CS 330: Network Applications & Protocols

Introduction to Computer Networks & the Internet

Department of Engineering and Computer Science

York College of Pennsylvania

Overview of Network Security

- **What is Network Security?**
- **Principles of Cryptography**
- **Message Integrity, Authentication**
- **Operational Security: Firewalls and IDS**

Overview of Network Security

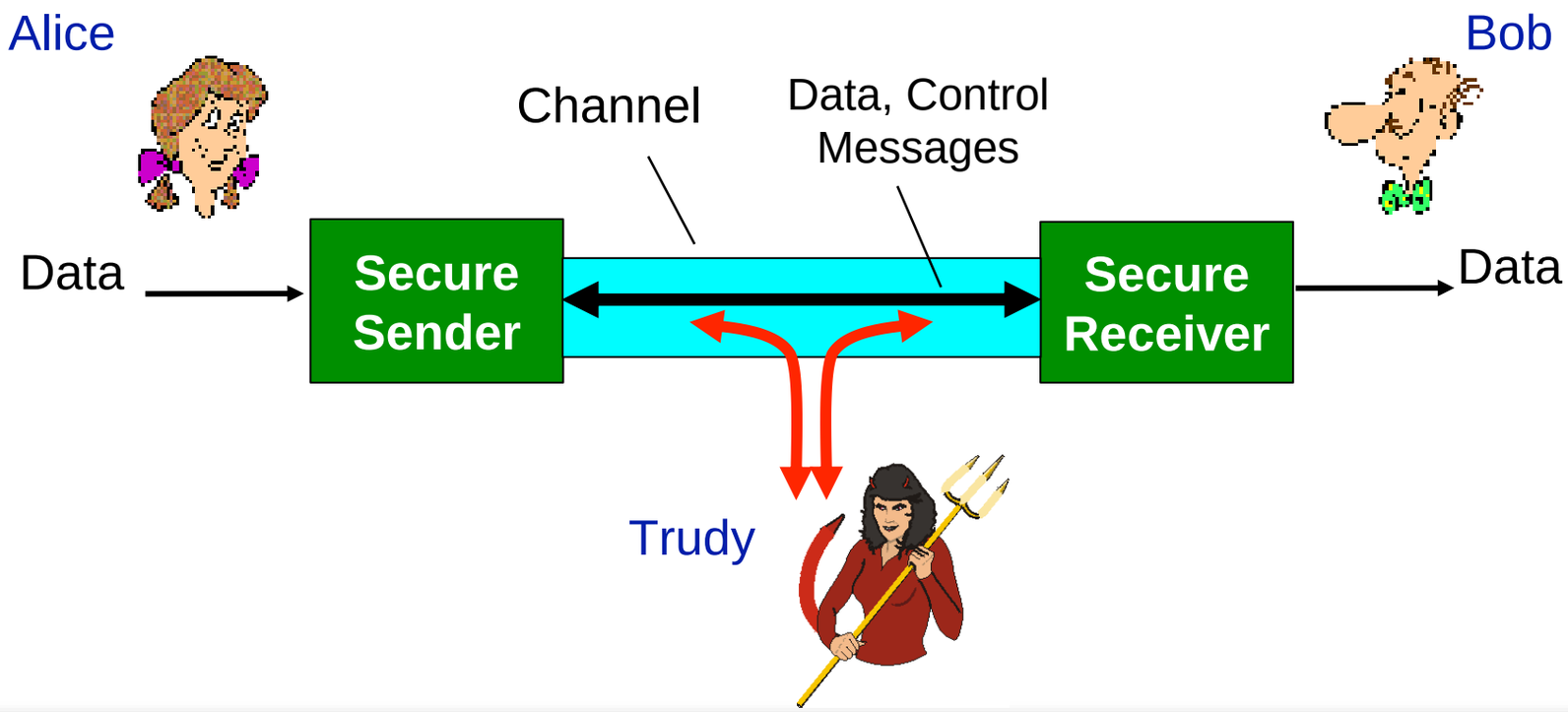
- **What is Network Security?**
- **Principles of Cryptography**
- **Message Integrity, Authentication**
- **Operational Security: Firewalls and IDS**

What is Network Security?

- **The following four items are desirable properties of secure communication:**
  - Confidentiality - only sender and intended receiver should "understand" message contents
    - Sender encrypts message
    - Receiver decrypts message
    - Eavesdropper should not be able to understand message
  - End-point Authentication - sender and receiver want to confirm identity of each other
    - Am I really talking to who I think I'm talking to?
  - Message Integrity - sender and receiver want to ensure message is not altered (in transit, or afterwards) without detection
  - Operational Security - services must be accessible and available to users
    - Protect network from downtime through redundancy
    - Protect network from attacks with firewalls, intrusion detection systems, etc.

Network Security

- **Bob and Alice want to communicate "securely" to prevent others from understanding their communication**
- **Trudy, the intruder, may intercept, delete, add messages**
  - Bob and Alice want to be able to detect changes made by an intruder
  - Bob and Alice don't want the intruder to be able to understand their messages



Network Security

- **In previous example:**
  - Bob and Alice don't necessarily have to represent 'users'
  - Can represent any number of machines that need to communicate with each other
- **Other examples of machines that may want secure communication**
  - Web browser/server for electronic transactions (e.g., on-line purchases)
  - On-line banking client/server
  - DNS servers
  - Routers exchanging routing table updates

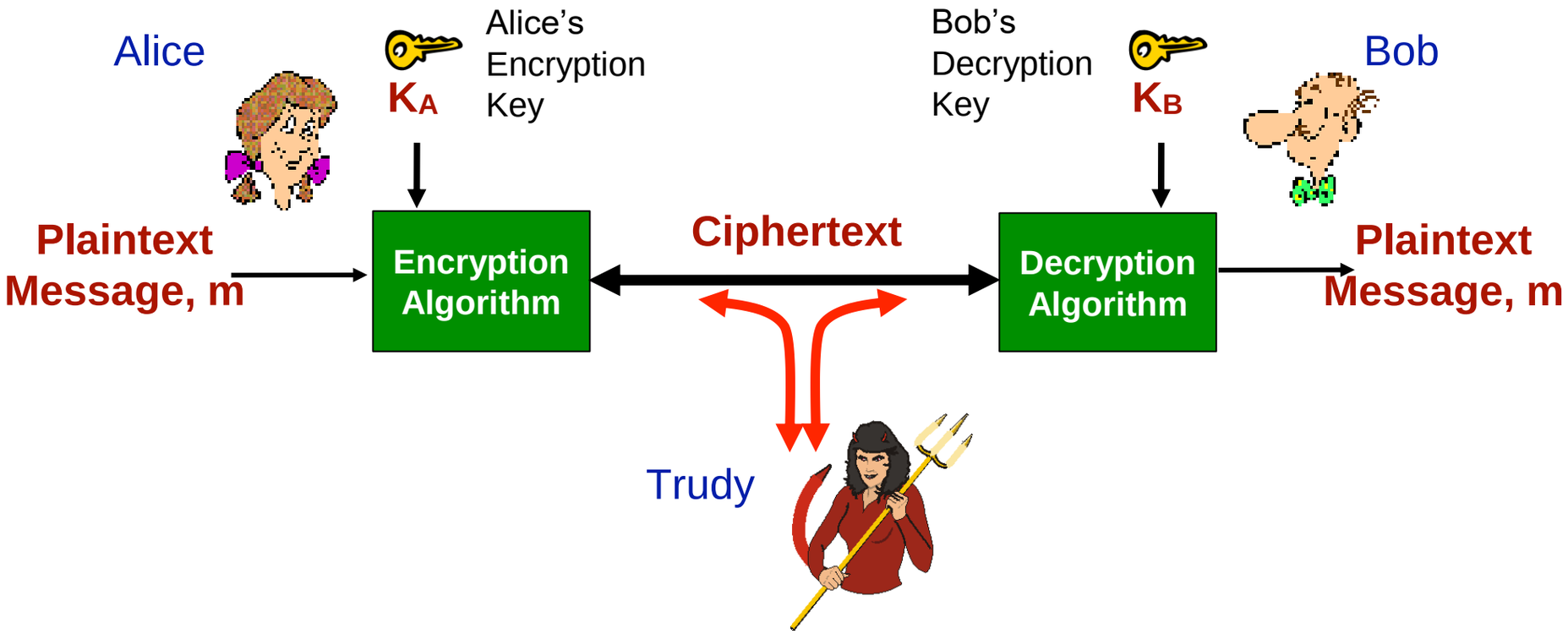
What Can an Intruder Do?

- **Eavesdrop - intercept or listen to messages**
- **Modification, Insertion, or Deletion of messages or message content**
- **Impersonation - can fake (spoof) source address in packet (or any field in packet)**
- **Hijacking - "take over" ongoing connection by removing sender or receiver, inserting himself in place**
- **Denial of Service - prevent service from being used by others (e.g., by overloading resources)**

Overview of Network Security

- **What is Network Security?**
- **Principles of Cryptography**
- **Message Integrity, Authentication**
- **Operational Security: Firewalls and IDS**

The Language of Cryptography



m - plaintext message

$K_A(m)$ - ciphertext, encrypted with key **K_A**

$m = K_B(K_A(m))$ - original plaintext message can be recovered with **K_B**

In **symmetric key systems**, both keys are the same

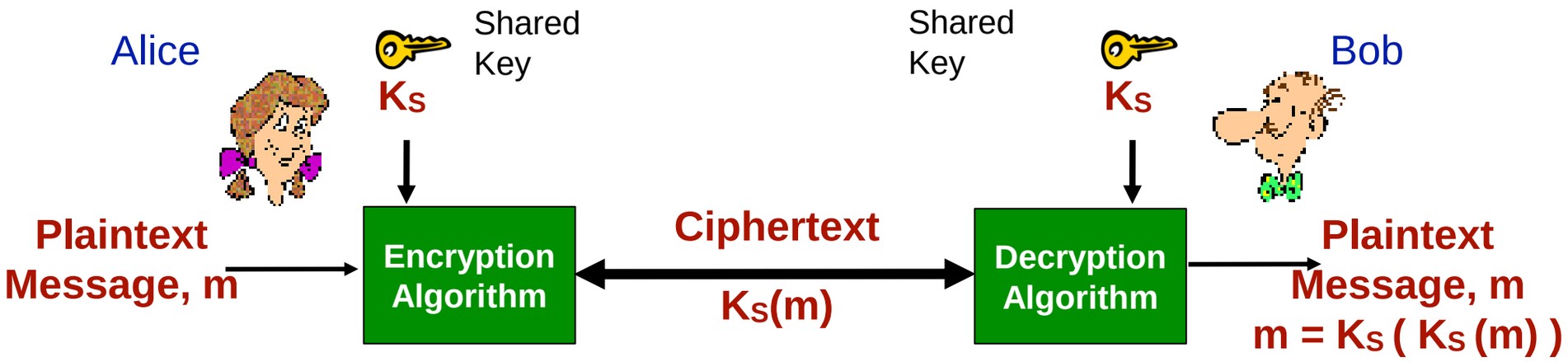
In **public key systems**, multiple keys are used:

