# CS306: Introduction to IT Security



Fall 2020



## Lecture 2: Symmetric-key Encryption



Instructor: **Nikos Triandopoulos**

September 8, 2020


STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

## 2.0 Announcements

# CS306: Staff

- Instructor
  - Nikos Triandopoulos, ntriando@stevens.edu
  - course organization / management, lectures, assignments, grades, …
    - all mistakes will be also mine ☺
  - office hours: **Thursdays 1 – 2pm** (**Zoom ID 91463728672**) or by appointment
- **Teaching assistants**
  - **Dean Rodman (drodman@stevens.edu), Devharsh Trivedi (dtrived5@stevens.edu), Joseph Iervasi (jiervasi@stevens.edu), Mohammad Khan (mkhan13@stevens.edu), Joshua Mimer (jmimer@stevens.edu), Uday Samavenkata (usamaven@stevens.edu)**
  - **assistance w/ labs, assignments, help sessions, grading, demos, …**

# CS306: TA hours

- Standard schedule, starting from tomorrow

SAME ZOOM ID

| Day | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|
| time | 13:00 – 14:00 | 13:00 – 14:00 | 13:00 – 14:00 | 13:00 – 14:00 | 13:00 – 14:00 |
| Zoom ID | 91463728672 | 91463728672 | 91463728672 | 91463728672 | 91463728672 |
| staff | Dean | Joshua | Joseph | **Nikos** | Uday |



- Additional TA hours to be added for homework assignments or before exams

# CS306: Lectures & labs

CS306 is offered in **2 required sessions**, each offered in **multiple sections**

- lectures
  - CS306-A Tue 2:00pm - 4:30pm **Online** 67 / 69
  - CS306-B Tue 6:30pm - 9:00pm **Online** 63 / 69 **last week**
- labs
  - CS306-Lx Thursdays



| x | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| time | 8 - 8:50 | 9:30 - 10:20 | 11:00 - 11:50 | 12:30 - 13:20 | 2:00 - 2:50 | 3:30 - 4:20 |
| enrollment | 1 | 18 | 29 | 29 | 29 | 24 |

# CS306: Lectures & labs (continued)

CS306 is offered in **2 required sessions**, each offered in **multiple sections**

- lectures
  - CS306-A Tue 2:00pm - 4:30pm **Online** 67 / 69
  - CS306-B Tue 6:30pm - 9:00pm **Online** 62 / 69
- labs
  - CS306-Lx Thursdays



**this week**

| x | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| time | 8 - 8:50 | 9:30 - 10:20 | 11:00 - 11:50 | 12:30 - 13:20 | 2:00 - 2:50 | 3:30 - 4:20 |
| enrollment | 1 | 17 | 29 | 28 | 28 | 26 |

# CS306: Lectures & labs (continued)

- Lecture/lab sections will cover the same materials
- Changes in lecture or lab sections
  - **allowed** (if need be), but **generally discouraged** (for planning purposes)
- In any case, if a section change is necessary
  - **students must let the TAs or instructor know well in advance**

# Our on-going semester-long project…

- Lectures take place in 2.5h slots
  - CS306-A Tue 2:00pm - 4:30pm **Online** 67 / 69
  - CS306-B Tue 6:30pm - 9:00pm **Online** 62 / 69
- Highly problematic & undesirable for both students & instructor
  - unfortunately unavoidable due to existing scheduling restrictions
- **Tentative countermeasures**
  - **two ~10-min breaks**
  - **Spending last 30min with demos, special topics of interest or offline materials**



**Please provide suggestions on what can make class experience better despite 2.5h lectures**

# CS306: Lab sections schedule

- labs
  - CS306-Lx Thursdays

**ZOOM ID: LAB SPECIFIC!**



| x | B | C | D | E | F |
|---|---|---|---|---|---|
| time | 9:30 - 10:20 | 11:00 - 11:50 | 12:30 - 13:20 | 14:00 - 14:50 | 15:30 - 16:20 |
| Zoom ID | 91573945614 | 93061161569 | 94976630644 | 92834271191 | 94520991826 |
| TAs | Dean, Joseph, Joshua, Uday | Dean, Devharsh, Joseph, Joshua | Dean/Devharsh, Joshua, Mohammad, Uday | Devharsh, Joseph, Mohammad, Uday | Dean, Joseph, Mohammad, Uday |

# CS306: Other announcements

- Canvas course materials are now updated
- Lab sessions start this week
- TA hours & office hours start tomorrow

# CS306: Tentative Syllabus

| Week | Date | Topics | Reading | Assignment |
|---|---|---|---|---|
| 1 | Sep 1 | Introduction | Lecture 1 | - |
| 2 | Sep 8 | Symmetric-key crypto I | | |
| 3 | Sep 15 | Symmetric-key crypto II | | |
| 4 | Sep 22 | Public-key crypto I | | |
| 5 | Sep 29 | Public-key crypto II | | |
| 6 | Oct 6 | Access control & authentication | | |
| - | Oct 13 | **No class (Monday schedule)** | | |
| 7 | Oct 20 | **Midterm** | All materials covered | |

# CS306: Tentative Syllabus (continued)

| Week | Date | Topics | Reading | Assignment |
|---|---|---|---|---|
| 8 | Oct 27 | Software & Web security | | |
| 9 | Nov 3 | Network security | | |
| 10 | Nov 10 | Database security | | |
| 11 | Nov 17 | Cloud security | | |
| 12 | Nov 24 | Privacy | | |
| 13 | Dec 1 | Economics | | |
| 14 | Dec 8 | Legal & ethical issues | | |
| 15 | Dec 10 (or later) | **Final** (closed "books") | All materials covered* | |





* w/ focus on what covered after midterm

# CS306: Course outcomes

- Terms
  - describe common security terms and concepts
- Cryptography
  - state basics/fundamentals about secret and public key cryptography concepts
- Attack & Defense
  - acquire basic understanding for attack techniques and defense mechanisms
- Impact
  - acquire an understanding for the broader impact of security and its integral connection to other fields in computer science (such as software engineering, databases, operating systems) as well as other disciplines including STEM, economics, and law
- Ethics
  - acquire an understanding for ethical issues in cyber-security

# Questions?

- Please ask questions during class!

