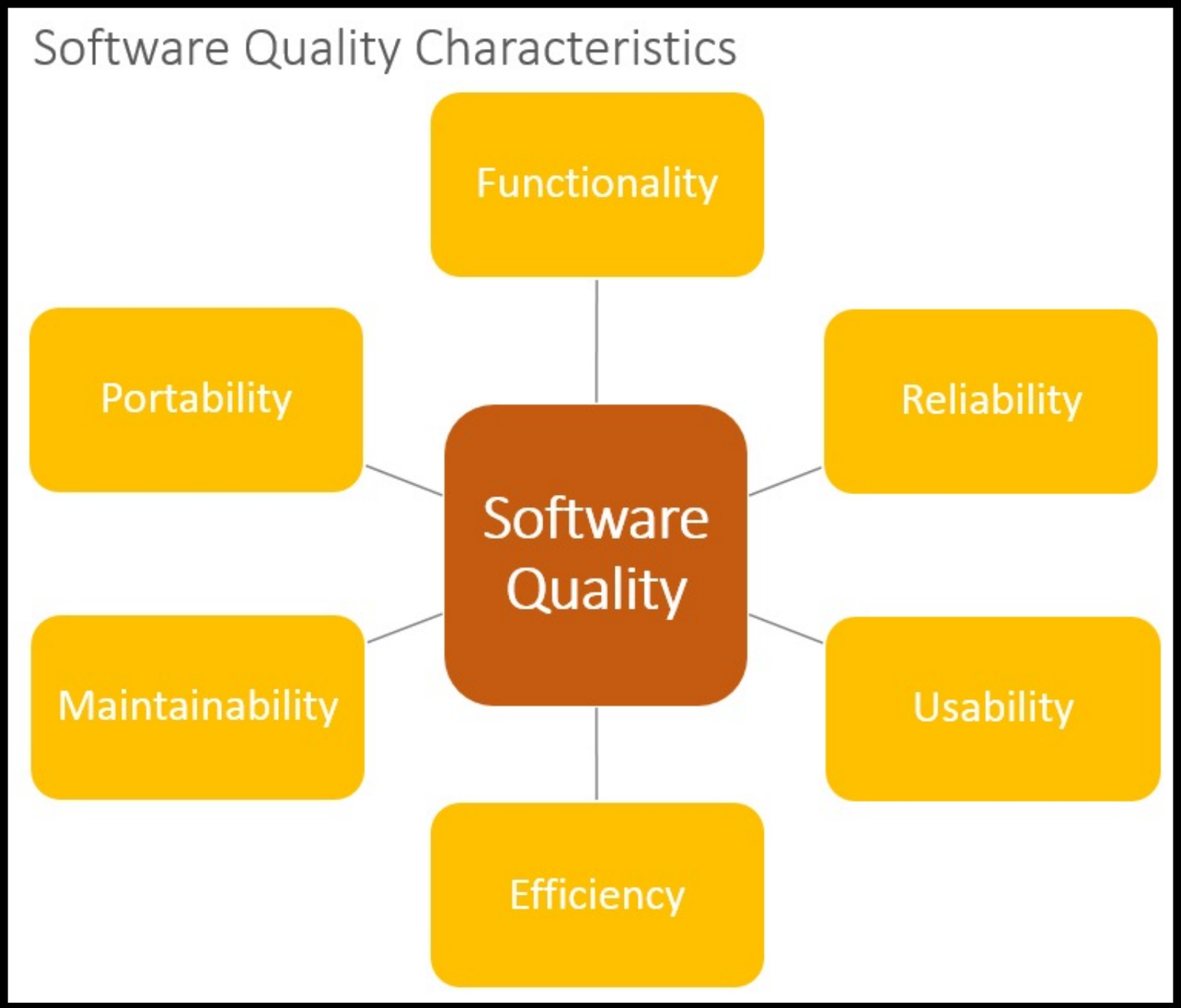
Security Principles

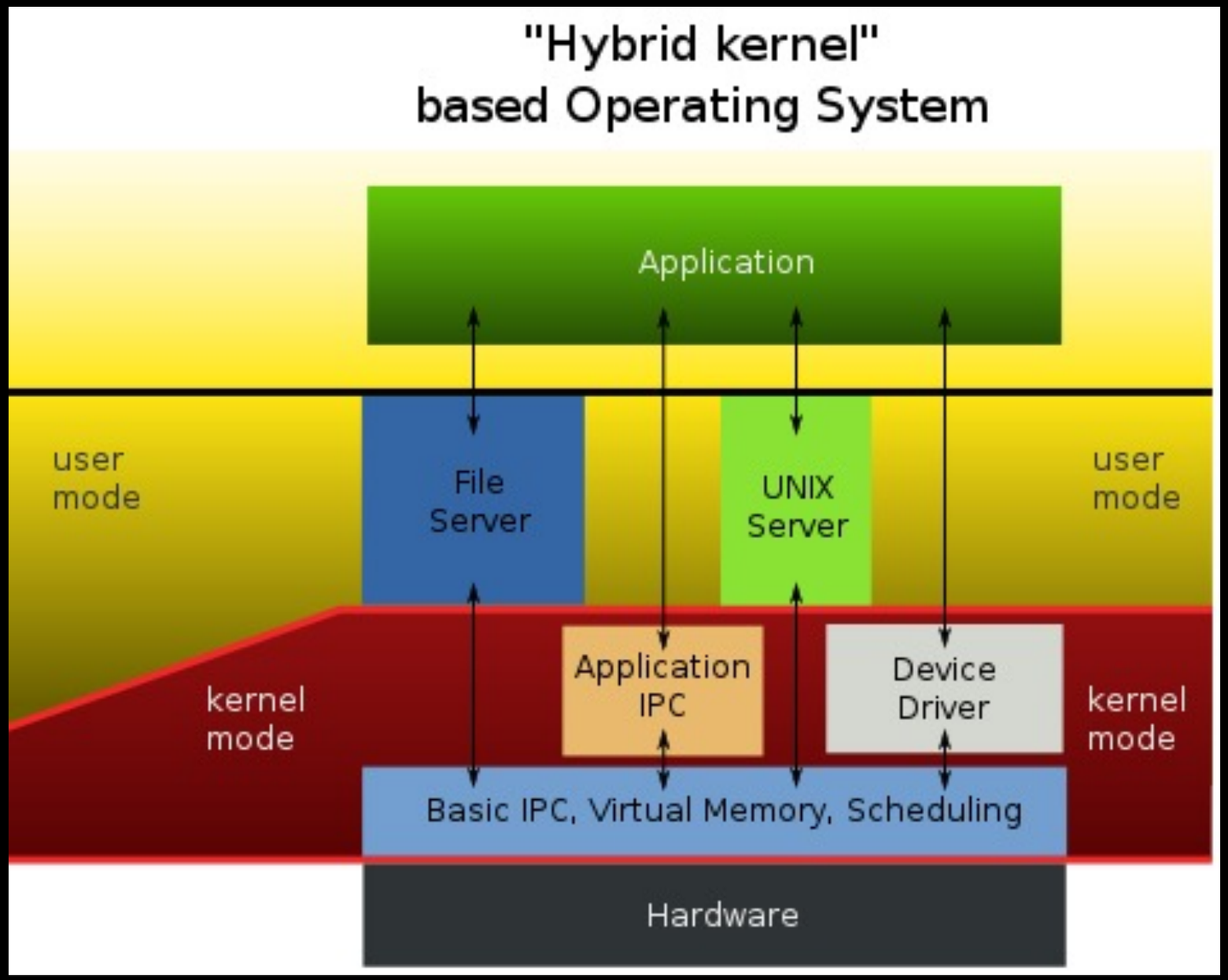
What is security?

Protecting systems against undesired behavior

What is security?



What needs security?



Can we achieve security?

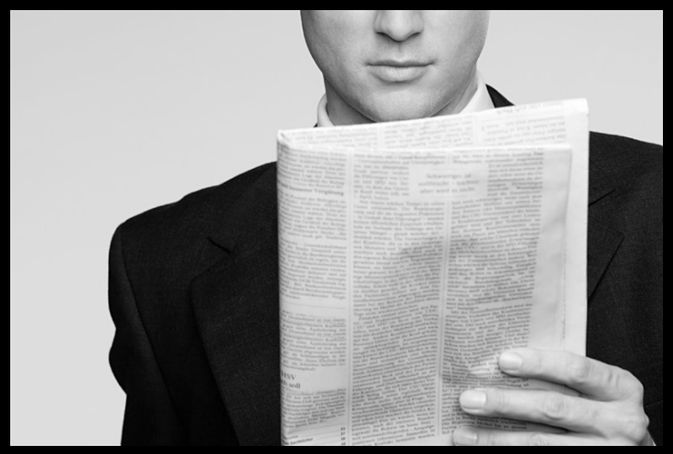
- Security is a relative property with various implications.
- Absolute security may not be practical or possible.
- Security is a quality characteristic that is known to be difficult to achieve.
- We may compromise other quality characteristics for improved security.

Why security is an issue?

