TCN 5080
Secure Telecom Transactions

Florida International University

Course Info

- Instructor: Dr. Deng Pan
- Email: pand@cs.fiu.edu
- Office: ECS-261A
- Office hours:
  - Monday 10am-12pm, Friday 3-5pm
  - Or by appointment

Course Info

- All the materials will be available at http://moodle.cis.fiu.edu.

- Course objectives:
  - Understand threats, principles, and mechanisms in secure network and telecom transactions.
  - Study cryptographic algorithms and their applications in security protocols in different layers of the Internet protocol stack.

Course Info

- References:
  - Charlie Kaufman, Radia Perlman, and Mike Speciner, Network Security: Private Communication in a Public World (2nd Edition), Prentice Hall, 2002.
  - William Stallings, Cryptography and Network Security : Principles and Practice (6th Edition), Prentice Hall, 2013.

Course Outline

- Cryptography
- Hashes and Digital Signatures
- Authentication Protocols
- IP Security
- SSL, PKI
- Firewalls
- Advanced topics

Grading

- Course Projects: 30%
- Midterm: 35%
- Final Exam: 35%

Chapter 1
Introduction

What is network security?

- By Google:
  - Network security is protection of the access to files and directories in a computer network against hacking, misuse and unauthorized changes to the system.
- By Cisco:
  - Network security consists of the policies adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.

What is network security?

confidentiality: only sender, intended receiver should "understand" message contents

- sender encrypts message
- receiver decrypts message

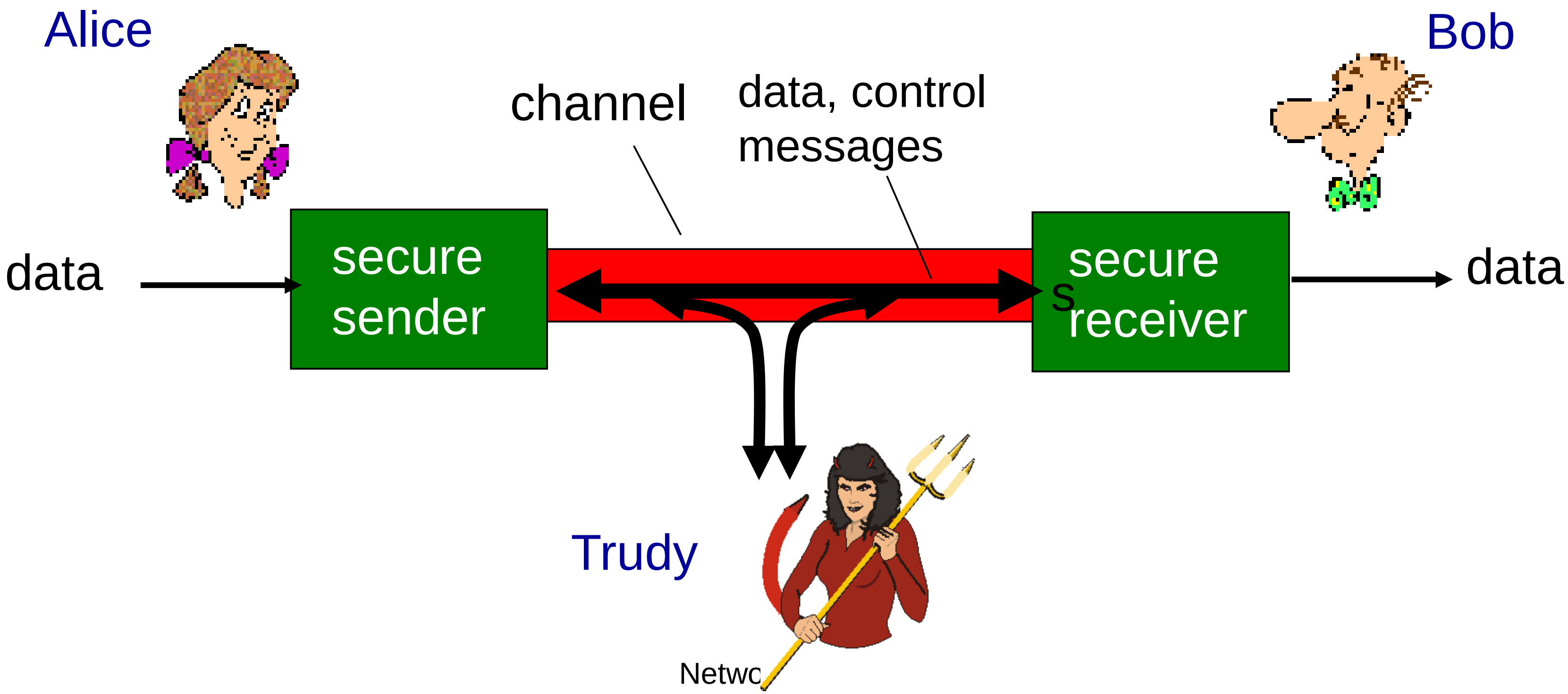
authentication: sender, receiver want to confirm identity of each other

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
- Alice and Bob want to communicate "securely"
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- 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 out there!

Q: What can a "Trudy" do?

A: A lot!

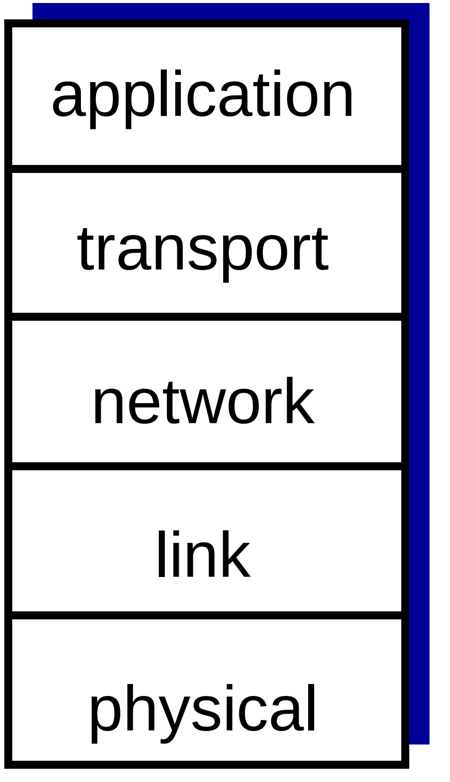
- *eavesdrop:* intercept messages
- actively *insert* messages into connection
- *impersonation:* can fake (spoof) source address in packet (or any field in packet)
- *hijacking:* "take over" ongoing connection by removing sender or receiver, inserting himself in place
- *denial of service*: prevent service from being used by others (e.g., by overloading resources)

Background requirements

- Computer networks
  - 5-layer Internet protocol stack
  - Protocols in each layer
- Mathematics
  - Number theory, probability...
- Programming
  - At least one programming language for course project

Internet protocol stack

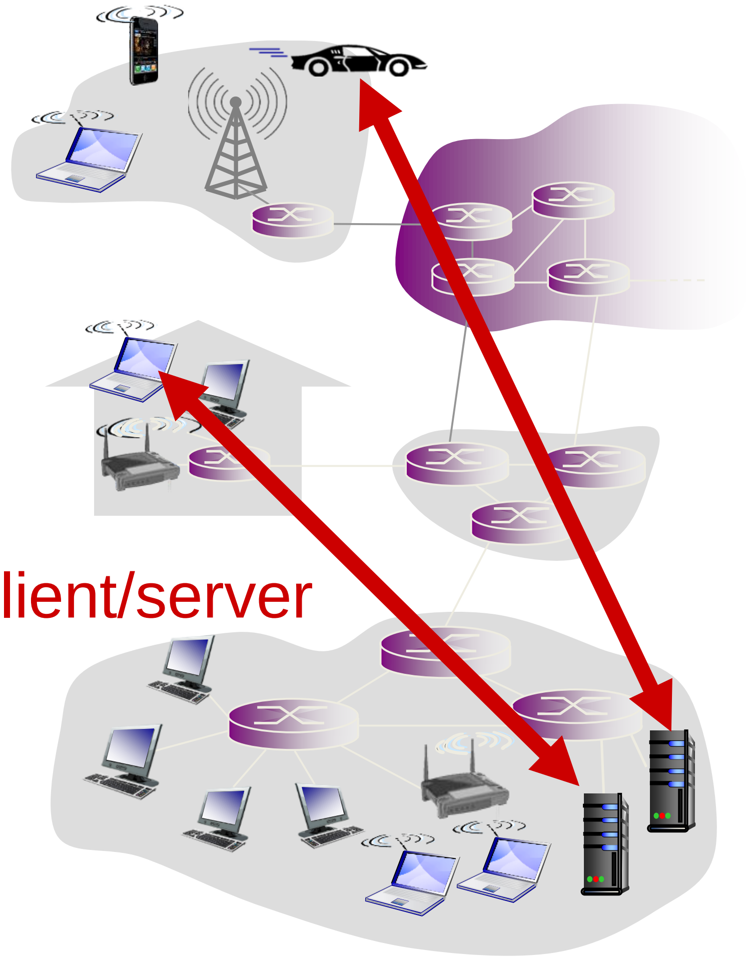
- Layered reference model
  - Change of implementation of one layer transparent to rest of system
- 5-layer TCP/IP protocol stack



Application Layer

- The application layer contains the higher-level protocols used by application programs for network communication.
  - Packet format, error handling, authentication…
- Two application architectures:
  - Client-server
  - Peer-to-peer (P2P)

