

<https://brown-csci1660.github.io>

# CS1660: Intro to Computer Systems Security Spring 2025

## Lecture 2: Cryptography I

Co-Instructor: **Nikos Triandopoulos**

January 28, 2025



BROWN

# CS1660: Announcements

- ◆ Override requests
  - ◆ Status update
- ◆ Course updates
  - ◆ Homework 0, Project 0
  - ◆ Ed Discussion, Top Hat, Gradescope
  - ◆ Lectures, online reading resources, in-class demos

# Today

- ◆ Computer Security
  - ◆ Motivation
  - ◆ Basic security concepts
- ◆ Cryptography
  - ◆ Symmetric-key ciphers
  - ◆ Classical ciphers & OTP
  - ◆ DES & AES block ciphers

## **2.0 Secure outsourced computation**

# Another example: Tax return preparation...

Involves information collection & processing

- ◆ calculate financial data
  - ◆ payroll, profits, stock quotes, ...
- ◆ manage data
  - ◆ search emails, store records, ...
- ◆ submit – done!



**... by many  
unknown machines!**

# Data & computation outsourcing

## Cloud-based services

- ◆ hardware, OS, software, apps, ...
- ◆ storage, computation, databases, analytics, ...

## Transformative multi-platform technology

- ◆ businesses, organizations or individuals
- ◆ client-server, distributed, P2P, Web-based, ...



## Internet protocols



## social networks



## big-data analytics



## sharing economy



## FinTech



# Security consequences



**Fact:** Untrusted interactions

- ◆ information is processed outside one's administration control or "trust perimeter"

**Risk:** Falsified / leaked information

- ◆ information may (un)intentionally altered by or shared with unauthorized entities

**Goal:** Integrity / privacy safeguards for outsourced assets

- ◆ need to protect information against change, damage / unauthorized access

# What can go wrong?



## **Fact:** Untrusted interactions

- ◆ information is processed outside one's administration control or "trust perimeter"

## **Risk:** Falsified / leaked information

- ◆ information may (un)intentionally altered by or shared with unauthorized entities

## **Goal:** Integrity / privacy safeguards for outsourced assets

- ◆ need to protect information against change, damage / unauthorized access

## **Threats:**

- ◆ misconfigurations, erroneous failures, limited liability
- ◆ economic incentives of cost-cutting providers
- ◆ compromises, attacks, advanced persistent threats (APTs)



# Limited liability

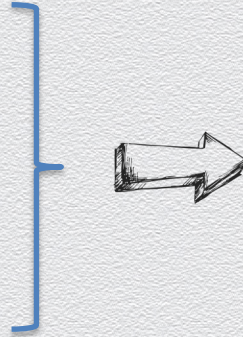
“[We will] not be responsible for any damages arising in connection with any unauthorized access to, alteration of, or the deletion, destruction, damage loss or failure to store any of your content or other data.”

**Amazon Web Services customer agreement**

# Advanced Persistent Threats (APTs)

Sophisticated well-targeted cyber-attack campaigns

- ◆ aim for unauthorized data manipulation or exfiltration
- ◆ employ rich attack vectors & highly adaptive strategies
  - ◆ social engineering
  - ◆ zero-day vulnerabilities
  - ◆ low-and-slow progression
  - ◆ intelligence



extremely hard-to-defend  
or even hard-to-detect

...	
RSA	(2011)
Bit9	(2013)
Dyn	(2016)
Equifax	(2017)
...	



# Real cases: Threats against integrity Vs. confidentiality

Figure 6: VERIS A<sup>4</sup> grid depicting associations between actors, actions, assets, and attributes

Server.Conf	35%	48%	23%	2%	.	1%		.	2%	2%	5%	1%	2%		.	.	.	1%		.
Server.Integ	35%	41%	23%	2%	.	1%		.	2%	2%	3%	1%	2%		.	.	.	.		.
Server.Avail	1%	2%	1%			.		.	.		.	.	.							
Network.Conf	.	.	.	.	1%	.		.	.		.									
Network.Integ	.	.	.	.	1%	.		.	.		.		.							
Network.Avail		.	.			.		.	.				.							
User.Conf	35%	36%	22%	1%	32%	.		.	.	.	3%	1%	.						.	
User.Integ	35%	34%	22%	1%	32%	.		.	.	.	1%	1%	.						.	
User.Avail	.	.	.	.	1%			.	.		1%	.								
Media.Conf	.	.	2%	2%	1%					2%	5%	2%	.						.	
Media.Integ	.	.	2%	2%	1%					2%	3%	1%	.							
Media.Avail			.	.	1%	✓				.	.	1%								
People.Conf	22%	24%	29%	4%	1%			.		4%	4%	1%						.	.	.
People.Integ	22%	24%	29%	4%	1%			.		4%	4%	1%						.	.	.
People.Avail	.	2%	2%	1%	1%			.		.	1%	1%								
External.Malware																				
External.Hacking																				
External.Social																				
External.Misuse																				
External.Physical																				
External.Error																				
External.Env																				
Internal.Malware																				
Internal.Hacking																				
Internal.Social																				
Internal.Misuse																				
Internal.Physical																				
Internal.Error																				
Internal.Env																				
Partner.Malware																				
Partner.Hacking																				
Partner.Social																				
Partner.Misuse																				
Partner.Physical																				
Partner.Error																				
Partner.Env																				

## Data Breach Investigations Report by Verizon (2013)

- ◆ servers are a high-value target
- ◆ compromises / attacks affect both confidentiality and integrity

# The “new” big threat: Data manipulation

Newest cyber threat will be data manipulation, US intelligence chief says 

- James Clapper calls data deletion or manipulation 'next push of the envelope'
- US digital networks currently threatened by wide-scale data theft

Cyber security chief:  
Manipulation of data by  
hackers may be next  
threat

PITTSBURGH  
TRIBUNE-REVIEW

Cybersecurity

Former NSA chief: Data manipulation an 'emerging art of war'

FCW  
THE BUSINESS OF FEDERAL TECHNOLOGY

But what happens when suddenly our data is manipulated, and you no longer can believe what you're physically seeing?

THE WALL STREET JOURNAL  
WSJ

## US Officials' View

- ◆ data manipulation is the new big threat

**a Digital Pearl Harbor**

## **2.1 Basic security concepts**

# What is Security?

**Security** is the prevention of, or protection against

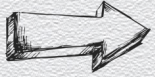

- ◆ access to information by unauthorized recipients
- ◆ intentional but unauthorized destruction or alteration of that information

Definition from: *Dictionary of Computing*, Fourth Ed.  
(Oxford: Oxford University Press 1996).

## **Security** (informal definition)

- ◆ the protection of information systems from
  - ◆ theft or damage to the hardware, the software, and to the information on them, as well as from disruption or misdirection of the services they provide
  - ◆ any possible threat

# The 'Security' game: What's at stake?

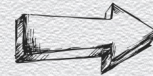
- ◆ Computer systems comprise assets that have (some) **value**
  - ◆ e.g., laptops store vast personal or important information (files, photos, email, ...)
  - ◆ personal, time dependent and often imprecise (e.g., monetary Vs. emotional)
- ◆ Valuable assets deserve **security protection**
  - ◆ to **preserve** their **value**,  expressed as a **security property**
    - ◆ e.g., personal photos should always be accessible by their owner
  - ◆ or to **prevent** (undesired) **harm**  examined as a concrete **attack**
    - ◆ e.g., permanent destruction of irreplaceable photos



# The 'Security' game: Who are the players?

## ◆ Defenders

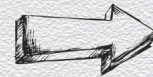
- ◆ system owners (e.g., users, administrators, etc.)
- ◆ seek to **enforce** one or more **security properties** or **defeat** certain **attacks**



**property-based view**

## ◆ Attackers

- ◆ external entities (e.g., hackers, other users, etc.)
- ◆ seek to launch attacks that **break** a **security property** or **impose** the system to certain **threats**



**attack-based view**

# Security properties

- ◆ General statements about the value of a computer system
- ◆ Examples
  - ◆ The C-I-A triad
    - ◆ **confidentiality, integrity, availability**
  - ◆ (Some) other properties
    - ◆ **authentication / authenticity**
    - ◆ **authorization / appropriate use**
    - ◆ **non-repudiation / accountability / auditability**
    - ◆ **anonymity**

# The C-I-A triad

- ◆ Captures the three fundamental properties that make any system valuable



Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data, while preserving access (availability)

# Confidentiality

- ◆ An asset is viewed only by authorized parties
  - ◆ e.g., conforming to originally-prescribed “read” rules  
<subject, object, access mode, policy> via access control
  - ◆ some other tools
    - ◆ encryption, obfuscation, sanitization, ...