- a shared public key, and
- a private key for each user

Breaking an encryption scheme

- **cipher-text only attack: Trudy has ciphertext she can analyze**
- **two approaches:**
  - brute force: search through all keys
  - statistical analysis
- **known-plaintext attack: Trudy has plaintext corresponding to ciphertext**
  - e.g., in monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,
- **chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext**

Symmetric Key Cryptography



- **Symmetric Key Cryptograph - Bob and Alice share the same (symmetric) key K_S**
  - Key may be a simple substitution pattern in monoalphabetic substitution cipher
- **How should Bob and Alice agree on a key?**
- **How should Bob and Alice exchange the shared key?**

Simple Encryption Scheme

- **Substitution cipher - substituting one thing for another**
  - Monoalphabetic cipher substitutes one letter for another
- **Encryption key is the mapping from set of 26 letters to set of 26 letters**

```
plaintext:   abcdefghijklmnopqrstuvwxyz
ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

- **Example:**

```
hello world
acggk rkogv
```

Pretty easy to break this type of cipher; same as crypto puzzles in weekly newspapers

A More Sophisticated Encryption Approach

- **Polyalphabetic encryption uses *n* monoalphabetic substitution ciphers**
  - Cycles through monoalphabetic ciphers in some pattern
    - For example, if *n*=4: M_1,M_3,M_4,M_3,M_2; M_1,M_3,M_4,M_3,M_2; …
  - For each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
    - For example, dog: d from M_1, o from M_3, g from M_4
    - Symbols may be substituted by ciphers throughout message
      - Much more difficult to break using crypto puzzle approach
- **Encryption key includes the *n* monoalphabetic substitution ciphers and the cyclic pattern in which they are applied**

Block Ciphers

- **Modern ciphers divide messages into k bit blocks and encrypt each of those block independently**
  - For small values of k, a simple lookup table is suitable

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Simple 3-bit block cipher

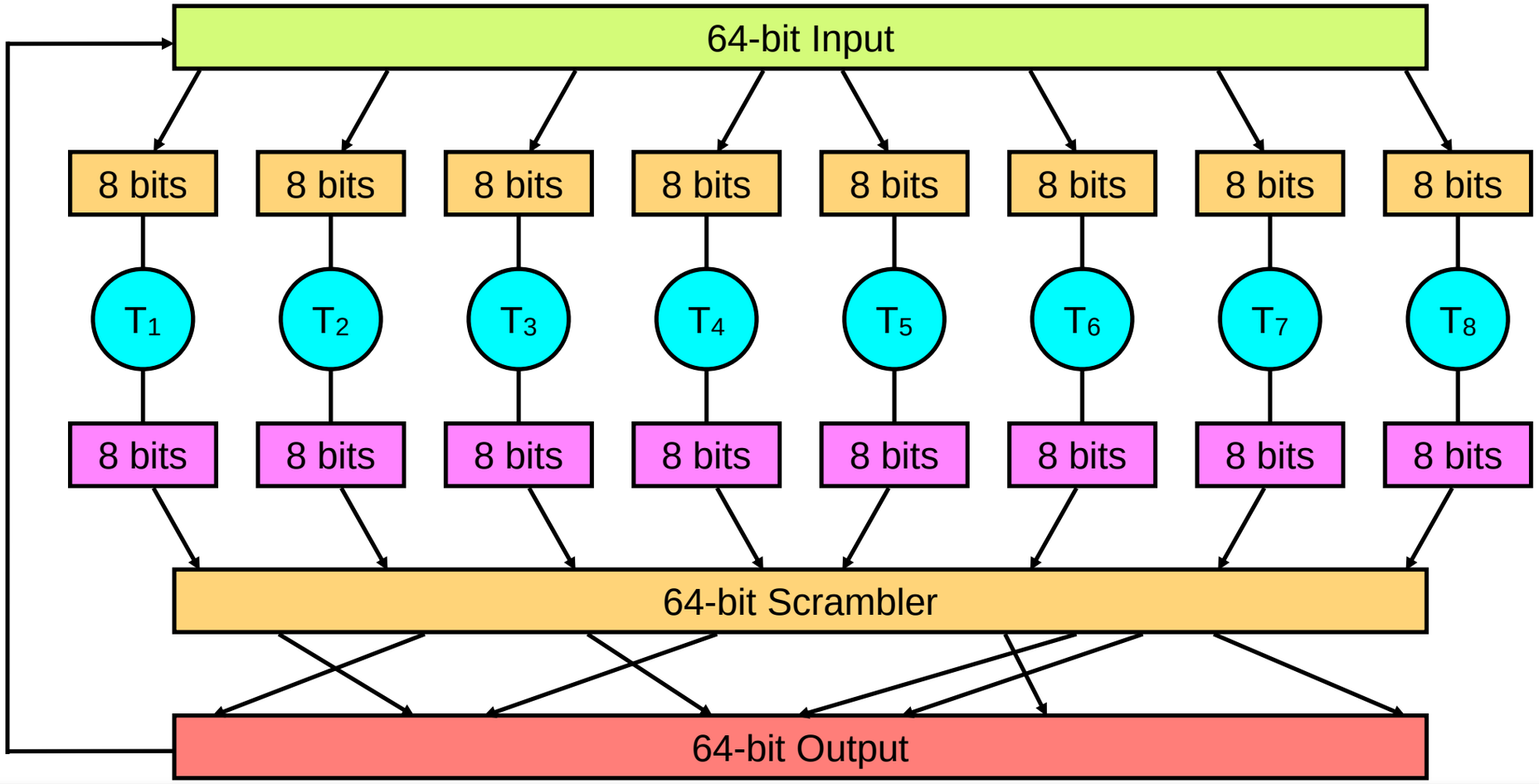
  - For large values of k (i.e. k=64, k=128, etc.), a lookup table would be too large
    - Instead, modern ciphers use mathematical functions to simulate these tables

Block Ciphers (Cont.)

- **Example of a block cipher function**
  - Divide input blocks into smaller 8-bit chunks
  - Use smaller, more manageable 8-bit lookup tables
  - Scramble the bits and feed them back around to the input
  - Loop this *n* times such that each input bit can affect the output bits

This is similar to the approach used by DES and AES

Loop for *n* rounds



Cipher Block Chaining

- **Since block cipher is a mathematical function, the same input will always produce the same output**
  - This is bad and provides an attack vector for an adversary