- There are political and economic motivations to break into systems, make systems unreachable, or steal data.
- Software development is erroneous.
- History shows that even the most secure systems can be compromised.
- Examples of large scale system breaches show large impacts on the operations of corporations.

Security Goals

Integrity



Confidentiality

Availability

Confidentiality

- **Confidentiality** is the avoidance of the unauthorized disclosure of information.
  - confidentiality involves the protection of data,
  - providing access for those who are allowed to see it
  - while disallowing others from learning anything about its content.

Approaching confidentiality

- Our goal is to disallow access to our private data
  - Make the data unreadable except for authorized principals
  - Restrict access to data
  - Monitor and control access to data
- Example applications and problems?

Unreadable data

- Data must be under control of data owner
- Data owner can choose to make the data unreadable

- The most important tool to achieve this is through encryption
  - Well studied mathematical theory used to achieve data confidentiality through strong properties

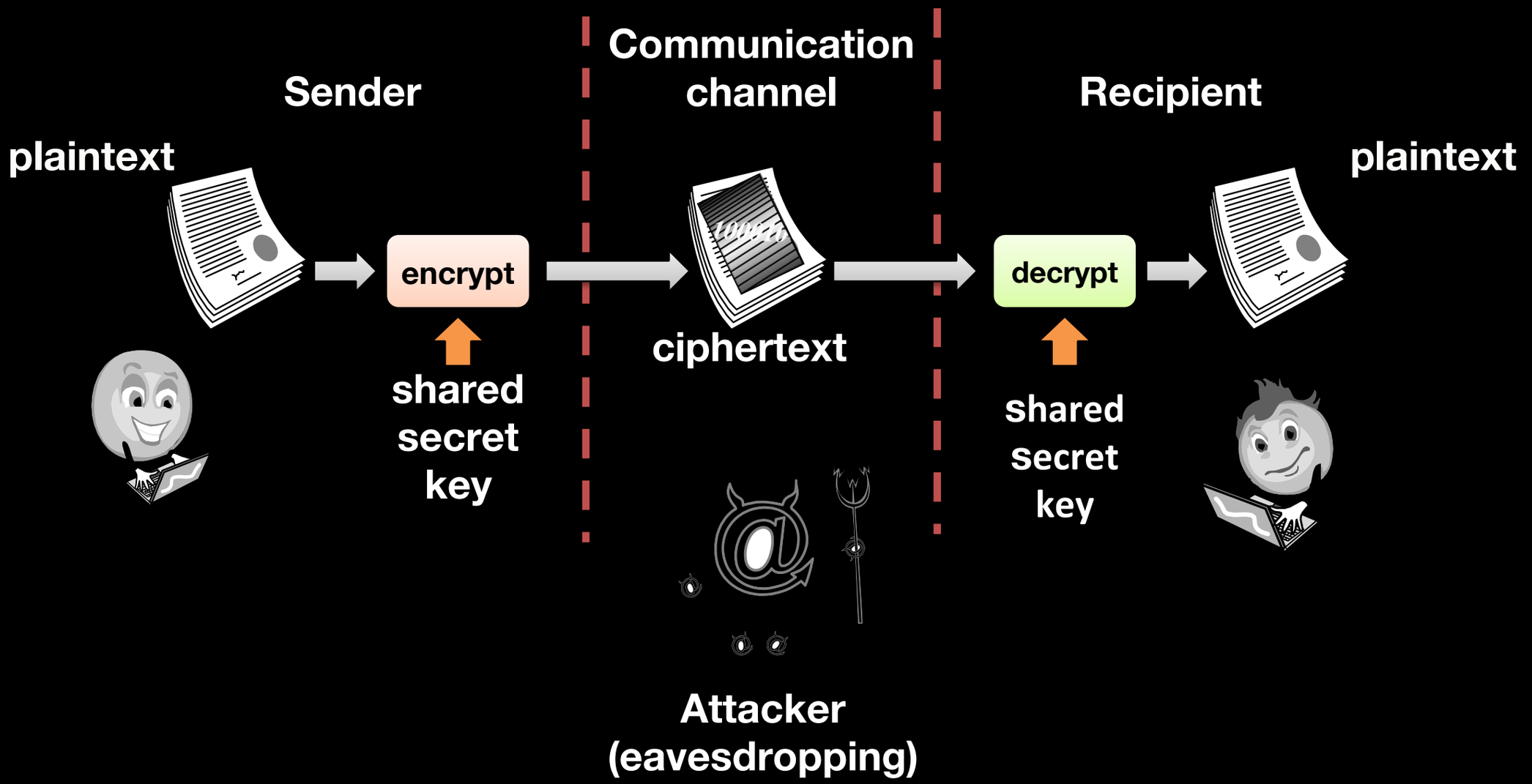
Cryptography

"*There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. This book is about the latter.*" —Bruce Schneier

