Basic Concepts and Tools for Cyber Security

Kehuan Zhang
© All Rights Reserved

IERG4130 2022

Basic Concepts

What is Information Security?

From NIST Special Publication 800-12):

- Information Security is:

  The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to ensure confidentiality, integrity, and availability.

- CIA Triad

The CIA Triad

- Confidentiality
  - Preventing unauthorized access or disclosure of sensitive information
- Integrity
  - Defending against improper information modification or fabrification
  - Authenticity: no un-authorized modification
  - Non-repudiation: non-deniable of certain operation
- Availability
  - Ensure a timely and reliable access to information and systems
  - The information or system is always available when needed

Examples of CIA in our daily life

- Confidentiality
  - Password, student grade, disease, etc.
  - business plan, core technology
  - Military intelligence
- Integrity
  - For Emails: is it really sent by someone? (fake or not)
  - Can someone deny if she or he had sent a specific email
  - Whether an email had been modified during transmission?
- Availability
  - Server is too slow to respond
  - A system was crash
  - Whether a network link is free from congestion

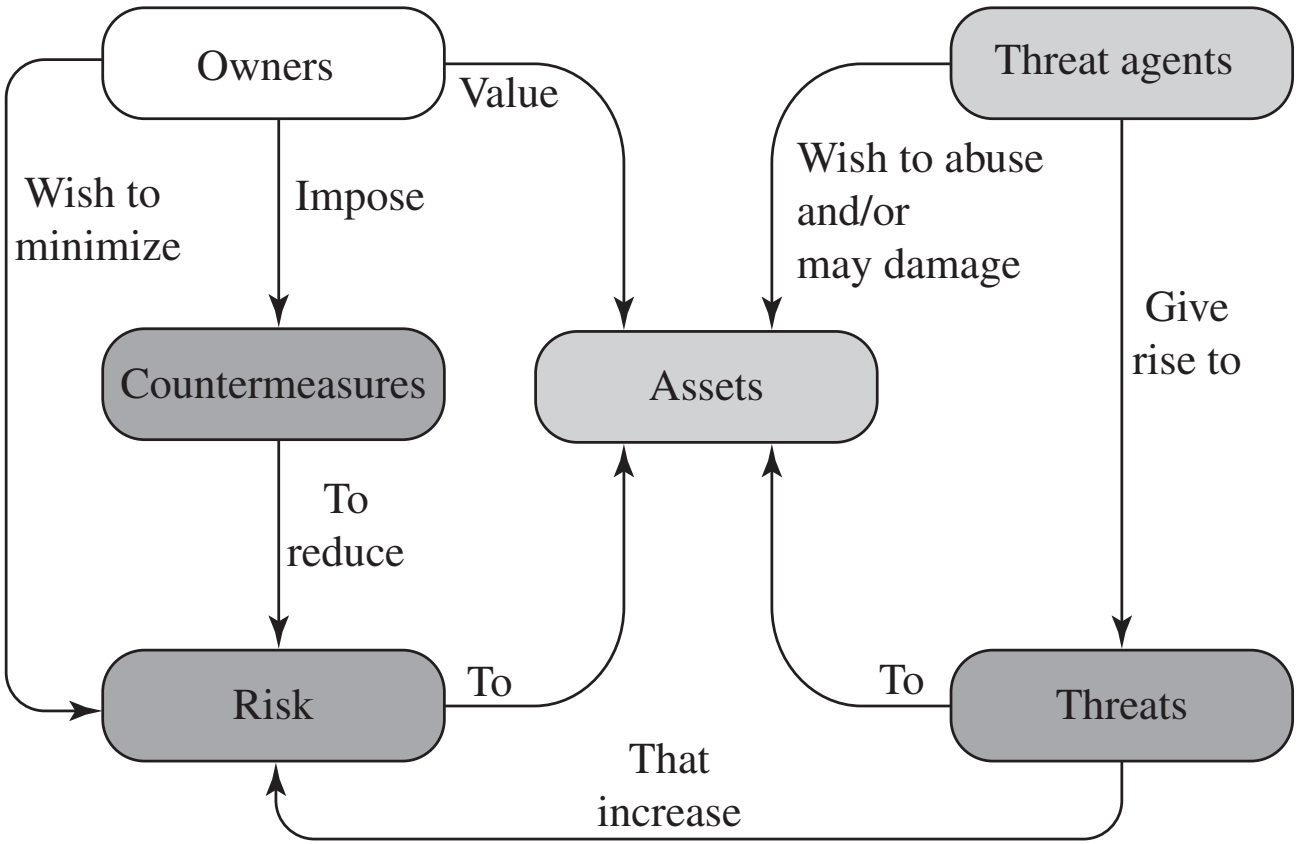
Additional Concepts related to CIA

- Authenticity
  - Being genuine and being able to be verified
  - Related to Integrity, but emphasize more on trusty
- Authenticate
  - To verify the true identity of a given user or machine
- Authorization
  - To assign certain permissions or privileges to an authenticated party
- Accountability
  - Actions of an entity can be traced uniquely to that entity
  - Related to integrity, but focus more on non-repudiation

A model for Computer Security

- Terms used in this model
  - Attack: an act to break security goals
  - Adversary (Threat agent): an entity to attack
  - Vulnerability: a weakness that can be exploited
  - Risk: potential loss due to attacks
  - Threat: a potential for violation of system security goals
  - Countermeasure: defense measures
  - Security Policy: a set of rules on how to defense
  - System Resource (Asset): data or service to be protected

The model


Owners
Value
Threat agents
Wish to
minimize
Impose
Wish to abuse
and/or
may damage
Give
rise to
Countermeasures
Assets
To
reduce
Risk
To
To
Threats
That
increase