Client-server architecture



server:
- always-on host
- permanent IP address
- data centers for scaling

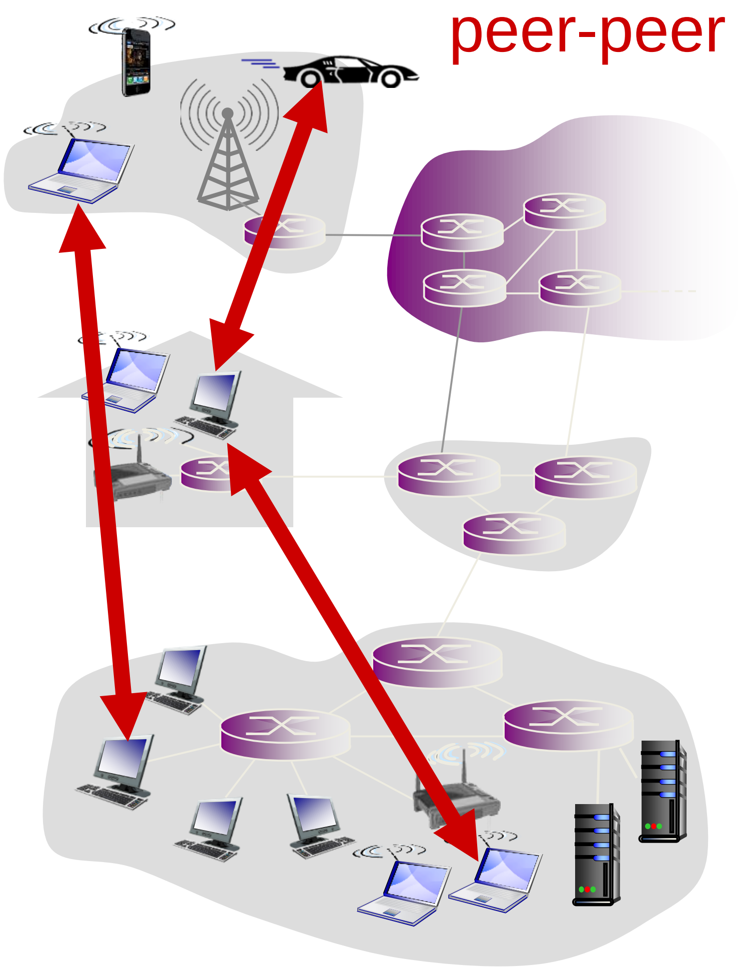
clients:
- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Client-server protocols

- Web
  - HTTP: Hypertext Transfer Protocol, RFC 2616
- File transfer
  - FTP: File Transfer Protocol, RFC 959
- Email
  - SMTP: Simple Mail Transfer Protocol, RFC 5321
- Remote login:
  - Telnet, RFC 854
- Supporting functions:
  - DNS: Domain Name System, RFC 1035

P2P architecture

- *no* always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - *self scalability* – new peers bring new service capacity, as well as new service demands
- peers are intermittently connected and change IP addresses
  - complex management



P2P protocols

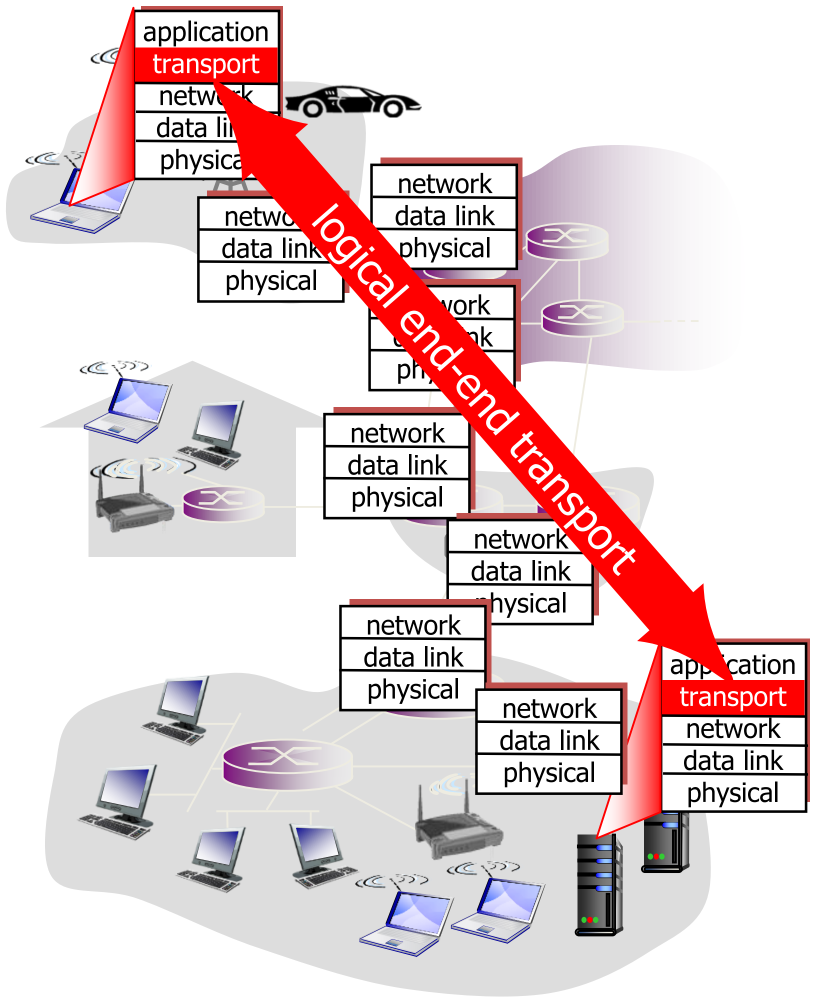
- File transfer
  - BitTorrent, RFC 5694
  - eMule
- Instant messaging
  - Skype
- Video streaming
  - PPStream

Transport layer

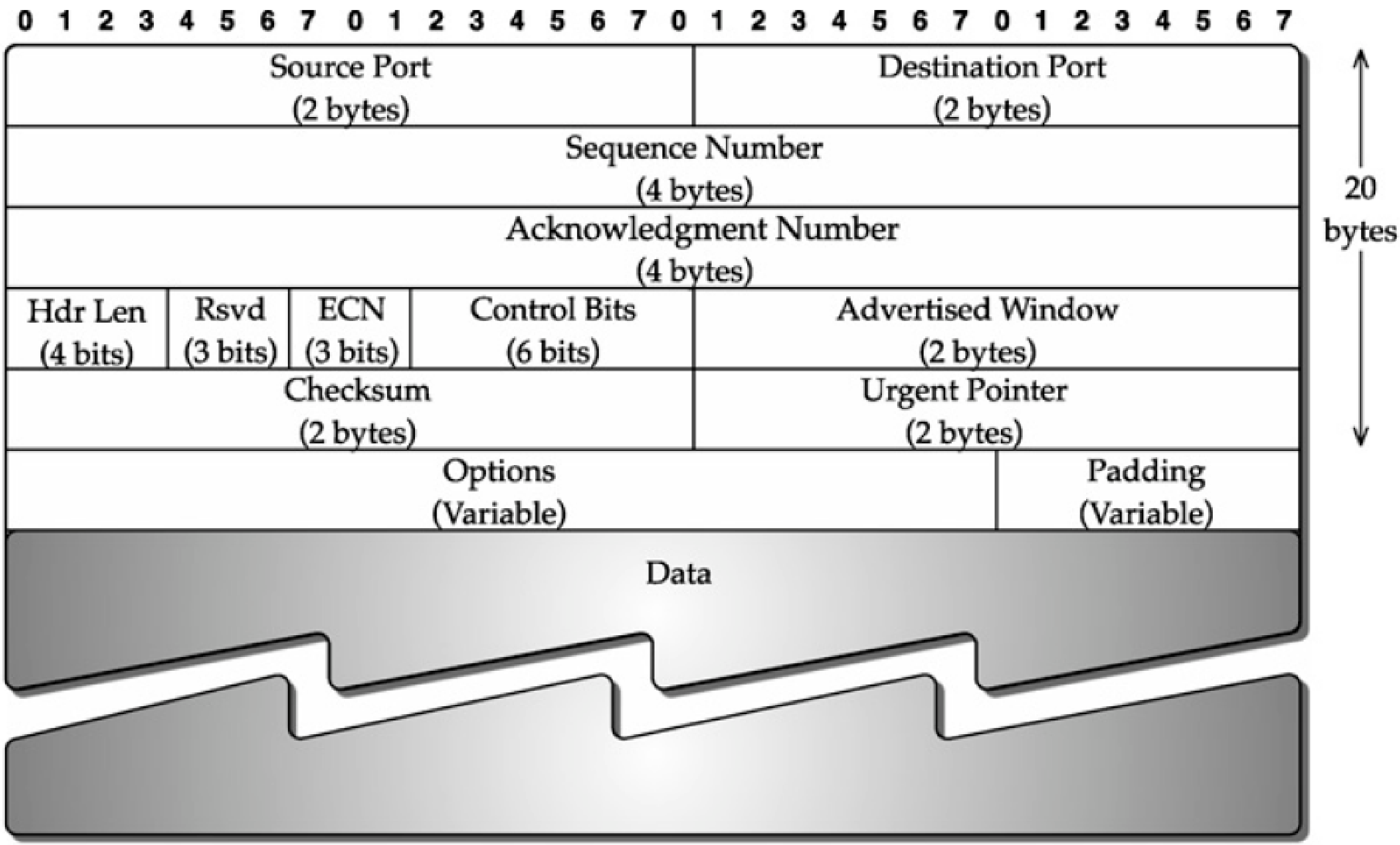
- The transport layer establishes a basic data channel that an application uses in its task-specific data exchange.
  - Implemented in end systems only
  - Logical communication between processes
- Two main protocols
  - TCP: transmission control protocol
  - UDP: user datagram protocol

Internet transport-layer protocols

- reliable, in-order delivery (TCP)
  - congestion control
  - flow control
  - connection setup
- unreliable, unordered delivery: UDP
  - no-frills extension of "best-effort" IP
- services not available:
  - delay guarantees
  - bandwidth guarantees

