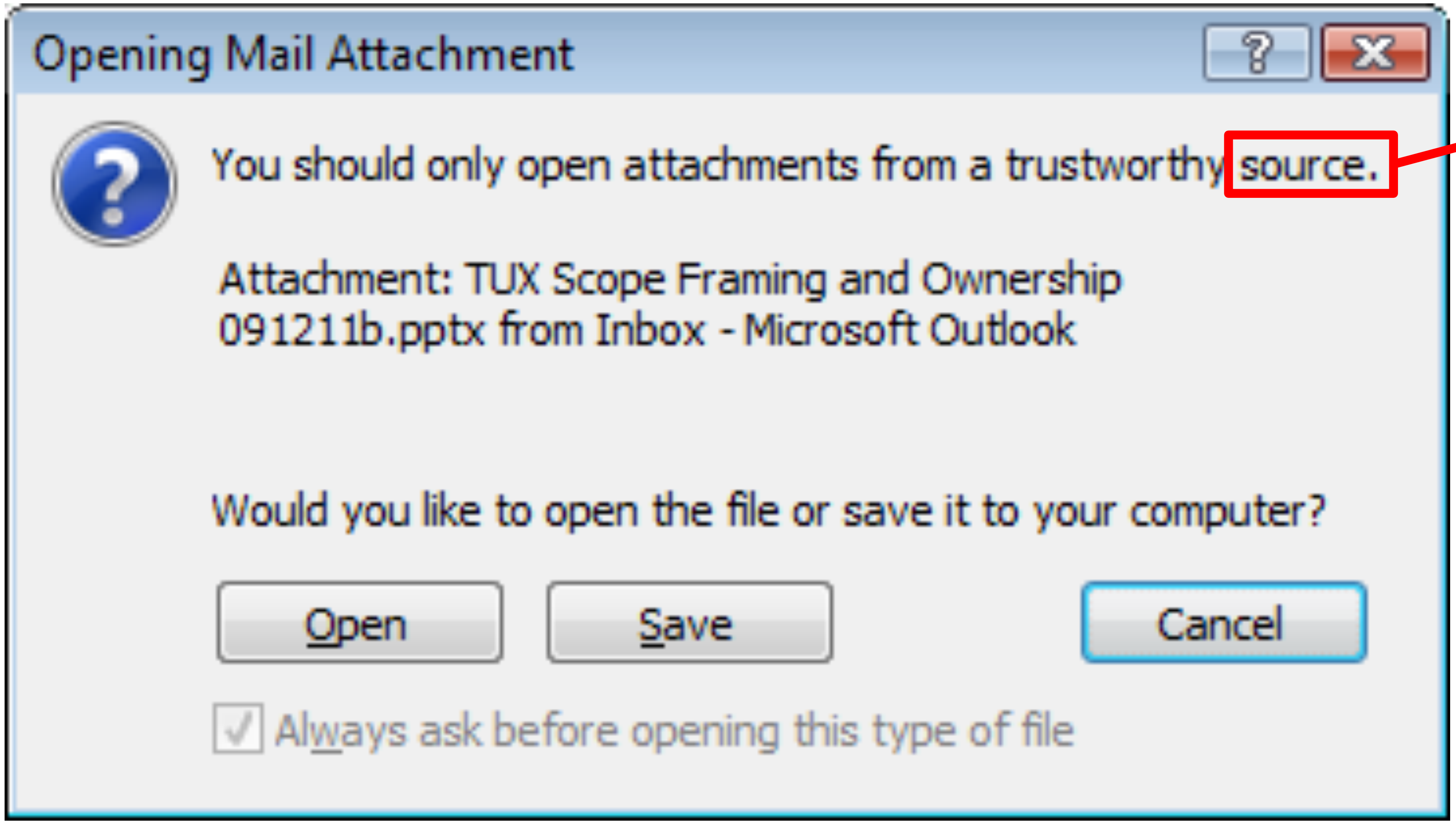
- Very highly targeted phishing
- Requires extensive knowledge and crafting
- **Extremely effective**