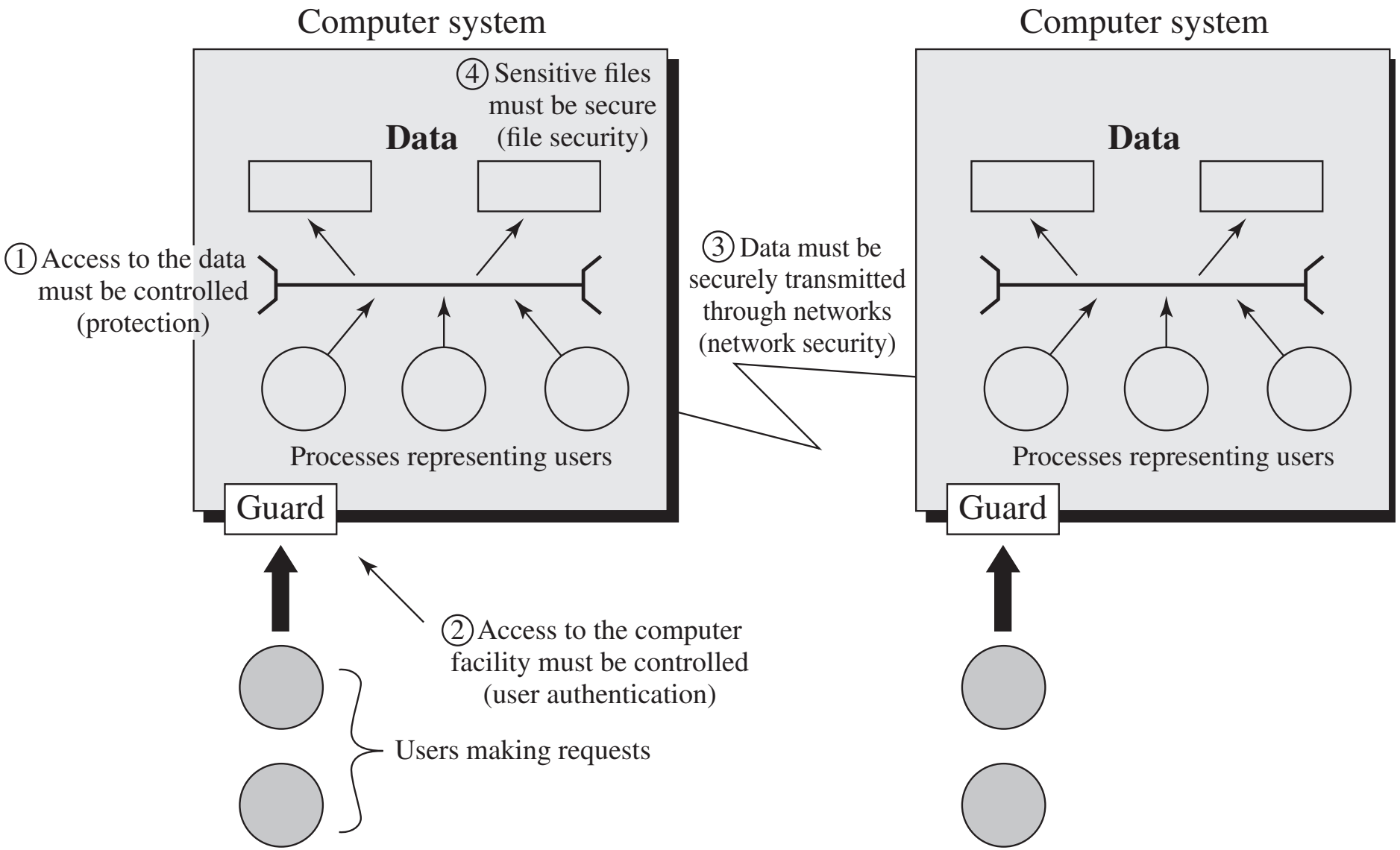
More on the Terms

- Assets Types
  - Not only the data and services
  - But also hardware, software, OS, network, etc.
- Vulnerability Types
  - Corrupted: integrity problem, different from expected
  - Leaky: confidentiality problem, lose access control
  - Unavailable: data inaccessible, service unusable
- Attack
  - Active vs. Passive attacks
  - Insider vs. Outsider
- Countermeasure
  - Preventive vs. Detective vs. Recovery

More on the Threats

- Unauthorized disclosure
  - Exposure: unauthorized release
  - Interception: unauthorized access during transmission
  - Inference: indirect access (e.g., via side-channel)
  - Intrusion: gain access with a successful attack
- Deception
  - Masquerade: pretending to be someone else
  - Falsification: deceiving with false data
  - Repudiation: denying certain actions
- Disruption
  - Incapacitation: make system mal- or non-function
  - Corruption: modify system data or functions
  - Obstruction: to interrupt normal operation
- Usurpation: misuse or take-over the control