# Integrity

- ◆ An asset is modified only by authorized parties
  - ◆ beyond conforming to originally-prescribed “write” access-control rules
  - ◆ precise, accurate, unmodified, modified in acceptable way by authorized people or processes, consistent, meaningful and usable
  - ◆ authorized actions, separation & protection of resources, error detection & correction
  - ◆ some tools
    - ◆ hashing, MACs

# Availability

- ◆ An asset can be used by any authorized party
  - ◆ usable, meets service's needs, bounded waiting/completion time, acceptable outcome
  - ◆ timely response, fairness, concurrency, fault tolerance, graceful cessation (if needed)
  - ◆ some tools
    - ◆ redundancy, fault tolerance, distributed architectures

# Authenticity

- ◆ The ability to determine that statements, policies, and permissions issued by persons or systems are genuine
  - ◆ some tools
    - ◆ digital signatures (cryptographic computations that allow entities to commit to the authenticity of their documents in a unique way)
      - ◆ achieve non-repudiation (authentic statements issued by some person or system cannot be denied)



# Anonymity

- ◆ The property that certain records/transactions cannot be attributed to any individual
- ◆ some tools
  - ◆ aggregation
    - ◆ disclosure of statistics on combined data from many individuals that cannot be tied to any individual
  - ◆ proxies
    - ◆ trusted agents interacting on behalf of an individual in an untraceable way
  - ◆ pseudonyms
    - ◆ fictional identities, known only to a trusted party, that fill in for real identities





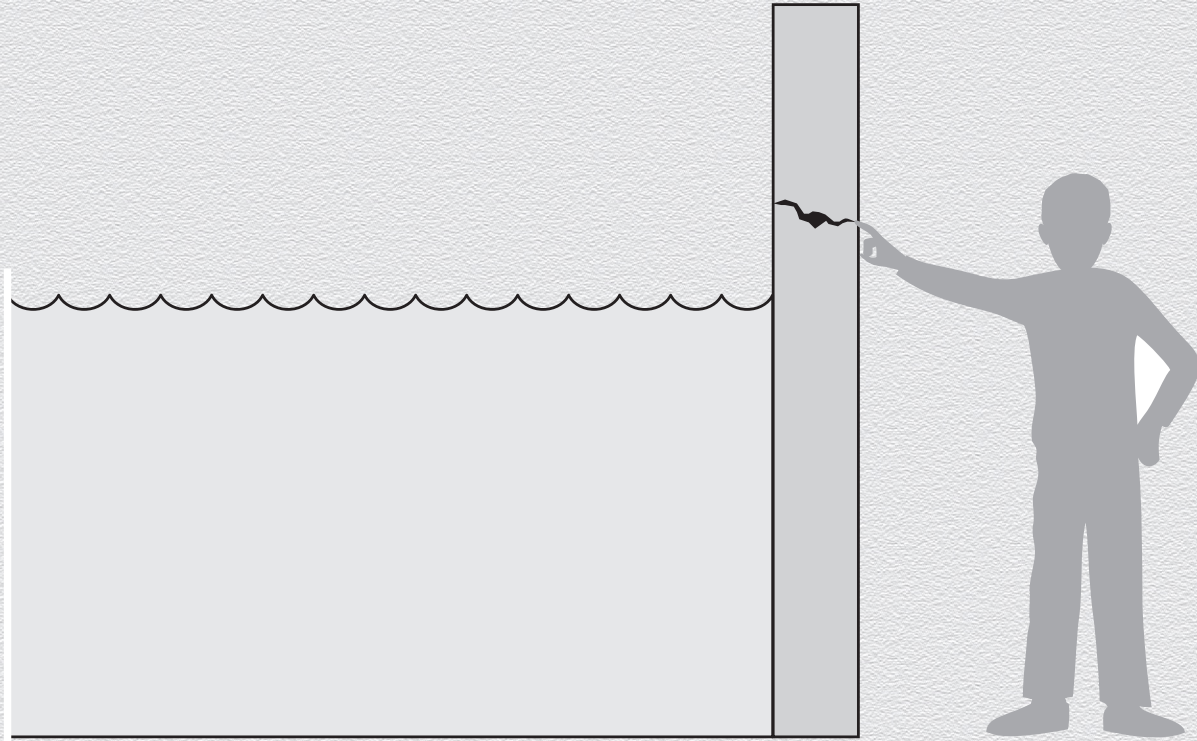
# The “Vulnerability - Threat - Control” paradigm

- ◆ A **vulnerability** is a weakness that could be exploited to cause harm
- ◆ A **threat** is a set of circumstances that could cause harm
- ◆ A **security control** is a mechanism that protects against harm
  - ◆ i.e., countermeasures designed to prevent threats from exercising vulnerabilities

Thus

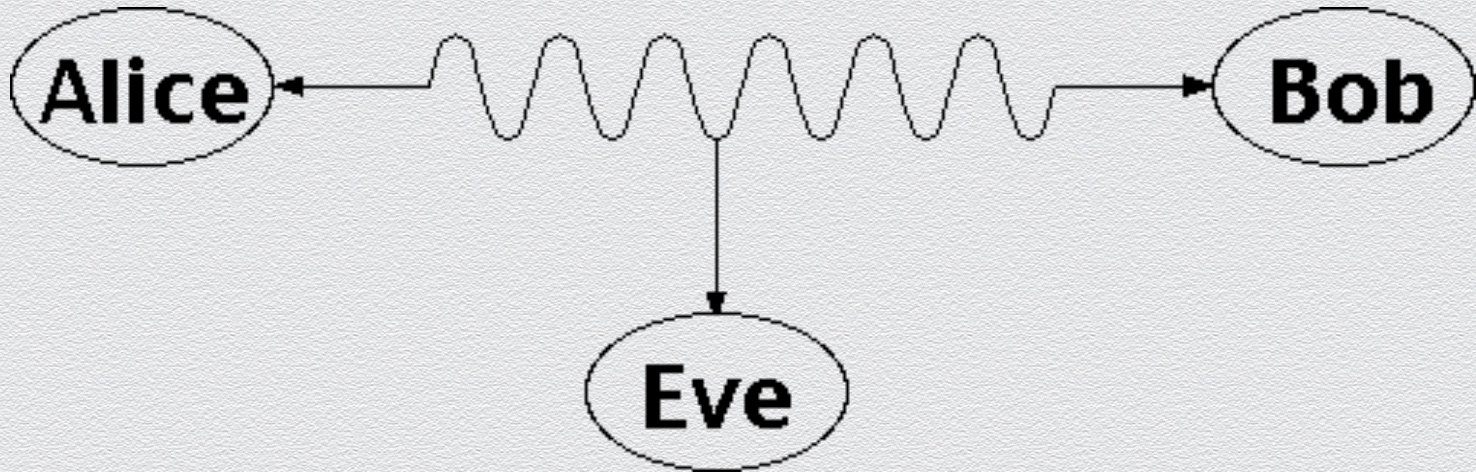
- ◆ **Attackers** seek to **exploit** vulnerabilities in order to **impose** threats
- ◆ **Defenders** seek to **block** these threats by **controlling** the vulnerabilities

