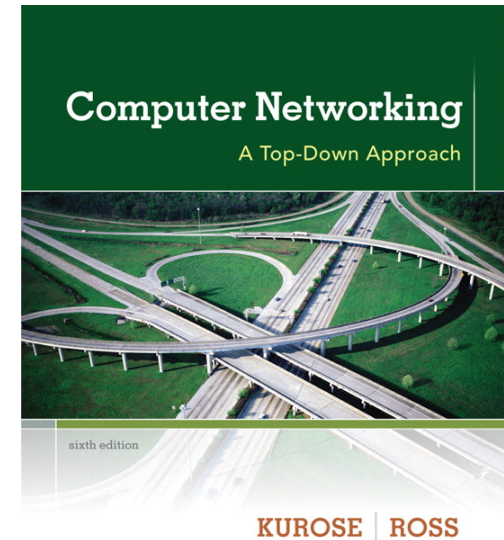


Chapter 8

Security



Computer
Networking: A Top
Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

All material copyright 1996-2012
J.F Kurose and K.W. Ross, All Rights Reserved

Chapter 8: Network Security

Chapter goals:

- ❖ understand principles of network security:
 - cryptography and its *many* uses beyond “confidentiality”
 - authentication
 - message integrity
- ❖ security in practice:
 - firewalls and intrusion detection systems
 - security in application, transport, network, link layers

Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity, authentication

8.4 ~~Securing e-mail~~

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 ~~Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

What is network security?

confidentiality: only sender, intended receiver should “understand” message contents

- sender encrypts message
- receiver decrypts message

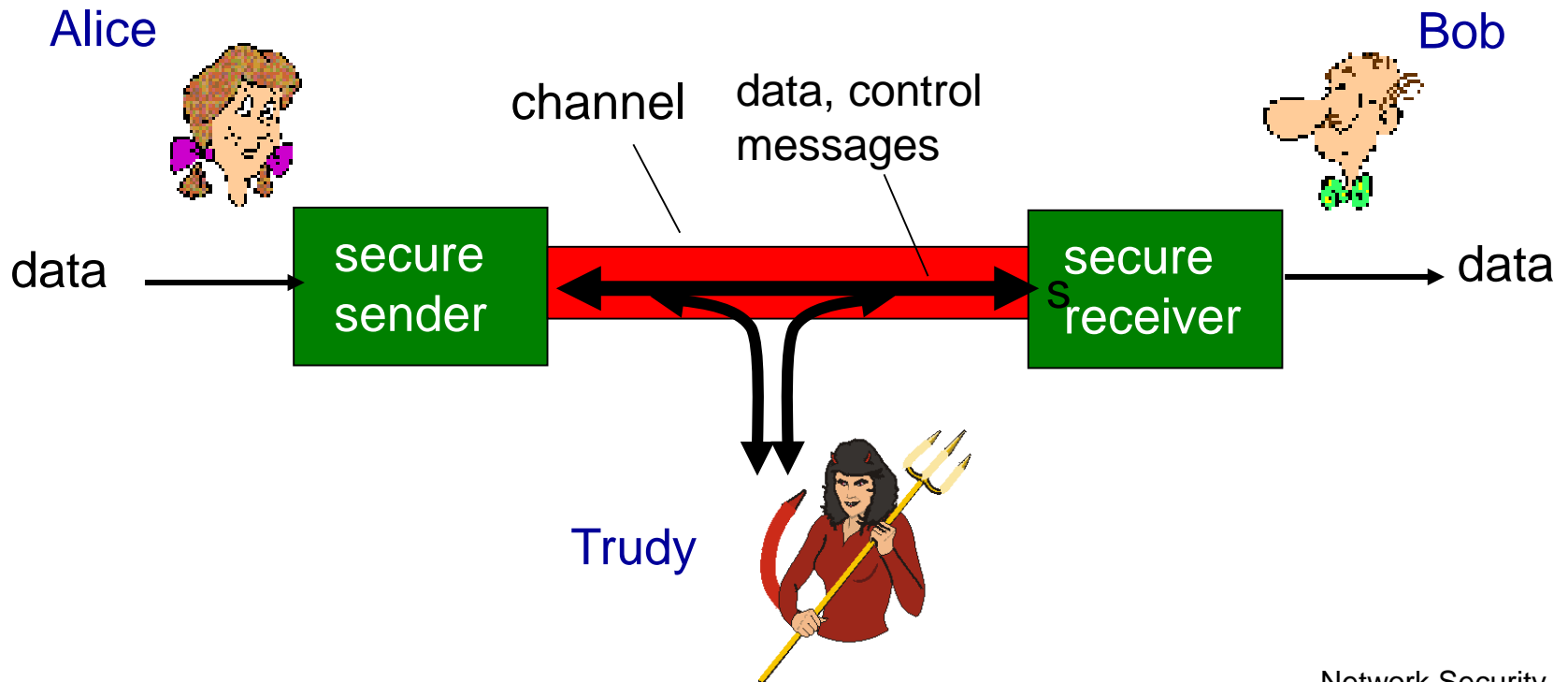
authentication: sender, receiver want to confirm identity of each other

message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

access and availability: services must be accessible and available to users

Friends and enemies: Alice, Bob, Trudy

- ❖ well-known in network security world
- ❖ Bob, Alice (lovers!) want to communicate “securely”
- ❖ Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- ❖ ... well, *real-life* Bobs and Alices!
- ❖ Web browser/server for electronic transactions (e.g., on-line purchases)
- ❖ on-line banking client/server
- ❖ DNS servers
- ❖ routers exchanging routing table updates
- ❖ other examples?

There are bad guys (and girls) out there!

Q: What can a “bad guy” do?

A: A lot! See section 1.6

- *eavesdrop*: intercept messages
- actively *insert* messages into connection
- *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)

Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity, authentication

8.4 Securing e-mail

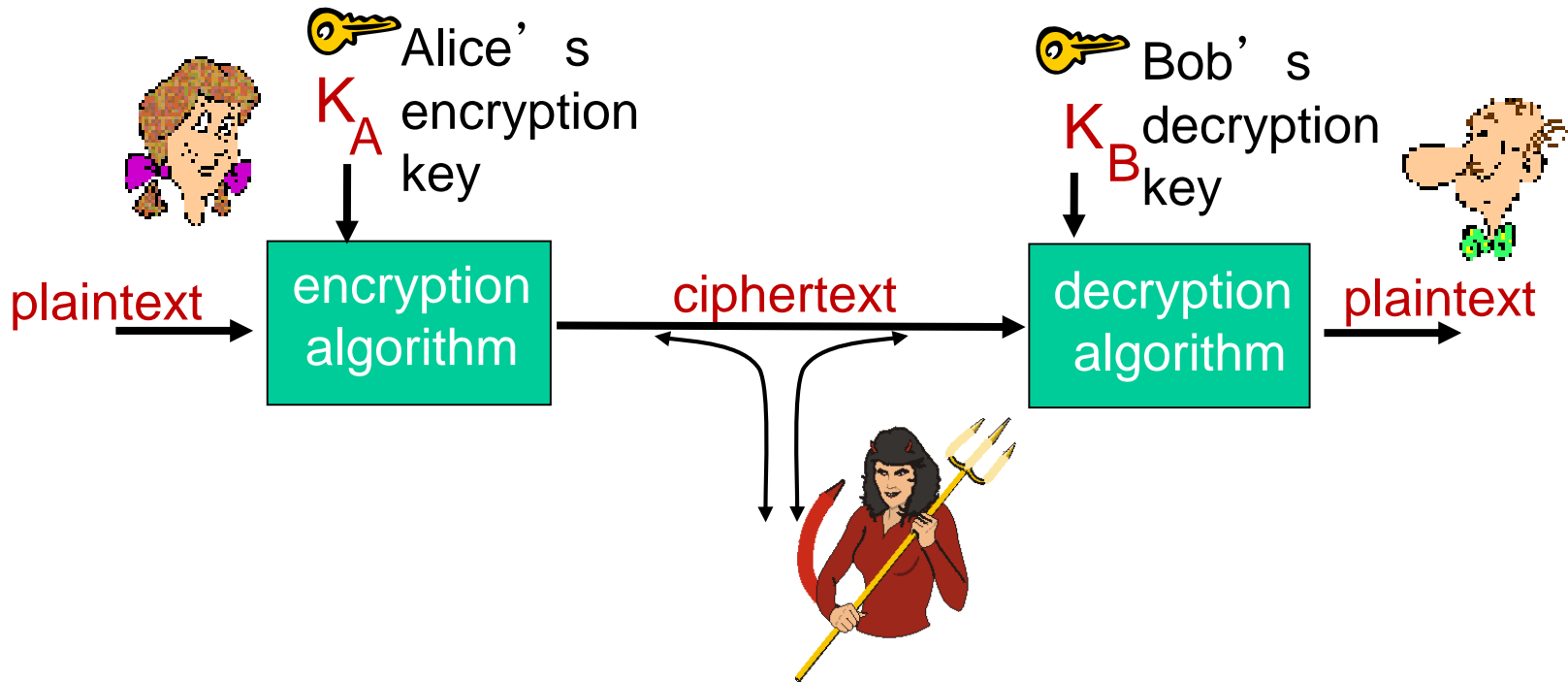
8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

The language of cryptography



m plaintext message

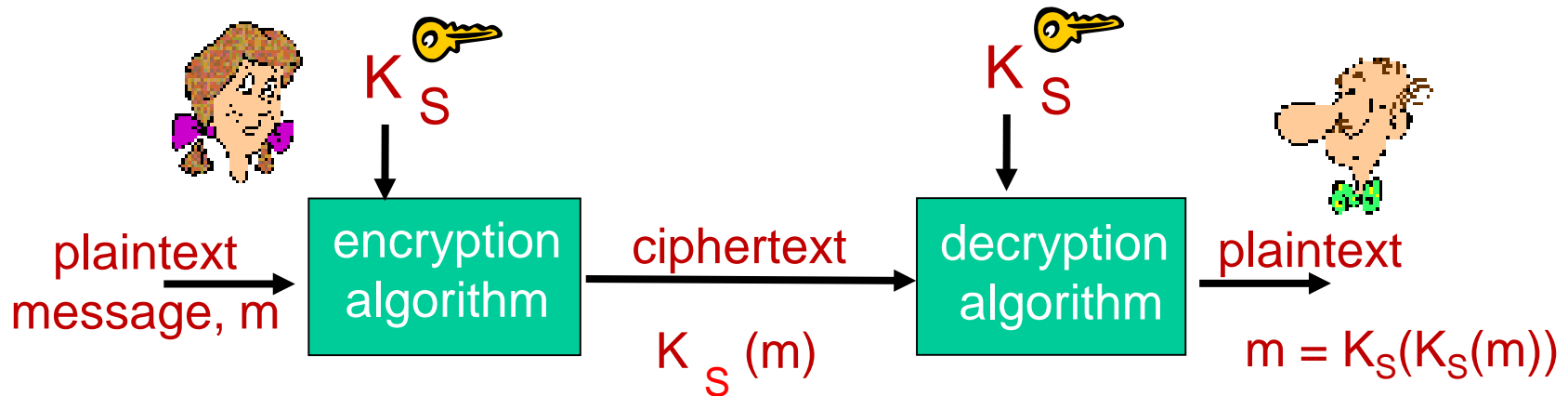
$K_A(m)$ ciphertext, encrypted with key K_A

$m = K_B(K_A(m))$

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 crypto: Bob and Alice share same (symmetric) key: K_S

- ❖ e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

Q: how do Bob and Alice agree on key value?

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	i	o	u	y	t	r	e	w	q

e.g.: Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc

 *Encryption key*: mapping from set of 26 letters
to set of 26 letters

Symmetric key crypto: DES

DES: Data Encryption Standard

- ❖ US encryption standard [NIST 1993]
- ❖ 56-bit symmetric key, 64-bit plaintext input
- ❖ block cipher with cipher block chaining
- ❖ how secure is DES?
 - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
 - no known good analytic attack
- ❖ making DES more secure:
 - 3DES: encrypt 3 times with 3 different keys

AES: Advanced Encryption Standard

- ❖ symmetric-key NIST standard, replaced DES (Nov 2001)
- ❖ processes data in 128 bit blocks
- ❖ 128, 192, or 256 bit keys
- ❖ brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Public Key Cryptography



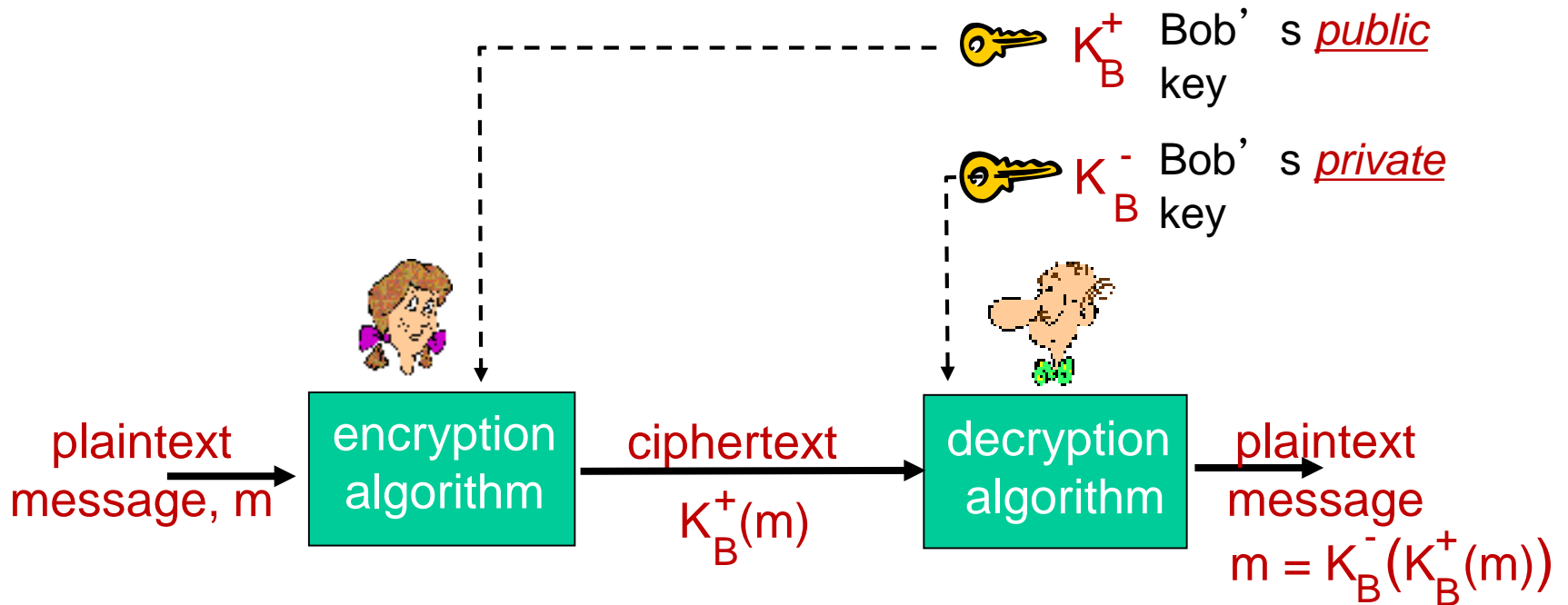
symmetric key crypto

- ❖ requires sender, receiver know shared secret key
- ❖ Q: how to agree on key in first place (particularly if never “met”)?