- **Cipher block chaining introduces randomness into the encrypted message using a randomly generated Initialization Vector (IV)**
  - IV is the same size as a block in the block cipher
  - The first block to be encrypted is XORed with the IV *before* being encrypted with the block cipher
    - The encrypted first block is XORed with second block to propagate randomness (output of second block is XORed with third, etc.)
  - IV is typically prepended as plaintext to encrypted message and sent along with message
    - Introduces a small overhead for sending encrypted messages
    - Receiver cannot decrypt the message without the IV

Common Block Cipher Algorithms

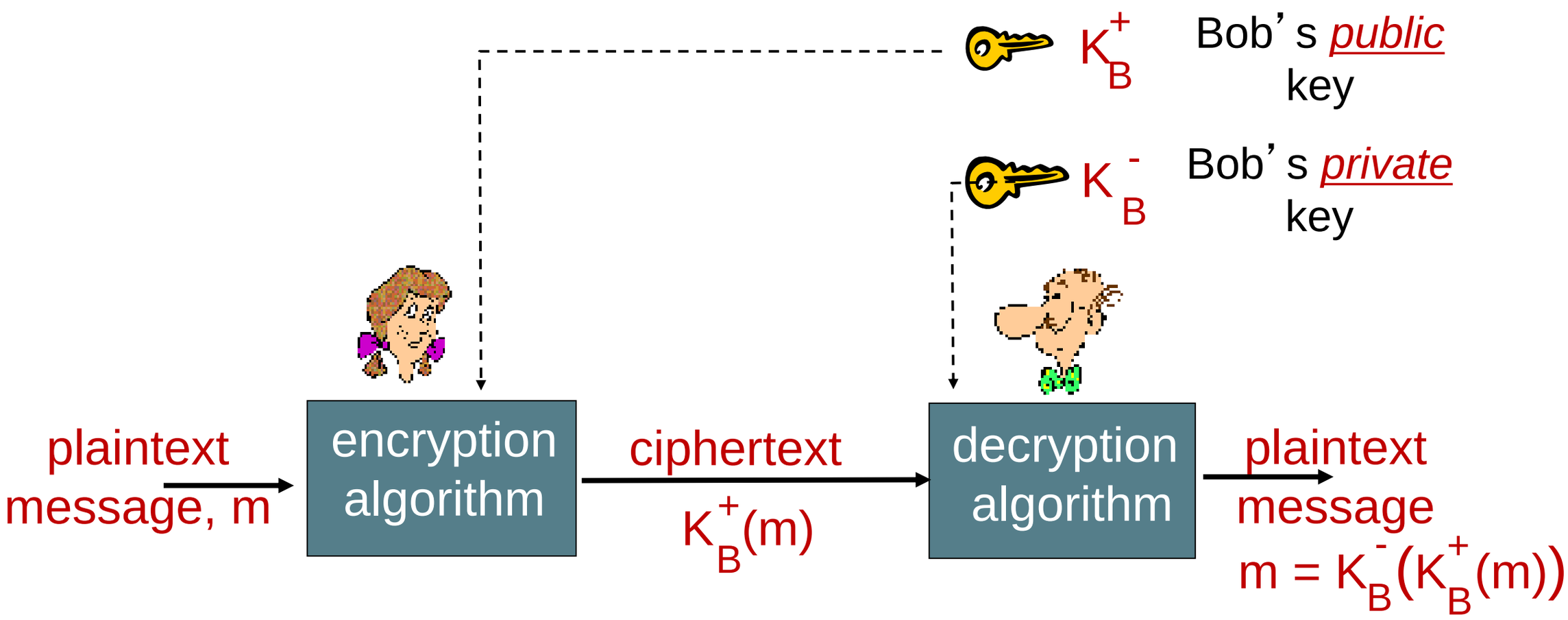
- **DES: Data Encryption Standard**
  - 56-bit symmetric key, 64-bit plaintext input
  - Block cipher with cipher block chaining
- **3DES: Triple Data Encryption Standard**
  - Same as DES, but encrypt message 3 times with 3 different keys
- **AES: Advanced Encryption Standard**
  - Replaced DES in most applications
  - Processes data in 128 bit blocks
  - 128, 192, or 256 bit keys

Nation Institute of Standards and Technology estimates that if theoretically had a machine that could crack DES in 1 second, it would take that same machine 149 trillion years to crack AES.

Public Key Cryptography

- **Symmetric Key Cryptography**
  - Requires sender and receiver to know a shared secret key
  - How should they agree on key in first place (particularly if never “met”)?
  - How should they share the key?
- **Public Key Cryptography**
  - Radically different approach
  - Sender and receiver do not share a secret key
  - Sender and receiver each have two keys: a shared public key and a private key
  - Public encryption key is known to all (even intruders)
  - Private decryption key known only to receiver

Public key cryptography



Public Key Cryptography (Cont.)

- **Sender determines a private key to use**
  - DOES NOT provide that private key to ANYONE
- **Receiver determines a private key to use**
  - DOES NOT provide that private key to ANYONE

- **Sender and receiver agree on a shared public key**
  - Does not matter if an intruder sees the shared public key
  - Public key can be exchanged over an unsecured channel
  - Great video provides general idea of how this works (Diffie Hellman key exchange)
    - http://www.youtube.com/watch?v=YEBfamv-_do

VIDEO:

http://www.youtube.com/watch?v=YEBfamv-_do

Diffie-Hellman Vulnerabilities

- **Does NOT provide authentication**
- **Vulnerable to man-in-the-middle attacks**
  - Intruder can establish one connection to Bob and another to Alice, intercept messages, re-encrypt and send

- **Another public key cryptography technique that avoids this problem is RSA**
  - Great video provides general idea of how RSA works
    - http://www.youtube.com/watch?v=wXB-V_Keiu8

