

# Toolbox:

## Authentication, Access Control, and Cryptography.

---

ICT 3156

# Introduction

---

- What is a toolbox?
- Why is a toolbox needed?
- A person (subject) has something of value. The attacker wants this, and employs means to achieve it. To defend the attacker, the defender needs tools.
- A security toolbox facilitates a system owner to establish a security policy, formally or informally, explicitly or implicitly, and begins taking measures to enforce that policy.

# Introduction

---

- Security strategies
  - Effective threats against the strategies
  - Countermeasures.
- 
- Fundamental Aspects of Computer Security? (From previous session)
  - Controlling threats and vulnerabilities involves a **policy** that specifies **who** (subject) can access **what** (objects) and **how** (by which means).
  - To be effective the policy enforcement must determine:
    - **Who** accurately.
    - Who can **access/not** access (example?) what.
    - How to safely limit access to intended subjects by transforming data.

# Security Tools

---

- Authentication and its techniques and technologies.
  - The property of accurate identification is called **authentication**.
- Access Control mechanisms.
  - Allowing exactly those accesses which is authorized is called **access control**.
- Cryptography or encryption primarily.
  - **Encryption** is a tool by which we can transform data so only intended receivers.

# Authentication

---

- The basis of computer security is controlled access: **someone** is authorized to take some **action** on **something**.
- For access control to work, what needs to be ensured?
- Determining who a person really is consists of two separate steps:
  - Identification** is asserting who a person is.
  - Authentication** is proving that asserted identity.

# Identification Versus Authentication

---

- Identities are often well known, public, and not protected.
- Authentication should be private, reliable and necessarily protected.
- Examples of identification and authentication?
- Identifiers may be widely known or easily guessed/determined. Ramifications?
- Case Studies: How by just knowing the identification, security can be compromised?

# Authentication mechanisms

---

Authentication mechanisms use any of **three qualities** to confirm a user's identity:

- Something the user knows.
  - Passwords, PIN numbers, passphrases, and so on.
- Something the user is.
  - Authenticators, called biometrics, are based on a physical characteristic of the user.
  - Fingerprint, voice signature, face, iris, and so on.
- Something the user has.
  - Identity badges, physical keys, a driver's license, or a uniform.

# Authentication Based on Phrases and Facts: Something You Know

---

- Vulnerabilities are rampant in the most common authentication parameter, the password.
- Nature of passwords,
- Criteria for selecting them, and
- Ways of using them for authentication.
- Vulnerability of information we share ourselves. Case Study.



# Password Use

---

- How is a password used (for authentication)?
- Difficulties in using:
  - Use:** Supplying a password for each access to an object can be inconvenient and time consuming.
  - Disclosure.**
  - Revocation:** To revoke one user's access right to an object, someone must change the password, thereby causing the same problems as disclosure.
  - Loss:** Depending on how the passwords are implemented, it may be impossible to retrieve a lost or forgotten password.

# Attacking and Protecting Passwords

---

- Passwords may be easily attacked. True/False? Why?

- Password guessing steps (in increasing degree of difficulty):

1. No password.
2. Same as the user ID.
3. Is, or is derived from, the user's name.
4. On a common word list.
5. Contained in a short college dictionary.
6. Contained in a complete English word list.
7. Contained in common non-English-language dictionaries.
8. Contained in a short college dictionary with capitalizations or substitutions.
9. Contained in a complete English dictionary with capitalizations or substitutions.
10. Contained in common non-English dictionaries with capitalization or substitutions.
11. Obtained by brute force, trying all possible combinations of alphabetic characters.
12. Obtained by brute force, trying all possible combinations from the full character set.

# Attacking Passwords

- Every password can be guessed; password strength is determined by how many guesses are required.

1. Dictionary Attacks
2. Inferring Passwords Likely for a User
3. Guessing Probable Passwords
4. Defeating Concealment
5. Exhaustive Attack

**Dictionary Attacks**

- General network users great deliberators of phrases, sentence forms, character names, places, mythological figures, Chinese words, Yiddish words, and other special-interests.
- Ability and frequency?
- These lists help site administrators identify users who have chosen weak passwords, but the same dictionaries can also be used by attackers of sites that do not have such offensive administration.

**Inferring Passwords Likely for a User**

- People typically choose personal passwords. Why?
- Trying this limited number of passwords by computer takes well under a second.
- Several tests conducted over password databases in different years showed that passwords are highly vulnerable and can easily be broken in a comparatively brief time.
- Once broken.

**Guessing Probable Passwords**

- Common passwords—such as 123456, 666666, 111111—are used astonishingly often.
- If answer to any of the following 3 questions is yes, then the password is not strong or isn't strong.
- Is the word you thought of long? Is it uncommon? Is it hard to spell or to pronounce?
- Example: There are only  $15^4 + 25^4 + 25^4 = 15,275$  passwords of length 4 or less. At an assumed rate of one password per millisecond, all these passwords can be checked in 28.175 seconds.
- Lots of common passwords are easy to find.