# Last week

- Course logistics
  - topic of study, enrollment eligibility, sessions
  - staff, learning materials, course organization
  - expectations, grading, policies, announcements
  - syllabus overview, course objectives/outcomes
- Introduction to the field of IT security
  - in-class discussion with a real-world example

# Today

- Introduction to the field of IT security
  - Basic concepts and terms
  - Symmetric encryption

# 2.1 Basic security concepts & terms

# What is IT security?

**IT 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).



**IT 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 ‘IT-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 ‘IT-security’ game: Who are the players?

- **Defenders**
  - system owners (e.g., users, administrators, etc.)
  - seek to **enforce** one or more **security properties**  **property-based view**

    or **defeat** certain **attacks**



- **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**
    - **non-repudiation / accountability / auditability**
    - **anonymity**

# The C-I-A triad

- Captures the three fundamental properties that make any system valuable

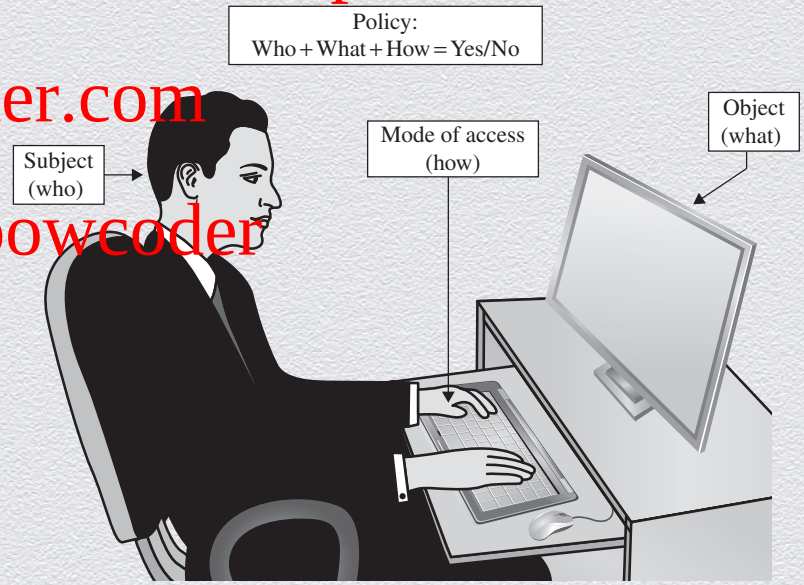


Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data while preserving access (availability)

# 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 on an individual in untraceable way
  - pseudonyms
    - fictional identities, known only to a trusted party, that fill in for real identities

# Discussion

1. Cloud-based storage
2. e-banking
   - What is a **valued asset**?
   - What does it mean to **preserve** this value?
   - What is a corresponding desired **security property**?
   - What is a **harm** that must be prevented?

# The “Vulnerability - Threat - Control” paradigm

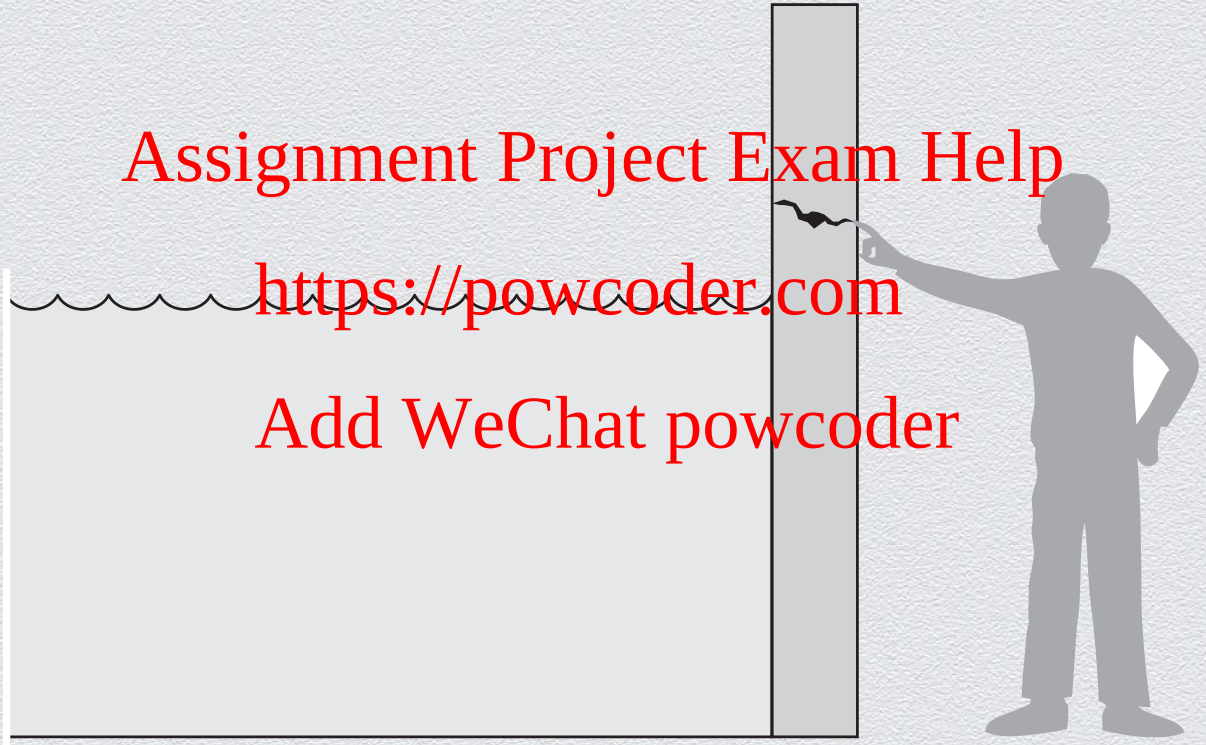
- A **vulnerability** is a weakness that could be exploited to cause harm
- A **threat** is a set of circumstances that could cause harm
- A **security control** is a mechanism that protects against harm
  - i.e., countermeasures designed to prevent threats from exercising vulnerabilities