Computer Security Scope

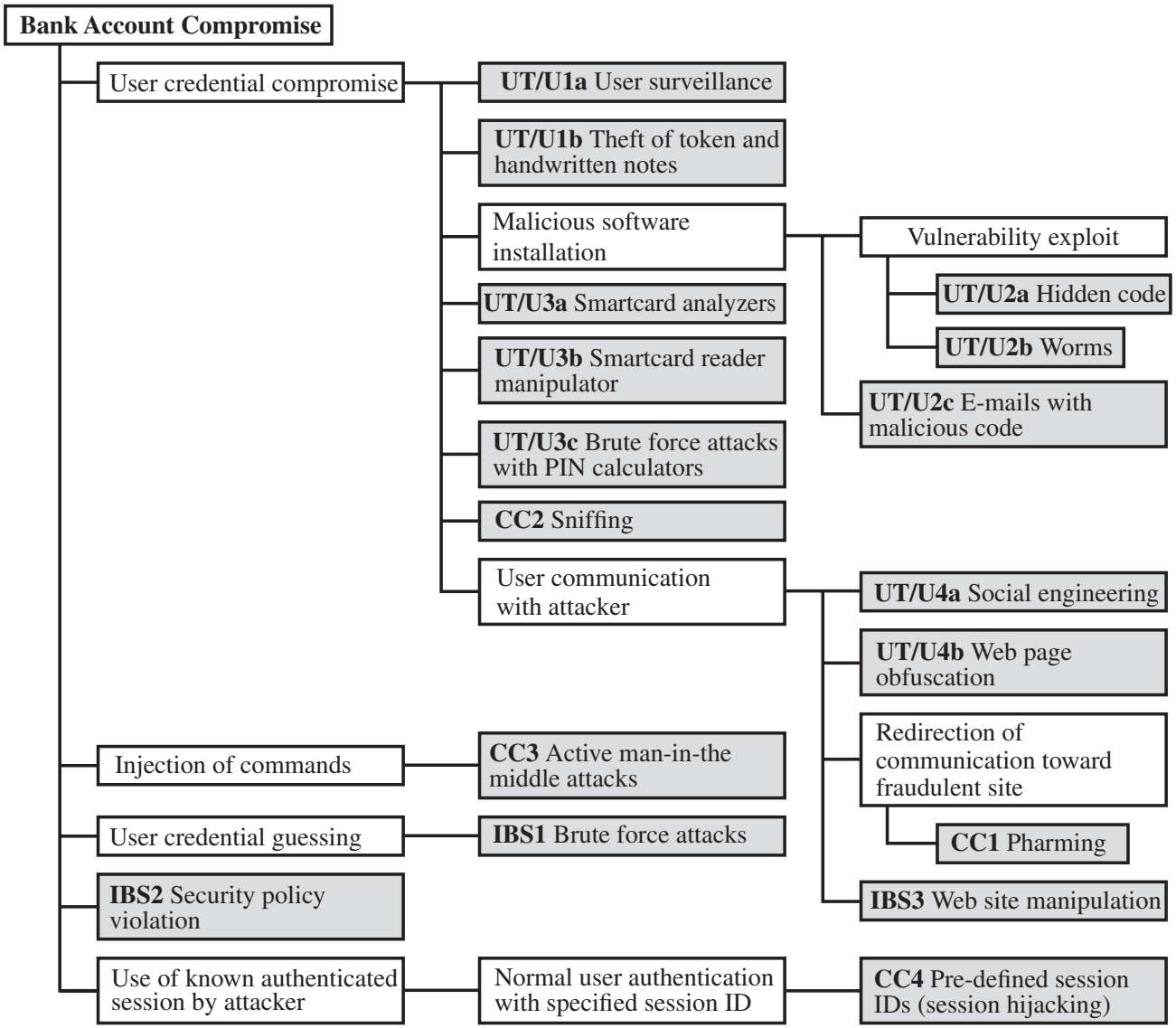


Adversary Model

- Define the adversary
  - Who would like to attack (e.g., inside or outside)?
  - Why do they want to attack?
  - What are they going after?
  - How capable are they to launch the attack?
- Identify the Attack Surface
  - What are your asset to protect?
  - How adversaries can access the asset?
  - (Network? Software? Data or PDF Documents?)
- Discover potential vulnerabilities
  - Patching and upgrading
  - Penetration testing (simulated attack)
  - Systematic analysis
- The concept of Attack Tree

Example of Attack Tree

- To depict all potential attacks (if possible)



Security Design Principles

Security Design Principles

1. Complete mediation
2. Layering
3. Fail-safe defaults
4. Economy of mechanism
5. Open design
6. Separation of privilege
7. Isolation
8. Least privilege
9. Least common mechanism
10. Usability

1. Complete mediation

- Why do we need to do complete mediation?
- Security of a system actually depends on weakest link
  - Cryptographic algorithm: perfect theory, poor coding
  - Any other example? (Consider human being's role)
- Thus it is necessary to take a wholistic approach

2. Layering

- It is also called **defense-in-depth**
- To to avoid any single-failing point
  - What is single-failing point?
  - E.g., authentication server
- Defending in multiple layers
  - If one layer fails, there is another layer
  - Examples?
  - Considering the firewalls
  - External firewall, secondary firewall, host firewall

3. Fail-safe defaults

- the default configuration should be safe