Key challenges

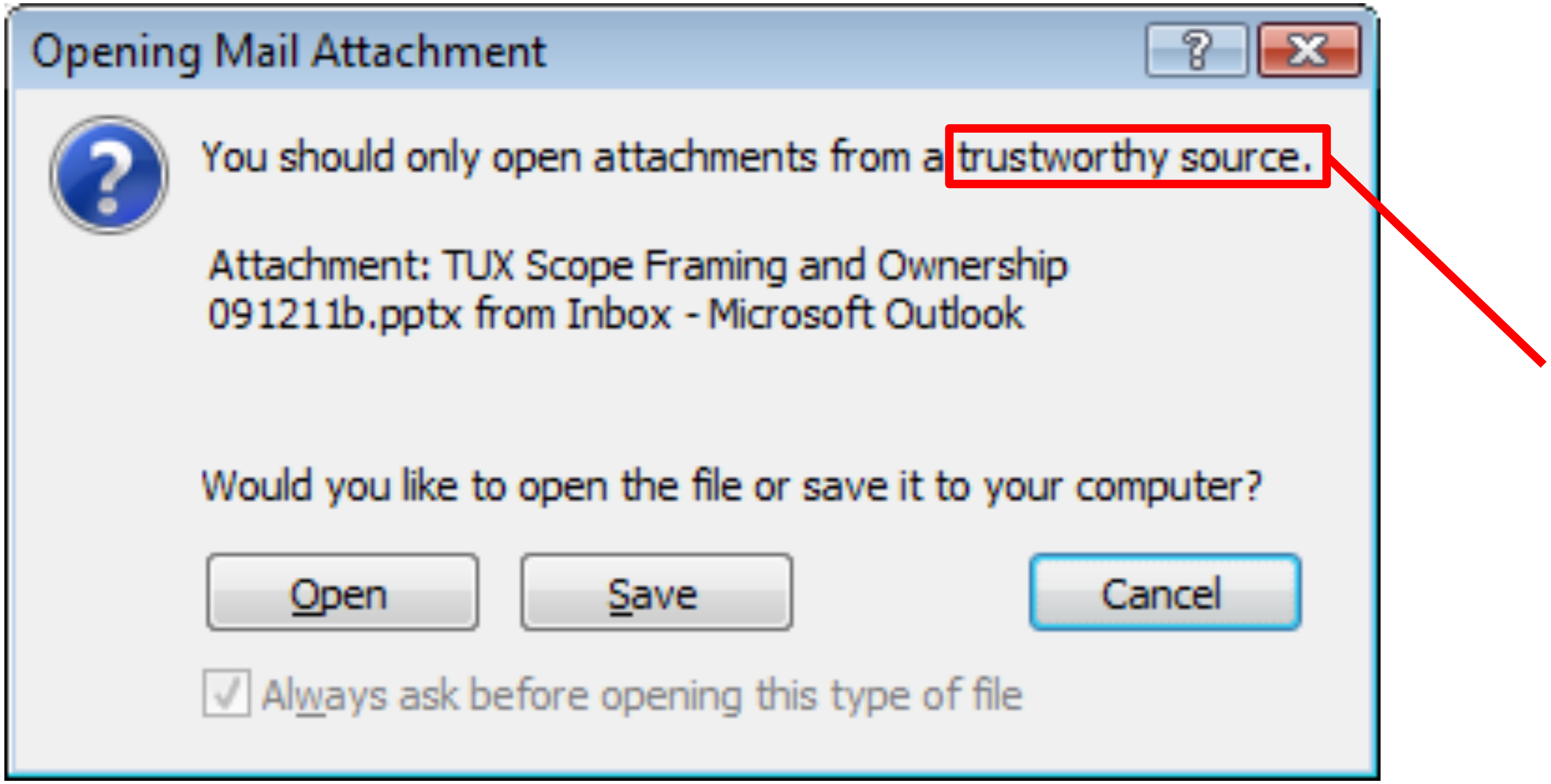
- Security is a secondary task
- Security concepts are hard
- Human capabilities are limited
- Misaligned priorities
- Active adversaries
  - Unlike ordinary UX
- Habituation
  - The "crying wolf" problem


Opening Mail Attachment
You should only open attachments from a trustworthy source.
Attachment: TUX Scope Framing and Ownership
091211b.pptx from Inbox - Microsoft Outlook
Would you like to open the file or save it to your computer?
Open
Save
Cancel
Always ask before opening this type of file