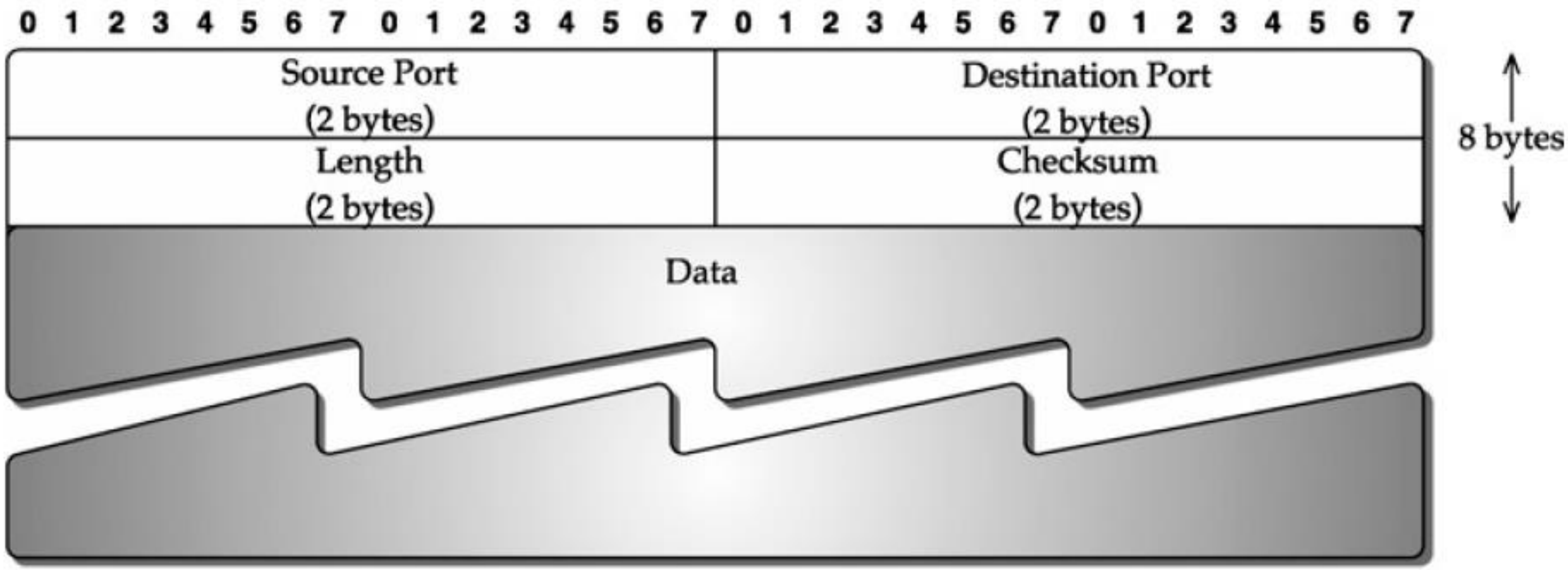


TCP Format



0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Source Port (2 bytes)
Destination Port (2 bytes)
Sequence Number (4 bytes)
Acknowledgment Number (4 bytes)
Hdr Len (4 bits)
Rsvd (3 bits)
ECN (3 bits)
Control Bits (6 bits)
Advertised Window (2 bytes)
Checksum (2 bytes)
Urgent Pointer (2 bytes)
Options (Variable)
Padding (Variable)
Data
20 bytes

UDP Format


0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Source Port
(2 bytes)
Destination Port
(2 bytes)
Length
(2 bytes)
Checksum
(2 bytes)
8 bytes
Data

Internet apps: application, transport protocols

application	application layer protocol	underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	TCP or UDP

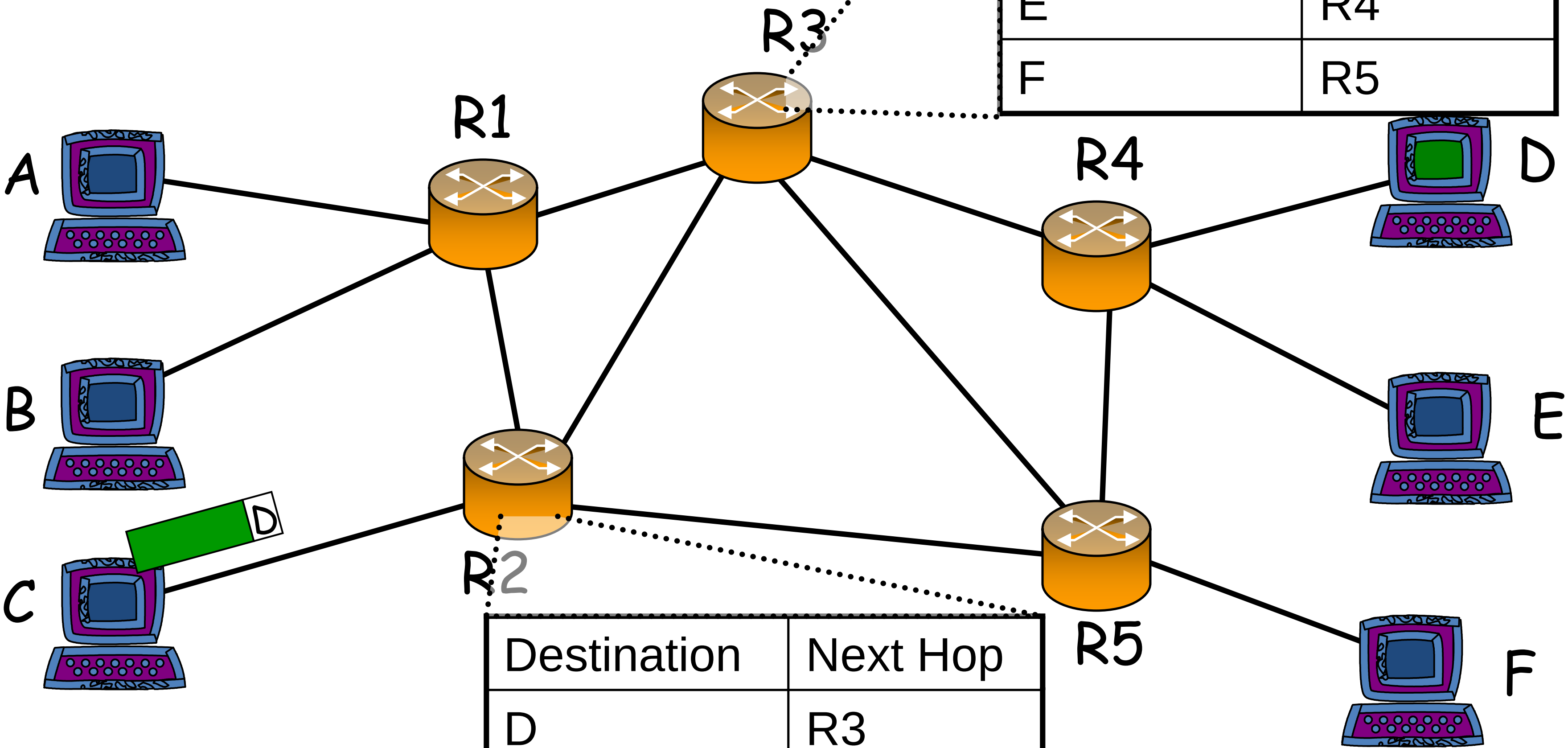
Network Layer

- The network layer has the responsibility of sending packets across potentially multiple networks.
  - Multi-hop transmission
  - Implemented in end systems and routers
  - Logical communication between hosts
- Single protocol
  - IP: Internet Protocol

Two key network-layer functions

- *forwarding:* move packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to dest.
  - *routing algorithms*

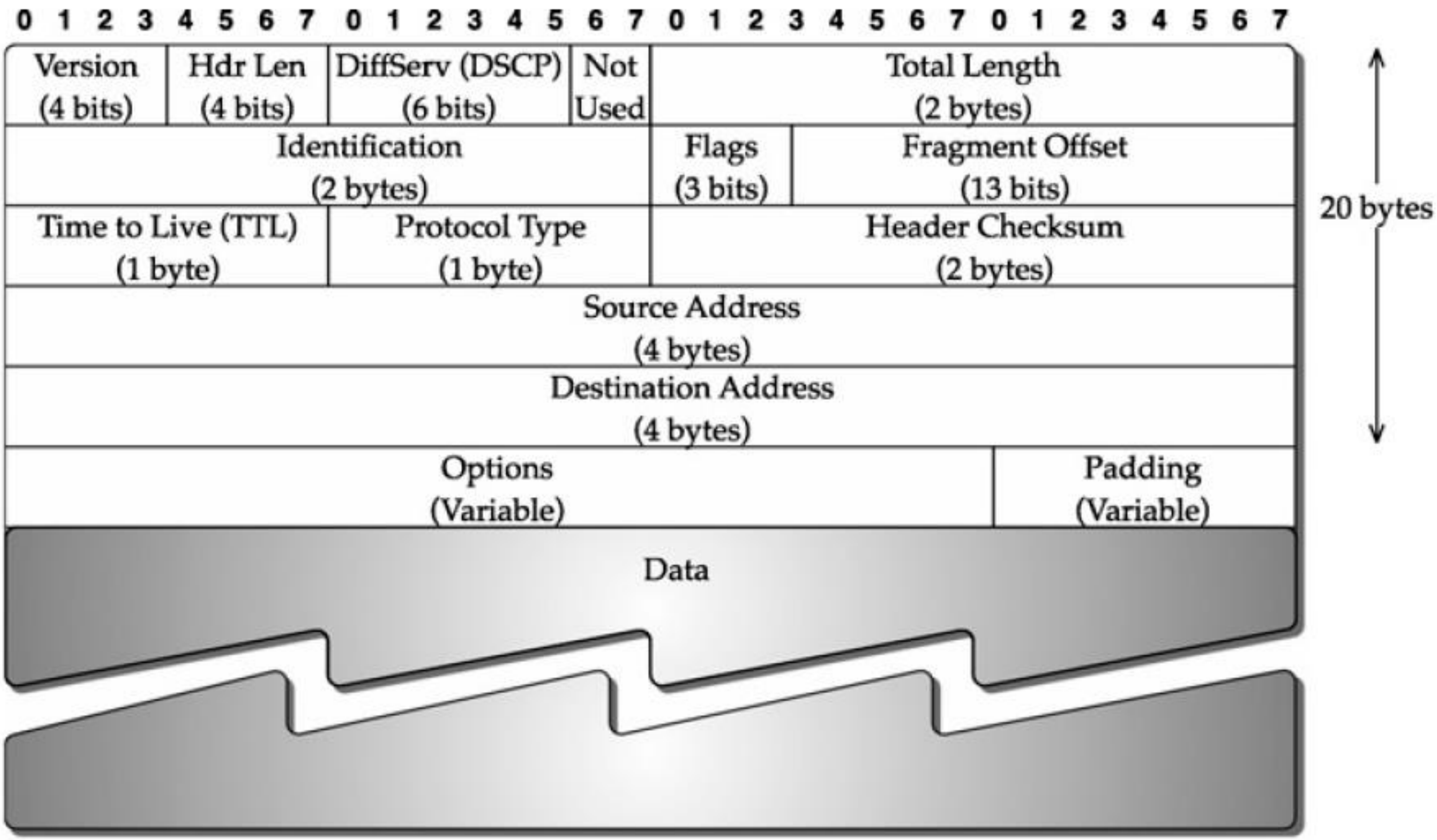
Interplay between routing and forwarding