- E.g., FTP server configuration: disable anonymous account by default
- E.g., Operating system: disable unnecessary services
- Any other example? (Hints: networked program listening on network ports: local vs. non-local)

4. Economy of mechanism (simplicity)

- Keep the design and implementation as simple as possible
  - To minimize the complexity, thus easier to perform security analysis

5. Open design

- Use open algorithms rather than secret one
- Why?
  - Open algorithms are relatively secure with many security analysis
  - Secret ones may not be the case
  - Secret ones may give you a false sense of security
- Discussion: how about military systems?
  - Do they open their algorithms?
  - Why?

6. Separation of Privilege

- It means to grant different permissions for different roles
  - E.g., admin vs. normal user vs. guest user
- Other related principles
  - 7 Isolation (also called compartmentalization)
    - ★ To isolate the data, functions and permissions
    - ★ To minimize interferences and control the damage
    - ★ Example: Sandbox in Browsers, dedicated server for authentication
  - 8 Least privilege
    - ★ Operate with minimum set of privileges necessary to perform a task
    - ★ To control the damage
    - ★ E.g., normal user may not be able to change security policies
    - ★ Any other examples? (Hint: which account to run a web server?)
  - 9 Least common mechanism
    - ★ If common part is compromised, huge impacts
    - ★ Considering the SDK, or libraries vulnerability
    - ★ Common parts among users may also be used for attack
    - ★ E.g., side-channel attack through *procfs* on Linux