public key crypto

- ❖ radically different approach [Diffie-Hellman76, RSA78]
- ❖ sender, receiver do *not* share secret key
- ❖ *public* encryption key known to *all*
- ❖ *private* decryption key known only to receiver

Public key cryptography



Public key encryption algorithms

requirements:

① need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

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

② given public key K_B^+ , it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

use public key first,
followed by
private key

use private key
first, followed by
public key

result is the same!

RSA in practice: session keys

- ❖ exponentiation in RSA is computationally intensive
- ❖ DES is at least 100 times faster than RSA
- ❖ use public key crypto to establish secure connection, then establish second key – symmetric session key – for encrypting data

session key, K_S

- ❖ Bob and Alice use RSA to exchange a symmetric key K_S
- ❖ once both have K_S , they use symmetric key cryptography

Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 *Message integrity*, authentication

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

Digital signatures

cryptographic technique analogous to hand-written signatures:

- ❖ sender (Bob) digitally signs document, establishing he is document owner/creator.
- ❖ *verifiable, nonforgeable*: 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, $K_B^-(m)$

Bob's message, m

Dear Alice
Oh, how I have missed you. I think of you all the time! ... (blah blah blah)
Bob

 K_B^- Bob's private key

Public key encryption algorithm

$m, K_B^-(m)$

Bob's message, m , signed (encrypted) with his private key

Digital signatures

- ❖ suppose Alice receives msg m , with signature: $m, K_B^-(m)$
- ❖ Alice verifies 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 thus verifies that:

- ✓ Bob signed m
- ✓ no one else signed m
- ✓ Bob signed m and not m'

non-repudiation:

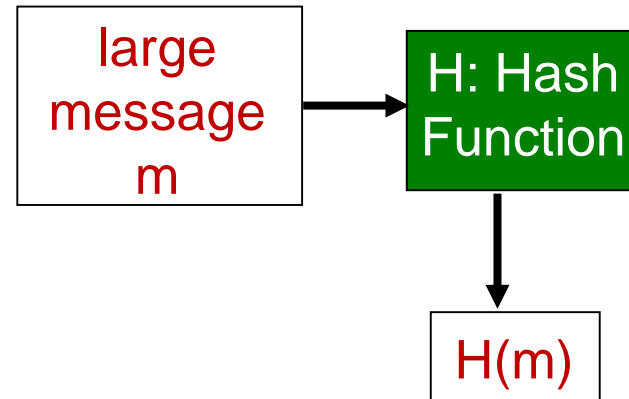
- ✓ Alice can take m , and signature $K_B^-(m)$ to court and prove that Bob signed m

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, $H(m)$.

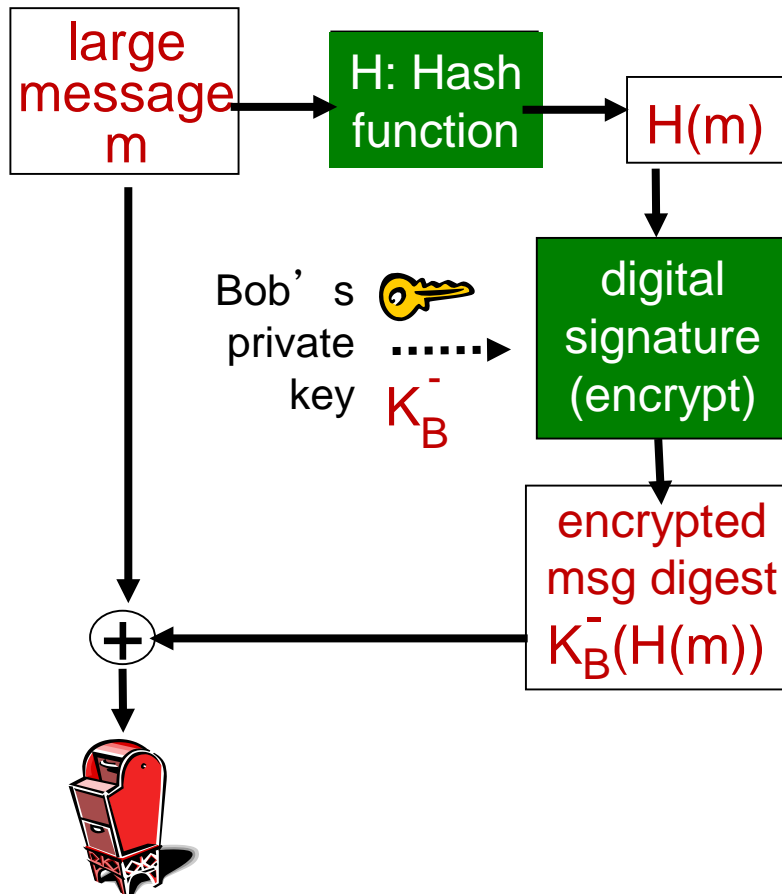


Hash function properties:

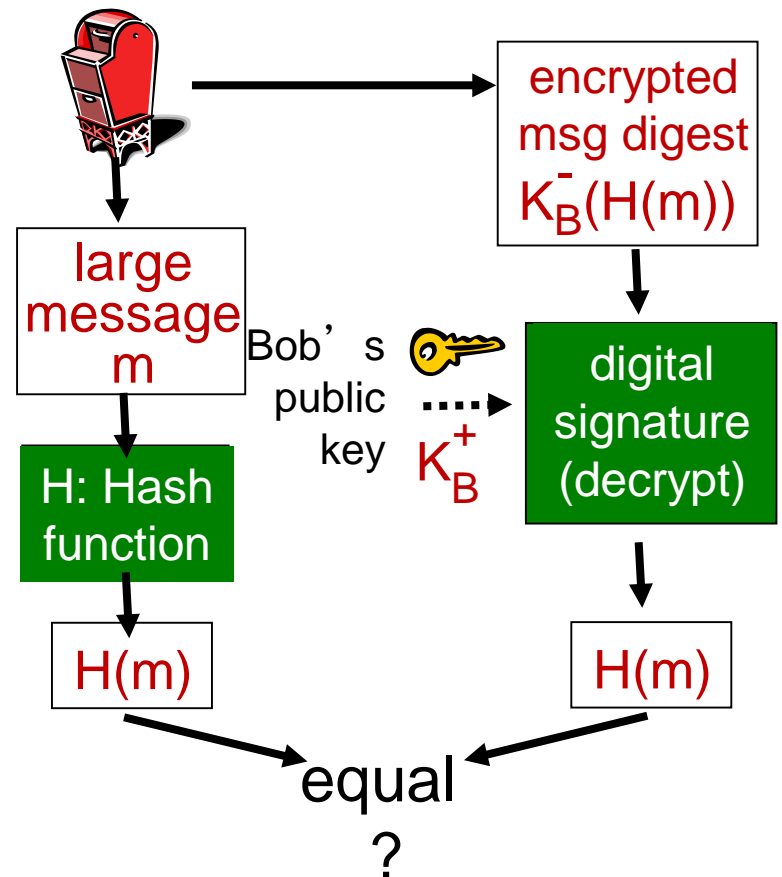
- ❖ many-to-1
- ❖ produces fixed-size msg digest (fingerprint)
- ❖ given message digest x , computationally infeasible to find m such that $x = H(m)$

Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:



Hash function algorithms

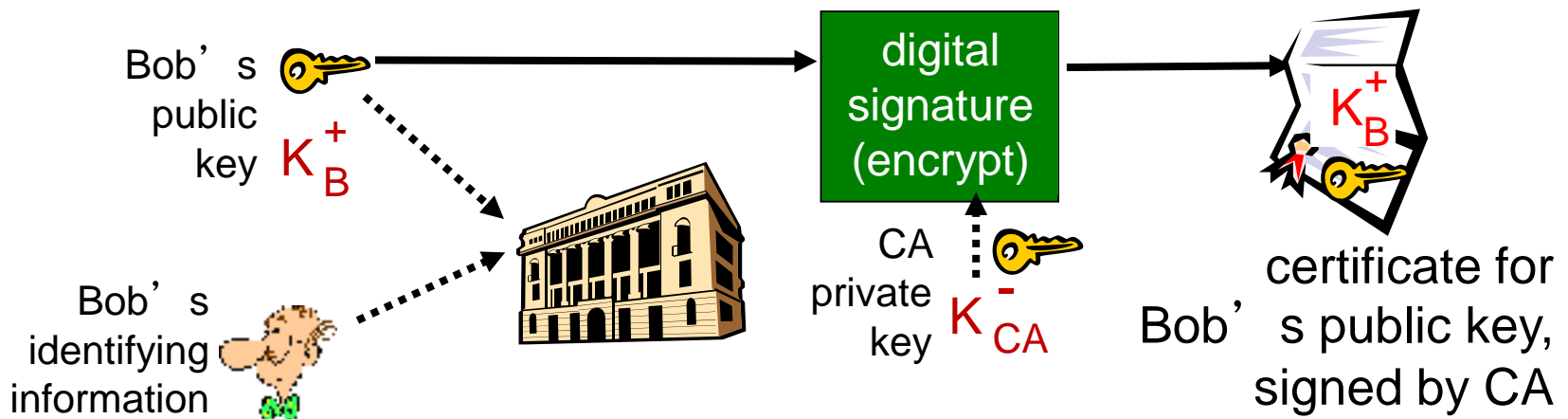
- ❖ **MD5 hash function widely used (RFC 1321)**
 - computes 128-bit message digest in 4-step process.
 - arbitrary 128-bit string x , appears difficult to construct msg m whose MD5 hash is equal to x
- ❖ **SHA-1 is also used**
 - US standard [NIST, FIPS PUB 180-1]
 - 160-bit message digest

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

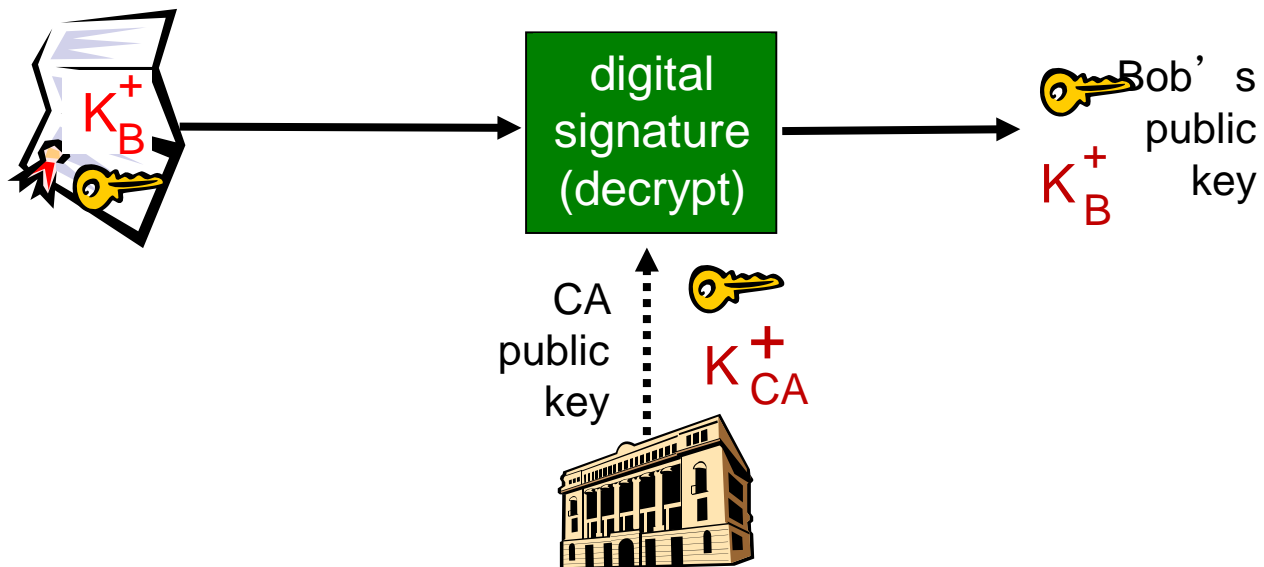
Certification authorities

- ❖ *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 digitally signed by CA – CA says “this is E’ s public key”

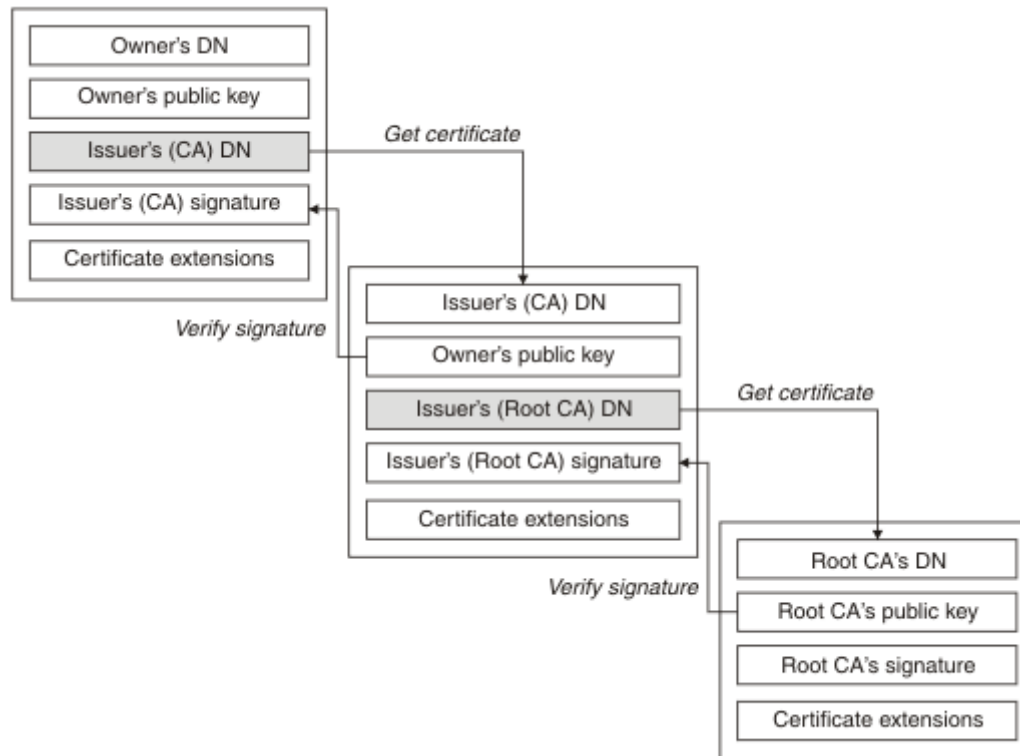


Certification authorities

- ❖ when Alice wants Bob's public key:
 - gets Bob's certificate (Bob or elsewhere).
 - apply CA's public key to Bob's certificate, get Bob's public key