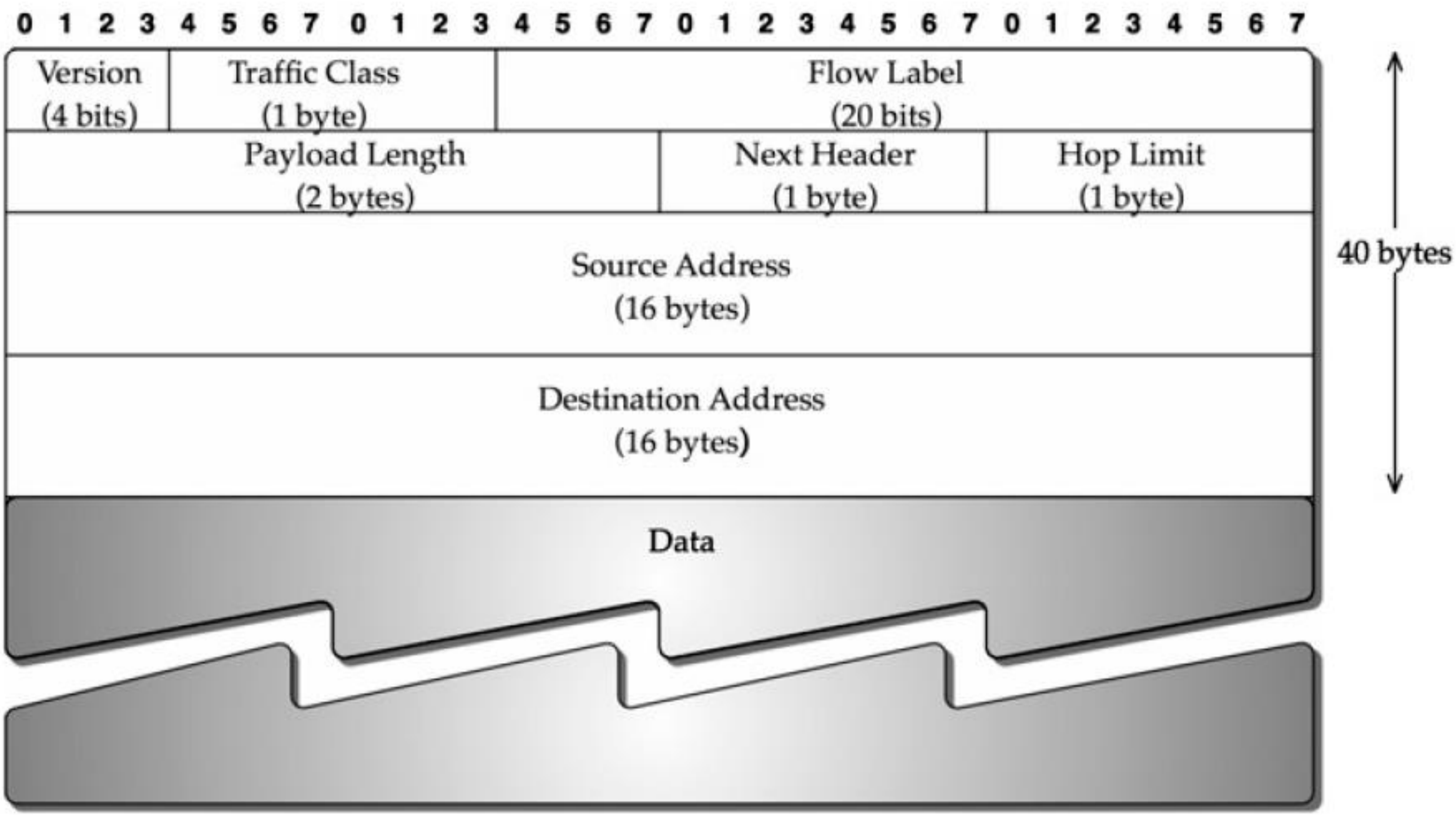
Destination	Next Hop
D	R4
E	R4
F	R5

Destination	Next Hop
D	R3
E	R3
F	R5

IP v4 Format


0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Version (4 bits)
Hdr Len (4 bits)
DiffServ (DSCP) (6 bits)
Not Used
Total Length (2 bytes)
Identification (2 bytes)
Flags (3 bits)
Fragment Offset (13 bits)
Time to Live (TTL) (1 byte)
Protocol Type (1 byte)
Header Checksum (2 bytes)
Source Address (4 bytes)
Destination Address (4 bytes)
Options (Variable)
Padding (Variable)
Data
20 bytes

IP v6 Format


0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Version (4 bits)
Traffic Class (1 byte)
Flow Label (20 bits)
Payload Length (2 bytes)
Next Header (1 byte)
Hop Limit (1 byte)
Source Address (16 bytes)
Destination Address (16 bytes)
Data
40 bytes

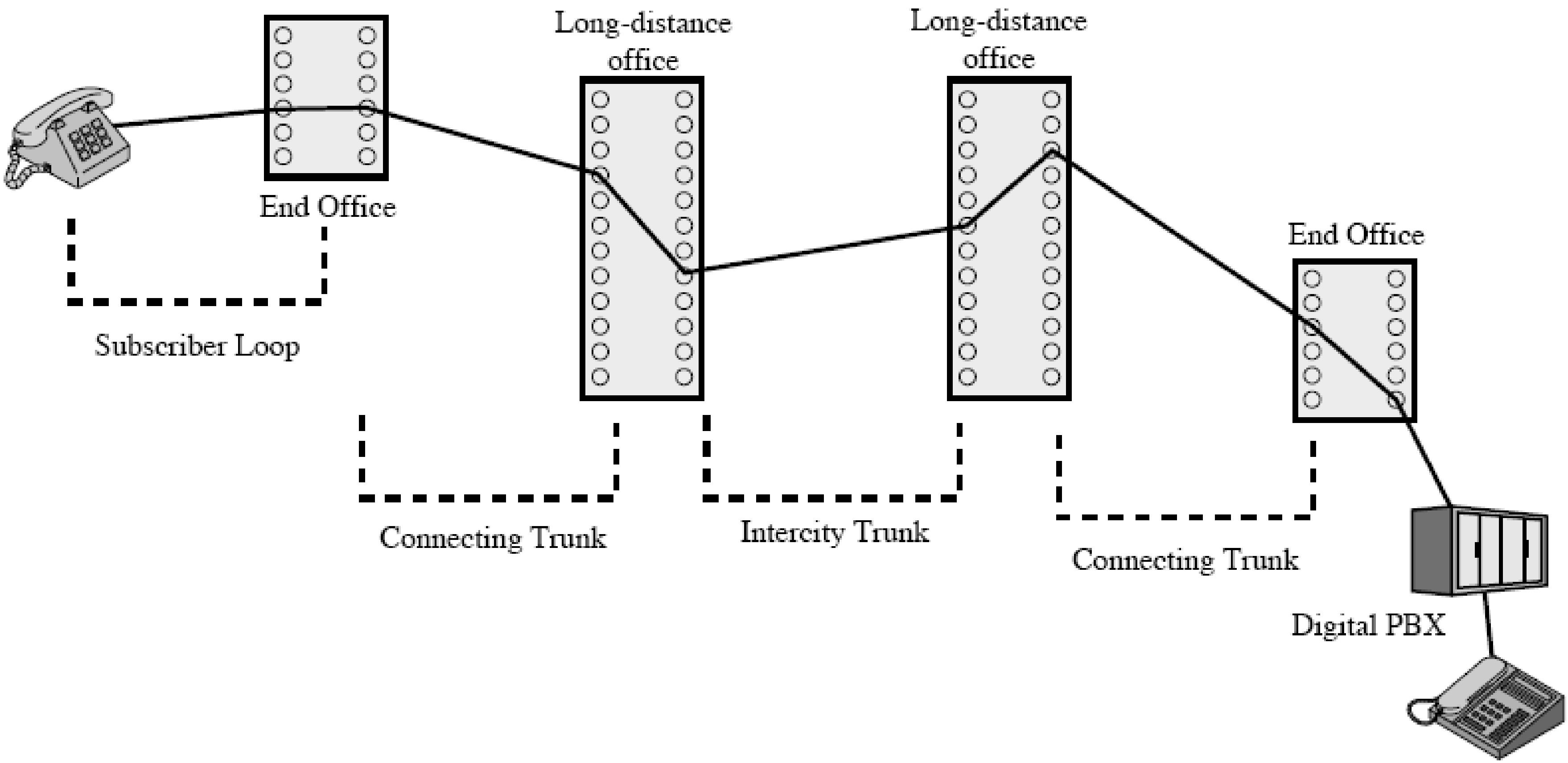
Data Link Layer

- The link layer transmits frames from one node to its adjacent nodes.
  - One-hop transmission
- Protocol used depends on type of network
  - Circuit switching
  - Packet switching
  - LANs (e.g., Ethernet, WIFI)