10. Usability

- Do **NOT** rely on user to make right decision to achieve security
- Even users with little or even no background can do thing right
- E.g.,
  - Bad Practice:
    - ★ On Android, asking user to accept or reject permission request when installing a new App
  - Good Practice:
    - ★ By default disallow user to access Web site with insecure SSL certificate

Basic Cryptographic Tools

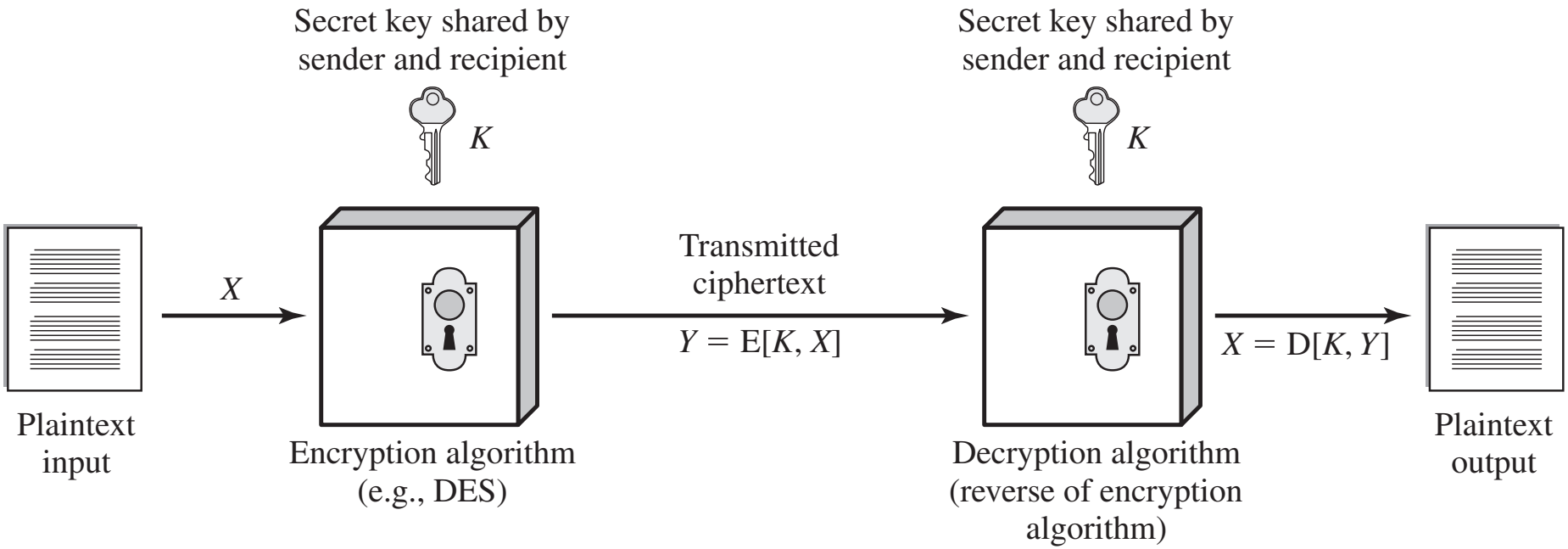
Recall the concept of CIA

- What is Confidentiality?
- What is Integrity?
- What is Availability?
- Question: How to achieve those security goals?
  - Cryptographic tools can help (partially)
  - A high overview here only
  - More details will be given in following weeks

Use encryption to achieve Confidentiality

- What is cryptography (or encryption)?
  - "Crypto" comes from Greek, means hidden or secret
  - "Graph"'s original Greek meaning is "write"
  - So, cryptography is to write some secret
- How it works?
  - To scramble the message controlled by a secret key
  - For example, rearrange the position, substitution
  - Only when the secret is known can the scrambled message be converted back to original form
- Convert the message confidentiality problem to the confidentiality problem of encryption key