Thus

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

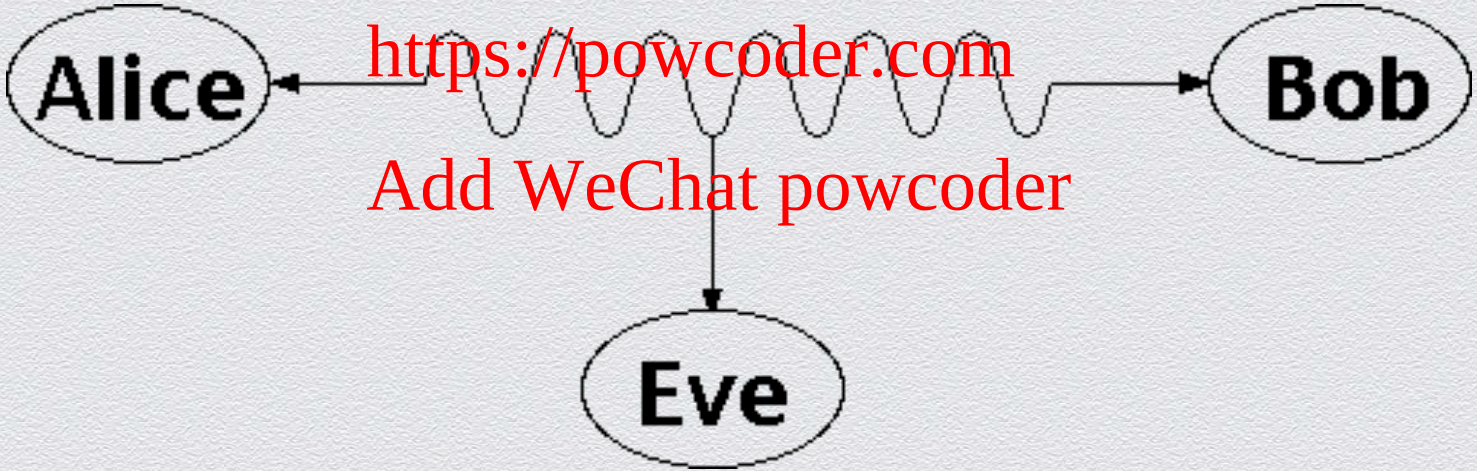
# A “Vulnerability - Threat - Control” example



# Example of threat

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

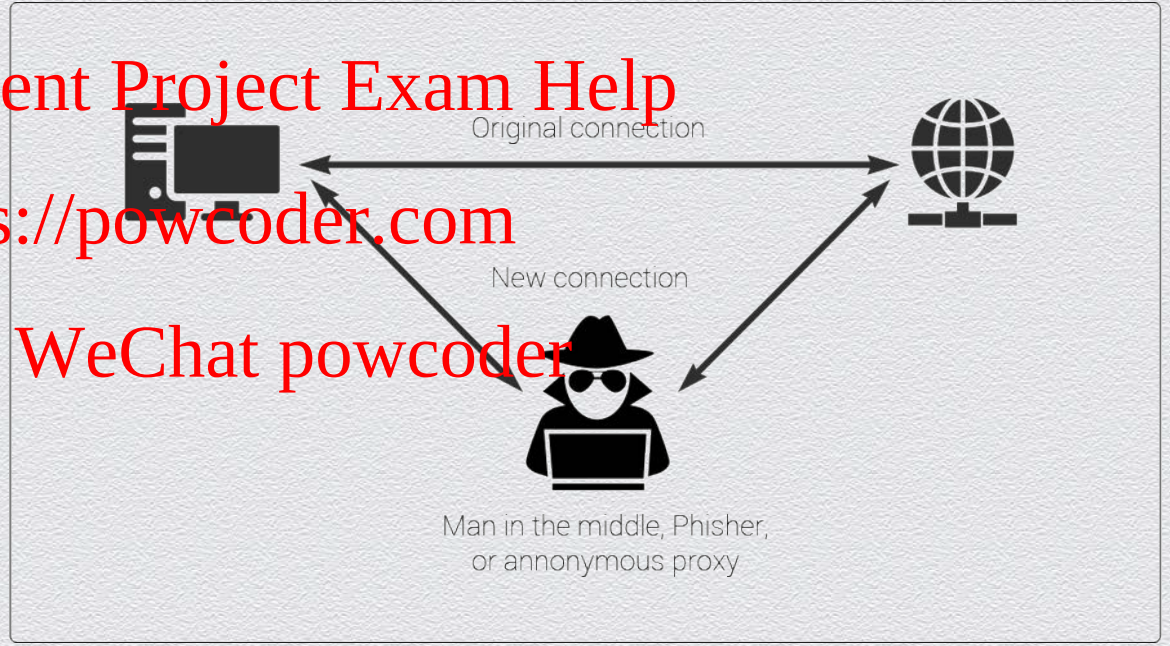




# Example of threat

- **Alteration:** unauthorized modification of information
  - **Example:** the man-in-the-middle attack, where a network stream is intercepted, modified, and retransmitted

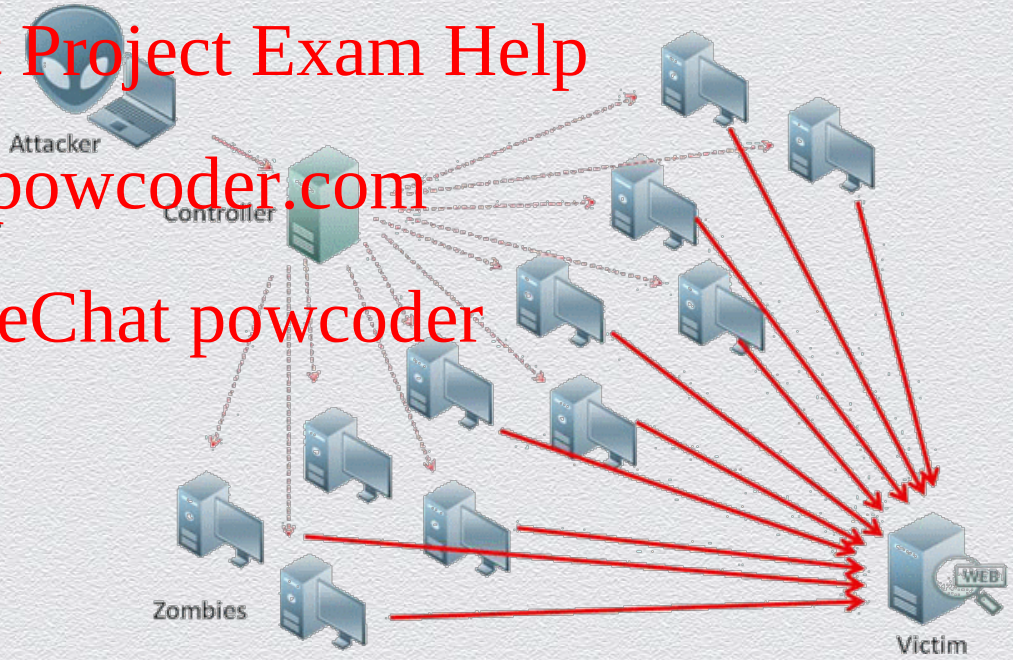




# Example of threat

- **Denial-of-service:** the 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