# A “Vulnerability - Threat - Control” example



## Example of threat

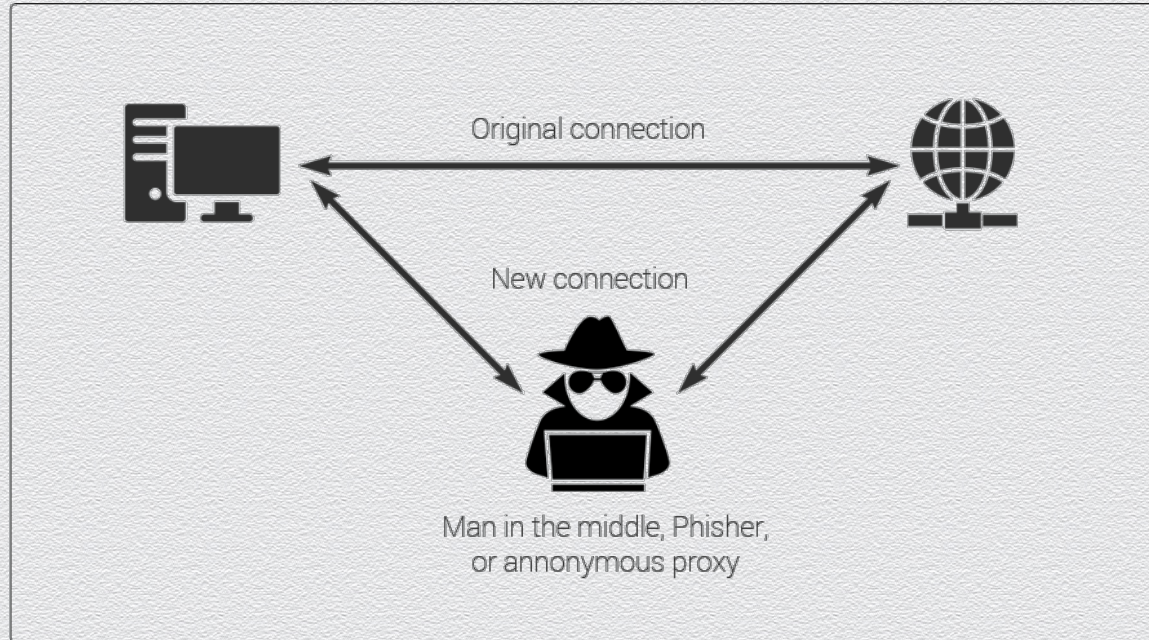
**Eavesdropping:** Interception of information intended for someone else during its transmission over a communication channel



# Example of threat

**Alteration:** Unauthorized modification of information

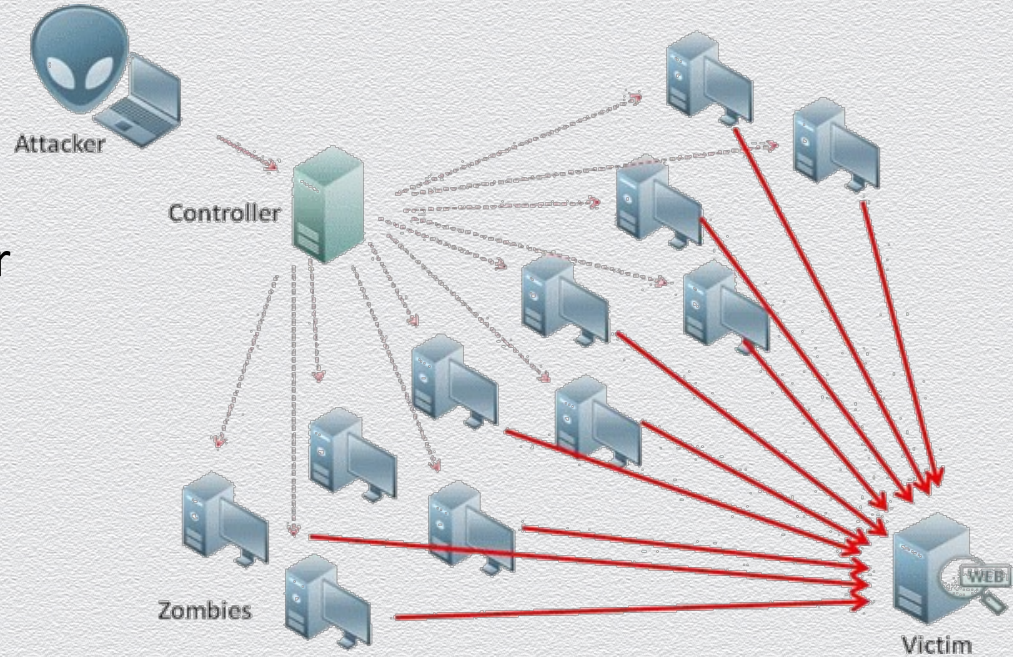
- ◆ **Example:** the attacker-in-the-middle attack, where a network stream is
  - ◆ intercepted and
    - ◆ modified and retransmitted; or
    - ◆ dropped



# Example of threat

**Denial-of-service:** Interruption or degradation of a data service or information access

- ◆ **Example:** email spam, to the degree that it is meant to simply fill up a mail queue and slow down an email server



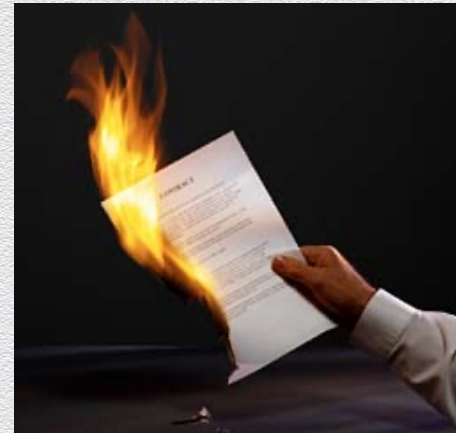
# Examples of threats

**Masquerading:** Fabrication of information that is purported to be from someone who is not actually the author

- ◆ e.g., IP spoofing attack: maliciously altering the source IP address of a message

**Repudiation:** Denial of a commitment or data receipt

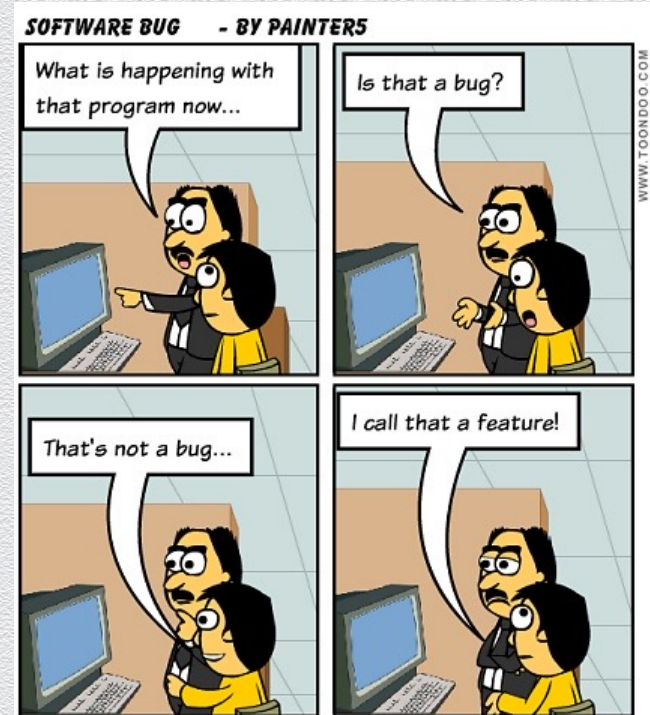
- ◆ an attempt to back out of a contract/protocol that, e.g., requires the different parties to provide receipts acknowledging that data has been received