Cryptography

"If I take a letter, lock it in a safe, hide the safe somewhere in New York, then tell you to read the letter, that's not security. That's obscurity. On the other hand, if I take a letter and lock it in a safe, and then give you the safe along with the design specifications of the safe and a hundred identical safes with their combinations so that you and the world's best safecrackers can study the locking mechanism—and you still can't open the safe and read the letter—that's security." —Bruce Schneier

Unreadable data



Restrict access to data

Restrict access to data

- **Access control:** rules and policies that limit access to confidential information to those people and/or systems with a “need to know.”
- This need to know may be determined by
  - identity, such as a person’s name or a computer’s serial number, or by
  - a role that a person has, such as being a manager or a computer security specialist.

Tools for Confidentiality

- **Authentication:** A process to determine a person's (or a system's) identity and role.

Something you are

Something you know

Something you have

Authentication

- Authentication systems required for secure systems
- Identify subjects
- Assign credentials to subjects
- Verify subject identities when needed
- Allow for flexibility of subjects entering or exiting a system
- Rules governing subject registration, identification, and modification of authentication primitives

Tools for Confidentiality

- **Authorization:** the determination if a person or system is allowed access to resources, based on an access control policy.
- Who can do what, when, and how?
- How to distinguish roles?
- How to grant/deny access?
- What system components are required?
- What are the various levels of trust?

Authorization Levels

1. No sharing at all (complete isolation).
2. Sharing copies of programs or data files.
3. Sharing originals of programs or data files.
4. Sharing programming systems or subsystems.
5. Permitting the cooperation of mutually suspicious subsystems—e.g., as with debugging or proprietary subsystems.
6. Providing "memoryless" subsystems—i.e., systems which, having performed their tasks, are guaranteed to have kept no secret record of the task performed (an income-tax computing service, for example, must be allowed to keep billing information on its use by customers but not to store information secretly on customers' incomes).
7. Providing "certified" subsystems—i.e., those whose correctness has been completely validated and is guaranteed a priori.

G. SCOTT GRAHAM and PETER J. DENNING 1972

Integrity

- **Integrity:** the property that information has not been altered in an unauthorized way.
- Example
  - Trusting a piece of information requires the integrity of its source.
- Systems with high integrity are developed with secure coding techniques, using safe languages, and under high software engineering standards.

Approaching integrity

- Software approach
- System backups and fault tolerance
- Integrity checking tools such as checksums
- Cryptographic tools such as digital signature

Availability

- **Availability:** the property that information is accessible and modifiable in a timely fashion by those authorized to do so.

- An important property: everyone wants the system to be available whenever needed.

Approaching availability

- Redundancy
- Fault tolerant code and automatic healing
- Appropriate authentication and authorization measures
- Physical protection
- Quality control for software and hardware

Security Tools

- A.A.A.

Assurance

Authenticity

Anonymity

Privacy in practice

- You're given a system that uses a national identification number to provide services. How can you improve the *privacy* of the system?

- You have access to personal data from patients in a hospital. You want to report this data to a company. How can you make sure you do not violate the *privacy* of the patients?

Assurance

- **Assurance** refers to how **trust** is provided and managed in computer systems.
- Policies, Permissions, Protections
- **Policies,** which specify behavioral expectations that people or systems have for themselves and others.
  - For example, the designers of an online music system may specify policies that describe how users can access and copy songs.
- **Permissions,** which describe the behaviors that are allowed by the agents that interact with a person or system.
  - For instance, an online music store may provide permissions for limited access and copying to people who have purchased certain songs.
- **Protections,** which describe mechanisms put in place to enforce permissions and polices.
  - We could imagine that an online music store would build in protections to prevent people from unauthorized access and copying of its songs.

Authenticity

- **Authenticity** is the ability to determine that statements, policies, and permissions issued by persons or systems are genuine.
- **Primary tool:**
  - **digital signatures.** These are cryptographic computations that allow a person or system to commit to the authenticity of their documents in a unique way that achieves
  - **nonrepudiation**, which is the property that authentic statements issued by some person or system cannot be denied.

Anonymity

- **Anonymity:** the property that certain records or transactions not to be attributable to any individual.
- **Tools:**
  - **Aggregation:** the combining of data from many individuals so that disclosed sums or averages cannot be tied to any individual.
  - **Mixing:** the intertwining of transactions, information, or communications in a way that cannot be traced to any individual.
  - **Proxies:** trusted agents that are willing to engage in actions for an individual in a way that cannot be traced back to that person.
  - **Pseudonyms:** fictional identities that can fill in for real identities in communications and transactions, but are otherwise known only to a trusted entity.