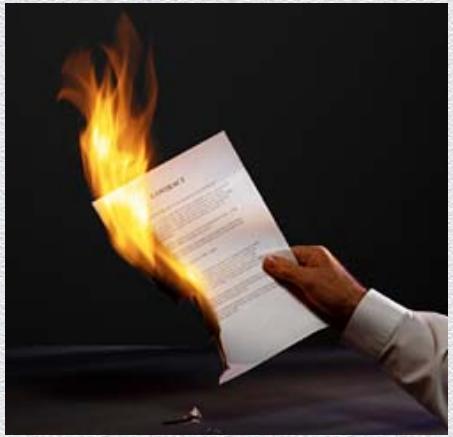


# Examples of threats

- **Masquerading**: the 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:** the denial of a commitment or data receipt
  - this involves 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

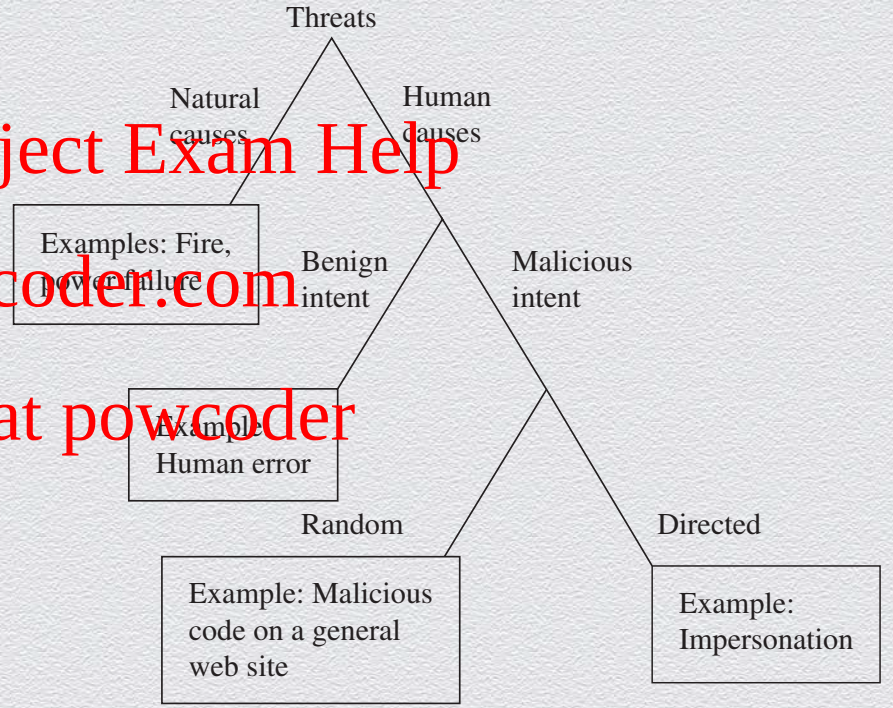
- **Software bugs:** Code is not doing what is supposed to be doing
  - **Example:** Some application code is mistakenly using an algorithm for encryption that has been broken
  - **Example:** There is no checking of array bounds



# An hard-to-win game: Varied threats

Threats

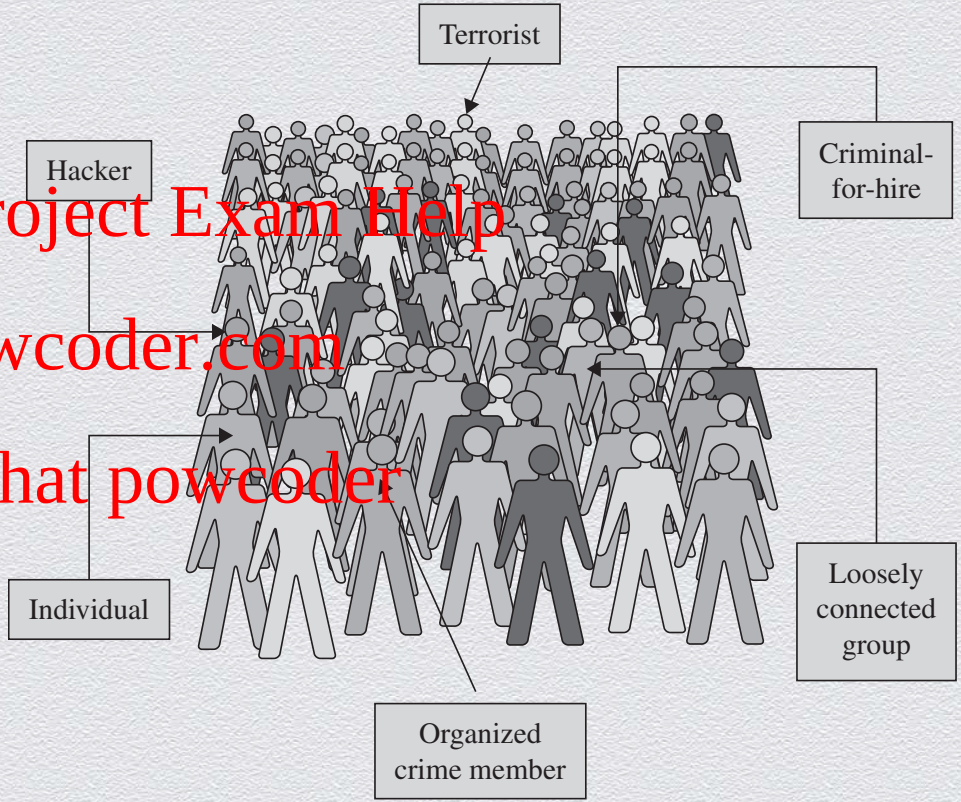
- from natural to human
- from benign to malicious
- from random to targeted (APTs)