Certificate chains



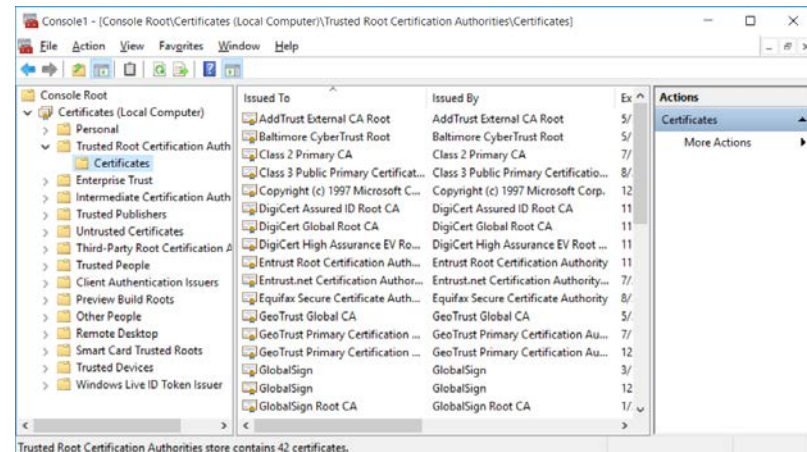
This server uses an deprecated hash algorithm SHA-1

https://www-01.ibm.com/support/knowledgecenter/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10600_.htm

Root certificates

- ❖ Certificates are verified using a chain of trust.
- ❖ The trust anchor is called a root certificate authority (CA)
- ❖ Root certificates are usually self signed
- ❖ Standard X.509
- ❖ Windows (and other operation systems) comes with a lot of root certificates installed.

- ❖ How to see your installed root certificates:
- ❖ Microsoft Management Console
 - Mmc from the command line
- ❖ Add/Remove Snap-in
 - Computer Account: finish + OK
- ❖ Open Trusted Root Certification Authorities
- ❖ Source
 - <http://www.jayway.com/2014/09/03/creating-self-signed-certificates-with-makecert-exe-for-development/>



Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

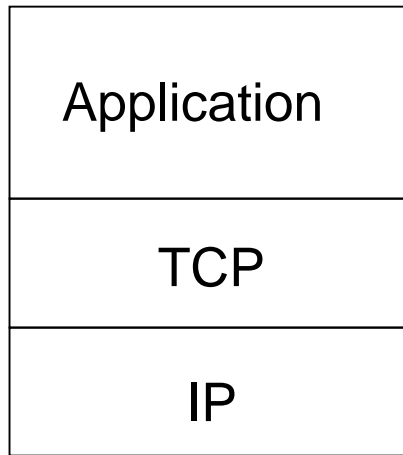
8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

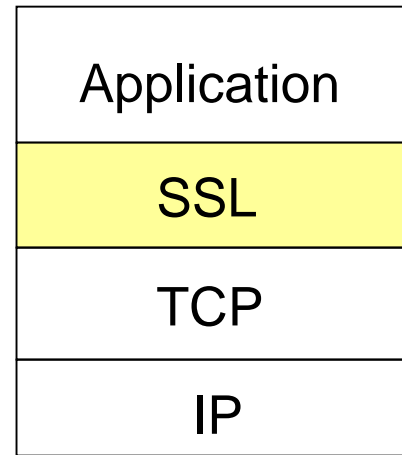
SSL: Secure Sockets Layer

- ❖ widely deployed security protocol
 - supported by almost all browsers, web servers
 - https
 - billions \$/year over SSL
 - TCP port 443
- ❖ mechanisms: [Woo 1994], implementation: Netscape
- ❖ Variation (newer) TLS: transport layer security, RFC 2246
- ❖ provides
 - *confidentiality*
 - *integrity*
 - *authentication*
- ❖ original goals:
 - Web e-commerce transactions
 - encryption (especially credit-card numbers)
 - Web-server authentication
 - optional client authentication
 - minimum hassle in doing business with new merchant
- ❖ available to all TCP applications
 - secure socket interface

SSL and TCP/IP



normal application



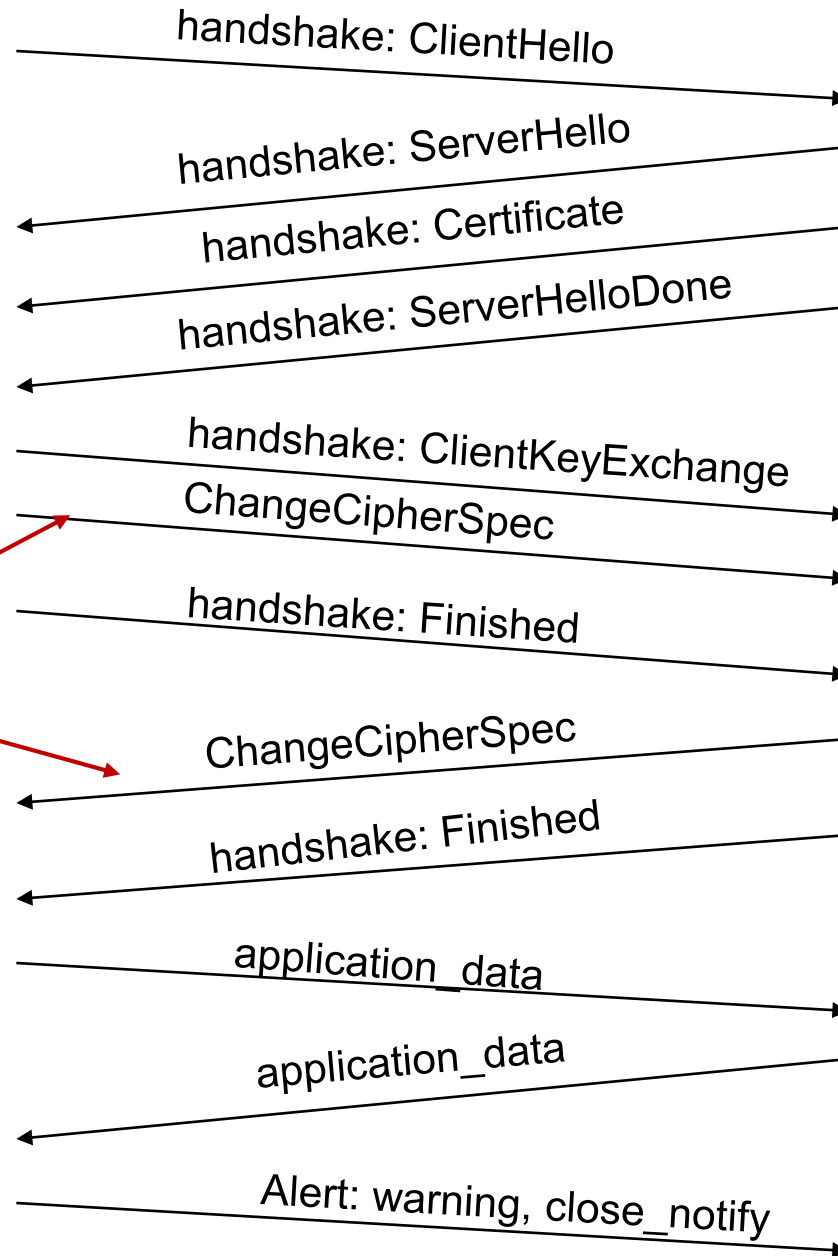
application with SSL

- ❖ SSL provides application programming interface (API) to applications
- ❖ C# and Java SSL libraries/classes readily available