Anonymity and Privacy

- Anonymity is a technique to achieve privacy.
- Privacy and Security two distinct but related concepts
- Software systems must be privacy preserving to gain users’ trust
- Privacy is particularly important when it comes to the web and mobile platforms.

"Know your enemy."

–Sun Tzu, Art of War

How can I know the enemy?

Adversary: one's opponent in a contest, conflict, or dispute.

Threat: a person or thing likely to cause damage or danger.

What are the threats to a *target system*?

What are the advantages of an adversary?

What are the advantages of the target system?

What are the motivations of the adversary?

How likely a threat can result in a compromise?

How can I know the enemy?

Attack:

An aggressive attempt to score a goal or point or otherwise gain an advantage.

Act against (someone or something) aggressively in an attempt to injure or kill.

My definition: A goal-oriented process through which a motivated adversary exploits a vulnerability in a computing system.

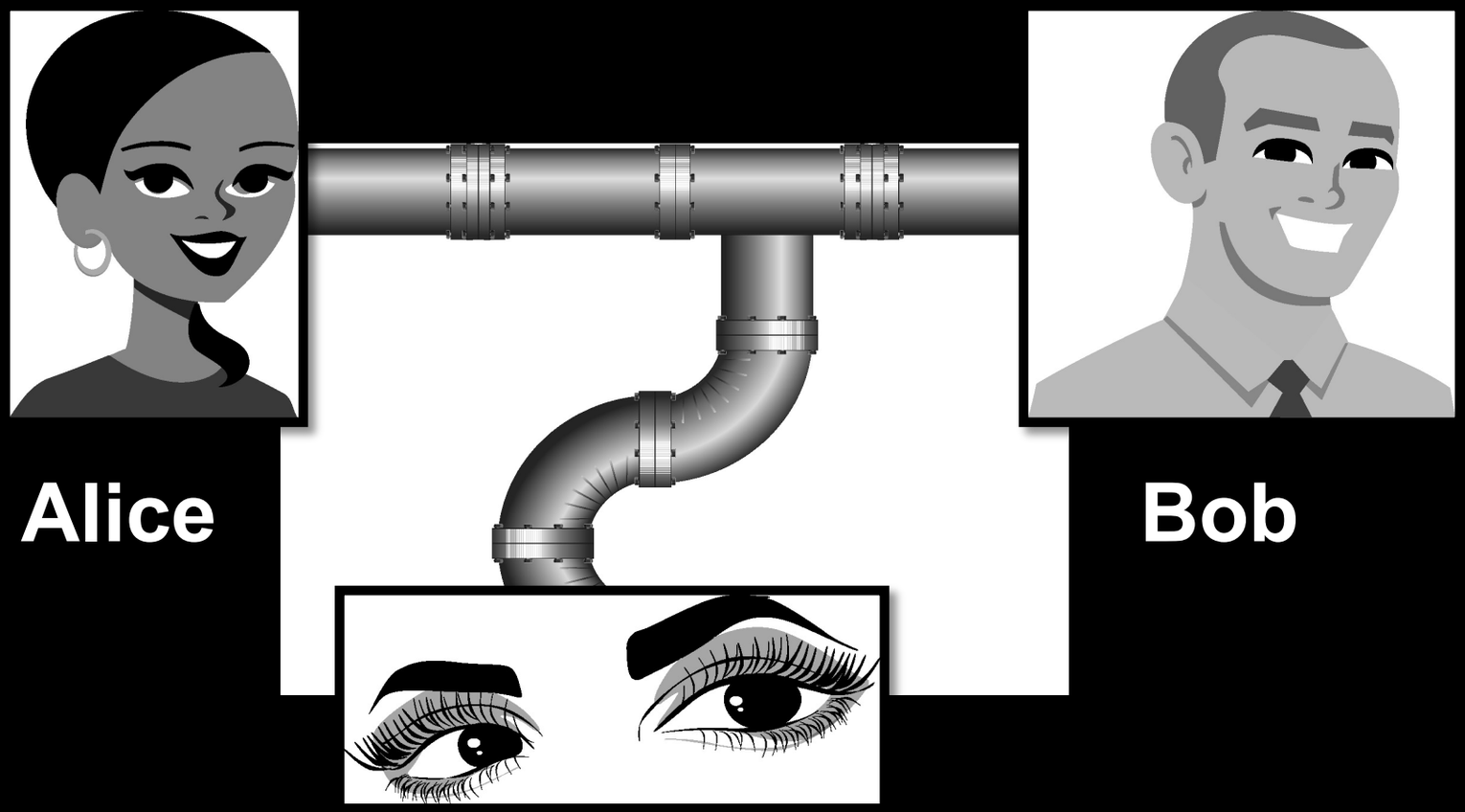
How can I know the enemy?