# 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

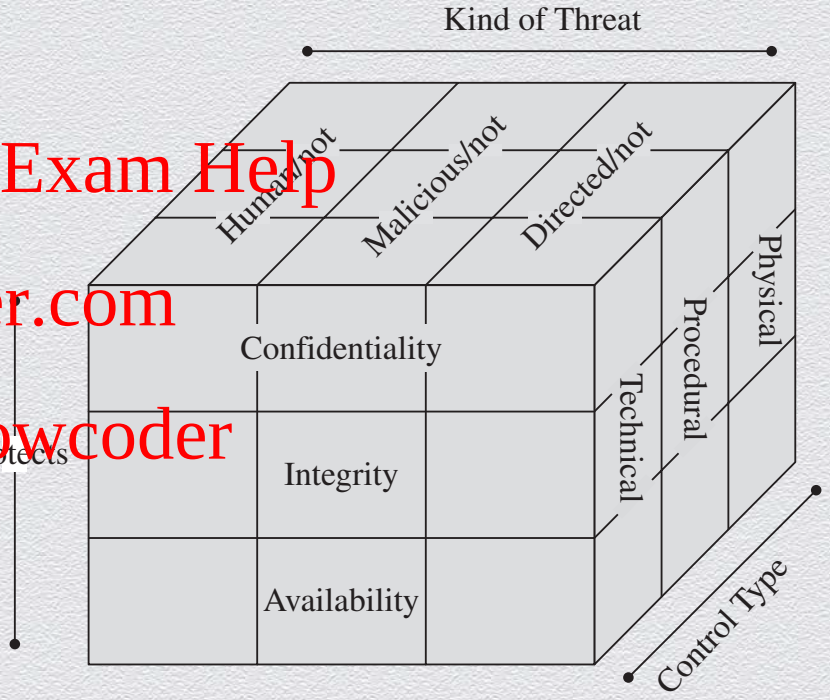


# A hard-to-win game: Choose your battle

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 transfering 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 (continued)

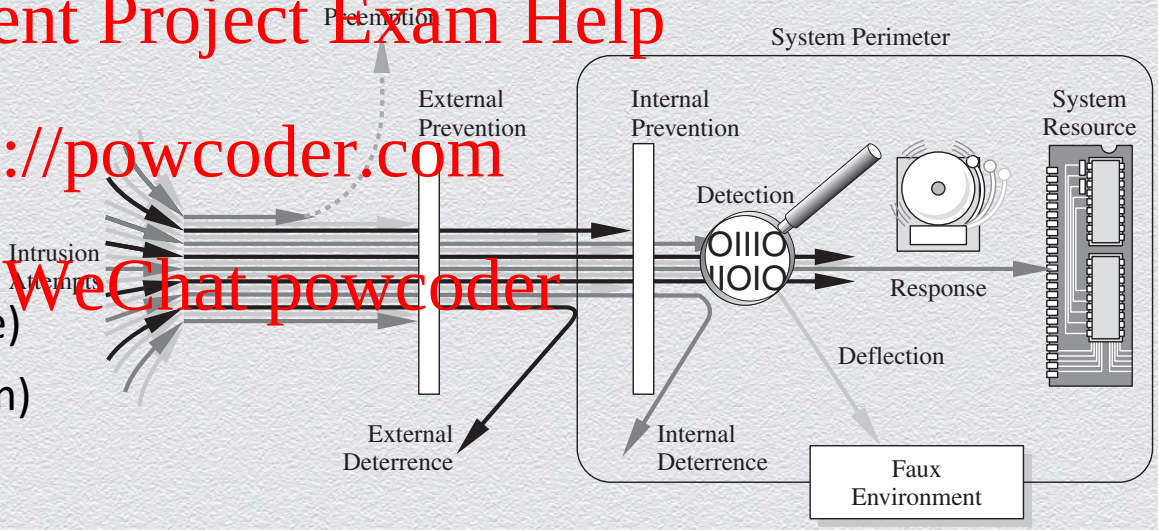
Controls offer a wide range of protection level / efficacy