Prerequisite: modular arithmetic

- **x mod n = remainder of x when divide by n**
- **facts:**

  [(a mod n) + (b mod n)] mod n = (a+b) mod n

  [(a mod n) - (b mod n)] mod n = (a-b) mod n

  [(a mod n) * (b mod n)] mod n = (a*b) mod n

- **thus**

  $(a \bmod n)^d \bmod n = a^d \bmod n$

- **example: x=14, n=10, d=2:**

  $(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$

  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

VIDEO:

http://www.youtube.com/watch?v=wXB-V_Keiu8

Why $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$?

follows directly from modular arithmetic:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{de} \bmod n$$

$$= (m^d \bmod n)^e \bmod n$$

Session Keys

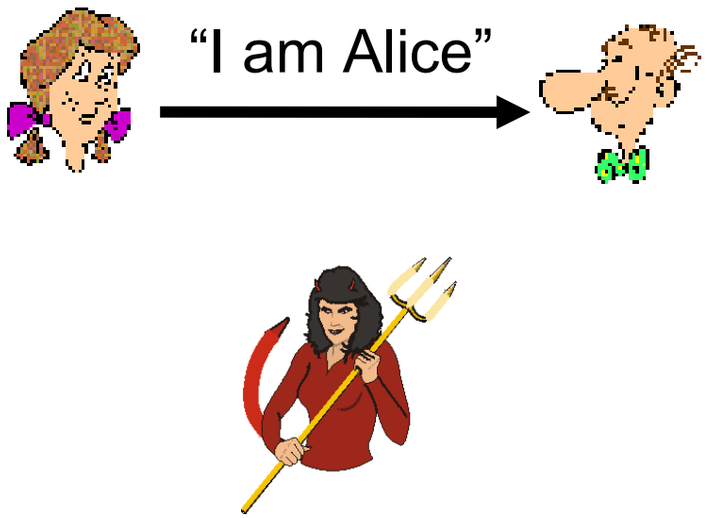
- **Exponentiation required by RSA is time-consuming process**
- **DES and AES can encrypt messages much faster than RSA**
- **So ... don't use RSA to encrypt *entire* communication between sender and receiver**
  - Use RSA to establish a secure connection between sender/receiver
    - The only data exchanged using RSA is a session key
    - The session key is used as the encryption key for one of the faster symmetric key cryptography methods such as DES or AES
  - Remainder of communication between sender and receiver is encrypted using the faster symmetric key cryptography
- **Example:**
  - Bob and Alice use RSA to exchange a symmetric key K_S
  - Once both have K_S, they use symmetric key cryptography

Overview of Network Security

- **What is Network Security?**
- **Principles of Cryptography**
- **Message Integrity, Authentication**
- **Operational Security: Firewalls and IDS**

Authentication

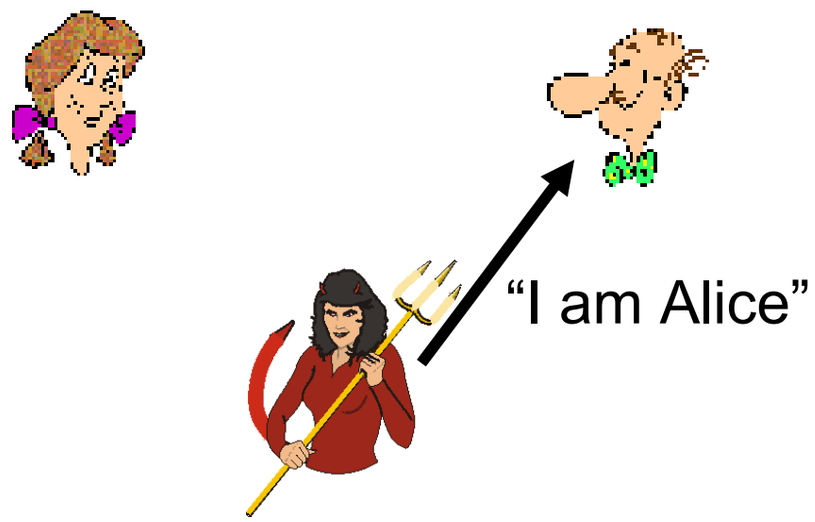
- **Goal: Bob wants Alice to "prove" her identity to him**
- **Authentication Protocol ap1.0: Alice says "I am Alice"**



What is the failure scenario?

Authentication

- **Goal: Bob wants Alice to "prove" her identity to him**
- **Authentication Protocol ap1.0: Alice says "I am Alice"**



In a network, Bob can not "see" Alice, so Trudy simply declares herself to be Alice

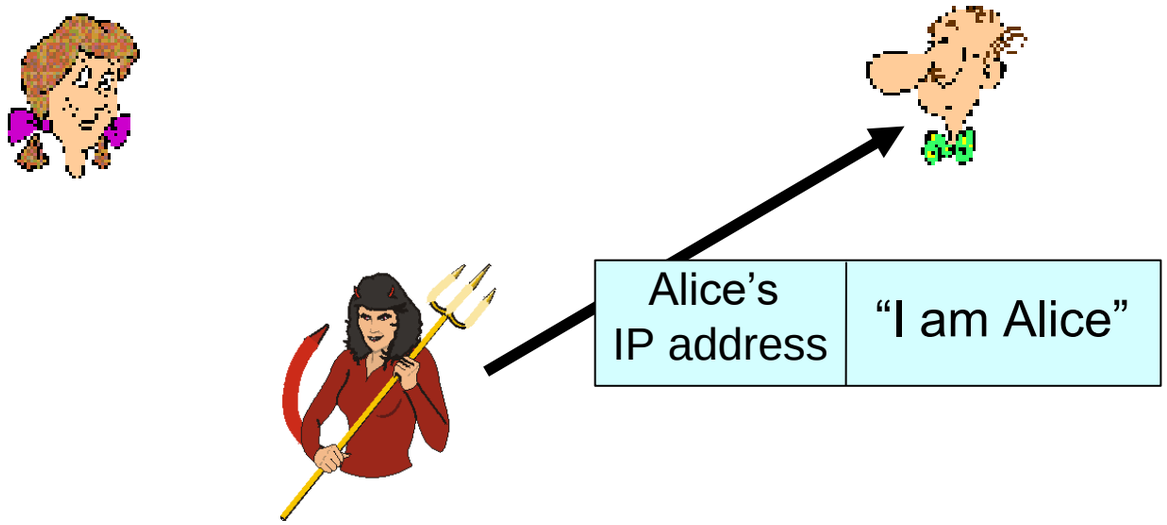
Authentication: Another Try

- **Authentication Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address**
- **Is an IP address enough to authenticate a sender?**

What is the failure scenario?

Authentication: Another Try

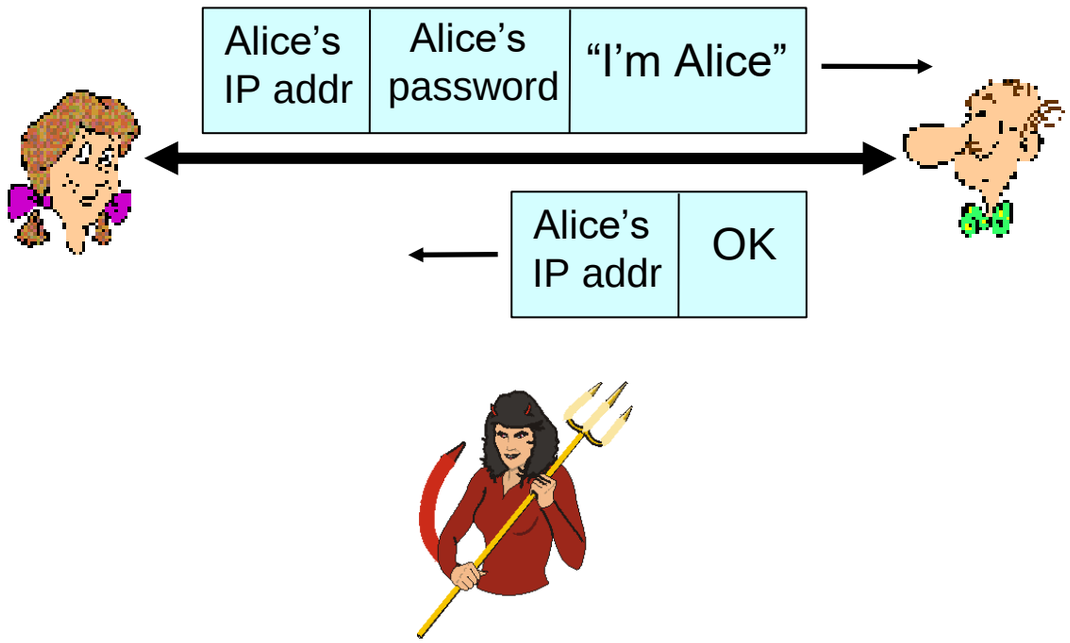
- **Authentication Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address**
- **Is an IP address enough to authenticate a sender? Of course NOT**



Trudy can create
a packet "spoofing"
Alice's address

Authentication: Yet Another Try

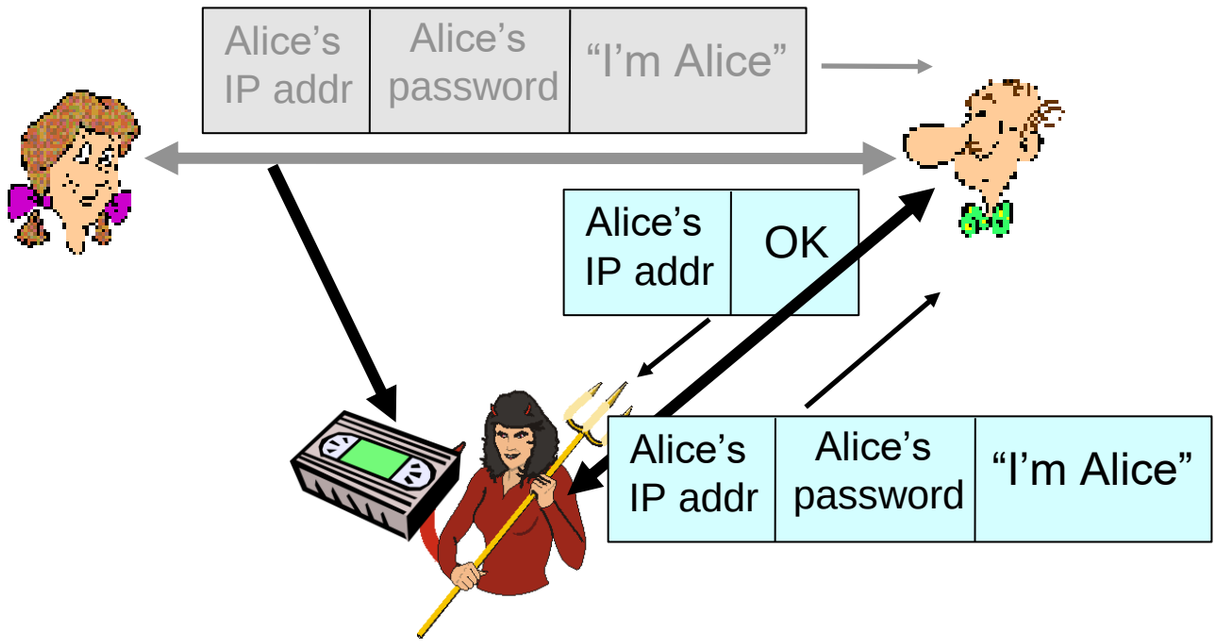
- **Authentication Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.**



What is the failure scenario?

Authentication: Yet Another Try

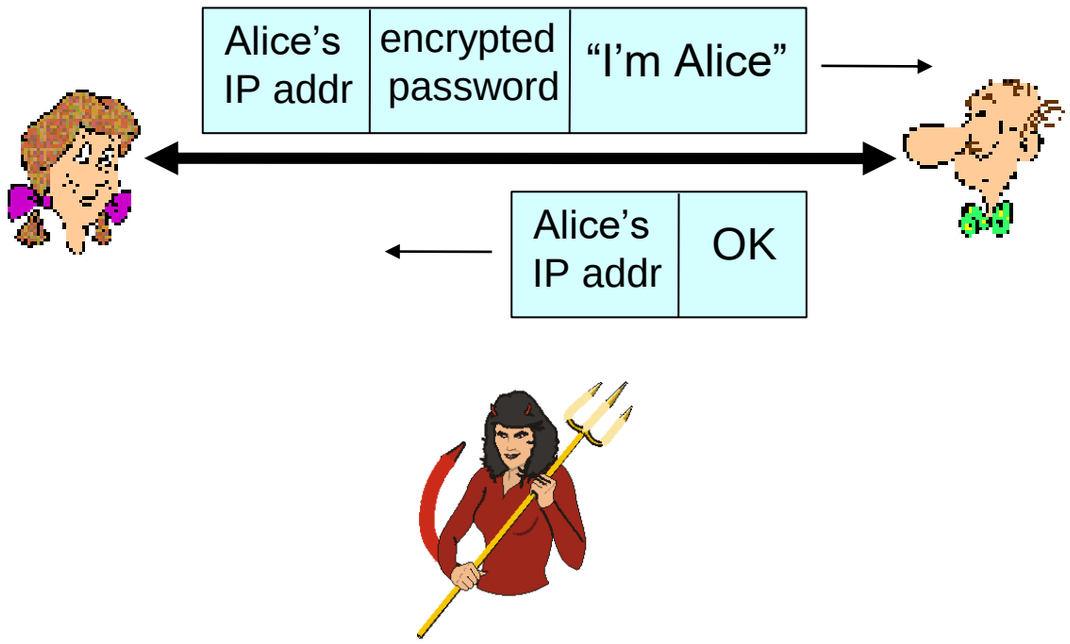