Typical Encryption Process

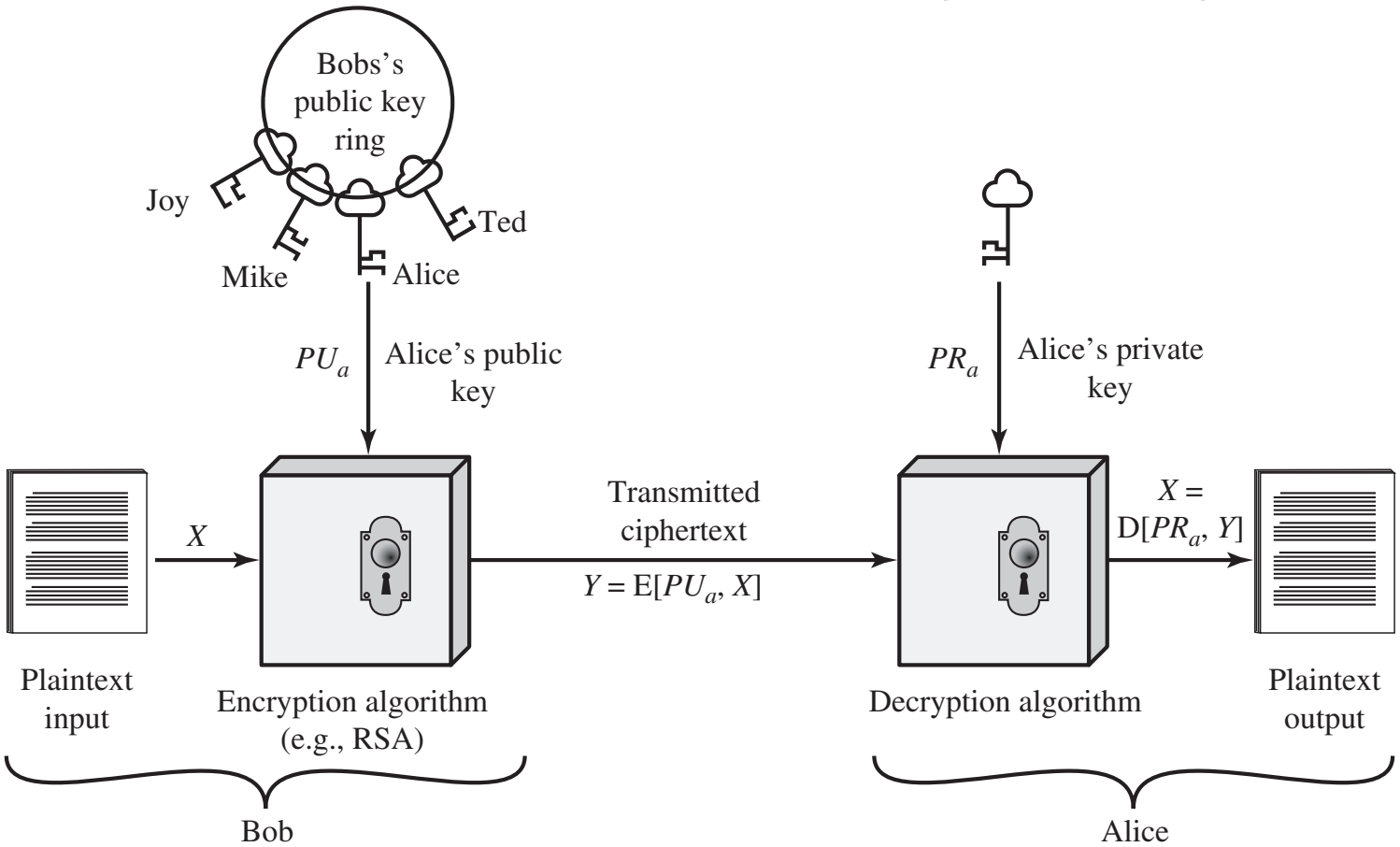


Encryption Algorithms

- Symmetric Encryption
  - Encryption and Decryption will use the same key
  - Current standard: AES(Advanced Encryption Standard)
  - Obsolete standard: DES
  - Many other algorithms exists
  - Warning: do not invent your own (before you become a Master)
- Asymmetric Encryption
  - Encryption and Decryption will use different keys
  - One key is secret, just like Symmetric encryptions
  - The other key is public, known to every body
  - Current standard: RSA, ECC