- 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

# Example of control: HTTPS protocol

**Hypertext Transfer Protocol Secure (HTTPS)**

- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity

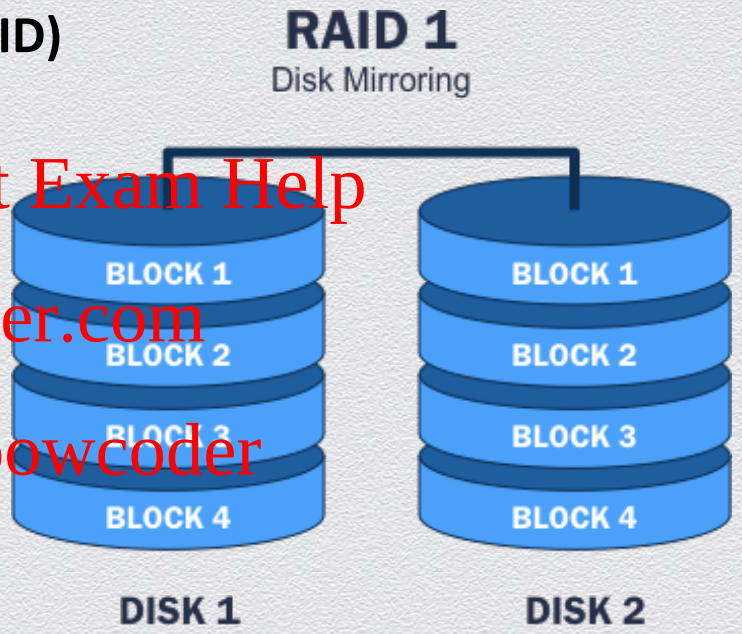


# Example of control: RAID technology

**Redundant Array of Independent Disks (RAID)**

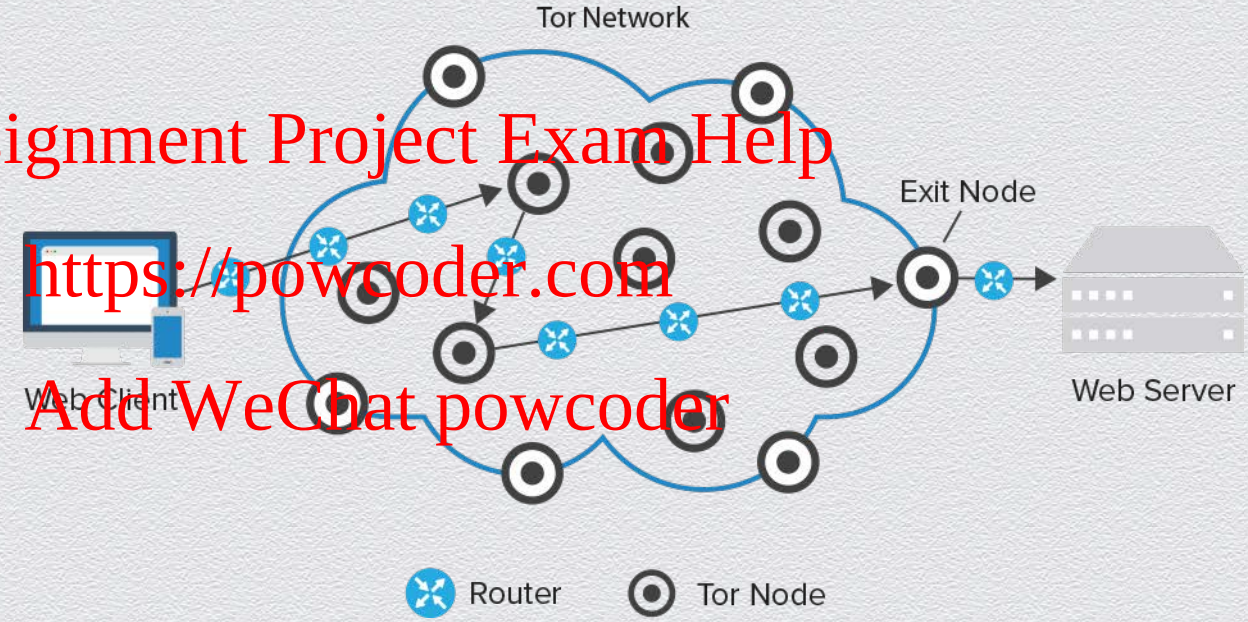
- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity





# Example of controls: TOR protocol

- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity



# As we will see: Exciting times to study (or work in) IT 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

# Recall: Confidentiality

Fundamental security property

- **an asset is viewed only by authorized parties**
- "C" in the CIA triad



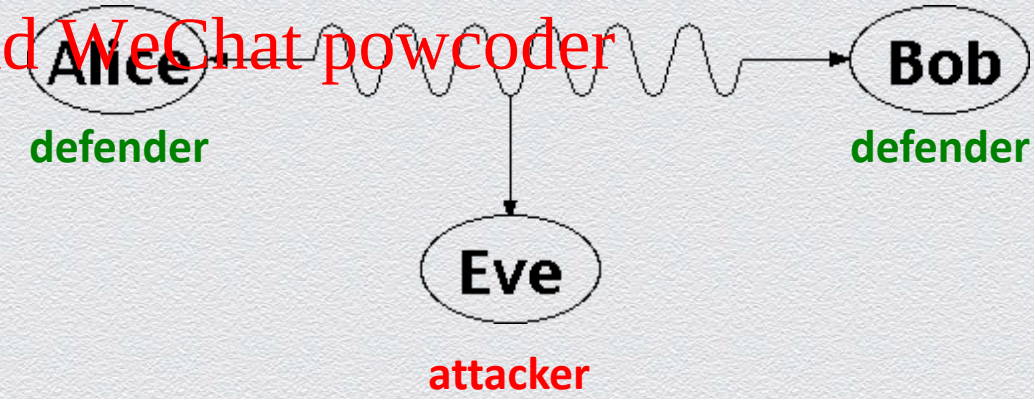
*"computer security seeks to prevent **unauthorized viewing (confidentiality)** or modification (integrity) of **data** while preserving access (availability)"*



**Eavesdropping**

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





# Problem setting: Secret communication

Two parties wish to communicate over a channel

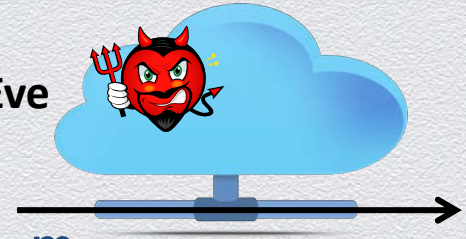
- Alice (sender/source) wants to send a message m to Bob (recipient/destination)