Switching Techniques

- Circuit switching
  - Dedicated communications path between two stations
  - E.g. public telephone network
- Packet switching
  - Message is broken into a series of packets
  - Each node determines next hop of transmission for each packet
  - E.g. IP

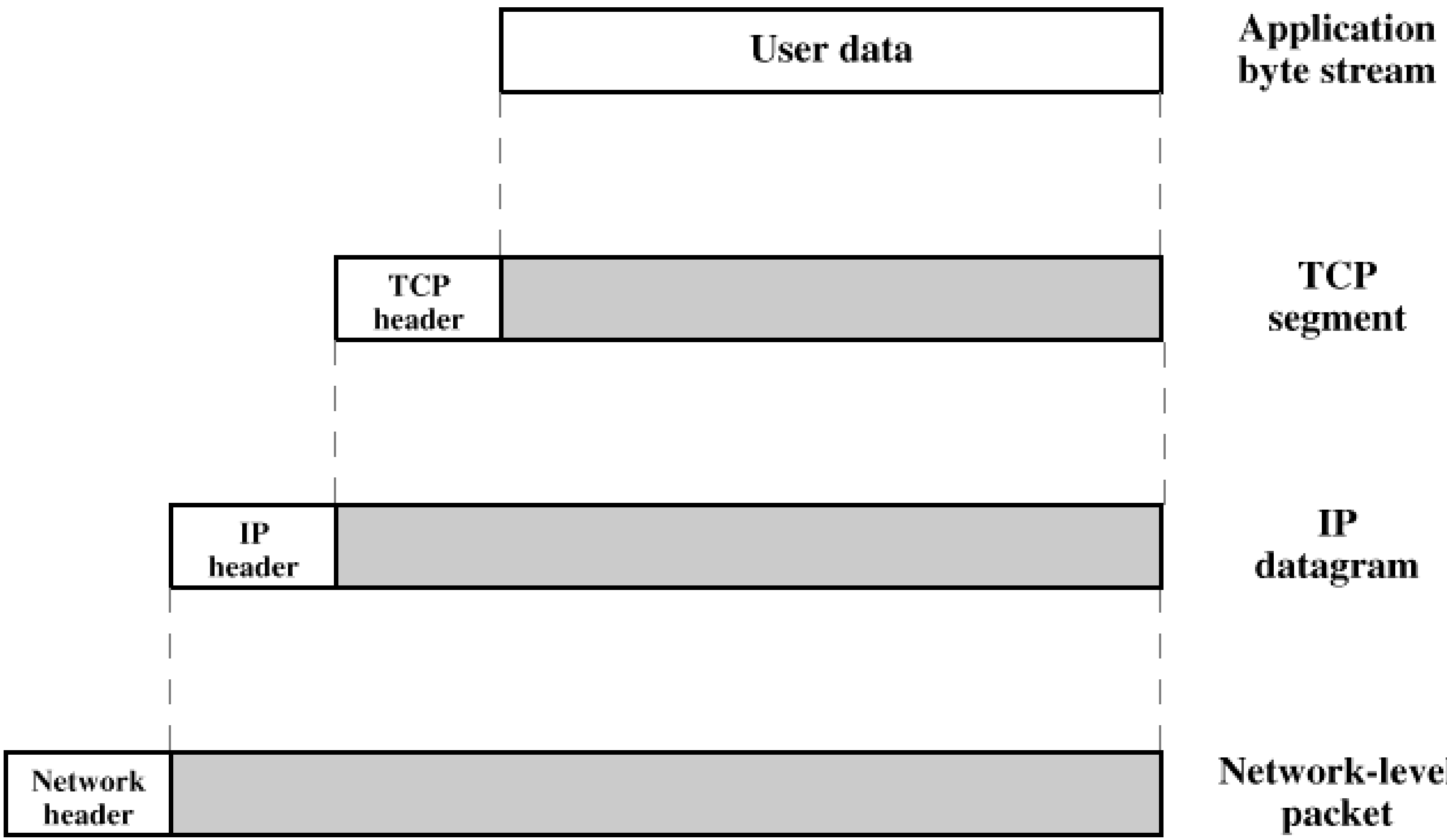
Public Switched Telephone Network

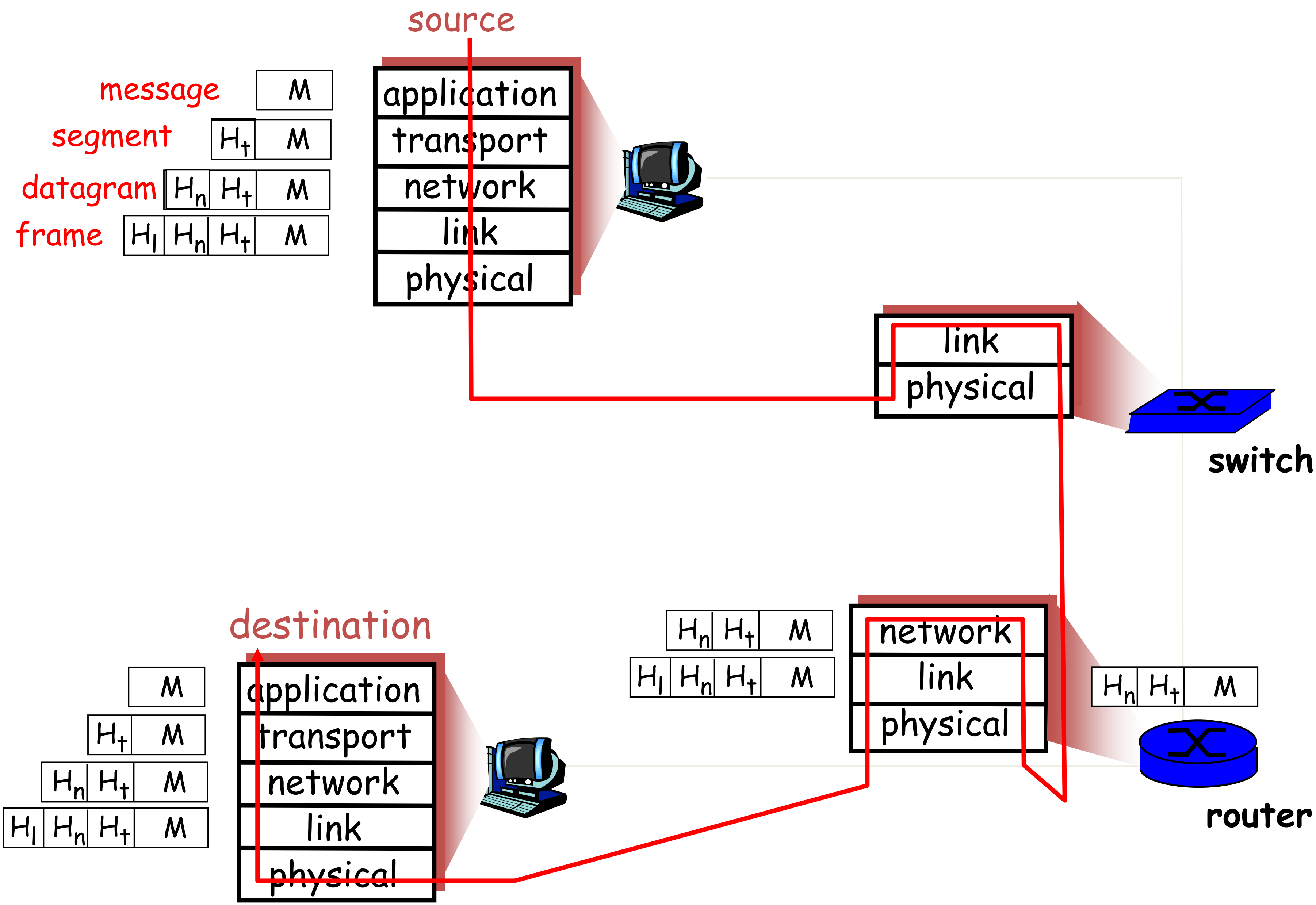


Physical Layer

- The physical layer consists of the basic networking hardware transmission technologies of a network.
- Covers the physical interface between a data transmission device and a transmission medium or network
- Physical layer specifies:
  - Characteristics of the transmission medium
  - The nature of the signals
  - The data rate
  - Other related matters

Protocol Data Units (PDUs)





source
message
M
segment
Ht
M
datagram
Hn
Ht
M
frame
Hl
Hn
Ht
M
application
transport
network
link
physical
link
physical
switch
destination
M
Ht
M
Hn
Ht
M
Hl
Hn
Ht
M
application
transport
network
link
physical
Hn
Ht
M
Hl
Hn
Ht
M
network
link
physical
Hn
Ht
M
router

Security protocols in each layer

- Application layer
  - SSH (secure telenet), SFTP
- Transport layer
  - SSL/TLS
- Network layer
  - IPSec
- Data link layer
  - WPA, WEP, PPP authentication
- Physical layer
  - Black listing

Security related Terminology

- Risk
- Threats
- Vulnerabilities
- Adversary
- Attacks
- Participants
- Trust
- Security Model

Risk

- At-risk valued resources that can be misused
  - Monetary
  - Data (loss or integrity)
  - Time
  - Confidence
  - Trust
- What does being misused mean?
  - Privacy (personal)
  - Confidentiality (communication)
  - Integrity (personal or communication)
- Availability
  - Denial of service