Hacker:

A **hacker** is any skilled computer expert that uses their technical knowledge to overcome a problem.

Threats and Attacks

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



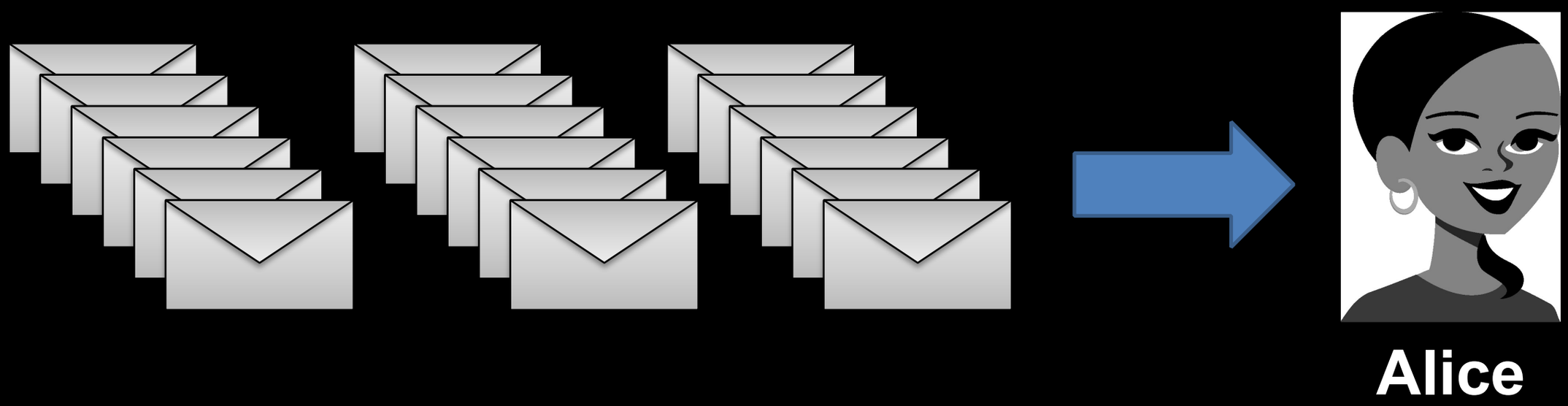
Man-in-the-middle

Threats and Attacks

- **Alteration:** unauthorized modification of information.
  - Intercepting, modifying, and retransmitting a network stream.
  - Modifying sensitive user data on disk.
  - Altering the behavior of a process
  - Remote injection attacks

Threats and Attacks

- **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.



Threats and Attacks

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

- **Example:** IP masquerading

“From: Alice”
(really is from Eve)