- **Authentication Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.**



Playback attack: Trudy records Alice's packet and later plays it back to Bob

Authentication: Yet Another Try (again)

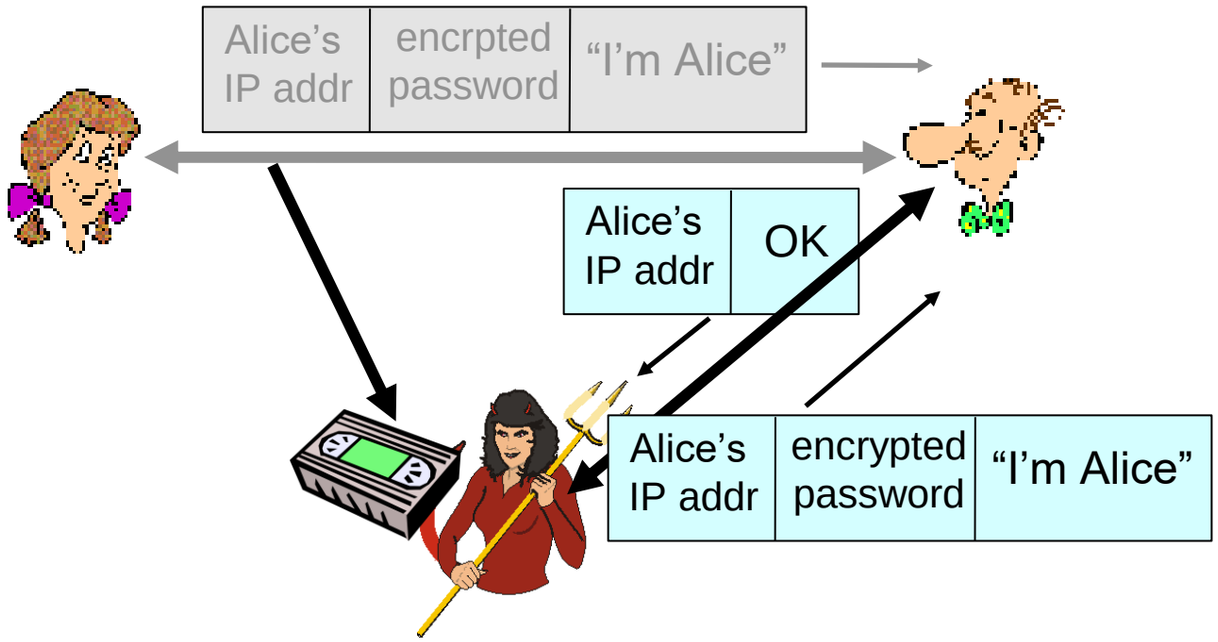
- **Authentication Protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.**



What is the failure scenario?

Authentication: Yet Another Try (again)

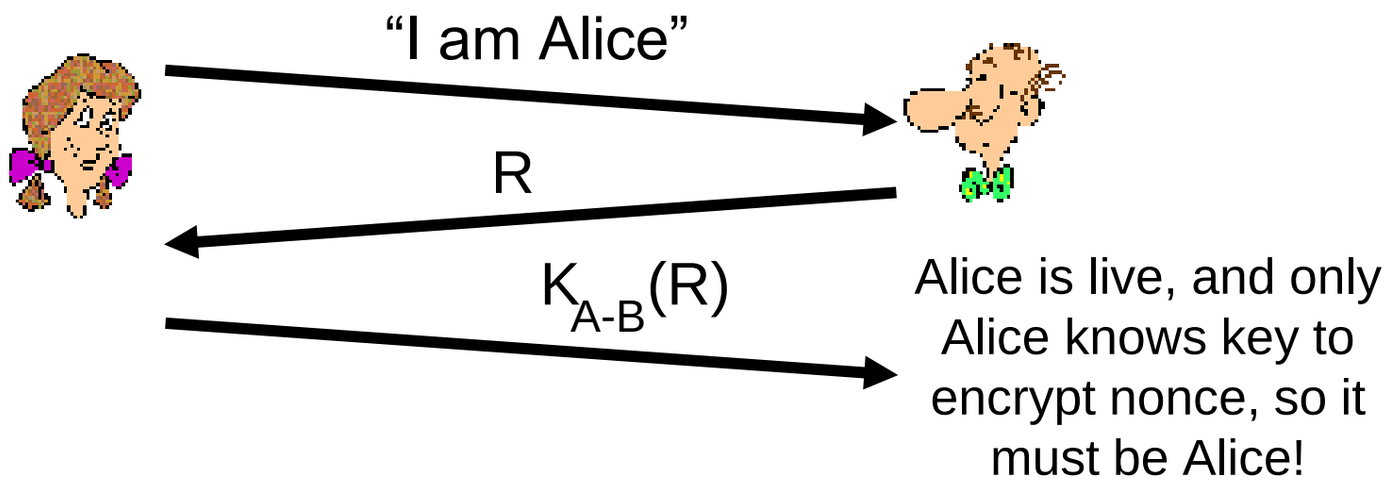
- **Authentication Protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.**



Record and playback still works!

Authentication: Still Trying

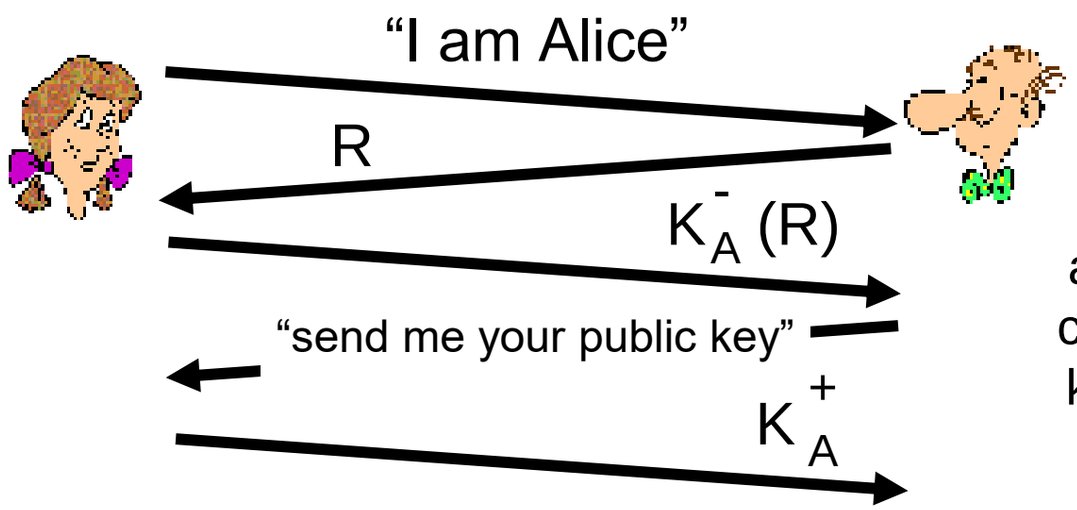
- **Goal: must avoid playback attacks**
- **Utilize a nonce - a number (R) used only once-in-a-lifetime**
- **Authentication Protocol ap4.0: to prove Alice is "live", Bob sends Alice a nonce, R. Alice must return R, encrypted with shared secret key**



Failures, drawbacks?

Authentication: Still Trying, Really

- **Authentication Protocol ap4.0 requires shared symmetric key**
  - Can we authenticate using public key techniques?
- **Authentication Protocol ap5.0: use nonce and public key cryptography**



Bob computes

$K_A^+(K_A^-(R)) = R$

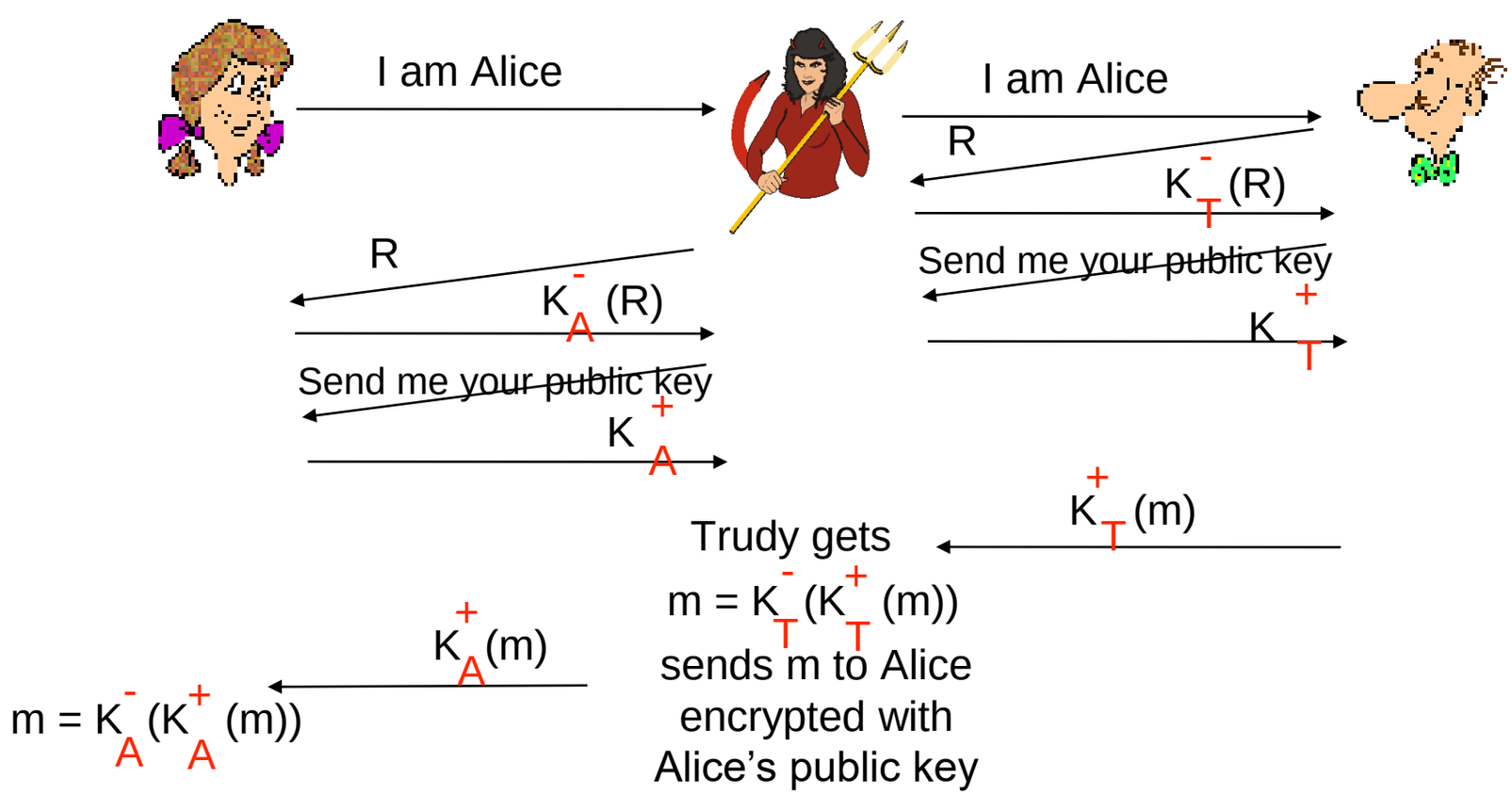
and knows only Alice could have the private key, that encrypted R such that

$K_A^+(K_A^-(R)) = R$

How's this?

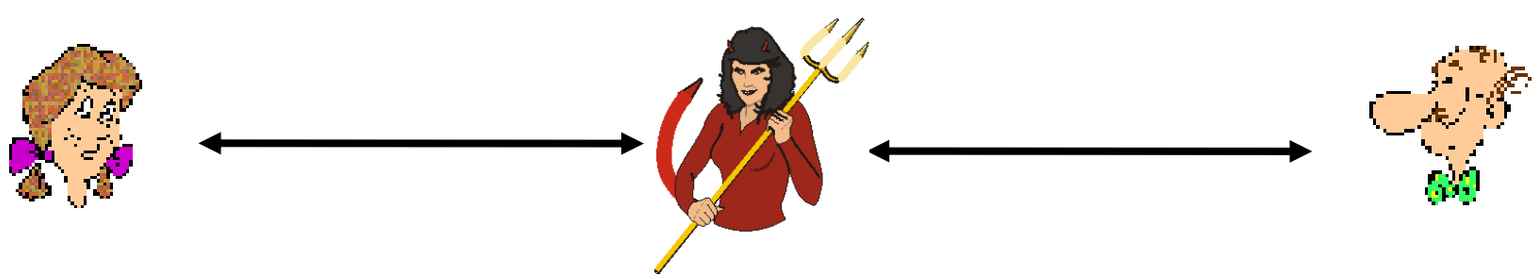
Authentication Protocol ap5.0: Security Hole

- **Man-in-the-middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)**



Authentication Protocol ap5.0: Security Hole

- **Man-in-the-middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)**
  - Difficult to detect:
    - Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation!)
    - Problem is that Trudy receives all messages as well!



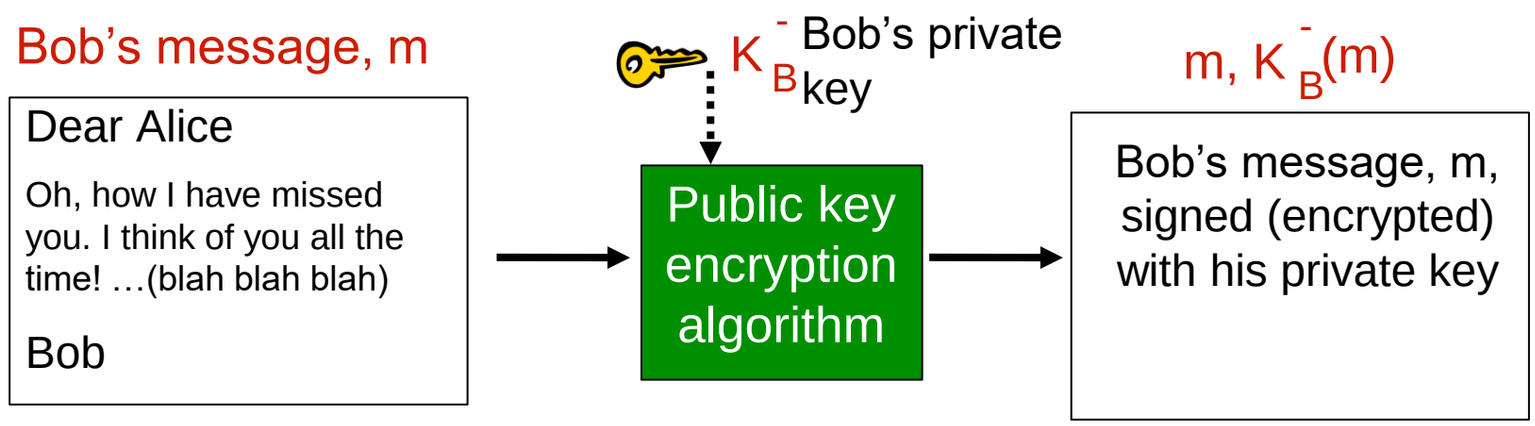
Nobody likes you Trudy

Digital Signatures

- **Cryptographic technique analogous to hand-written signatures**
  - Sender (Bob) digitally signs document, establishing he is document owner/creator
  - Verifiable and non-forgeable
    - Recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

Digital Signatures

- **Simple digital signature for message m**
  - Bob signs m by encrypting with his private key K_B^-, creating "signed" message, $K_B^-(m)$



Digital Signatures

- **Suppose Alice receives message m, with signature: m, $K_B^-(m)$**
- **Alice can verify m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$**
- **If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key**

- **Alice can verify that:**
  - Bob signed m
  - No one else signed m
  - Bob signed m and not m' (i.e. m was not altered)
- **Non-repudiation:**
  - Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m

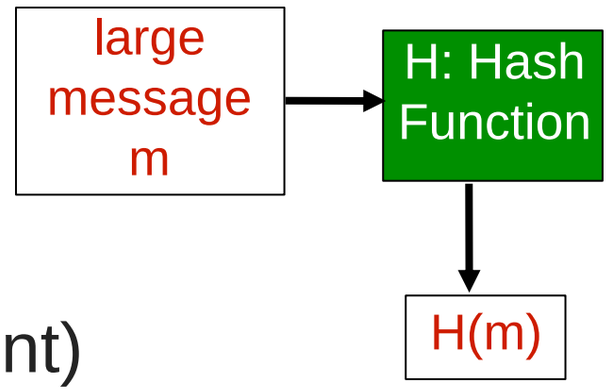
Digital Signatures

- **Digitally signing messages using encryption is computationally expensive**