# Example of vulnerability

**Software bugs:** Code is not doing what is supposed to be doing

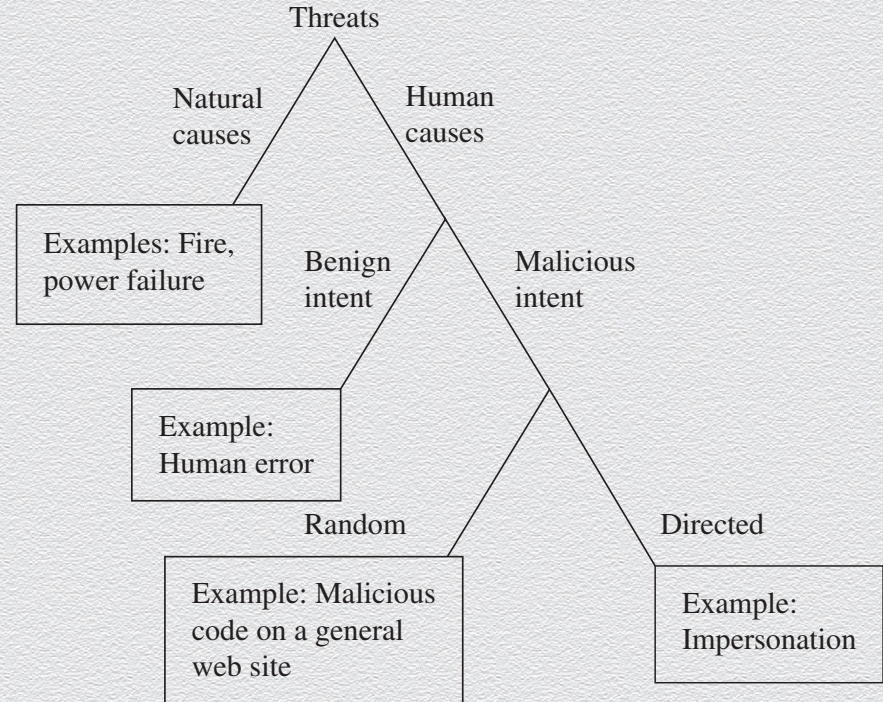
- ◆ **Example:** Some application code is mistakenly using an algorithm for encryption that has been broken
- ◆ **Example:** There is no checking of array bounds



# A hard-to-win game: Varied threats

## Threats

- ◆ from natural to human
- ◆ from benign to malicious
- ◆ from random to targeted (APTs)

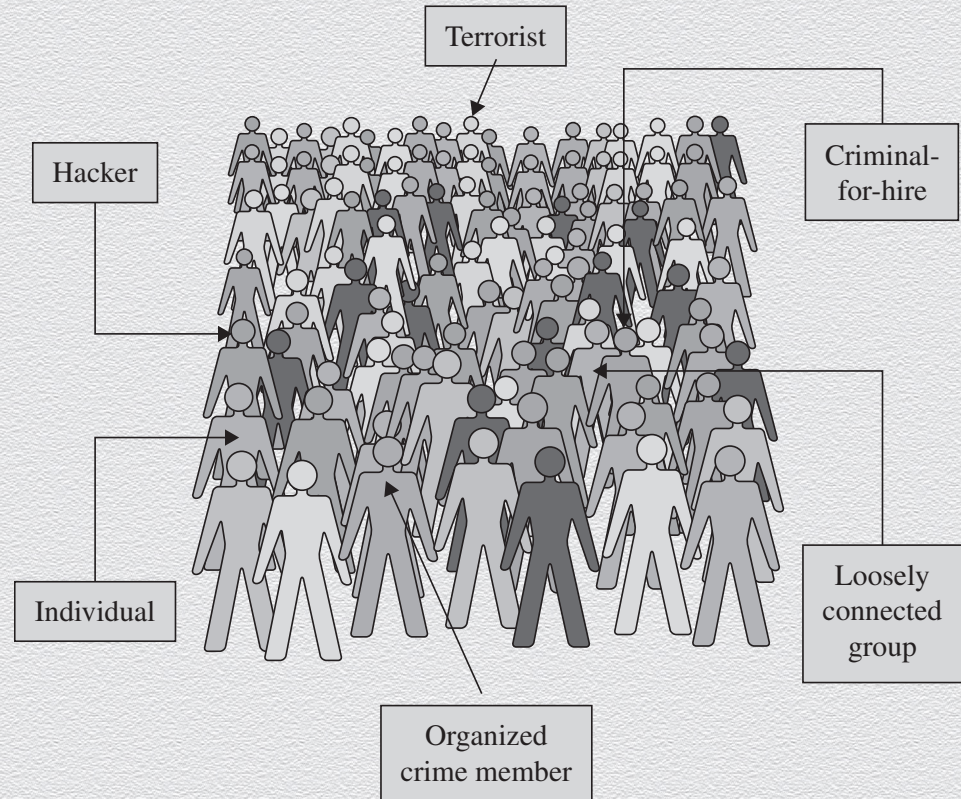




# A hard-to-win game: Unknown enemy

## Attackers

- ◆ beyond isolated “crazy” hackers
- ◆ organized groups/crime
  - ◆ may use computer crime (e.g., stealing CC#s) in order to finance other crimes
- ◆ terrorists
  - ◆ computers/assets as target, method, enabler, or enhancer

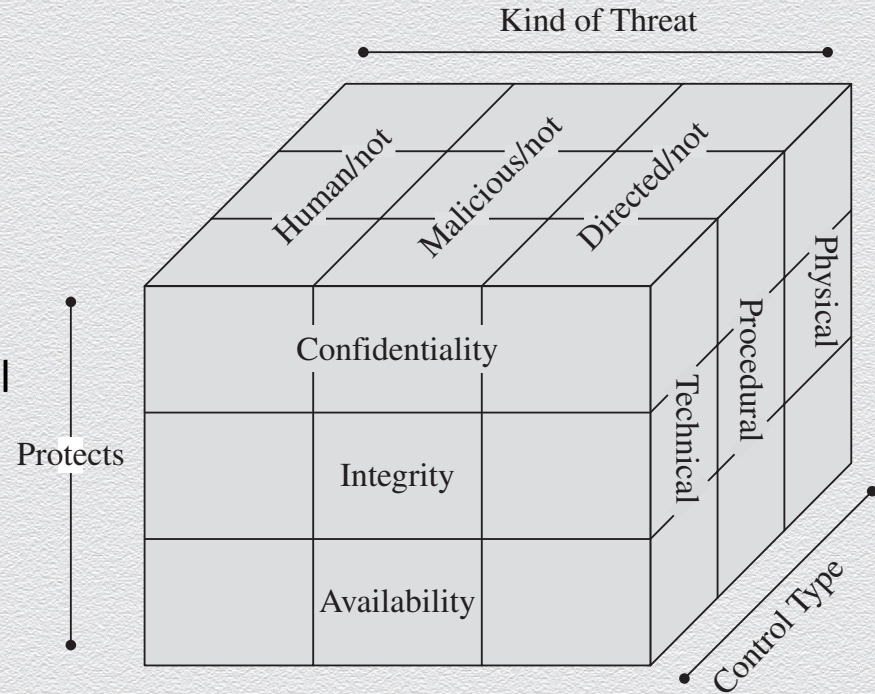


# A hard-to-win game: Choose your battle

## Risk management

- ◆ choose priorities
  - ◆ which threats to control
    - ◆ estimate possible harm & impact
  - ◆ what / how many resources to devote
    - ◆ estimate solution cost & protection level
- ◆ consider trade-offs balancing cost Vs. benefit
- ◆ compute the residual risk
  - ◆ decide on transferring risk or doing nothing

Never a “one-shot” game



# A hard-to-win game: Best-effort approach

Deciding on controls relies on incomplete information

- ◆ likelihood of attack and impact of possible harm is impossible to measure perfectly
- ◆ full set of vulnerabilities is often unknown
  - ◆ weak authentication, lack of access control, errors in programs, etc.
- ◆ system's attack surface is often too wide
  - ◆ physical hazards, malicious attacks, stealthy theft by insiders, benign mistakes, impersonations, etc.