Model of Public Key Encryption

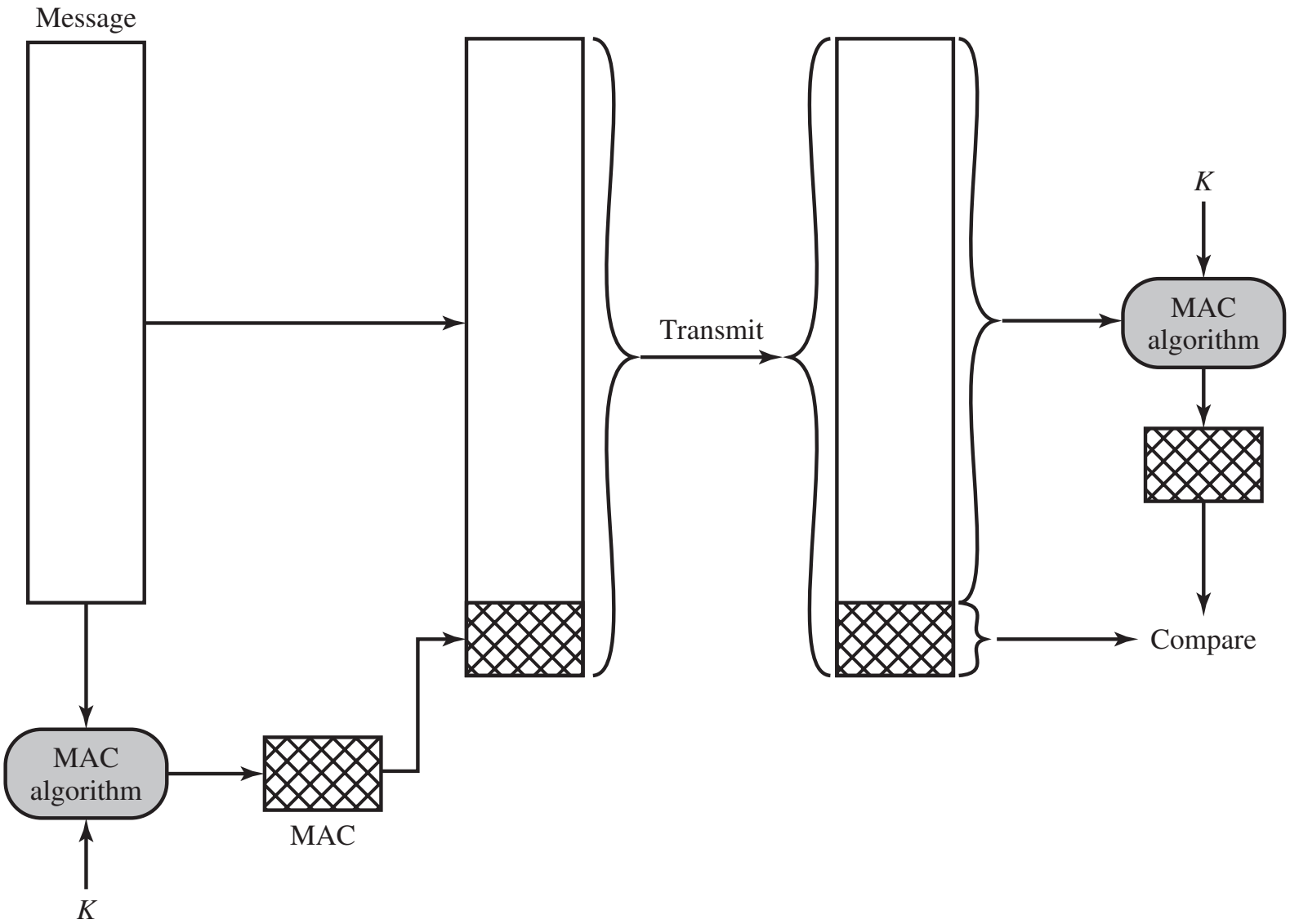
- Note the difference when using different keys for encryption
- Both of keys can be used to encrypt and decrypt message, as long as they are used in pairs
- But for confidentiality, the public key has to be used
- Private key is used for digital signature (later slides)