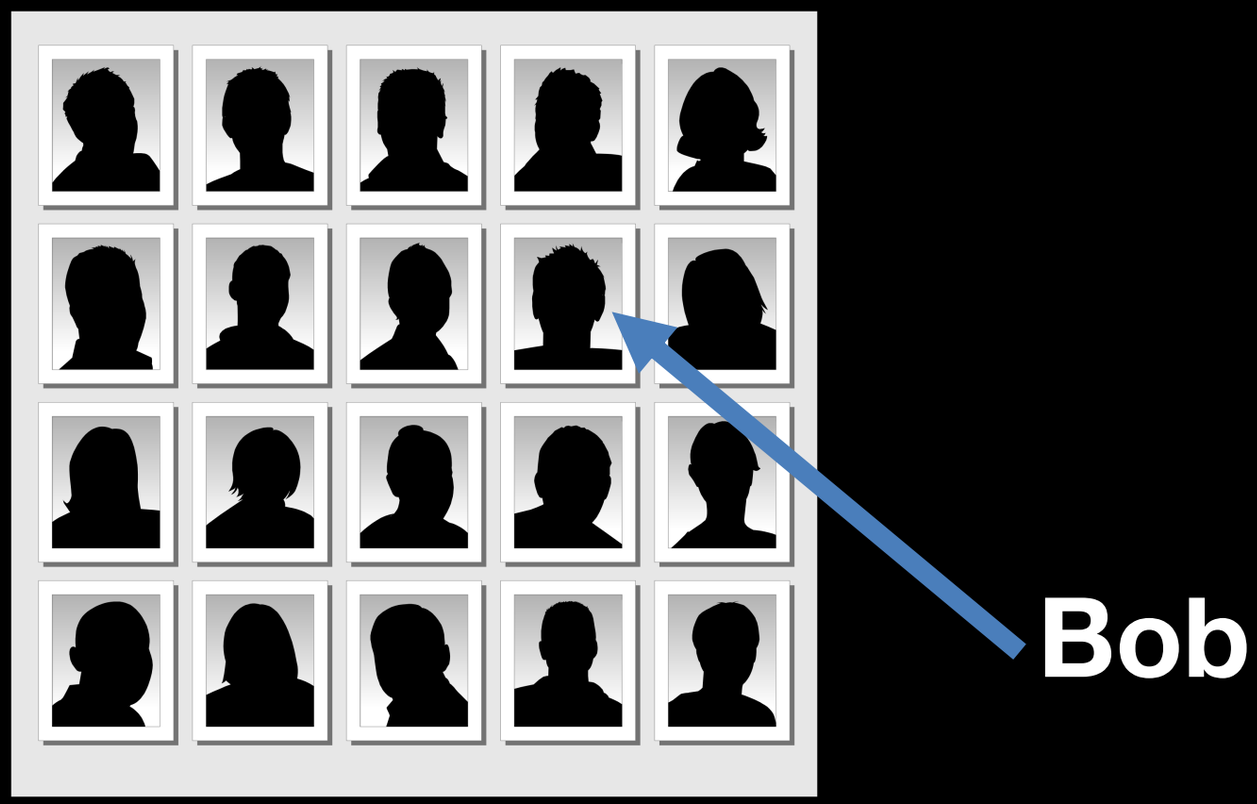
Threats and Attacks

- **Repudiation:** the denial of a commitment or data receipt.
  - This involves an attempt to back out of a contract or a protocol that requires the different parties to provide receipts acknowledging that data has been received.

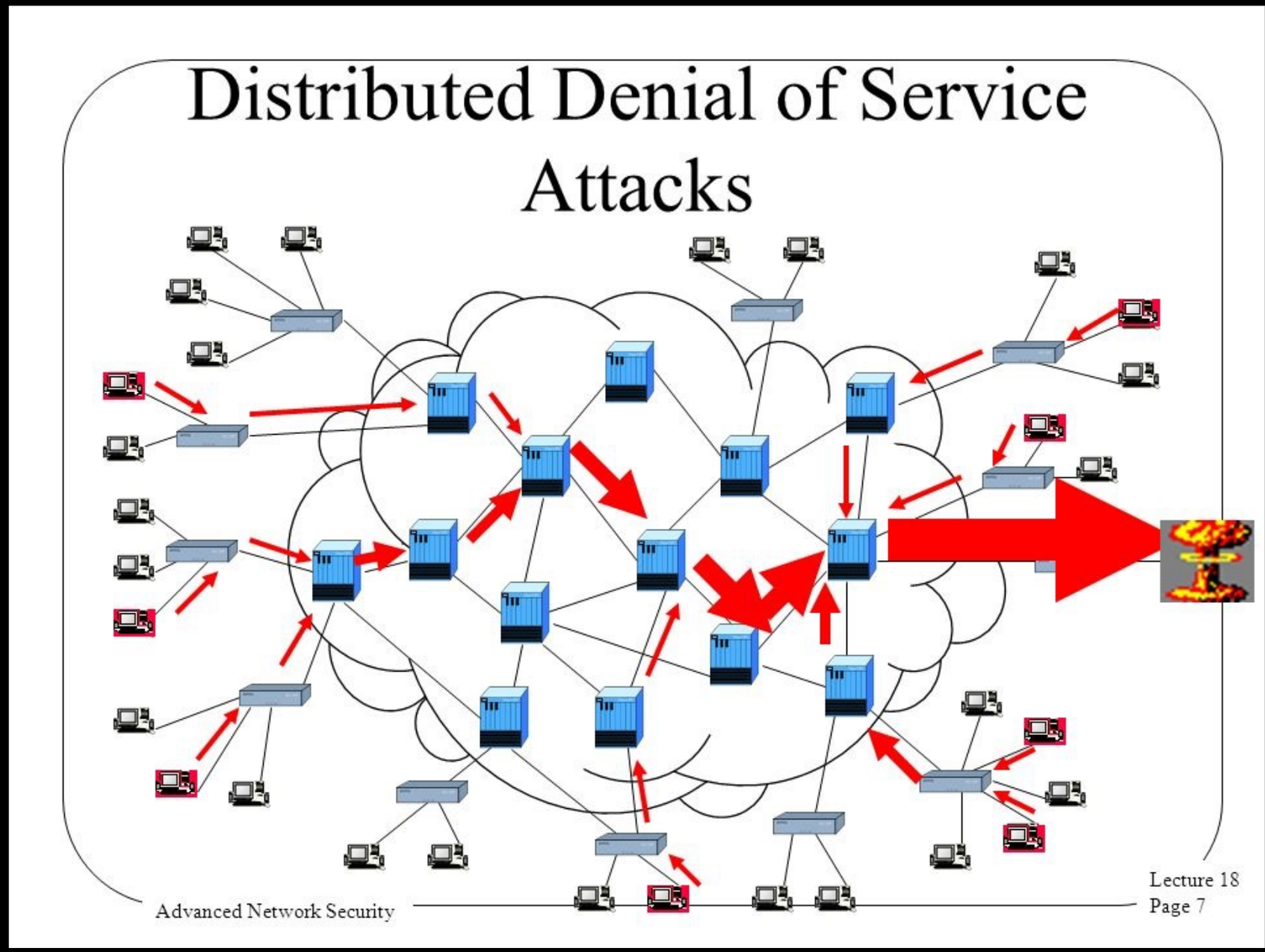
Public domain image from http://commons.wikimedia.org/wiki/File:Plastic_eraser.jpeg