A useful strategy: The “method – opportunity – motive” view of an attack

- ◆ **deny any of them and the attack will (likely) fail**

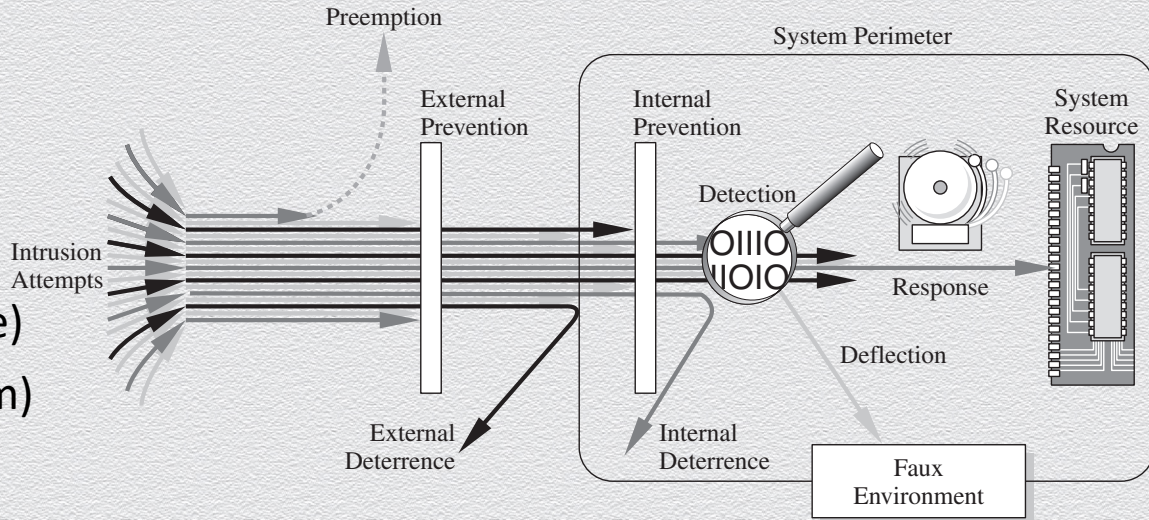
# A hard-to-win game: Best-effort approach (cont.)

Controls offer a wide range of protection level / efficacy

- ◆ they counter or neutralize threats or remove vulnerabilities in different ways

## Types of controls

- ◆ prevent (attack is blocked)
- ◆ deter (attack becomes harder)
- ◆ deflect (change target of attack)
- ◆ mitigate (make impact less severe)
- ◆ contain (stop propagation of harm)
- ◆ detect (real time/after the fact)
- ◆ recover (from its effects)



Hard to balance cost/effectiveness of controls with likelihood/severity of threats

# A hard-to-win game: Security tradeoffs

Often complete security against all conceivable adversaries is unfeasible

- ◆ More often than not, tradeoffs are unavoidable
  - ◆ Risk mitigation Vs. cost of deploying defense mechanisms
    - ◆ Here, cost refers to many other aspects (beyond monetary expenses)
    - ◆ Human factors, e.g., user acceptance and usability of defense mechanisms

# Example of control: HTTPS protocol

## Hypertext Transfer Protocol Secure (HTTPS)

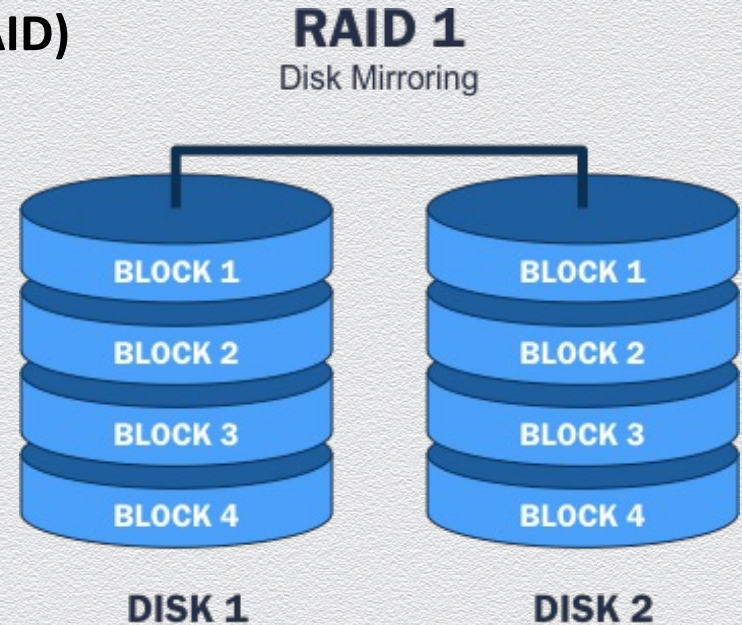
- ◆ Confidentiality
- ◆ Integrity
- ◆ Availability
- ◆ Authenticity
- ◆ Anonymity



# Example of control: RAID technology

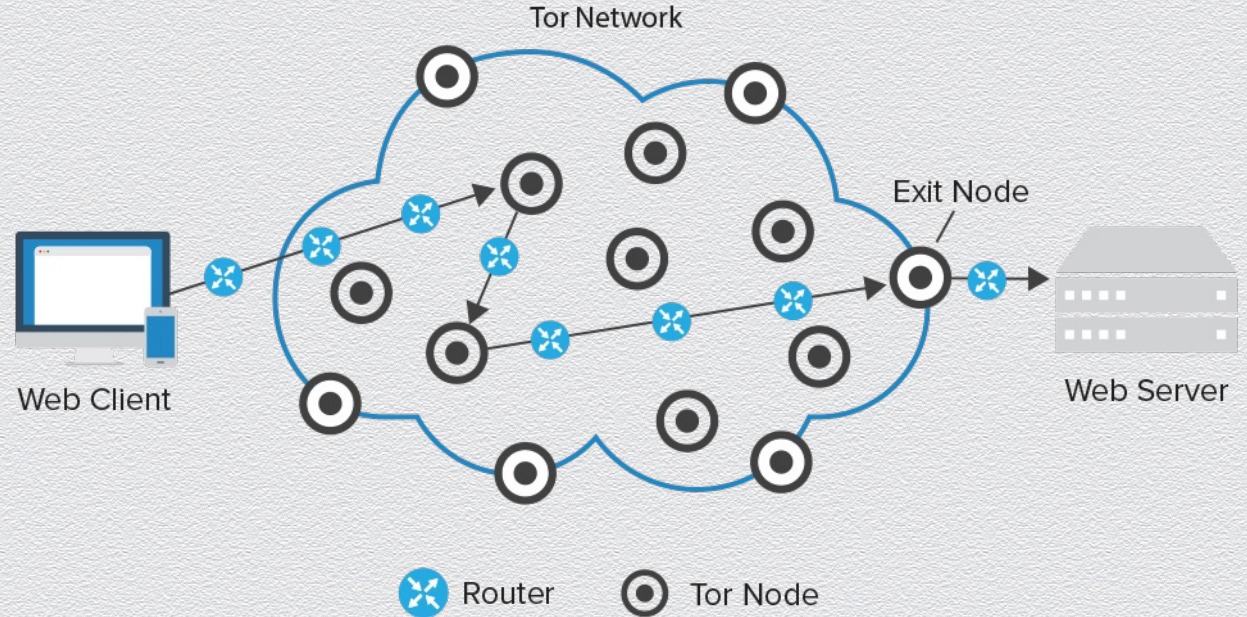
## Redundant Array of Independent Disks (RAID)