SSL connection



*everything
henceforth
is encrypted*

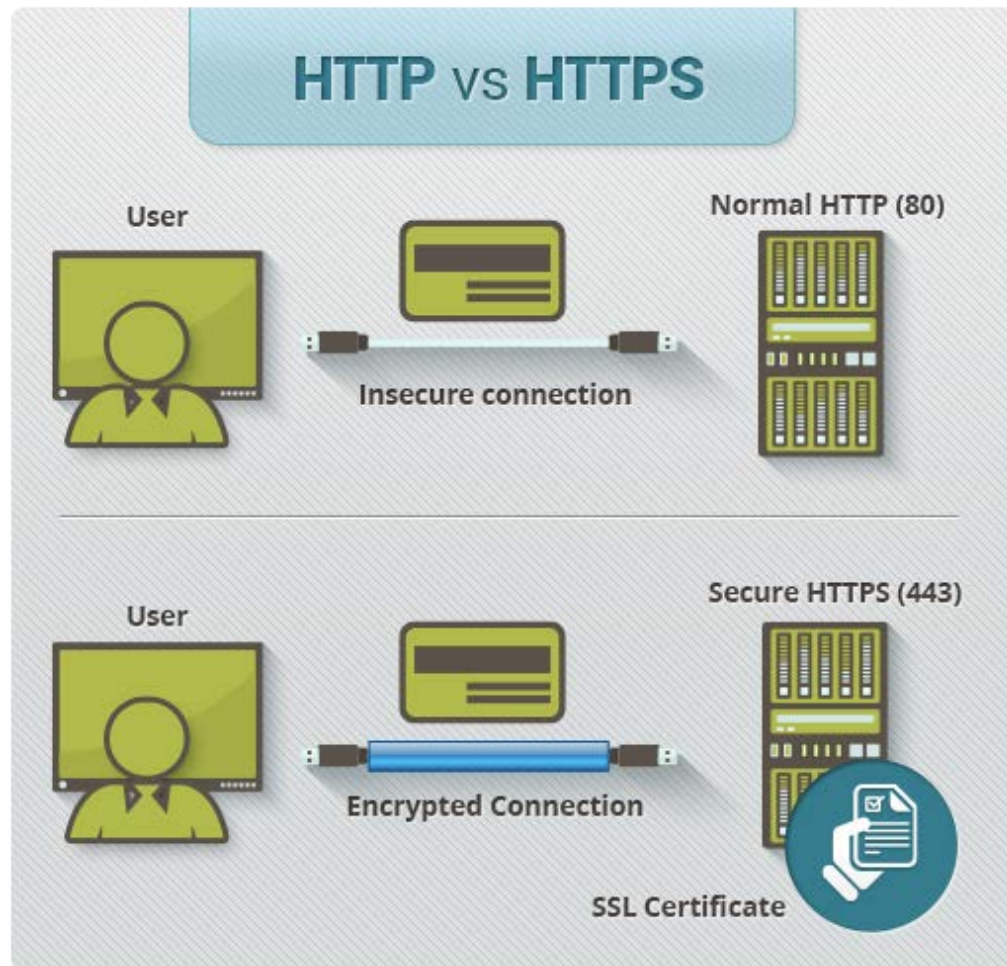


TCP FIN follows

C# API, SslStream

- ❖ The class SslStream wraps an ordinary Stream providing secure communication
- ❖ Example
 - SimpleSslClient, to access a secure web site
 - EchoTcpSecure, client + server

HTTP secure



HTTP secure



<https://www.instantssl.com/ssl-certificate-products/https.html>

Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

What is network-layer confidentiality ?

between two network entities:

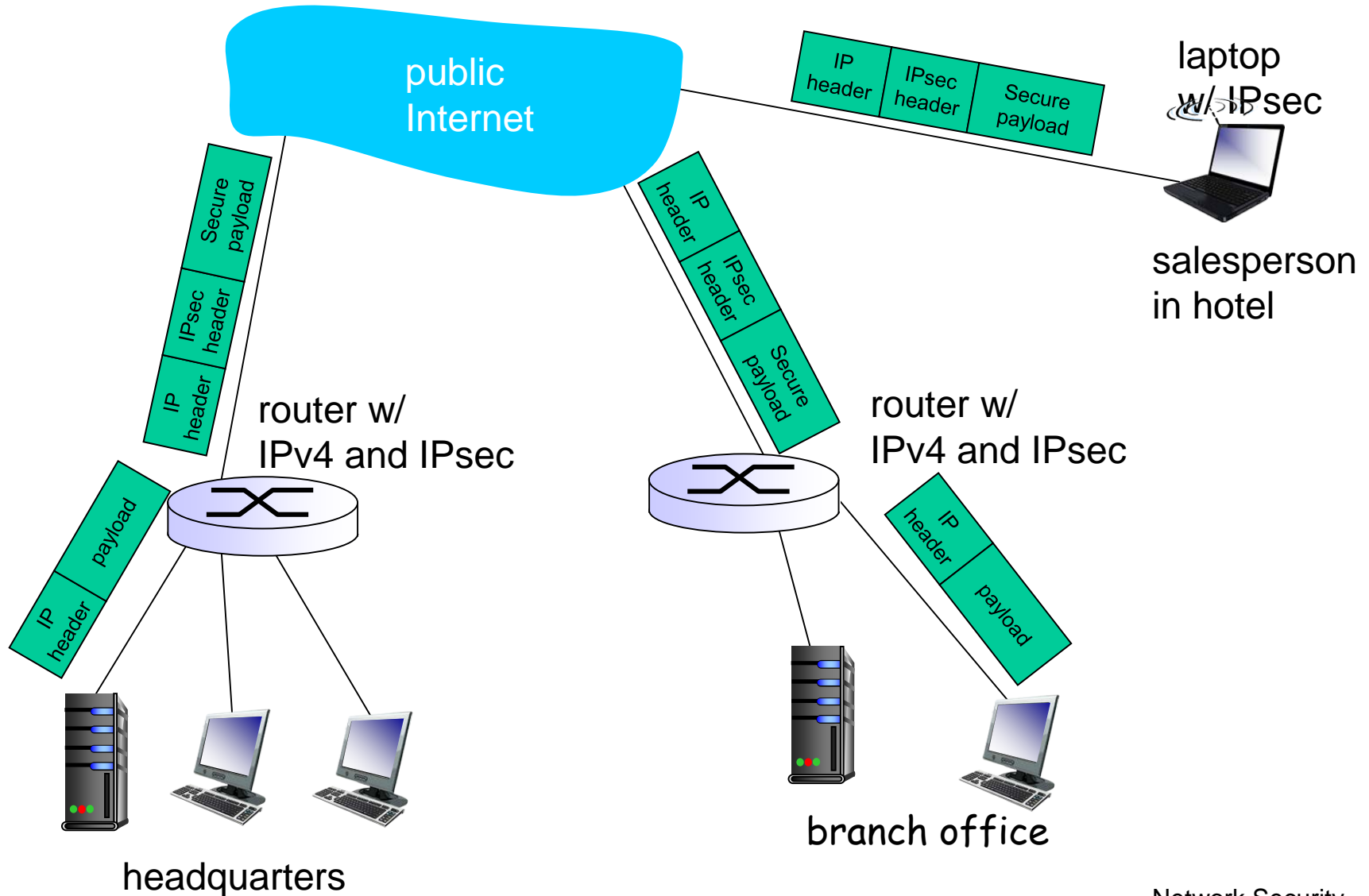
- ❖ sending entity encrypts datagram payload, payload could be:
 - TCP or UDP segment, ICMP message, OSPF message
- ❖ all data sent from one entity to other would be hidden:
 - web pages, e-mail, P2P file transfers, TCP SYN packets
...
- ❖ “blanket coverage”