Threats

- A threat is a specific means by which an attacker can put a system at risk
  - An ability of an attacker (e.g., eavesdrop on a communication channel)
  - Independent of what can be compromised
- A threat model is a collection of threats that deemed important for a particular environment
  - A collection of attacker(s) abilities
  - E.g. a powerful attacker can read and modify all communications and generate messages on a communication channel

Vulnerabilities

- Vulnerabilities are systemic artifacts that expose the user, data or system to a threat.
  - Buffer overflows, WEP key leakage, etc
- Where do vulnerabilities come from?
  - Bad software or hardware
  - Poor understanding of requirements/bad design
  - Bad policy/configuration
  - System misuse
  - Unintended purpose or environment

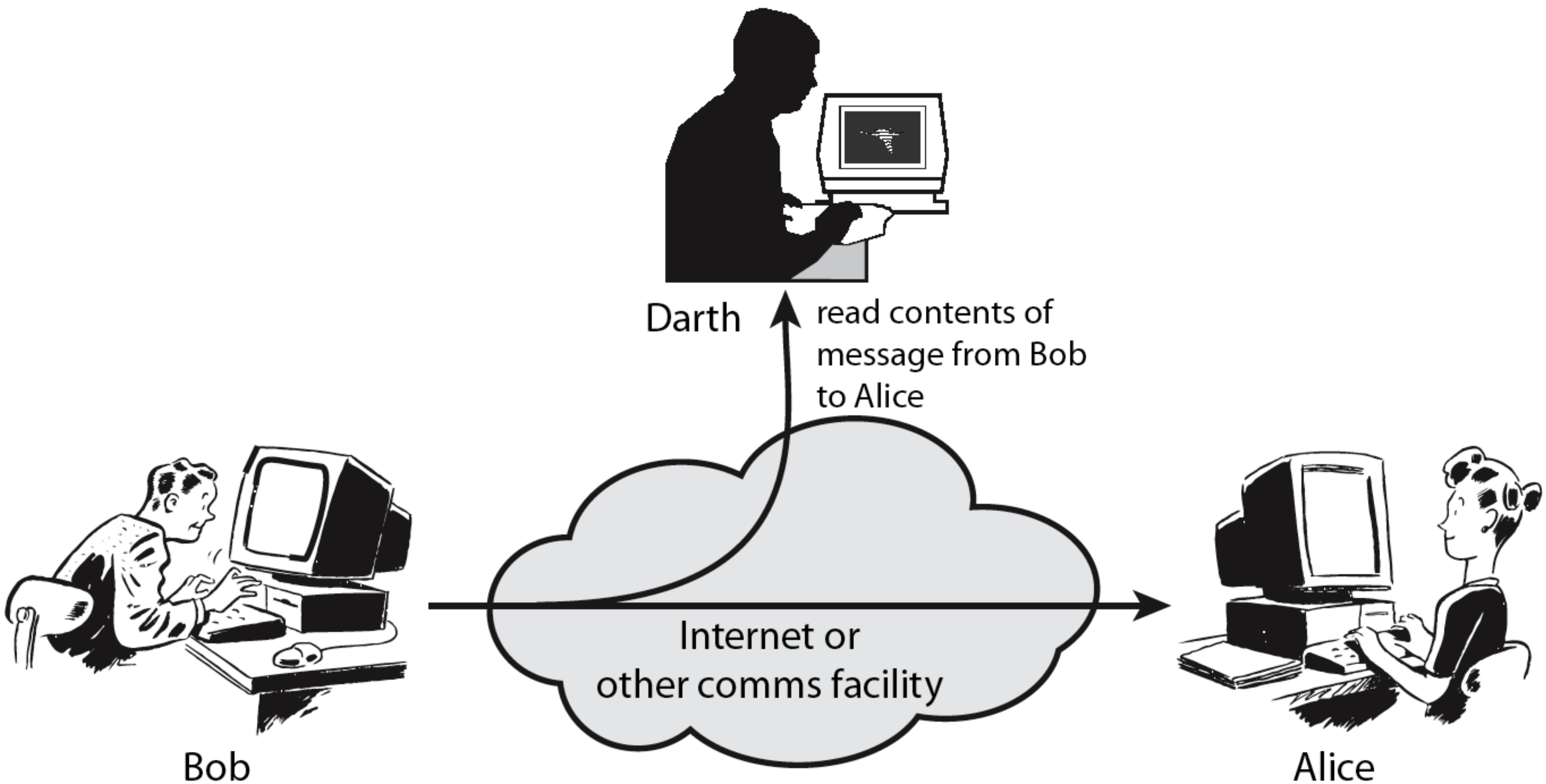
Adversary

- An adversary is anyone attempting to circumvent the security infrastructure.
  - The curious and generally clueless (e.g., script-kiddies)
  - Casual attackers seeing to understand systems
  - People with an axe to grind
  - Malicious groups with sophisticated users (e.g., chaos clubs)
  - Competitors (industrial espionage)
  - Governments (seeking to monitor or disrupt activities)

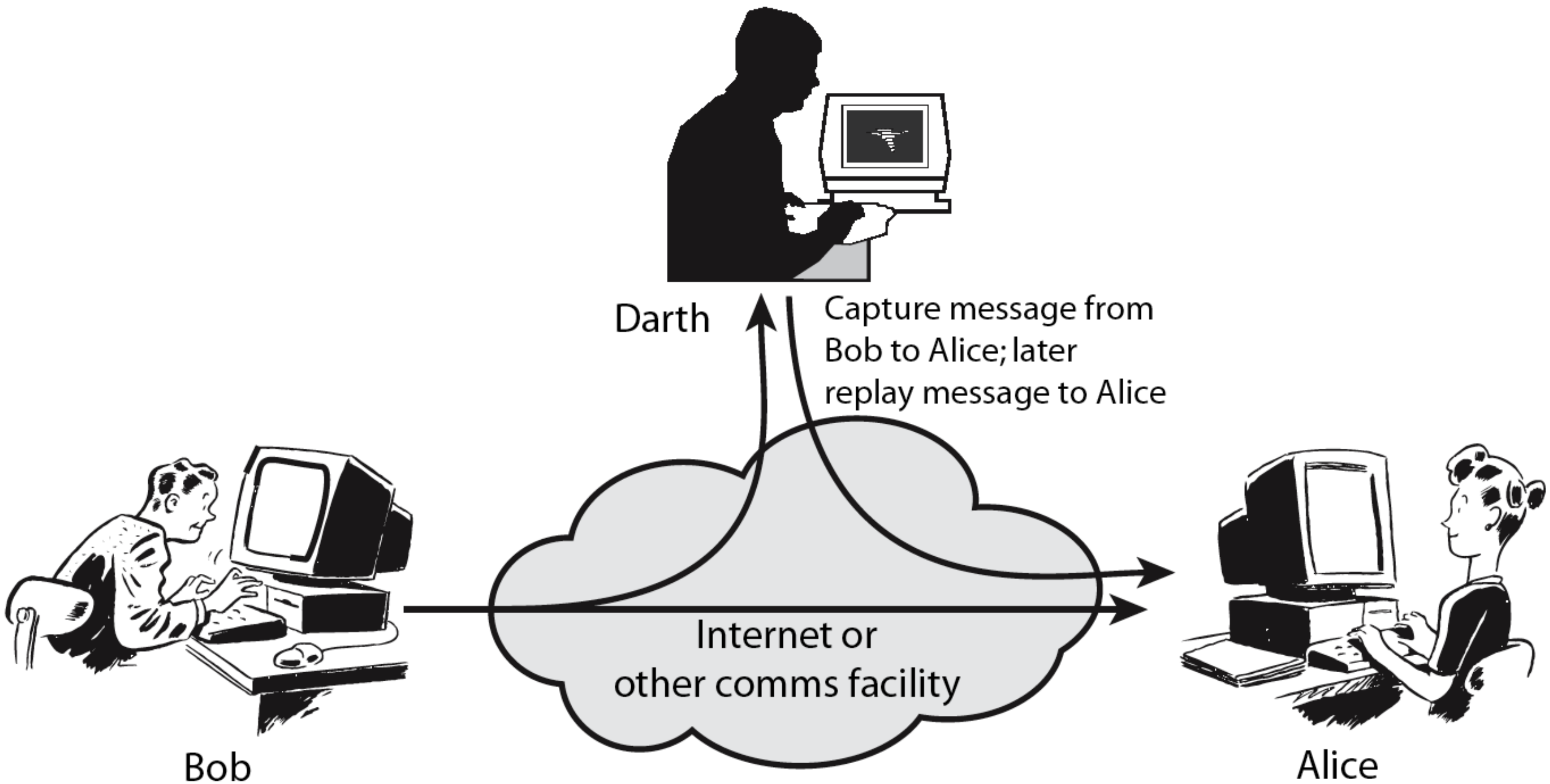
Attacks

- An attack occurs when someone attempts to exploit a vulnerability
- Kinds of attacks
  - Passive (e.g., eavesdropping)
  - Active (e.g., replay attack)
- A compromise occurs when an attack is successful
  - Typically associated with taking over/altering resources

Passive Attacks



Active Attacks



Participants

- Participants are expected system entities
  - Computers, agents, people, enterprises, …
  - Depending on context referred to as: servers, clients, users, entities, hosts, routers, …
- Security is defined with respect to these entitles
  - Implication: every party may have unique view
- A trusted third party
  - Trusted by all parties for some set of actions
  - Often used as introducer or arbiter