- ◆ Confidentiality
- ◆ Integrity
- ◆ Availability
- ◆ Authenticity
- ◆ Anonymity



# Example of control: TOR protocol

- ◆ Confidentiality
- ◆ Integrity
- ◆ Availability
- ◆ Authenticity
- ◆ Anonymity





# Exciting times to study (or work in) Security!

## Relevance to practice & real-world importance

- ◆ plethora of real-world problems & real needs for security solutions
- ◆ combination of different research areas within CS and across other fields
- ◆ multi-dimensional topic of study
  - ◆ protocol design, system building, user experience, social/economic aspects
- ◆ wide range of perspectives
  - ◆ practical / systems – foundations / theory, attacker's Vs. defender's view

## 2.2 Symmetric-key encryption

# Confidentiality

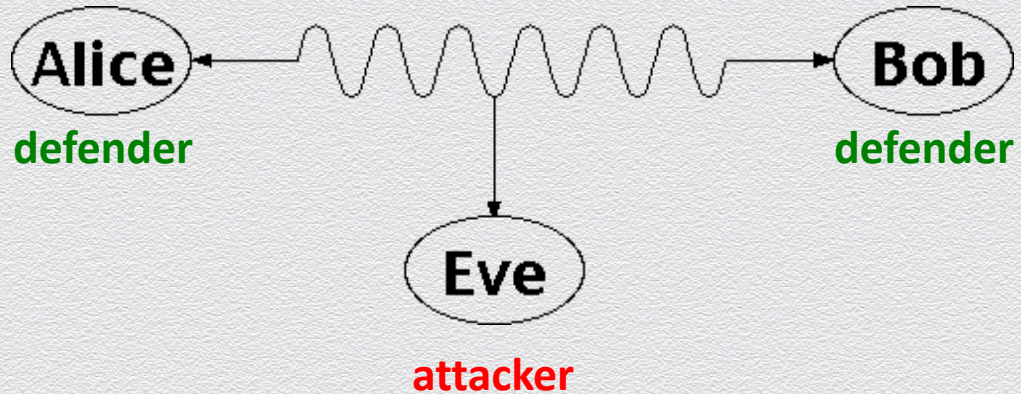
Fundamental security property

- ◆ **an asset is viewed only by authorized parties**
- ◆ “C” in the CIA triad

*“computer security seeks to prevent **unauthorized viewing (confidentiality)** or **modification (integrity)** of **data** while preserving access (**availability**)”*

## Eavesdropping

- ◆ main threat against confidentiality of **in-transit** data



# Problem setting: Secret communication

Two parties wish to communicate over a channel

- ◆ Alice (sender/source) wants to send a message  $m$  to Bob (recipient/destination)

Underlying channel is unprotected

- ◆ Eve (attacker/adversary) can eavesdrop any sent messages
- ◆ e.g., packet sniffing over networked or wireless communications



# Solution concept: Symmetric-key encryption

## Main idea

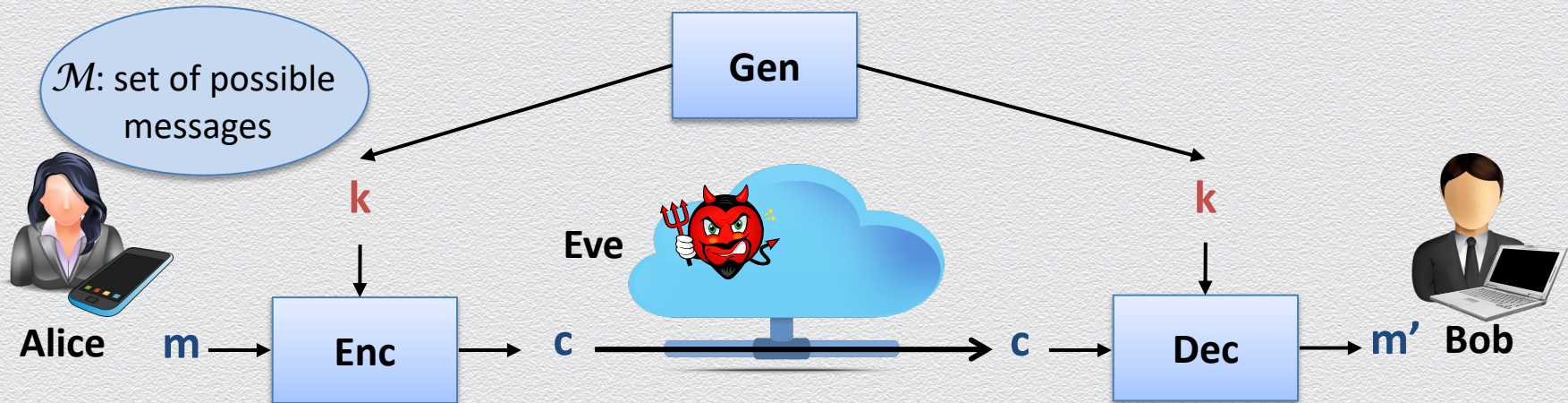
- ◆ secretly transform message so that it is **unintelligible** while in transit
  - ◆ Alice **encrypts** her message  $m$  to **ciphertext**  $c$ , which is sent instead of **plaintext**  $m$
  - ◆ Bob **decrypts** received message  $c$  to original message  $m$
  - ◆ Eve can intercept  $c$  but “**cannot learn**”  $m$  from  $c$
  - ◆ Alice and Bob share a **secret key**  $k$  that is used for both message transformations



# Security tool: Symmetric-key encryption scheme

Abstract cryptographic primitive, a.k.a. **cipher**, defined by

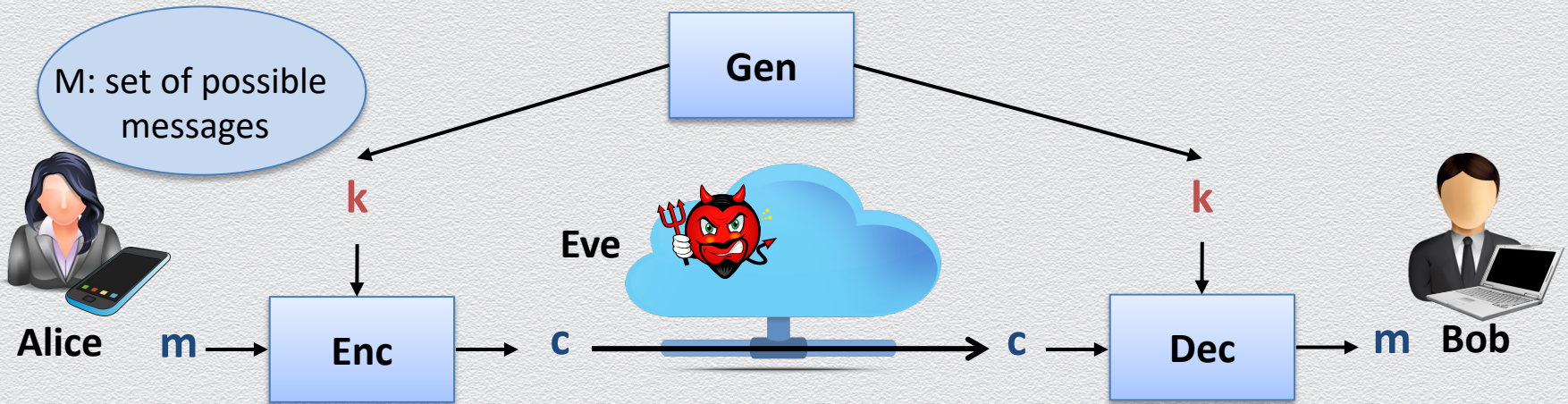
- ◆ a **message space**  $\mathcal{M}$ ; and
- ◆ a triplet of algorithms (**Gen**, **Enc**, **Dec**)
  - ◆ Gen is randomized algorithm, Enc may be randomized, whereas Dec is deterministic
  - ◆ Gen outputs a uniformly random key  $k$  (from some key space  $\mathcal{K}$ )