Underlying channel is unprotected

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



Alice m



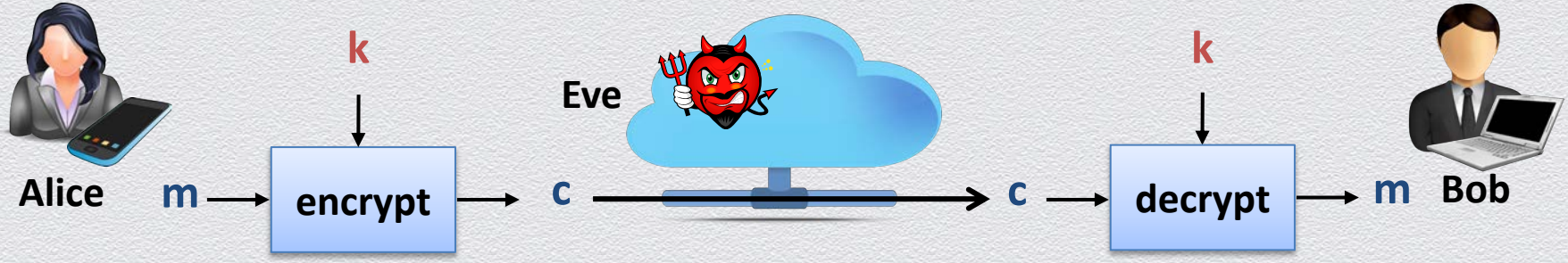

m Bob

# 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



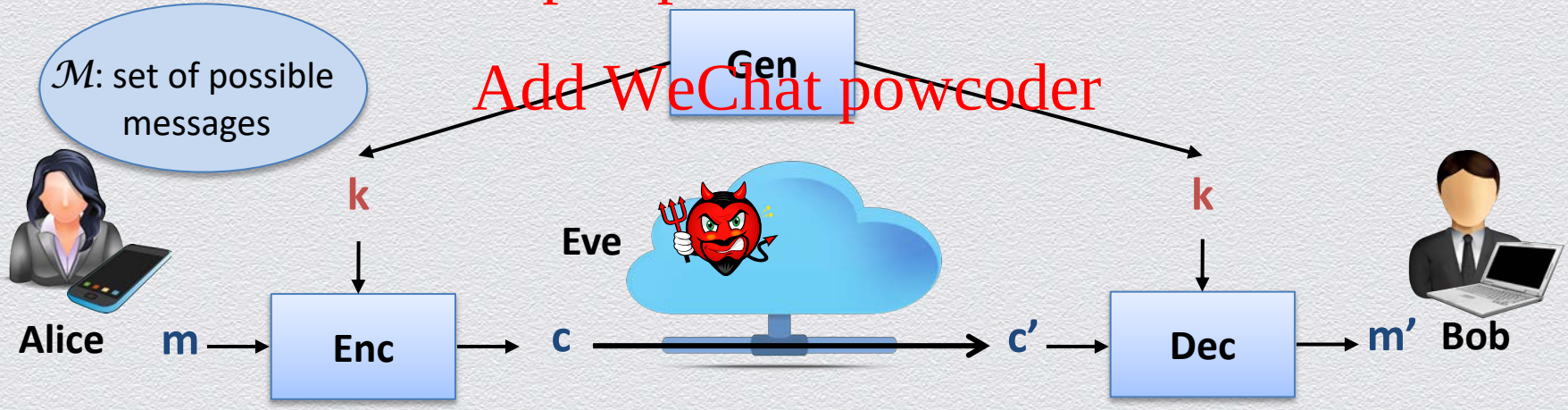


# Security tool: Symmetric-key encryption scheme

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

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



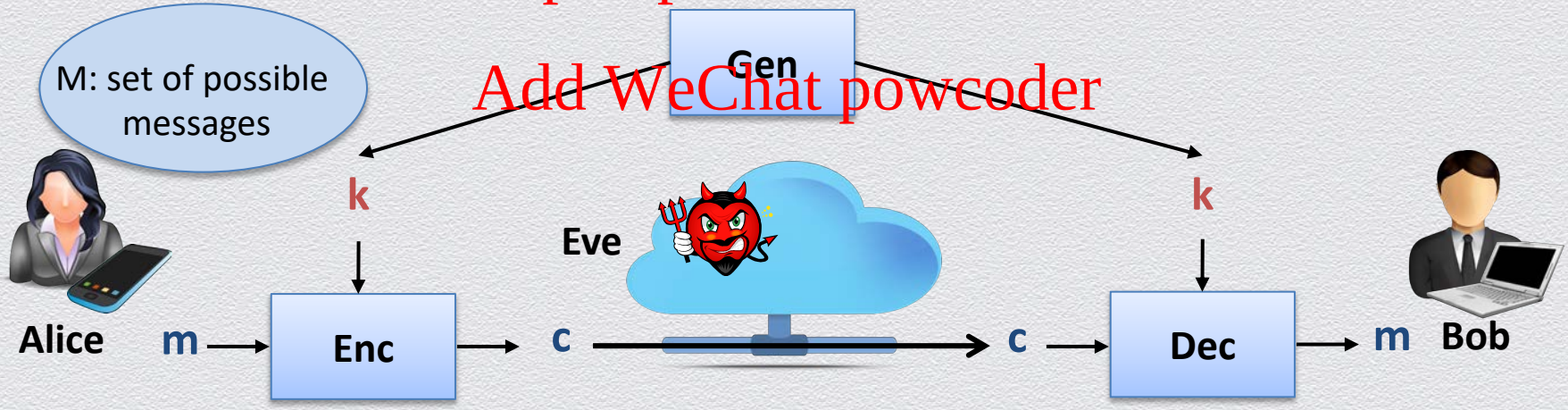


# Desired properties for symmetric-key encryption scheme

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 Dec(k, Enc(m, k), k) = m
- **security**: one "cannot learn" plaintext m from ciphertext c





# Kerckhoff's principle

*"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

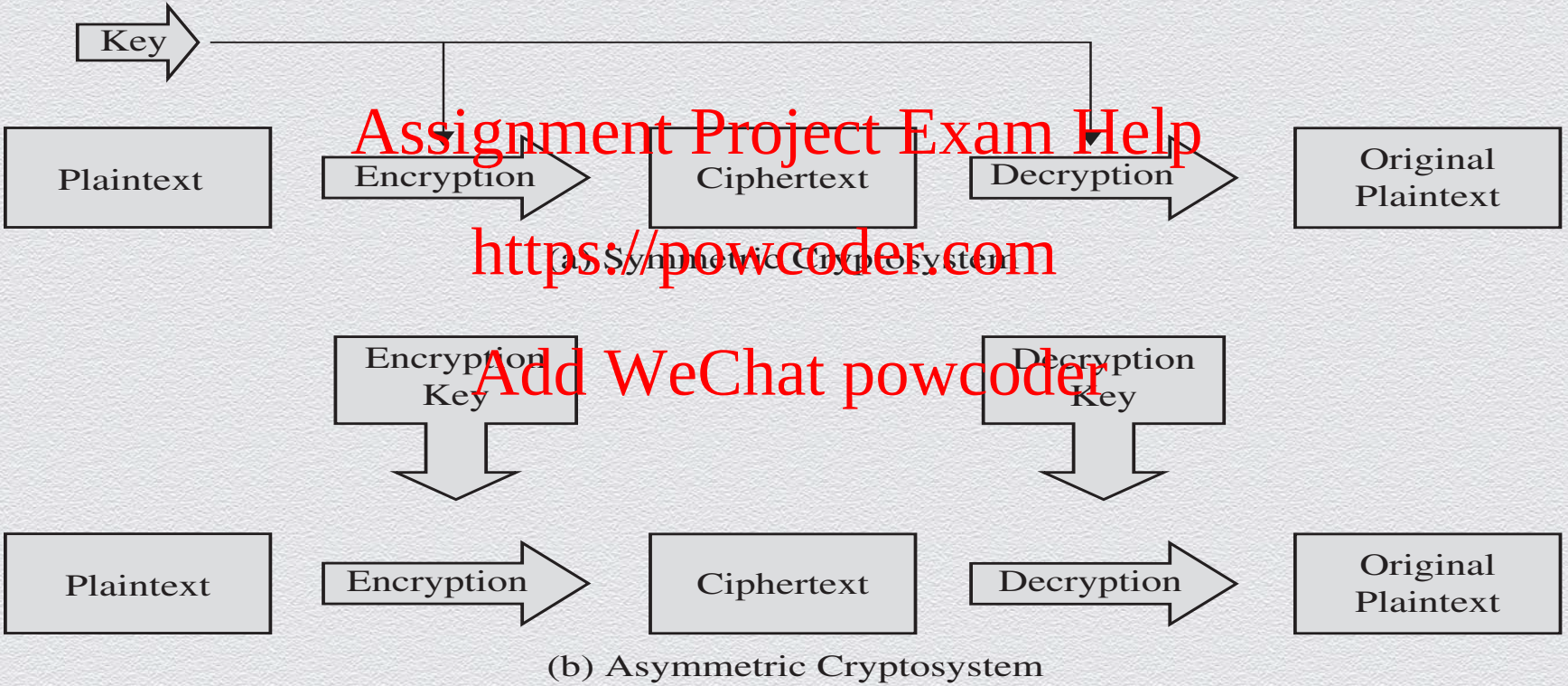
- due to security & correctness, Alice & Bob must share some secret info
- if no shared key captures this secret info, it must be captured by Enc, Dec
- but keeping Enc, Dec secret is problematic
  - harder to keep secret an algorithm than a short key (e.g., after user revocation)
  - harder to change 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