Virtual Private Networks (VPNs)

motivation:

- ❖ institutions often want private networks for security.
 - costly: separate routers, links, DNS infrastructure.
- ❖ VPN: institution's inter-office traffic is sent over public Internet instead
 - encrypted before entering public Internet
 - logically separate from other traffic

Virtual Private Networks (VPNs)



Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

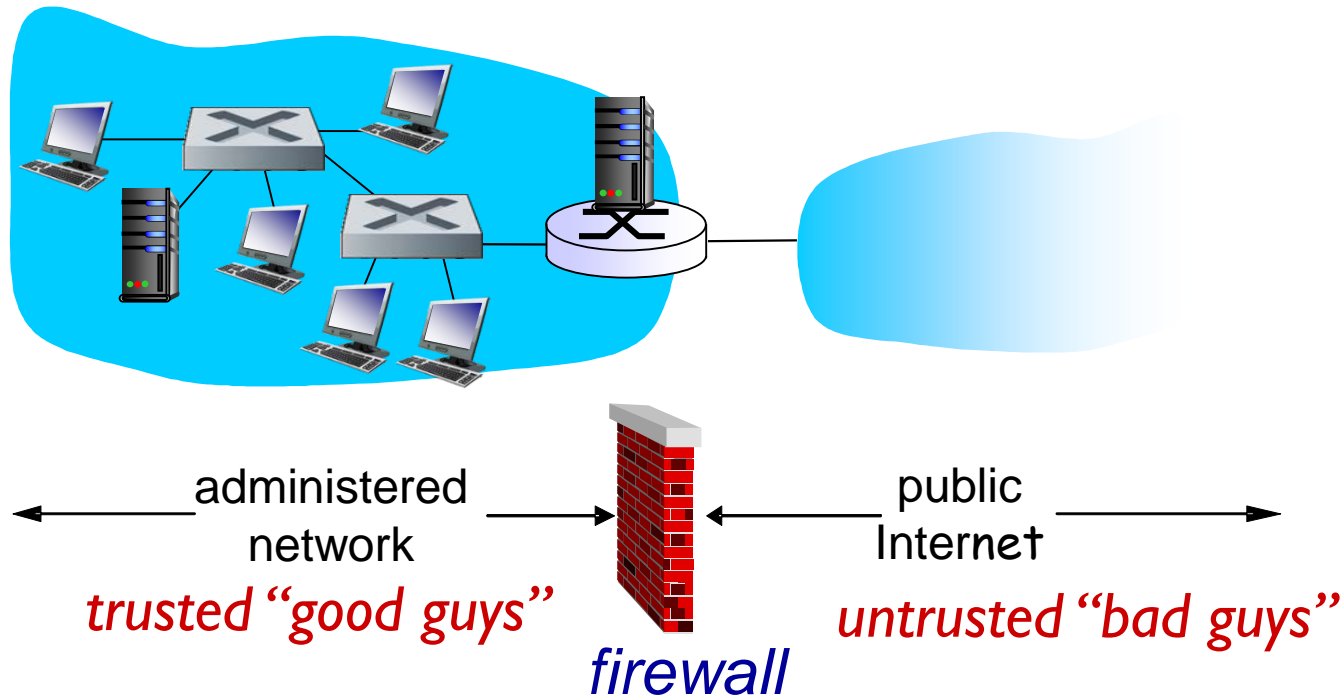
8.7 ~~Securing wireless LANs~~

8.8 *Operational security: firewalls and IDS*

Firewalls

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Firewalls: why

prevent denial of service attacks:

- ❖ attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data

- ❖ e.g., attacker replaces CIA’ s homepage with something else

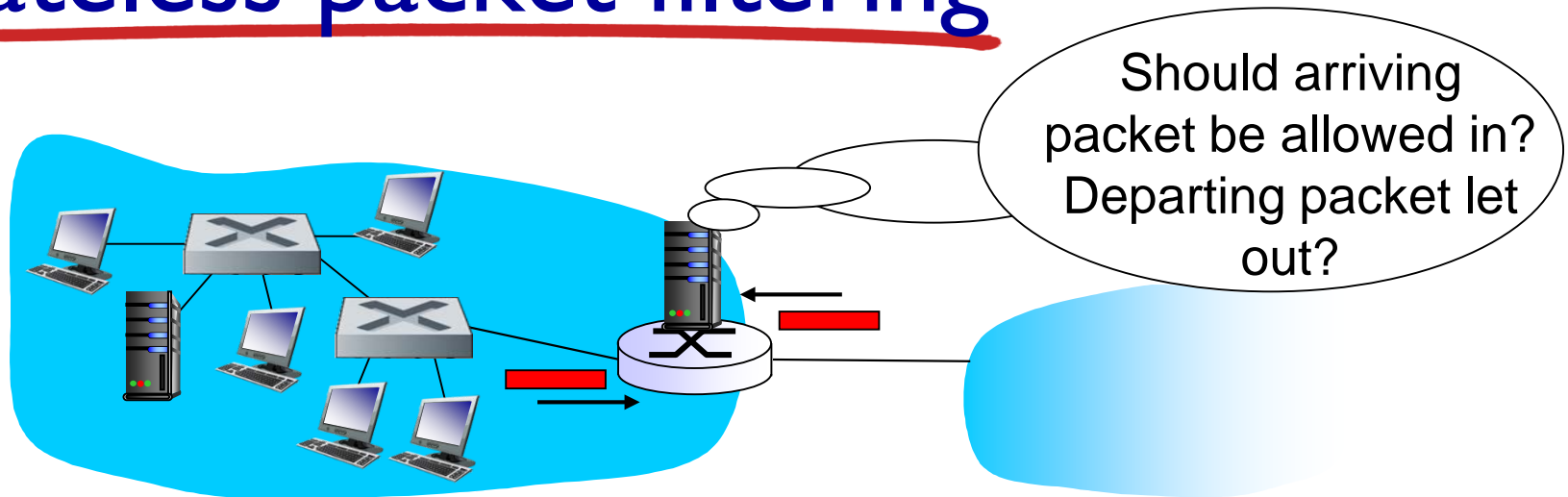
allow only authorized access to inside network

- ❖ 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
 - TCP/UDP source and destination port numbers
 - ICMP message type
 - TCP SYN and ACK bits

Stateless packet filtering: example

- ❖ *Warriors of the Net, movie 8:55*
 - https://www.youtube.com/watch?v=PBWhzz_GnI0
- ❖ *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