Opening Mail Attachment
You should only open attachments from a trustworthy source.
Attachment: TUX Scope Framing and Ownership 091211b.pptx from Inbox - Microsoft Outlook
Would you like to open the file or save it to your computer?
Open
Save
Cancel
Always ask before opening this type of file

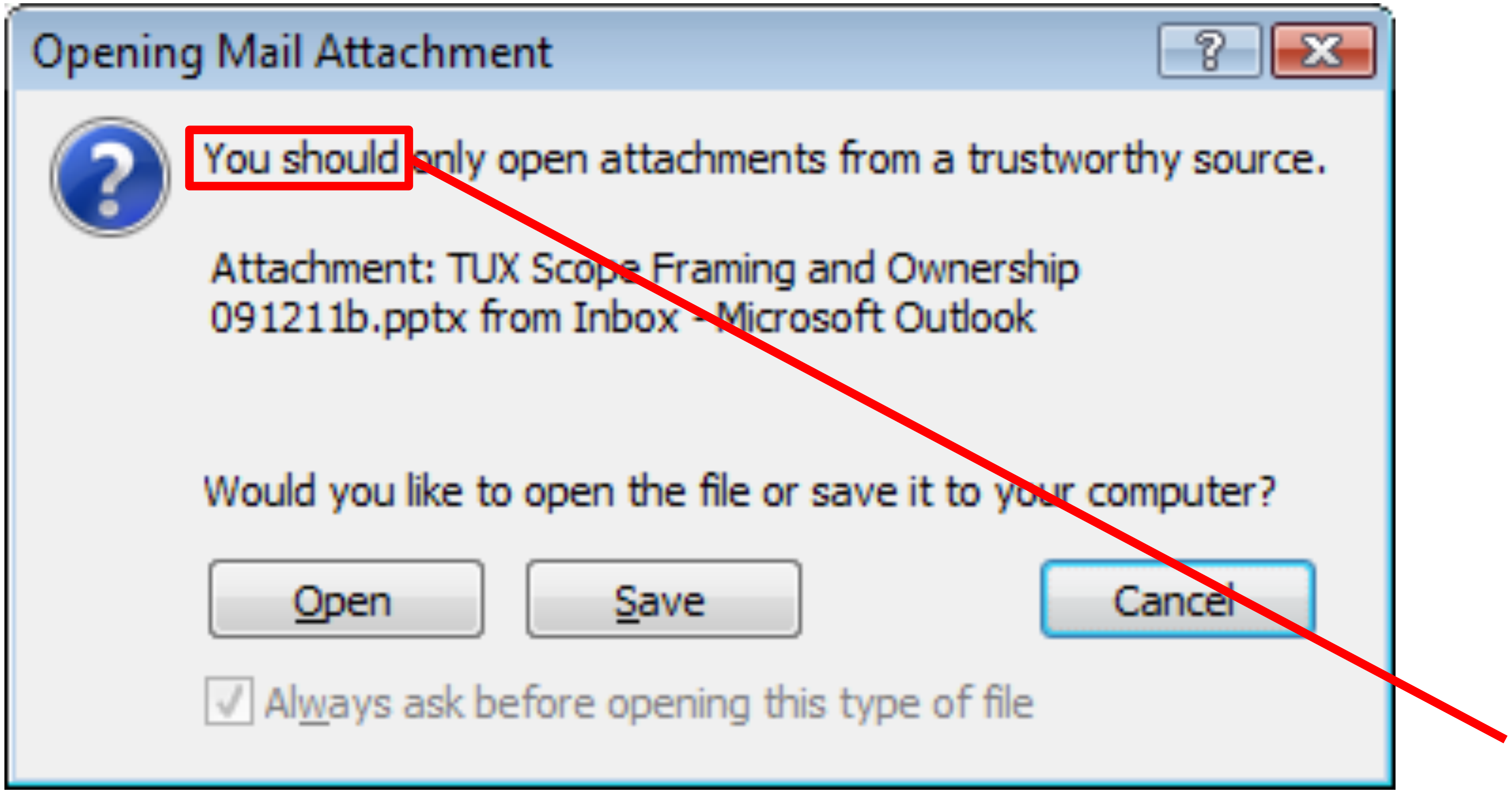


What's the source of this attachment?