Threats and Attacks

- **Correlation** and **traceback:** the integration of multiple data sources and information flows to determine the source of a particular data stream or piece of information.



Denial of Service



Source: Peter Reiher

Security Principles

- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege
- Least common mechanism
- Psychological acceptability
- Work factor
- Compromise recording

Economy of mechanism

- This principle stresses **simplicity** in the **design** and **implementation** of security measures.
  - While applicable to most engineering endeavors, the notion of simplicity is especially important in the security domain, since a simple security framework facilitates its understanding by developers and users and enables the efficient development and verification of enforcement methods for it.

Fail-safe defaults

- This principle states that the default configuration of a system should have a **conservative protection scheme**.
  - For example, when adding a new user to an operating system, the default group of the user should have minimal access rights to files and services. Unfortunately, operating systems and applications often have default options that favor usability over security.
  - This has been historically the case for a number of popular applications, such as web browsers that allow the execution of code downloaded from the web server.

Complete mediation

- The idea behind this principle is that every access to a resource must be checked for **compliance with a protection scheme**.
  - As a consequence, one should be wary of performance improvement techniques that save the results of previous authorization checks, since permissions can change over time.
  - For example, an online banking web site should require users to sign on again after a certain amount of time, say, 15 minutes, has elapsed.

Open design

- According to this principle, the security architecture and **design** of a system should be made **publicly available**.
  - Security should rely only on keeping cryptographic keys secret.
  - Open design allows for a system to be scrutinized by multiple parties, which leads to the early discovery and correction of security vulnerabilities caused by design errors.
- The open design principle is the opposite of the approach known as **security by obscurity,** which tries to achieve security by keeping cryptographic algorithms secret and which has been historically used without success by several organizations.

Separation of privilege

- This principle dictates that **multiple conditions** should be required to achieve access to restricted resources or have a program perform some action.

Least privilege

- Each program and user of a computer system should operate with the bare **minimum privileges necessary** to function properly.
  - If this principle is enforced, abuse of privileges is restricted, and the damage caused by the compromise of a particular application or user account is minimized.
  - The military concept of **need-to-know** information is an example of this principle.

Least common mechanism

- In systems with multiple users, mechanisms allowing resources to be **shared by more than one user should be minimized**.
  - For example, if a file or application needs to be accessed by more than one user, then these users should have separate channels by which to access these resources, to prevent unforeseen consequences that could cause security problems.

Psychological acceptability

- This principle states that user interfaces should be **well designed and intuitive**, and all security-related settings should adhere to what an ordinary user might expect.

Work factor

- According to this principle, the **cost of overcoming** a security mechanism should be compared with the resources of an attacker when designing a security scheme.
  - A system developed to protect student grades in a university database, which may be attacked by snoopers or students trying to change their grades, probably needs less sophisticated security measures than a system built to protect military secrets, which may be attacked by government intelligence organizations.

Compromise recording

- This principle states that sometimes it is more desirable to **record the details** of an intrusion than to adopt more sophisticated measures to prevent it.
  - Internet-connected surveillance cameras are a typical example of an effective compromise record system that can be deployed to protect a building in lieu of reinforcing doors and windows.
  - The servers in an office network may maintain logs for all accesses to files, all emails sent and received, and all web browsing sessions.