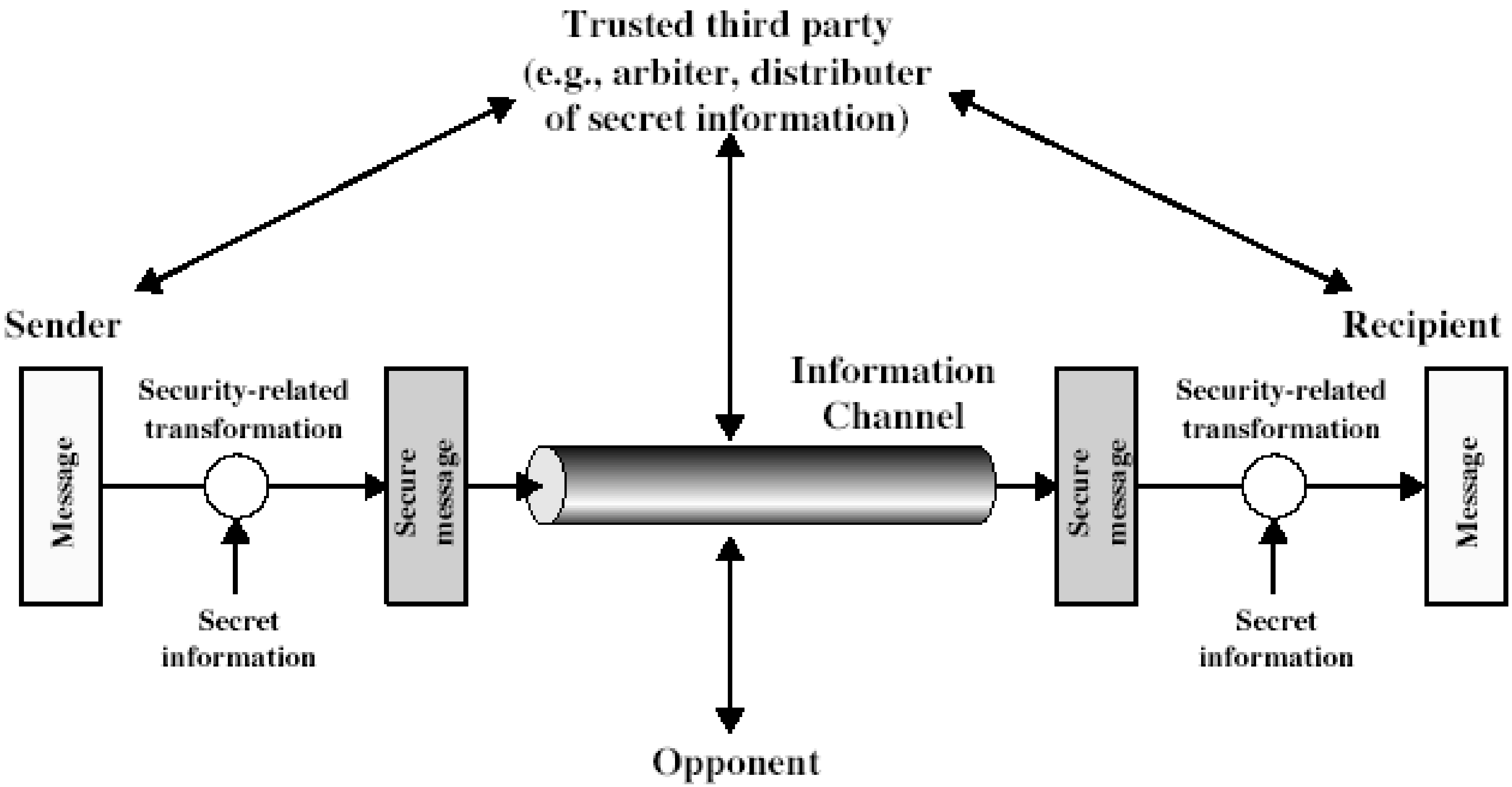
Trust

- Trust refers to the degree to which an entity is expected to behave.
- What is an entity not expected to do?
- A trust model describes, for a particular environment, who is trusted to do what.
- You make trust decisions every day...
  - What are they?
  - Whom do you trust?
- Can you measure trust?

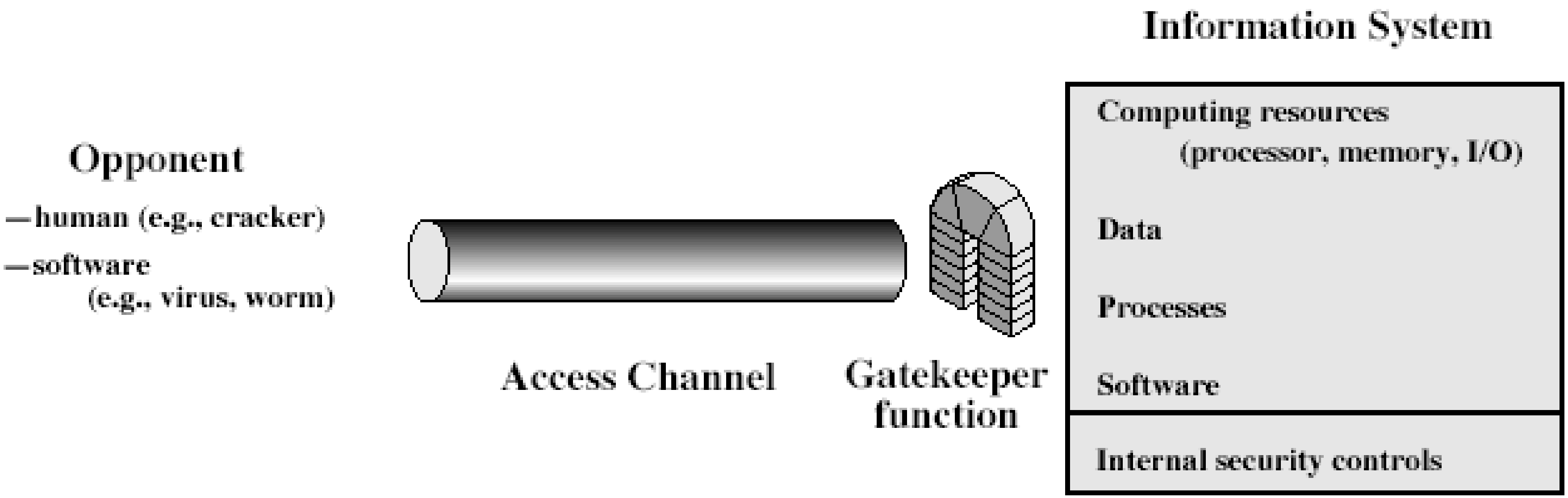
Security Model

- A security model is the combination of a trust and threat models that address the set of perceived risks
  - The “security requirements” used to develop some cogent and comprehensive design
  - Every design must have security model: LAN network or global information system? Java applet or operating system?
- Systems must be explicit about these things to be secure.
  - What are the security concerns (risks)? Threats?
  - Who are our adversaries?
  - Who do we trust and to do what?

Model for Network Transmission Security



Model for Network Access Security



Security Service

- Enhance security of data processing systems and information transfers of an organization
- Using one or more security mechanisms
  - Encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization
- Often replicates functions normally associated with physical documents
  - which, for example, have signatures, dates; need protection from disclosure, tampering, or destruction; be notarized or witnessed; be recorded or licensed

Security Services (X.800)

- Authentication - assurance that the communicating entity is the one claimed
- Access Control - prevention of the unauthorized use of a resource
- Data Confidentiality –protection of data from unauthorized disclosure
- Data Integrity - assurance that data received is as sent by an authorized entity
- Non-Repudiation - protection against denial by one of the parties in a communication

Chapter 2
Secret Key Cryptography

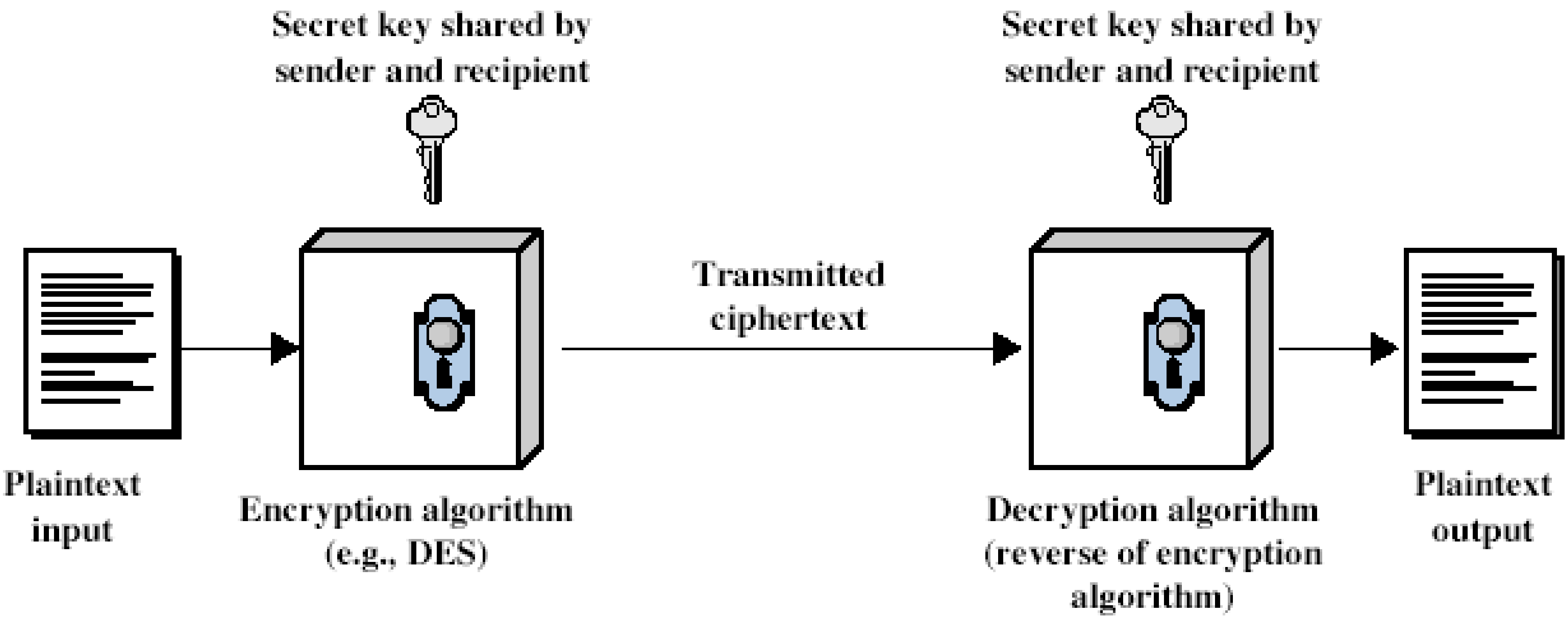
Secret Key Encryption

- or conventional / private-key / single-key
- sender and recipient share a common key
- all classical encryption algorithms are secret key based
- was only type prior to invention of public-key in 1970's