**Defeating Concealment**

- Easier than guessing a password is just to read one from a table. Not the ones found in the.
- Storing systems store passwords in hidden (encrypted) form.
- The only valid point is that the process is **one-way**. No it and Denial?
- Limited number of attempts.
- If the attacker obtains an encrypted password table and knows the corresponding algorithm, a computer program can easily test hundreds of thousands of guesses in a matter of minutes.
- Rainbow table: precomputed list of popular values, such as passwords.
- Right user-specific component joined to an encrypted password to distinguish identical passwords.

**Exhaustive Attack**

- In an exhaustive or brute force attack, the attacker tries all possible passwords, usually in order submitted below.
- For example, assume passwords are words consisting of the 26 characters A-Z and can be of any length from 1 to 8 characters. How much time will it take to exhaust the search at the rate of 1 password per millisecond?

# Dictionary Attacks

---

- Several network sites post dictionaries of phrases, science fiction character names, places, mythological names, Chinese words, Yiddish words, and other specialized lists.
- Merits and Demerits?
- These lists help site administrators identify users who have chosen weak passwords, but the same dictionaries can also be used by attackers of sites that do not have such attentive administrators.

# Inferring Passwords Likely for a User

---

- People typically choose personal passwords. Why?
- Trying this limited number of passwords by computer takes well under a second!
- Several tests conducted over password datasets in different years showed that passwords are highly vulnerable and can easily be broken in a comparably lesser time.
- Case studies.

# Guessing Probable Passwords

---

- Common passwords—such as qwerty, password, 123456—are used astonishingly often.
- If answer to any of the following 3 questions is No, then the password is not strong or safe enough.
- Is the word you thought of long? Is it uncommon? Is it hard to spell or to pronounce?
- Example: There are only  $26^1 + 26^2 + 26^3 = 18,278$  passwords of length 3 or less. At an assumed rate of one password per millisecond, all these passwords can be checked in 18.278 seconds.
- Lists of common passwords are easy to find.

# Defeating Concealment

---

- Easier than guessing a password is just to read one from a table, like the ones stored in OS.
- Operating systems store passwords in hidden (encrypted) form.
- The only critical point is that the process be **one-way**. Merit and Demerit?
- Limited number of attempts.
- If the attacker obtains an encrypted password table and learns the concealment algorithm, a computer program can easily test hundreds of thousands of guesses in a matter of minutes.
- Rainbow table**: precomputed list of popular values, such as passwords.
- Salt**: user-specific component joined to an encrypted password to distinguish identical passwords.

# Exhaustive Attack

---

- In an exhaustive or brute force attack, the attacker tries all possible passwords, usually in some automated fashion.
- For example, assume passwords are words consisting of the 26 characters A–Z and can be of any length from 1 to 8 characters. How much time will it take to exhaust the search at the rate of 1 password per microsecond?



# Example

## Rainbow Table

Identity	Password
Jane	qwerty
Pat	aaaaaa
Phillip	oct31witch
Roz	aaaaaa
Herman	guessme
Claire	aq3wm\$oto!4

Original Password	Encrypted Password
asdfg	0x023c94fc
p@55w0rd	0x04ff38d9
aaaaaa	0x13b9c32f
password	0x2129f30d
qwerty	0x471aa2d2
12345678	0x4f2c4dd8
123456	0x5903c34d
aaaaa	0x8384a8c8

## Salt

Identity	ID+password (not stored in table)	Stored Authentication Value
Jane	Jan+qwerty	0x1d46e346
Pat	Pat+aaaaaa	0x2d5d3e44
Phillip	Phi+oct31witch	0xc23c04d8
Roz	Roz+aaaaaa	0xe30f4d27
Herman	Her+guessme	0x8127f48d
Claire	Cla+aq3wm\$oto!4	0x5209d942

Identity	Password
Jane	0x471aa2d2
Pat	0x13b9c32f
Phillip	0x01c142be
Roz	0x13b9c32f
Herman	0x5202aae2
Claire	0x488b8c27

# Attacking Passwords

---

- Every password can be guessed; password strength is determined by how many guesses are required.
  1. Dictionary Attacks
  2. Inferring Passwords Likely for a User
  3. Guessing Probable Passwords
  4. Defeating Concealment
  5. Exhaustive Attack
- All these techniques to defeat passwords, combined with usability issues, indicate that we need to look for other methods of authentication.