- **Why not just encrypt a portion of the message to act as a digital signature?**
  - Still need to ensure that content of message hasn't changed
  - Use encrypted message digests as signature

Message Digests

- **Computationally expensive to public-key-encrypt long messages**
- **Goal: fixed-length, easy-to-compute digital "fingerprint"**
  - Apply hash function H to m, get fixed size message digest, H(m)

- **Hash function properties:**
  - Many-to-1
  - Produces fixed-size message digest (fingerprint)
  - Given message digest x, computationally infeasible to find m such that x = H(m)



Internet checksum: poor crypto hash function

Internet checksum has some properties of hash function:

- **produces fixed length digest (16-bit sum) of message**
- **is many-to-one**

But given message with given hash value, it is easy to find another message with same hash value:

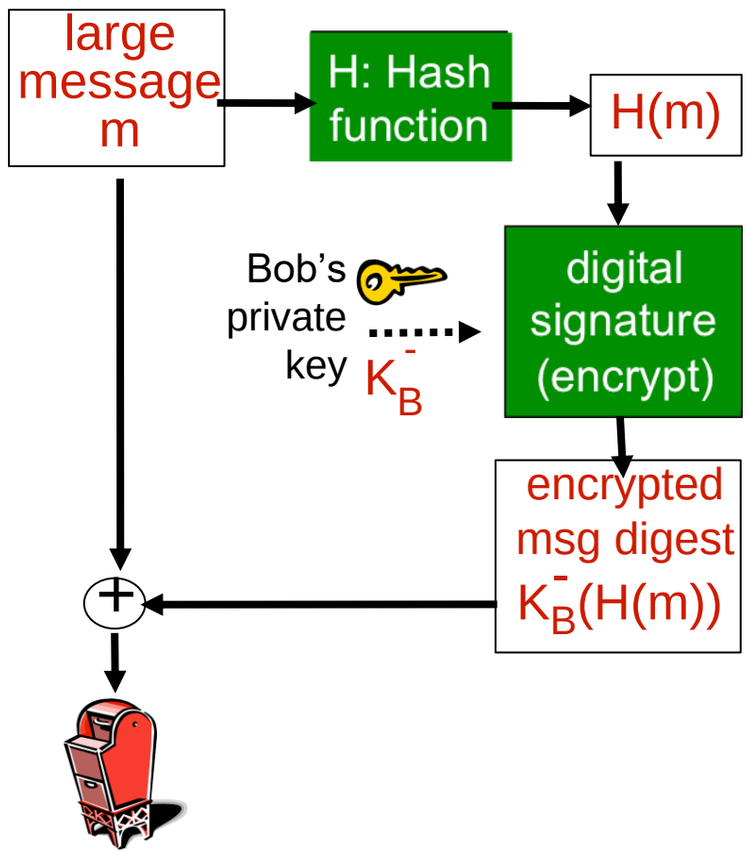
message	ASCII format
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	B2 C1 D2 AC

message	ASCII format
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42
	B2 C1 D2 AC

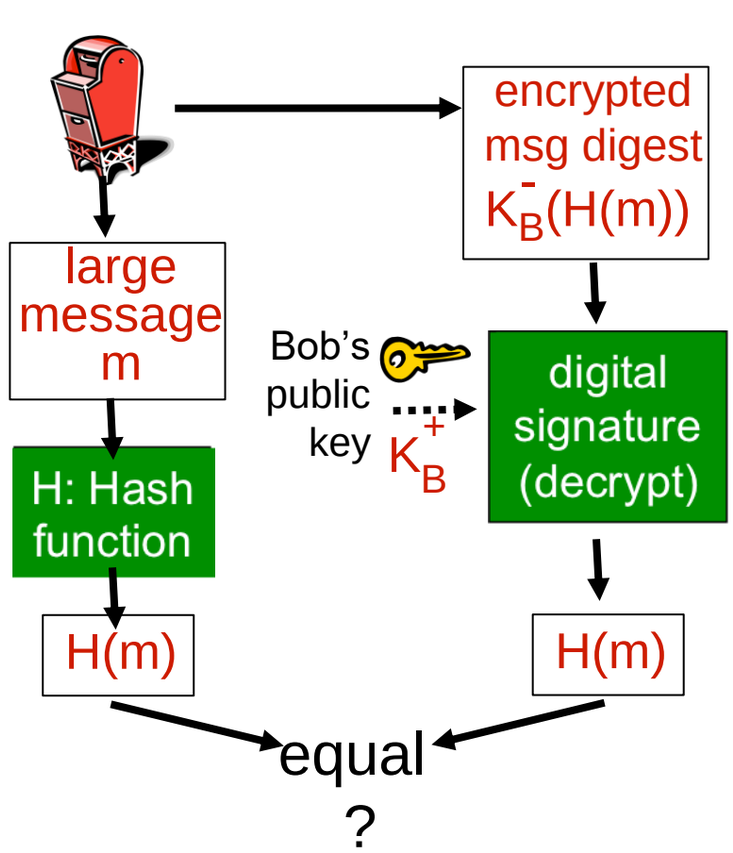
different messages but identical checksums!

Digital Signature = Signed Message Digest

Bob sends digitally signed message



Alice verifies signature, integrity of digitally signed message

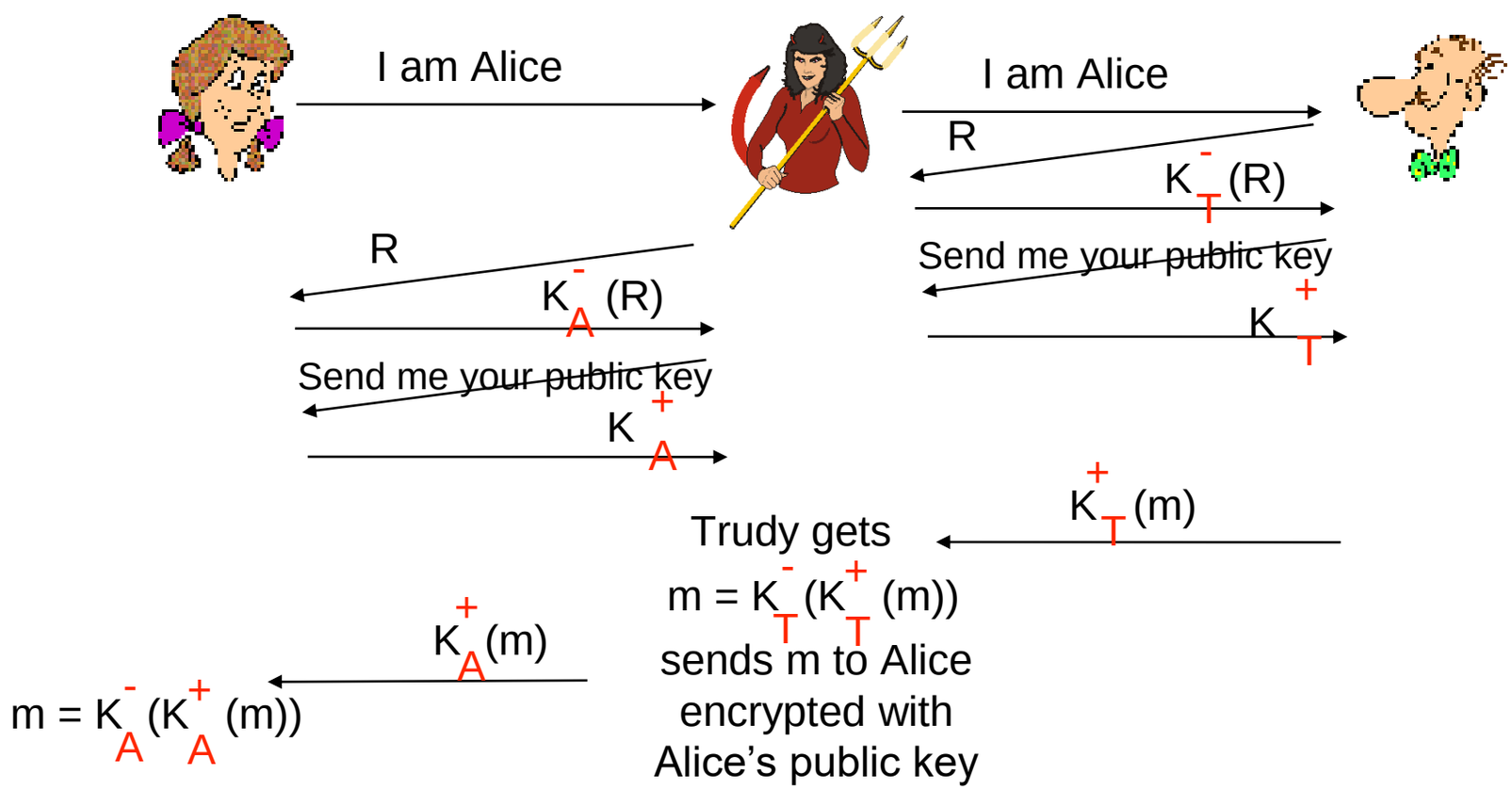


Hash Function Algorithms

- **MD5 hash function widely used**
  - Computes 128-bit message digest in 4-step process.
  - Arbitrary 128-bit string x, appears difficult to construct msg m whose MD5 hash is equal to x
- **SHA-1 is also used**
  - US standard (used by government bodies)
  - 160-bit message digest

Recall: ap5.0 Security Hole

- **Man-in-the-middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)**

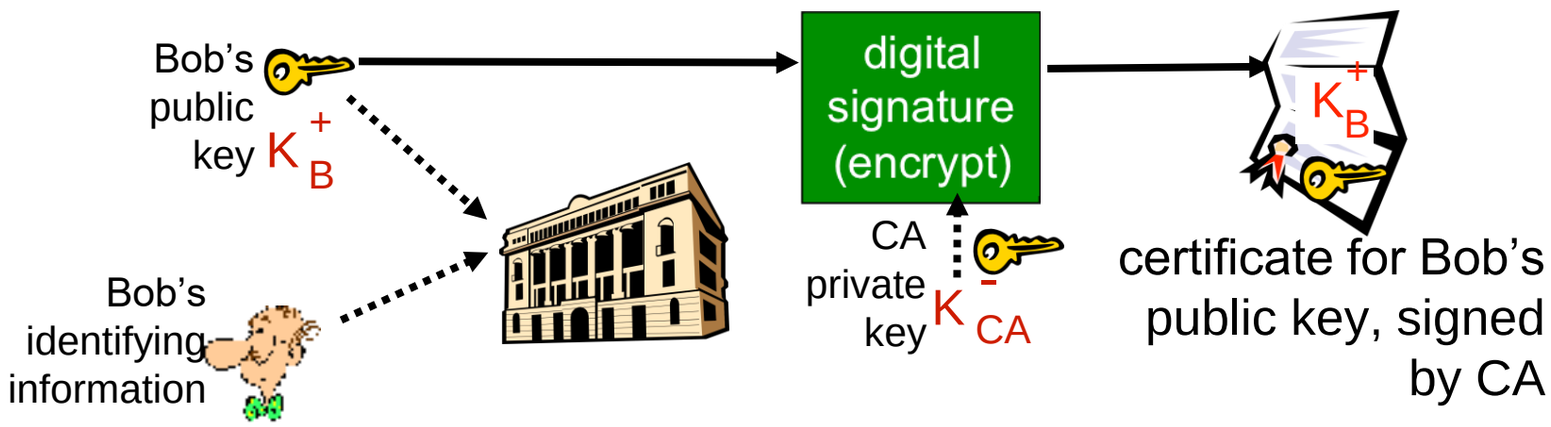


Public-key certification

- **motivation: Trudy plays pizza prank on Bob**
  - Trudy creates e-mail order:
    Dear Pizza Store, Please deliver to me four pepperoni pizzas. Thank you, Bob
  - Trudy signs order with her private key
  - Trudy sends order to Pizza Store
  - Trudy sends to Pizza Store her public key, but says it's Bob's public key
  - Pizza Store verifies signature; then delivers four pepperoni pizzas to Bob
  - Bob doesn't even like pepperoni

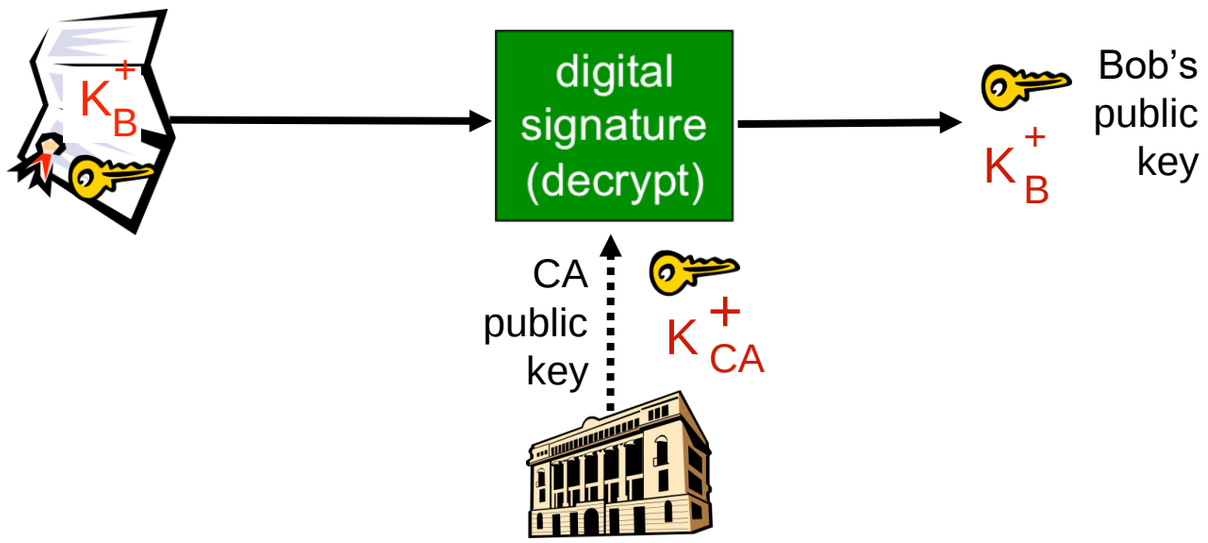
Certification Authorities

- **Certification authority (CA): binds public key to particular entity, E**
- **E (person, router) registers its public key with CA**
  - E provides “proof of identity” to CA
  - CA creates certificate binding E to its public key
  - Certificate containing E’s public key is digitally signed by CA – CA says “this is E’s public key”



Certification Authorities

- **When Alice wants Bob's public key**
  - Gets Bob's certificate (from Bob or elsewhere)
  - Apply CA's public key to Bob's certificate, get Bob's public key