# Desired properties for symmetric-key encryption scheme

By design, any symmetric-key encryption scheme should satisfy the following

- ◆ **efficiency:** key generation & message transformations “are fast”
- ◆ **correctness:** for all  $m$  and  $k$ , it holds that  $\text{Dec}(\text{Enc}(m, k), k) = m$
- ◆ **security:** one “cannot learn” plaintext  $m$  from ciphertext  $c$



# (Auguste) Kerckhoff's principle (1883)

*"The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience."*



## Reasoning

- ◆ due to security & correctness, Alice & Bob must share some secret info
- ◆ if no shared key captures this secret info, it must be captured by Enc, Dec
- ◆ but keeping Enc, Dec secret is problematic
  - ◆ harder to keep secret an algorithm than a short key (e.g., after user revocation)
  - ◆ harder to change an algorithm than a short key (e.g., after secret info is exposed)
  - ◆ riskier to rely on custom/ad-hoc schemes than publicly scrutinized/standardized ones



# (Auguste) Kerckhoff's principle (1883)

*“The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.”*

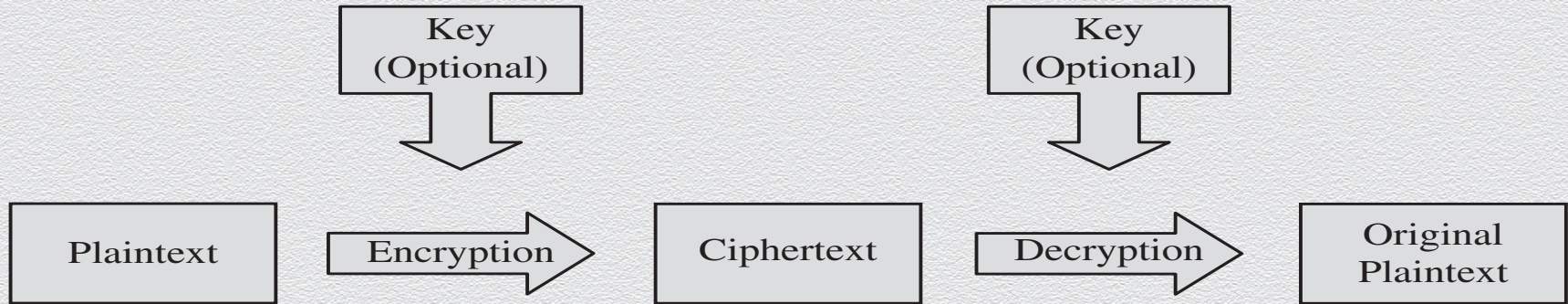
General good-hygiene principle (beyond encryption)

- ◆ Security relies solely on keeping secret keys
- ◆ System architecture and algorithms are publicly available
- ◆ Claude Shannon (1949): *“one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them”*
- ◆ Opposite of “security by obscurity” practice

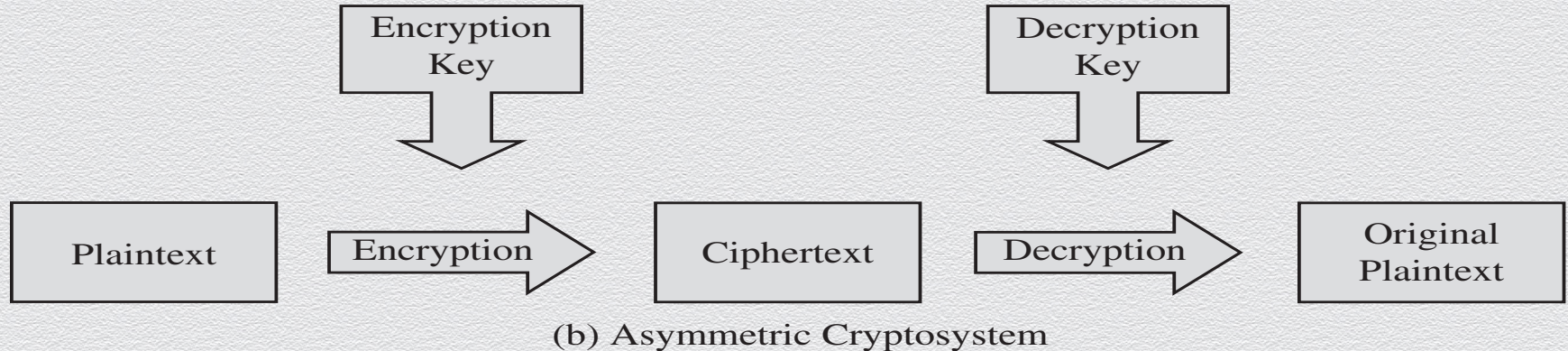
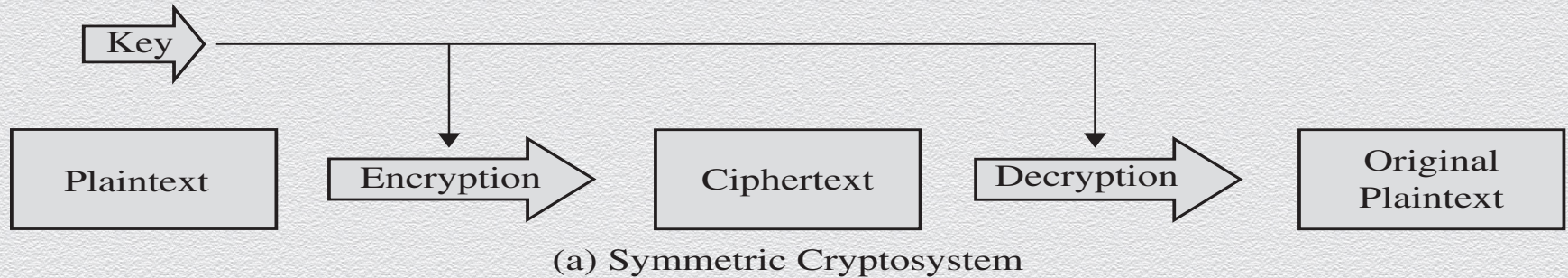


# Symmetric-key encryption

- ◆ Also referred to as simply “symmetric encryption”



# Symmetric Vs. Asymmetric encryption



# Main application areas

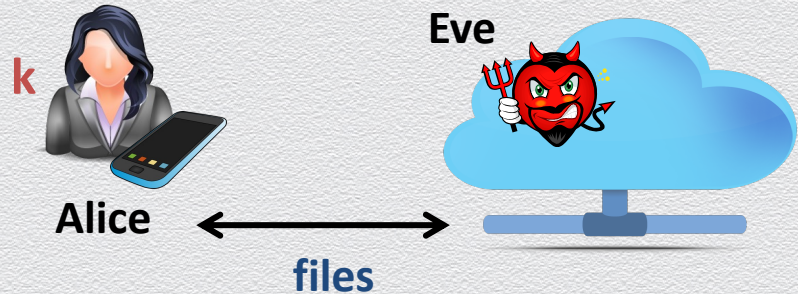
## Secure communication

- ◆ **encrypt messages** sent among parties
- ◆ assumption
  - ◆ Alice and Bob **securely generate, distribute & store shared key  $k$**
  - ◆ attacker does not learn key  $k$