# Protecting Passwords

---

- Good Passwords
- Other Things Known
- Security Questions

While choosing passwords:

1. Use characters other than just a–z.
2. Choose long passwords.
3. Avoid actual names or words.
4. Use a string you can remember.
5. Use variants for multiple passwords.
6. Change the password regularly.
7. Don't write it down.
8. Don't tell anyone else.

# Authentication Based on Biometrics: Something You Are

---

- Biometrics are biological properties, based on some physical characteristic of the human body.
- Examples: fingerprint, hand geometry (shape and size of fingers), retina and iris, voice, handwriting, signature, hand motion, typing characteristics, blood vessels in the finger or hand face, facial features, such as nose shape or eye spacing.
- Advantages?
- Biometric cannot be lost, stolen, forgotten, or shared and is always available, always at hand, so to speak. These characteristics are difficult, if not impossible, to forge.





# Problems with Use of Biometrics

---

- Biometrics are relatively *new*, and some people find their use *intrusive*.
- Biometric recognition devices are *costly*.
- Biometric readers and comparisons can become a *single point of failure*.
- All biometric readers use *sampling* and establish a *threshold* for acceptance of a close match.
- Although equipment accuracy is improving, *false readings* still occur.
- The *speed* at which a recognition must be done limits accuracy.
- Although we like to think of biometrics as unique parts of an individual, *forgeries* are possible. Case Study.
- Any other?

# False Positives and Negatives

- False positive: incorrectly confirming an identity.
- False negative: incorrectly denying an identity.
- Dichotomous system or test : When a screening system compares something it has with something it is measuring.

} Errors to be avoided.

Reference Standard for Describing Dichotomous Tests

	Is the Person Claimed	Is Not the Person Claimed
Test is positive (There is a match.)	True Positive = a	False Positive = b
Test is negative (There is no match.)	False Negative = c	True Negative = d

# False Positives and Negatives

	Is the Person Claimed	Is Not the Person Claimed
Test is positive (There is a match.)	True Positive = a	False Positive = b
Test is negative (There is no match.)	False Negative = c	True Negative = d

Four standard measures to measure success:

- Sensitivity  $= a / (a + c)$

The proportion of positive results among all possible correct matches.

- Specificity  $= d / (b + d)$

The proportion of negative results among all people who are not sought.

- Prevalence  $= (a + c) / (a + b + c + d)$

Tells us how common a certain condition or situation is.

- Accuracy  $= (a + d) / (a + b + c + d)$

Measures the degree to which the test correctly flags the condition or situation.



# Problems with Use of Biometrics

---

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:
  1. Registration:
  2. Authentication:

# Problems with Use of Biometrics

---

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:
  1. Registration:
    - A characteristic of the user (hand, for example) is captured and reduced to a set of data points.
    - The user may be asked to present the hand several times.
    - Registration produces a pattern, called a **template**, of the data points particular to a specific user.
  2. Authentication:

# Problems with Use of Biometrics

---

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:
  1. Registration:
  2. Authentication:
    - The system remeasures the hand and compares the new measurements with the stored template.
    - If the new measurement is close enough to the template, the system accepts the authentication; otherwise, the system rejects it.

# Accuracy of Biometrics

---

- Biometric authentication means a subject matches a template closely enough. “Close” is a system parameter that can be tuned.
- Measuring the accuracy of biometric authentication is difficult because the authentication is not unique.
- Case Studies.

# Authentication Based on Tokens: Something You Have

---

- Something you have means that you have a physical object in your possession.