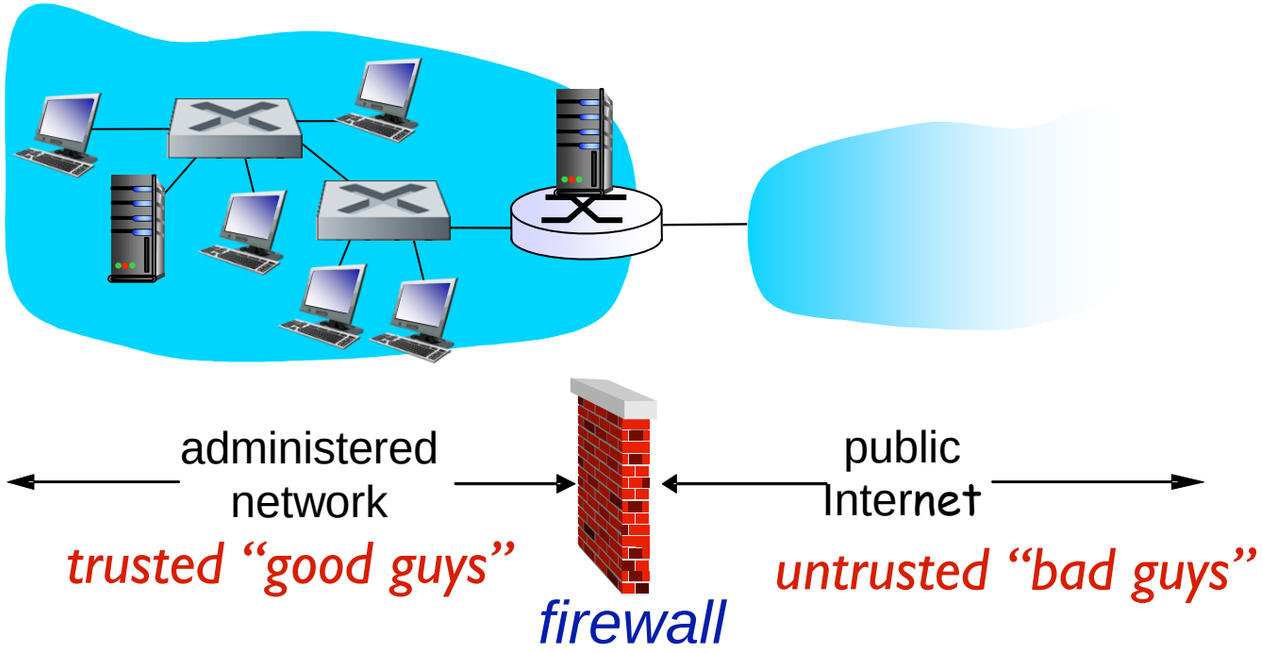


Overview of Network Security

- **What is Network Security?**
- **Principles of Cryptography**
- **Message Integrity, Authentication**
- **Operational Security: Firewalls and IDS**

Firewalls

- **Isolate an organization's internal net from larger Internet, allowing some packets to pass, blocking others**

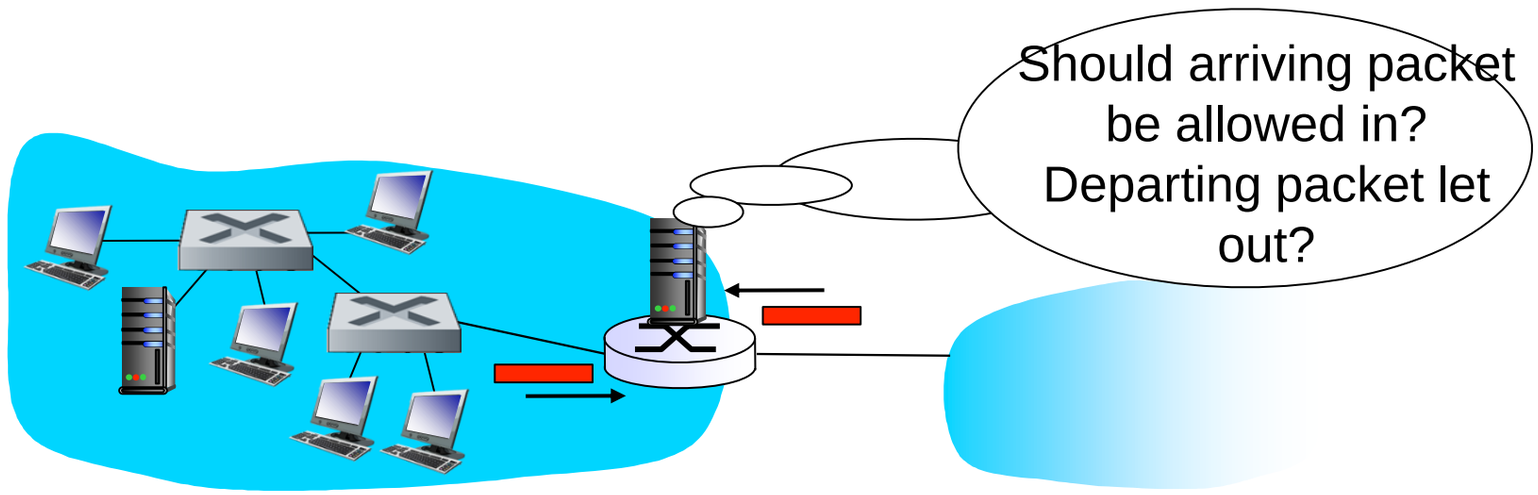


Why Use a Firewall?

- **Prevent denial of service attacks**
  - SYN flooding - attacker establishes many bogus TCP connections, no resources left for “real” connections
- **Prevent illegal modification/access of internal data**
  - For example, attacker replaces CIA’s homepage with something else
- **Allow only authorized access to inside network**
  - Set of authenticated users/hosts
- **Three types of firewalls:**
  - Stateless packet filters
  - Stateful packet filters
  - Application gateways

Stateless Packet Filtering

- **Internal network connected to Internet via router firewall**
- **Router filters packet-by-packet, decision to forward/drop packet based on:**
  - Source IP address, destination IP address
  - Protocol type in IP datagram (i.e. TCP, UDP, ICMP, etc.)
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits



Stateless Packet Filtering: Example

- **Example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23**
  - Result: all incoming, outgoing UDP flows and telnet connections are blocked

- **Example 2: block inbound TCP segments with ACK=0**
  - Result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

Stateless Packet Filtering: More Examples

Policy	Firewall Setting
No outside Web access	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth	Drop all incoming UDP packets - except DNS and router broadcasts
Prevent your network from being used for a smurf DoS attack	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255)
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists

- **ACL: table of rules, applied top to bottom to incoming packets: (action, condition) pairs**

Action	Source Address	Destination Address	Protocol	Source Port	Destination Port	Flag Bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateful Packet Filtering

- **Stateless packet filter: heavy handed tool**
  - Admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

Action	Source Address	Destination Address	Protocol	Source Port	Destination Port	Flag Bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- **Stateful packet filter: tracks status of every TCP connection**
  - Track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets “makes sense”
  - Timeout inactive connections at firewall: no longer admit packets

Stateful Packet Filtering: ACL

- **ACL augmented to indicate need to check connection state table before admitting packet**