Some Basic Terminology

- **plaintext/cleartext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** – recovering plaintext from ciphertext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key

Symmetric Cipher Model



Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- mathematically have:

  $Y = \mathrm{E}_K(X)$

  $X = \mathrm{D}_K(Y)$
- assume encryption algorithm is known

Cryptography

- characterize cryptographic system by:
  - type of encryption operations used
    - substitution / transposition / product
  - number of keys used
    - single-key or private / two-key or public
  - way in which plaintext is processed
    - block / stream

Cryptanalysis

- objective to recover key not just message
- general approaches:
  - cryptanalytic attack
  - brute-force attack

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/µs	Time required at 10^6 decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^6 years

Classical Substitution Ciphers

- where letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- earliest known substitution cipher
- by Julius Caesar
- first attested use in military affairs
- replaces each letter by 3rd letter on
- example:

```
PHHW PH DIWHU WKH WRJD SDUWB
```

Caesar Cipher

- can define transformation as:

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
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- mathematically give each letter a number

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
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

- then have Caesar cipher as:

c = E(p) = (p + k) mod (26)

p = D(c) = (c – k) mod (26)

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
  - A maps to A,B,..Z
- could simply try each in turn
- a brute force search
- given ciphertext, just try all shifts of letters
- do need to recognize when have plaintext
- eg. break ciphertext "GCUA VQ DTGCM"

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: dkvqfibjwpescxhtmyauolrgzn

Plaintext:  ifwewishtoreplaceletters
Ciphertext: wirfrwajuhyftsdvfsfuufya
```

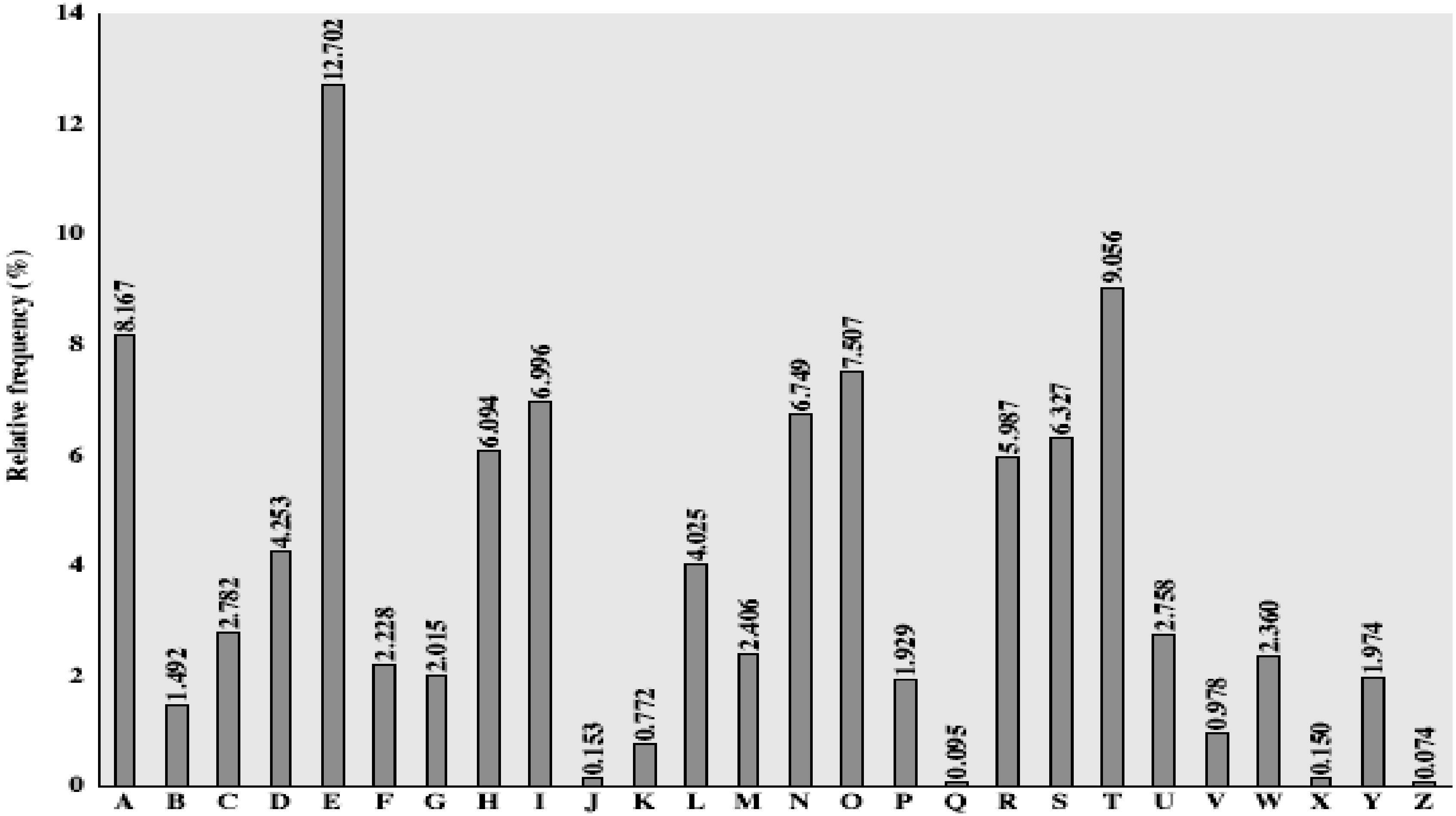
Monoalphabetic Cipher Security

- now have a total of
  - $26! = 4 \times 10^{26}$ keys
- with so many keys, might think is secure
- but would be wrong
- problem is language characteristics

Language Redundancy and Cryptanalysis

- human languages are **redundant**
- eg "th lrd s m shphrd shll nt wnt"
- letters are not equally commonly used
- in English E is by far the most common letter
  - followed by T,R,N,I,O,A,S
- other letters like Z,J,K,Q,X are fairly rare
- have tables of single, double & triple letter frequencies for various languages

English Letter Frequencies



Use in Cryptanalysis

- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- if caesar cipher look for common peaks/troughs
  - peaks at: A-E-I triple, NO pair, RST triple
  - troughs at: JK, X-Z
- for monoalphabetic must identify each letter
  - tables of common double/triple letters help

Example Cryptanalysis

- given ciphertext:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

- count relative letter frequencies (see text)
- guess P & Z are e & t
- guess ZW is th and hence ZWP is the
- proceeding with trial and error finally get:

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

Example

- A generalization of the Caesar cipher, known as the affine cipher is as follows:

  $C=E([a,b],p)=(ap+b) \mod 26$

- A ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is ‘B’, and the second most frequent is ‘U’. Break the code.

Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improving security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (sans duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Encrypting and Decrypting

- plaintext is encrypted two letters at a time
  1. if a pair is a repeated letter, insert filler like 'X’
  2. if both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
  3. if both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom)
  4. otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair

Security of Playfair Cipher

- security much improved over monoalphabetic
- since have 26 x 26 = 676 digrams
- would need a 676 entry frequency table to analyse (verses 26 for a monoalphabetic)
- and correspondingly more ciphertext
- was widely used for many years
  - eg. by US & British military in WW1
- it can be broken, given a few hundred letters
- since still has much of plaintext structure

Polyalphabetic Ciphers

- **polyalphabetic substitution ciphers**
- improve security using multiple cipher alphabets
- make cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

Vigenère Cipher

- simplest polyalphabetic substitution cipher
- effectively multiple caesar ciphers
- key is multiple letters long $K = k_1 \, k_2 \, \ldots \, k_d$
- i^{th} letter specifies i^{th} alphabet to use
- use each alphabet in turn
- repeat from start after d letters in message
- decryption simply works in reverse

Example of Vigenère Cipher

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

```
key:        deceptivedeceptivedeceptive
plaintext:  wearediscoveredsaveyourself
ciphertext: zicvtwqngrzgvtwavzhcqyglmgj
```

Security of Vigenère Ciphers

- have multiple ciphertext letters for each plaintext letter
- hence letter frequencies are obscured
- but not totally lost
- start with letter frequencies
  - see if look monoalphabetic or not
- if not, then need to determine number of alphabets, since then can attack each

Autokey Cipher

- ideally want a key as long as the message
- Vigenère proposed the **autokey** cipher
- with keyword is prefixed to message as key
- knowing keyword can recover the first few letters
- use these in turn on the rest of the message
- eg. given key *deceptive*

```
key:        deceptivewearediscoveredsav
plaintext:  wearediscoveredsaveyourself
ciphertext: zicvtwqngkzeiigasxstslvvwla
```

Transposition Ciphers

- now consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- write message letters out diagonally over a number of rows
- then read off cipher row by row
- eg. write message out as:

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

- giving ciphertext

```
MEMATRHTGPRYETEFETEOAAT
```

Row Transposition Ciphers

- a more complex transposition
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

```
Key:            3 4 2 1 5 6 7
Plaintext:      a t t a c k p
                o s t p o n e
                d u n t i l t
                w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Product Ciphers

- ciphers using substitutions or transpositions are not secure because of language characteristics
- hence consider using several ciphers in succession to make harder, but:
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- this is bridge from classical to modern ciphers