<i>Policy</i>	<i>Firewall Setting</i>
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	dest address	protocol	source port	dest 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	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- ❖ *stateful packet filter*: track 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

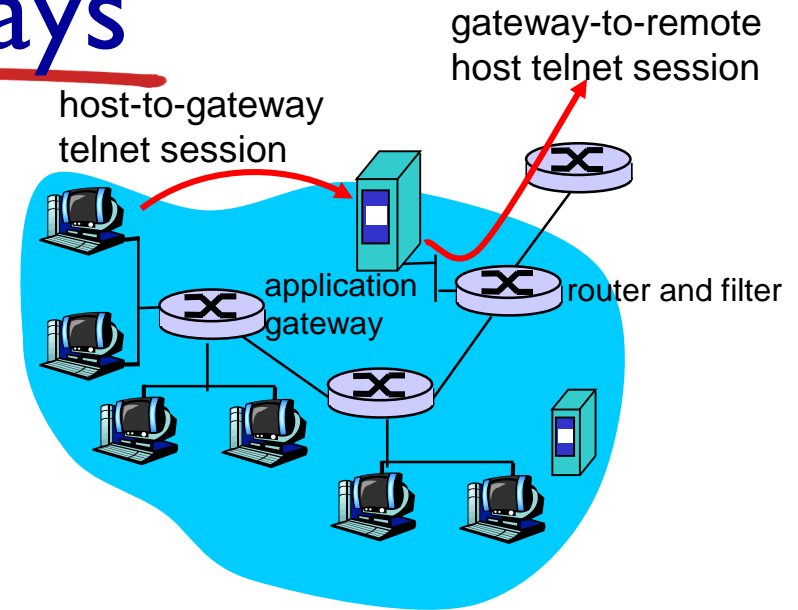
Stateful packet filtering

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

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
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	

Application gateways

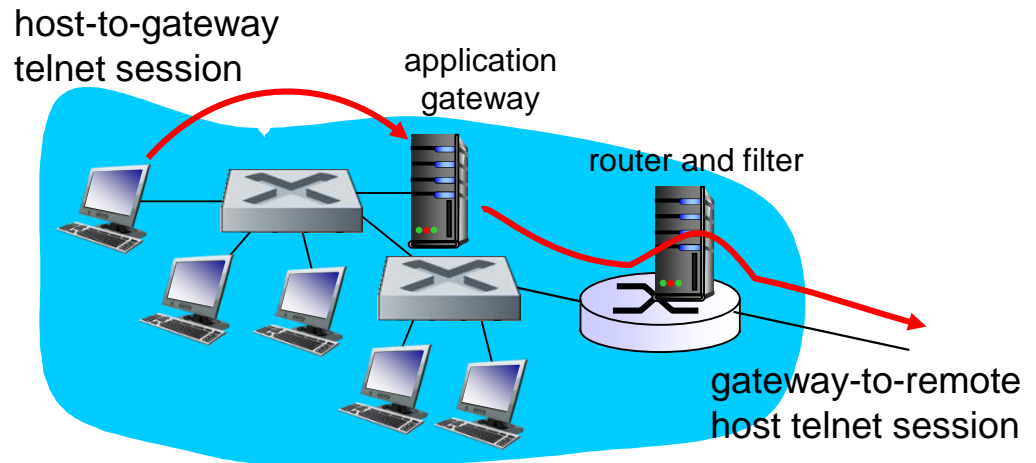
- ❖ Warriors of the net, movie 5:35
- ❖ filters packets on application data as well as on IP/TCP/UDP fields.
- ❖ **example:** allow select internal users to telnet outside



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

Application gateways

- ❖ filter packets on application data as well as on IP/TCP/UDP fields.
- ❖ *example:* allow select internal users to telnet outside



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

Limitations of firewalls, gateways

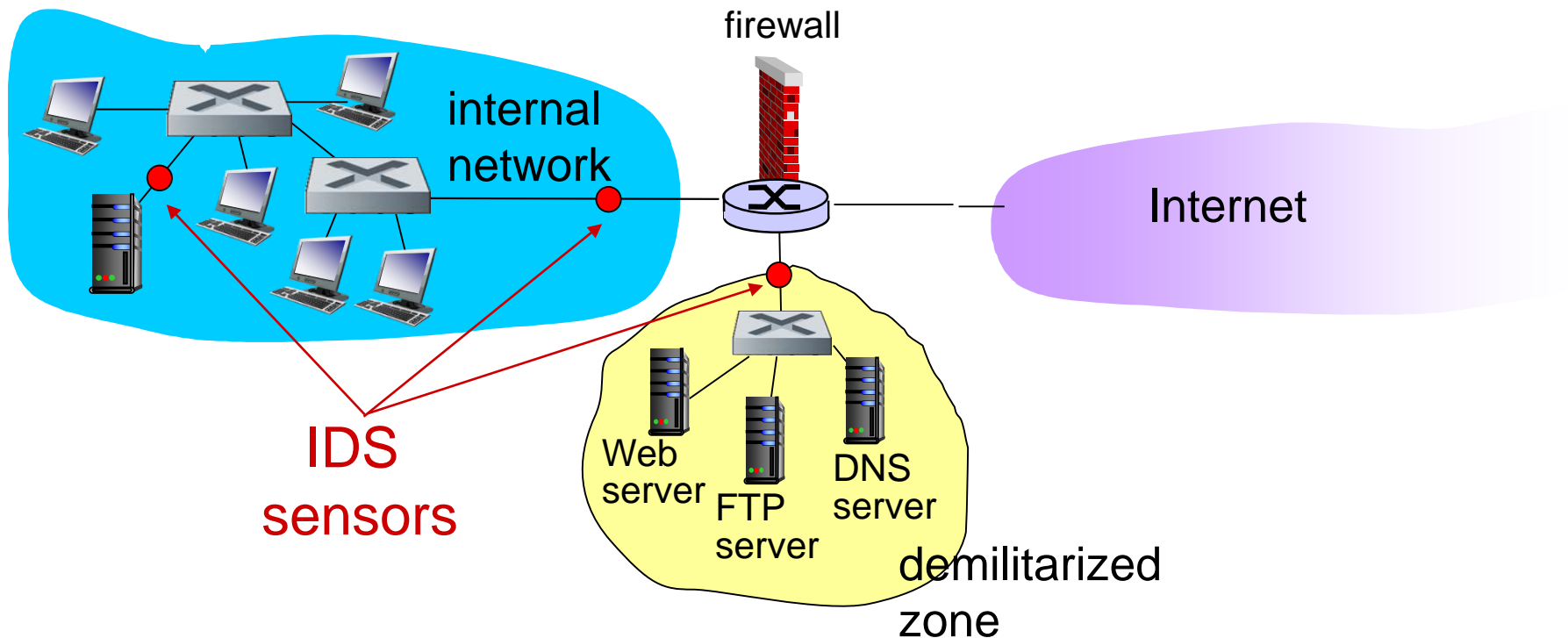
- ❖ *IP spoofing*: router can't know if data “really” comes from claimed source
- ❖ if multiple app's. need special treatment, each has own app. gateway
- ❖ client software must know how to contact gateway.
 - e.g., must set IP address of proxy in Web browser
- ❖ filters often use all or nothing policy for UDP
- ❖ *tradeoff*: degree of communication with outside world, level of 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*
 - *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
 - <http://map.norsecorp.com/>

Intrusion detection systems

- ❖ multiple IDSs: different types of checking at different locations



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