What's the source of this attachment?

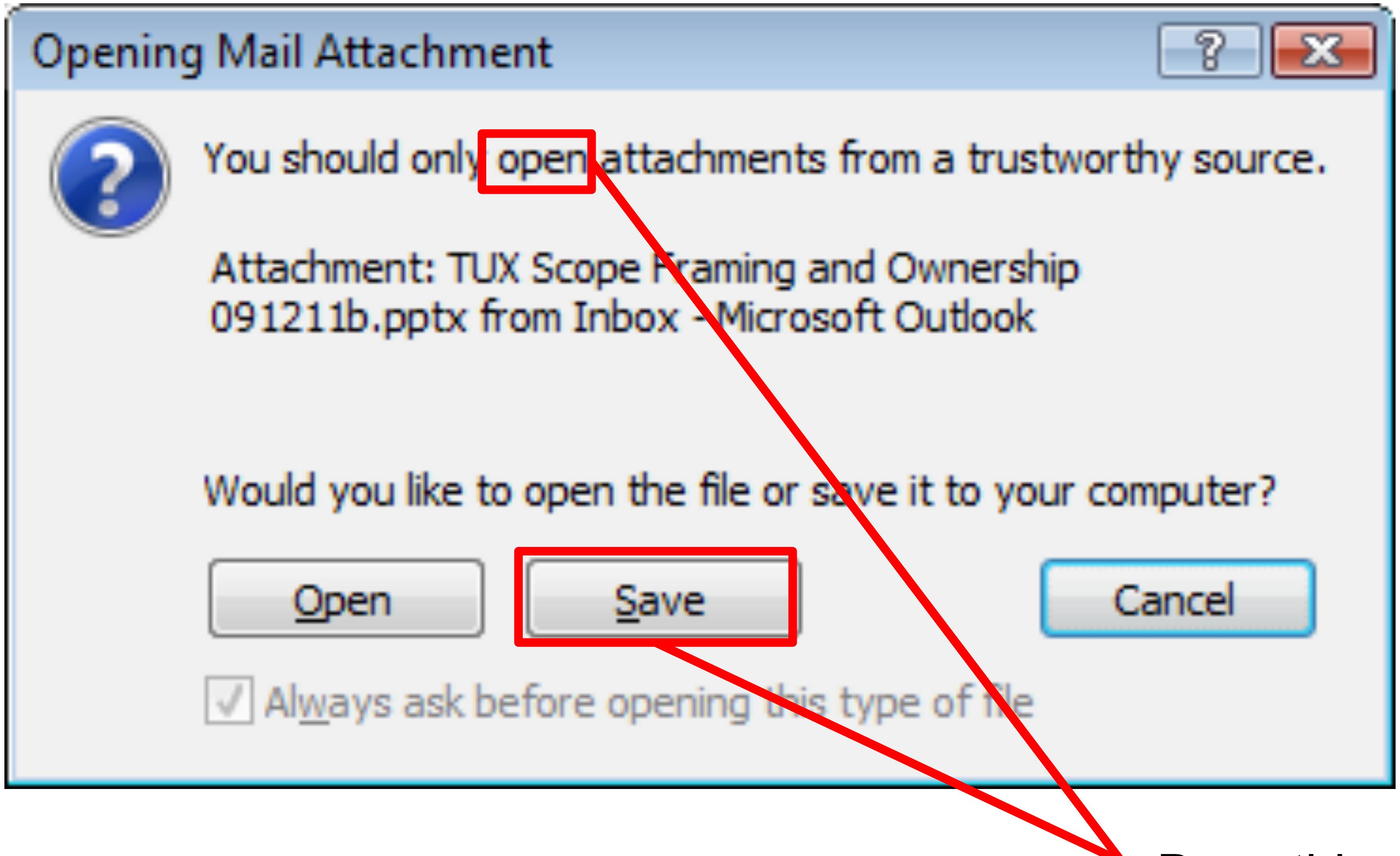
What makes a source trustworthy or not trustworthy?



What's the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don't follow this advice?