Action	Source Address	Destination Address	Protocol	Source Port	Destination Port	Flag Bit	Check Connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Stateful Packet Filtering: Connection Table

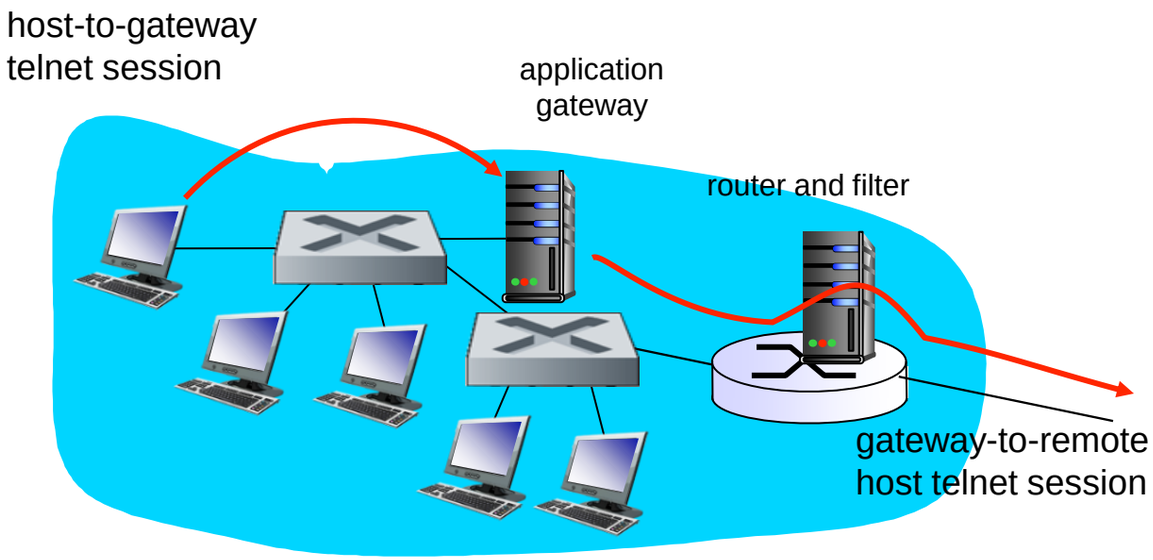
- **Given the following connection table and the previous ACL**

Source Address	Destination Address	Source Port	Destination Port
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

- **Allow a packet from 37.96.87.123, port 80 to 222.22.1.7, port 12699**
- **Block a packet from 12.1.18.83, port 80 to 222.22.1.7, port 12699**
  - According to the connection table, no connection has been established

Application Gateways

- **Filters packets on application data as well as on IP/TCP/UDP fields**
- **Example, allow select internal users to telnet out of the network**
  - Require all telnet users to telnet through gateway
  - For authorized users:
    - Gateway sets up telnet connection to destination host
    - Gateway relays data between 2 connections
  - Router filter blocks all telnet connections not originating from gateway



Limitations of Firewalls, Gateways

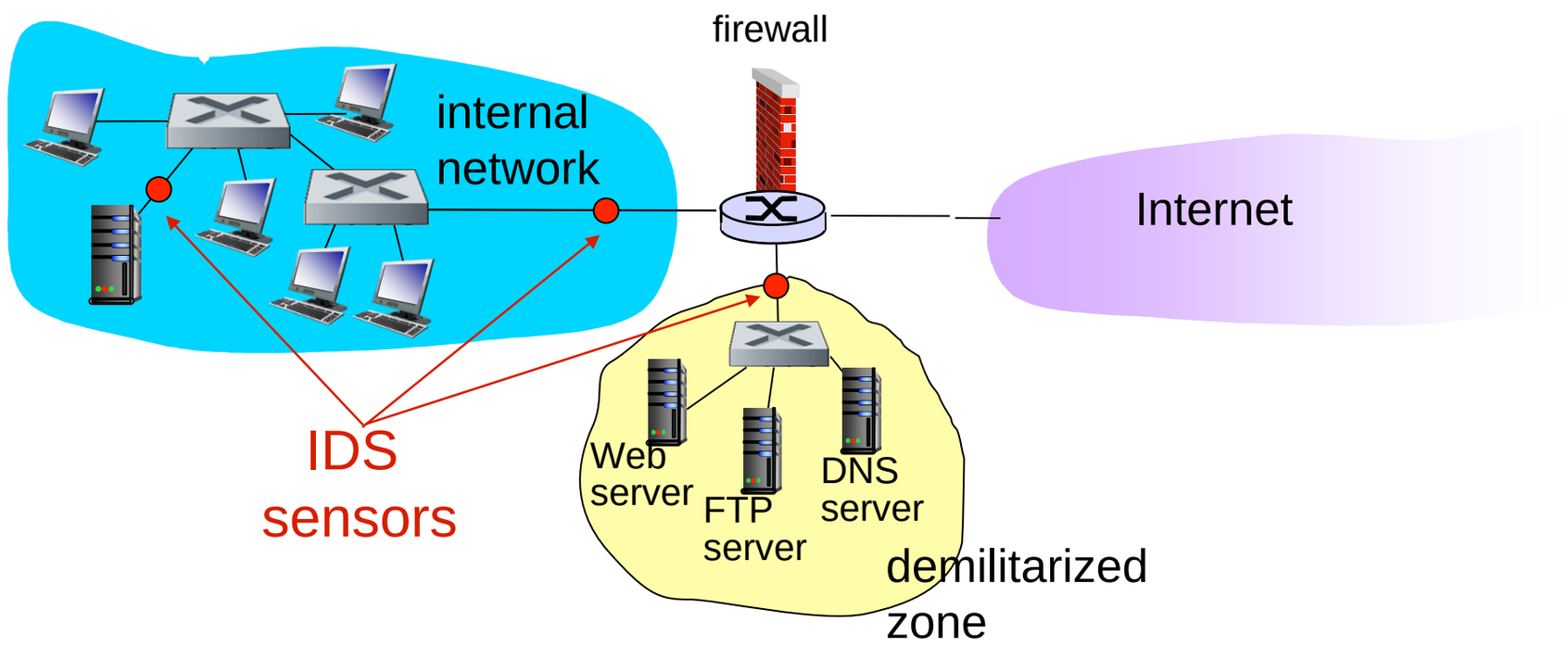
- **IP spoofing - router can't know if data "really" comes from claimed source**
- **If multiple applications need special treatment, each has own application gateway**
- **Client software must know how to contact the application gateway**
  - e.g. must set IP address of proxy in Web browser
- **Filters often use all or nothing policy for UDP**
- **Tradeoff between communication with outside world and security**
- **Many highly protected sites still suffer from attacks**

Intrusion Detection Systems

- **Packet filtering**
  - Operates on TCP/IP headers only
  - No correlation check among sessions
- **IDS: Intrusion Detection System**
  - Perform deep packet inspection - look at packet contents (e.g. check character strings in packet against database of known virus, attack strings)
  - Examine correlation among multiple packets
    - Port scanning
    - Network mapping
    - DoS attack

Intrusion Detection Systems

- **Multiple IDSs perform different types of checking at different locations**
  - Distribute work load of IDS throughout network
  - IDS may potentially need to scan thousands of signatures that represent known network attacks or viruses



Intrusion Detection Systems

- **Example of an IDS rule:**

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any
  (msg: "ICMP PING NMAP"; dsize: 0; itype:8;)
```

  - Raise an alert for an ICMP packet from any external IP address to any internal IP address that is of ICMP type 8, and has an empty payload
  - Send the alert message, "ICMP PING NMAP"

Network Security (summary)

basic techniques……

- cryptography (symmetric and public)
- message integrity
- end-point authentication

…. used in many different security scenarios

- secure email
- secure transport (SSL)
- IP sec
- 802.11

operational security: firewalls and IDS