## Secure storage

- ◆ **encrypt files** outsourced to the cloud
- ◆ assumption
  - ◆ Alice **securely generates & stores key  $k$**
  - ◆ attacker does not learn key  $k$



# Brute-force attack

## Generic attack

- ◆ given a captured ciphertext  $c$  and known key space  $\mathcal{K}$ , Dec
- ◆ strategy is an **exhaustive search**
  - ◆ for all possible keys  $k$  in  $\mathcal{K}$ 
    - ◆ determine if  $\text{Dec}(c,k)$  is a likely plaintext  $m$
- ◆ **requires some knowledge on the message space  $\mathcal{M}$** 
  - ◆ i.e., structure of the plaintext (e.g., PDF file or email message)

## Countermeasure

- ◆ key should be a **random** value from a **sufficiently large** key space  $\mathcal{K}$  to make exhaustive search attacks **infeasible**

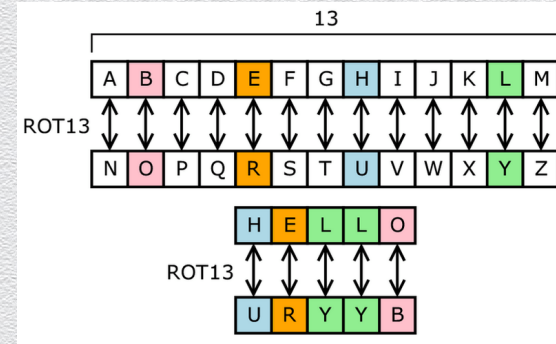


## 2.3 Classical ciphers

# Substitution ciphers

## Large class of ciphers

- ◆ each letter is uniquely replaced by another
- ◆ there are  $26!$  possible substitution ciphers
  - ◆ e.g., one popular substitution “cipher” for some Internet posts is ROT13
- ◆ historically
  - ◆ all classical ciphers are of this type



# General structure of classical ciphers

## Based on letter substitution

- ◆ message space  $\mathcal{M}$  is “valid words” from a given alphabet
  - ◆ e.g., English text without spaces, punctuation or numerals
  - ◆ characters can be represented as numbers in  $[0:25]$
- ◆ encryption
  - ◆ mapping each plaintext character into another character
  - ◆ character mapping is typically defined as a “shift” of a plaintext character by a number of positions in a canonical ordering of the characters in the alphabet
  - ◆ character shifting occurs with “wrap-around” (using mod 26 addition)
- ◆ decryption
  - ◆ undo character shifting with “wrap-around” (using mod 26 subtraction)

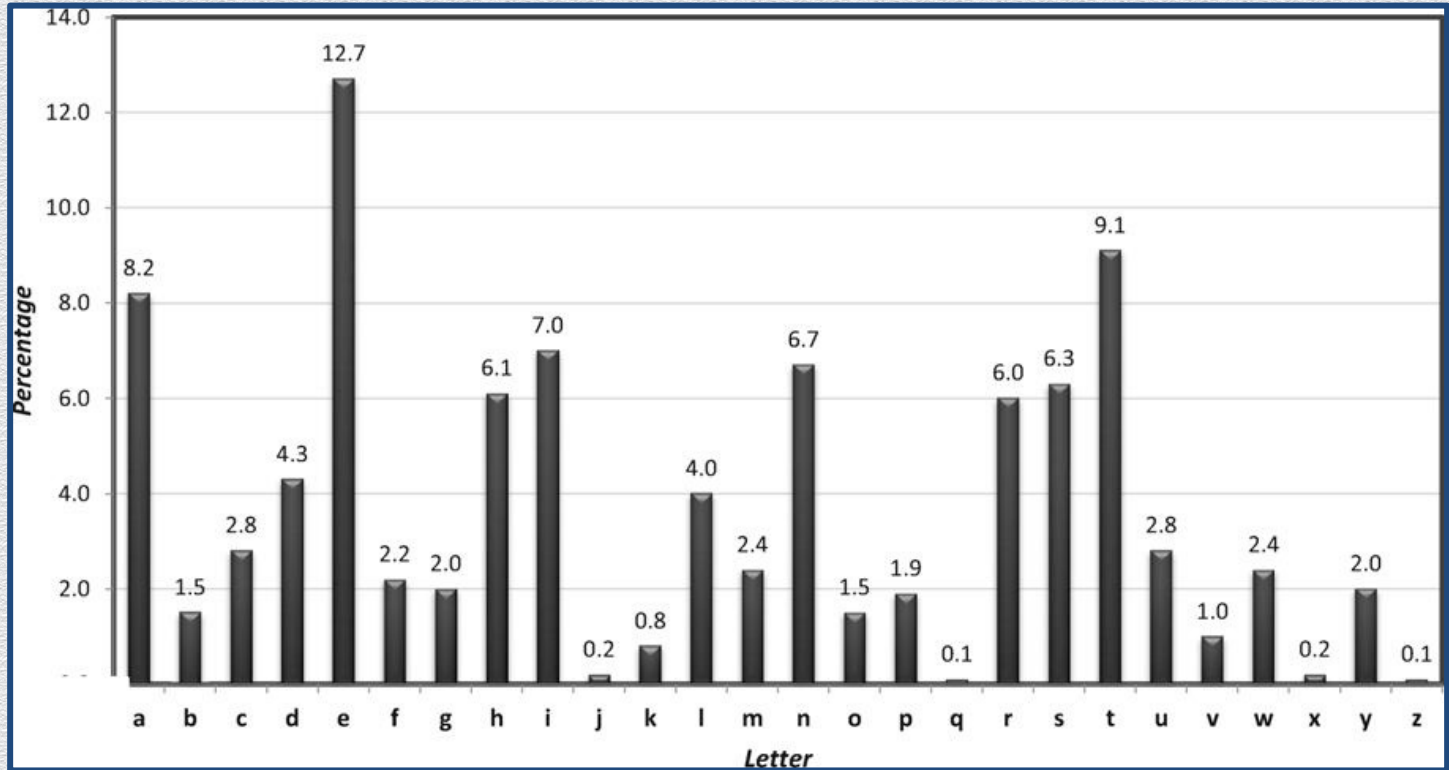


# Limitations of substitution ciphers

Generally, susceptible to frequency (and other statistical) analysis

- ◆ letters in a natural language, like English, are not uniformly distributed
- ◆ cryptographic attacks against substitution ciphers are possible
  - ◆ e.g., by exploiting knowledge of letter frequencies, including pairs and triples

# Letter frequency in (sufficiently large) English text



# Classical ciphers – examples

## Caesar's cipher

- ◆ shift each character in the message by 3 positions
  - ◆ or by 13 positions in ROT-13
- ◆ cryptanalysis
  - ◆ **no secret key is used** – based on “security by obscurity”
  - ◆ thus the code is trivially insecure once knows Enc (or Dec)

# Classical ciphers – examples (II)

## Shift cipher

- ◆ **keyed extension** of Caesar's cipher
- ◆ randomly set key  $k$  in  $[0:25]$ 
  - ◆ shift each character in the message by  $k$  positions
- ◆ cryptanalysis
  - ◆ **brute-force attacks** are effective given that
    - ◆ **key space is small** (26 possibilities or, actually, 25 as 0 should be avoided)
    - ◆ message space  $M$  is **restricted to “valid words”**
      - ◆ e.g., corresponding to valid English text