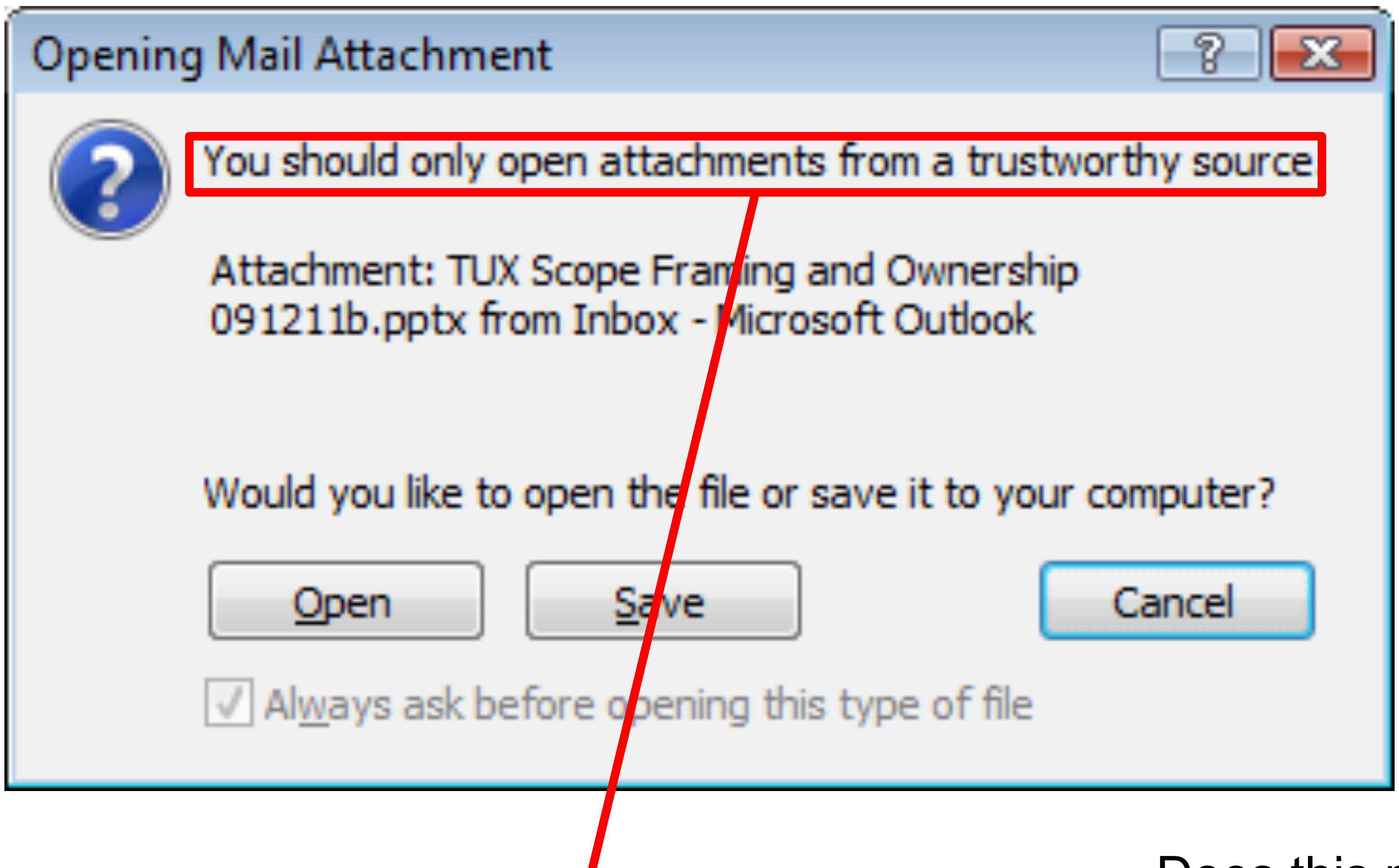


What's the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don't follow this advice?

Does this mean that opening is dangerous but saving is safe?



What's the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don't follow this advice?

What steps can I take to decide what to do?

Does this mean that opening is dangerous but saving is safe?

Use psychology in your favor

- Limit memory requirements
- Grab attention when you need it
- Make critical information stand out / avoid habituation
- Minimize effort:
  - To get users to take action, make it easy
  - To get users to avoid danger, make it difficult

Fin.

Course Eval

https://tufts.bluera.com/tufts/