# Symmetric-key encryption

- Also referred to as simply “symmetric encryption”



# Symmetric Vs. Asymmetric encryption



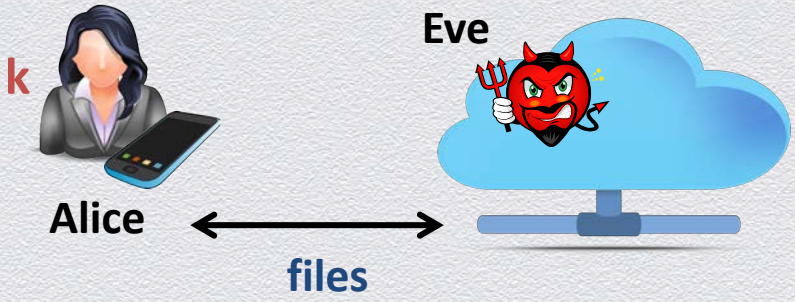
# Main application areas

**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 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**

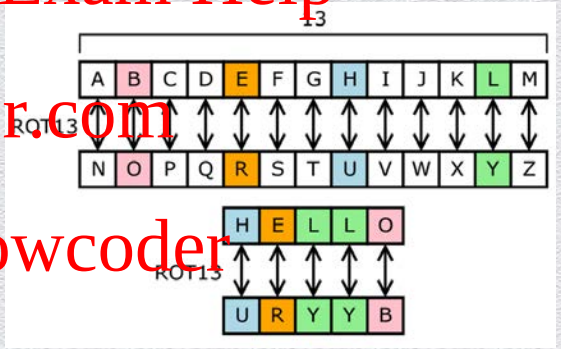
## 2.3 Classical ciphers

# 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

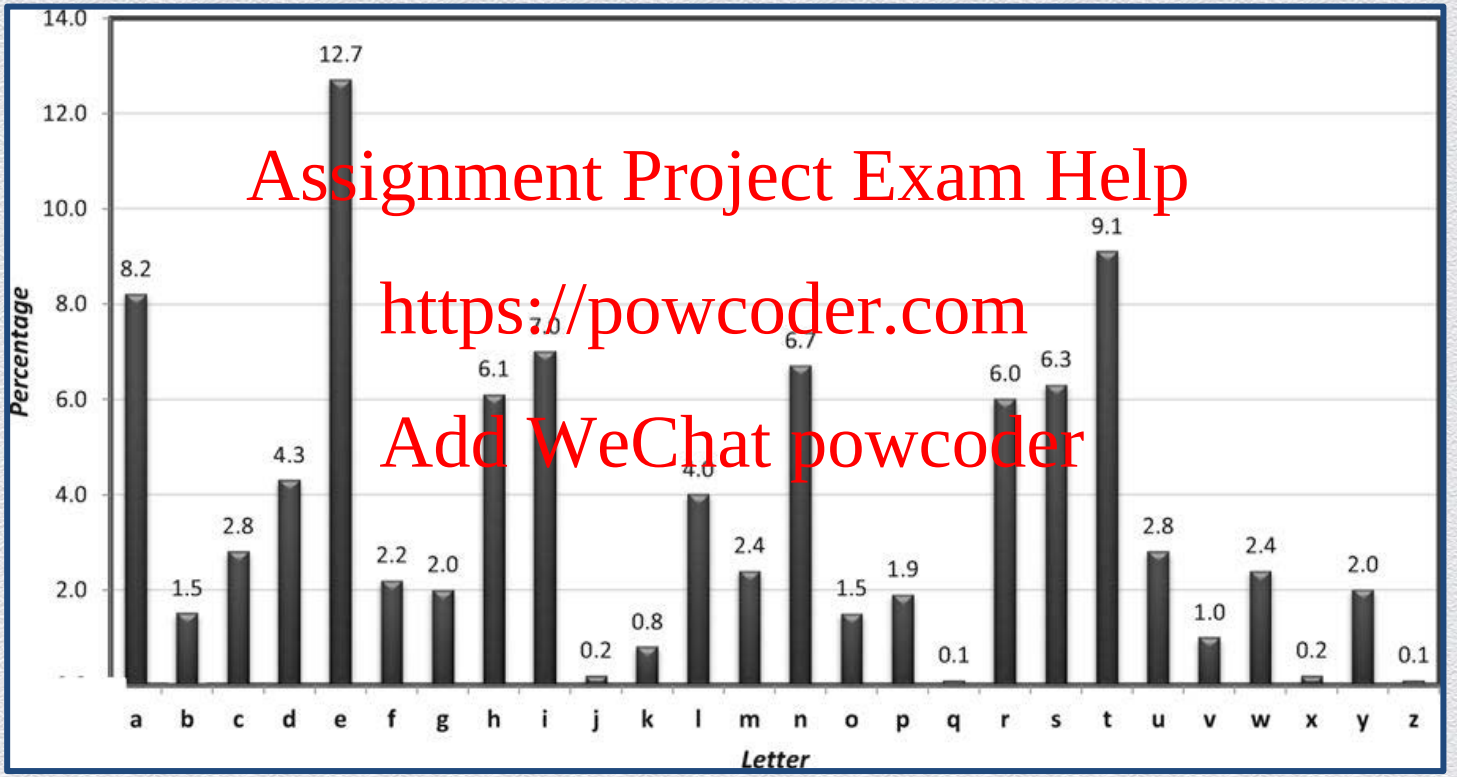
- message space $\mathcal{M}$ is "valid words" from a given alphabet
  - e.g., English text without spaces, punctuation or numerals
  - characters can be represented as numbers in [0:25]
- encryption
  - mapping each plaintext character into another character
  - character mapping is typically defined 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