Use Message Authentication Code to Achieve Integrity

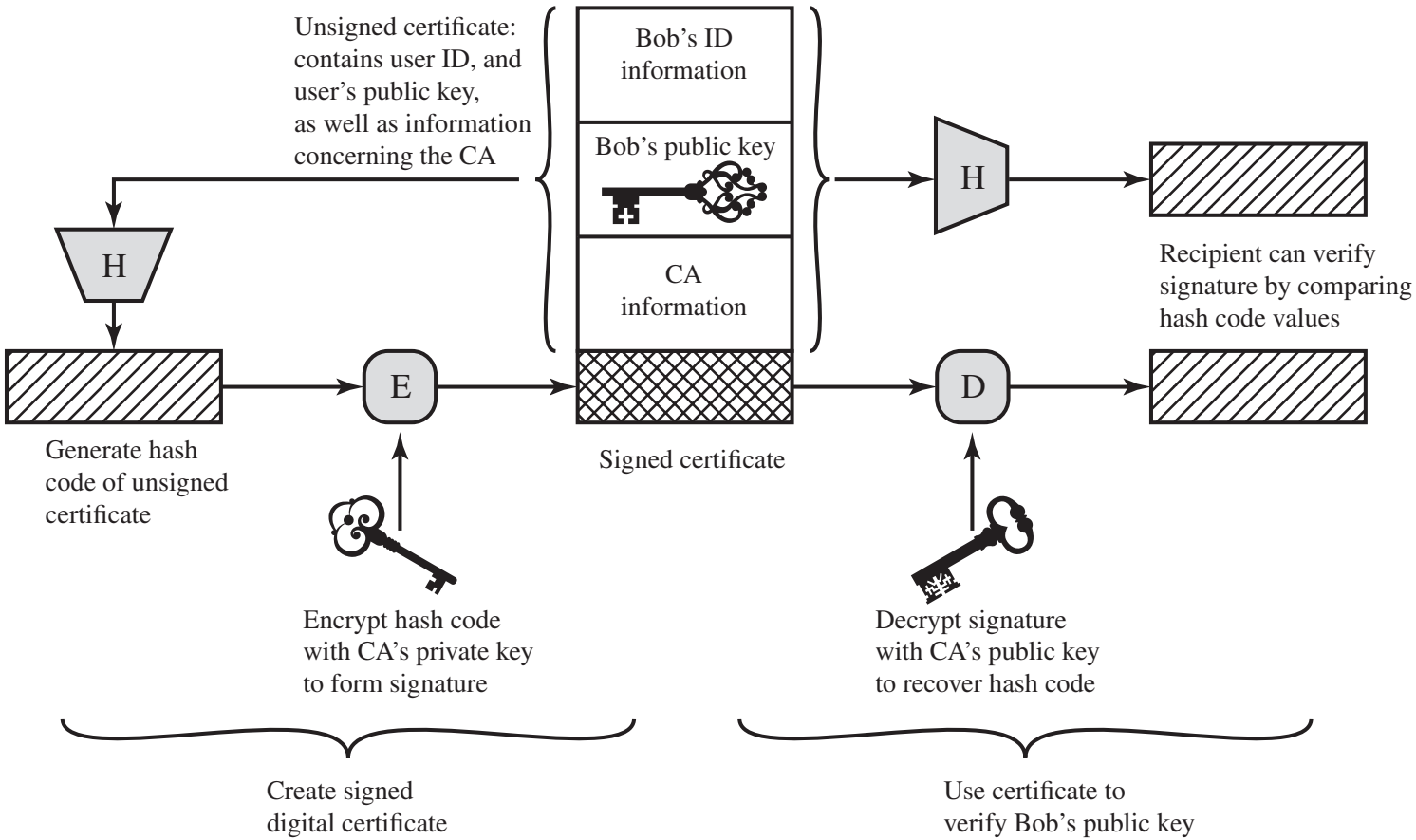
- How to verify the integrity a given message?
  - Can we verify it by encrypt it?
  - What is potential limitations of this method?
- Hash Function
  - Generate a digest with fixed length for arbitrary length
  - Security requirement for Hash function?
  - Reversible?
  - Collisions?
- Message Authentication Code (MAC)
  - MAC = Hash + Encryption

A Typical Model of MAC



Use Digital Signature to Achieve Non-repudiation

- Digital signature is built on public-key encryptions
- Pay attention to the encryption key
- As mentioned earlier, private key is used to encrypt (sign) documents, while public is used to verify



About Random Numbers

- Why do we need to care about random numbers?
- Random numbers have huge impacts on security
  - Remember encryption? Problem was converted to confidentiality of encryption keys.
  - But how to ensure the secrecy of encryption key?
  - Randomness! (besides other protection measures)
- How to evaluation randomness?
  - There are some math criteria
  - E.g., uniform distribution: chance of each number is equal
  - Independence: cannot use known sequence to infer
- True Random vs. Pseudorandom
  - True random numbers rely on certain physical process
    - ★ can not be re-generated, and generation speed may be slow
    - ★ Perfect in terms of randomness
    - ★ Often used as pre-shared master keys
  - Pseudorandom numbers are generated with certain algorithms
    - ★ Can be re-generated given the same seed
    - ★ Used for tokens, session keys, or keys in steaming cipher