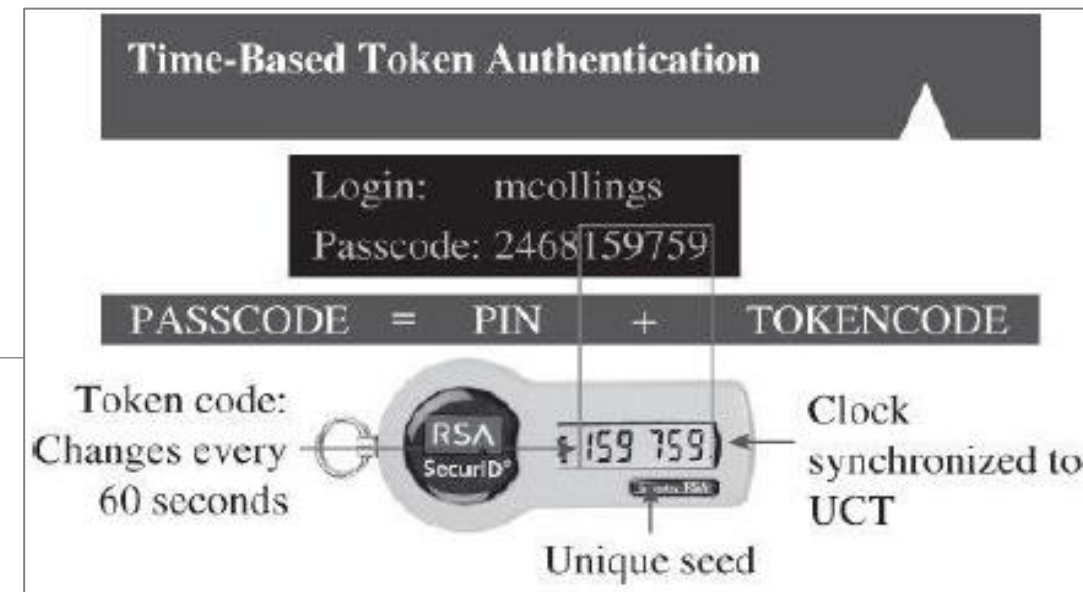


# Active and Passive Tokens

---

- **Passive** tokens do not change.
- Example: Key, ID card.
- **Active** tokens communicate with a sensor.
- Have some variability or interaction with its surroundings.
- Example: Metro card.

# Static and Dynamic Tokens



- The value of a **static token** remains fixed.
- Examples: Keys, identity cards, passports, credit and other magnetic-stripe cards, and radio transmitter cards (called RFID devices).
- Static tokens are most useful for **onsite authentication**.
- Dynamic tokens** have computing power on the token to change their internal state.
- A dynamic authentication token is essentially a device that generates an unpredictable value often called as a **pass number**.
- Dynamic token generators are useful for **remote authentication**, especially of a person to a computer.
- Skimming**

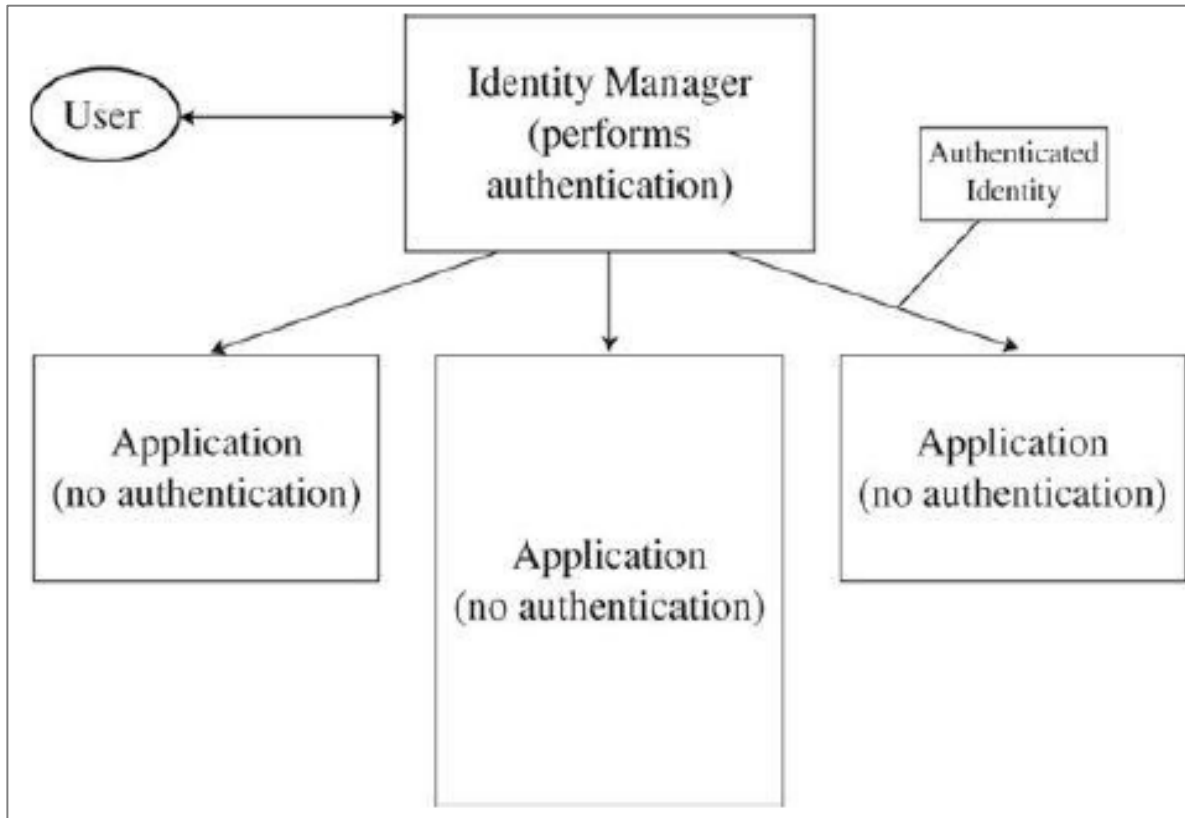
# Federated Identity Management

---

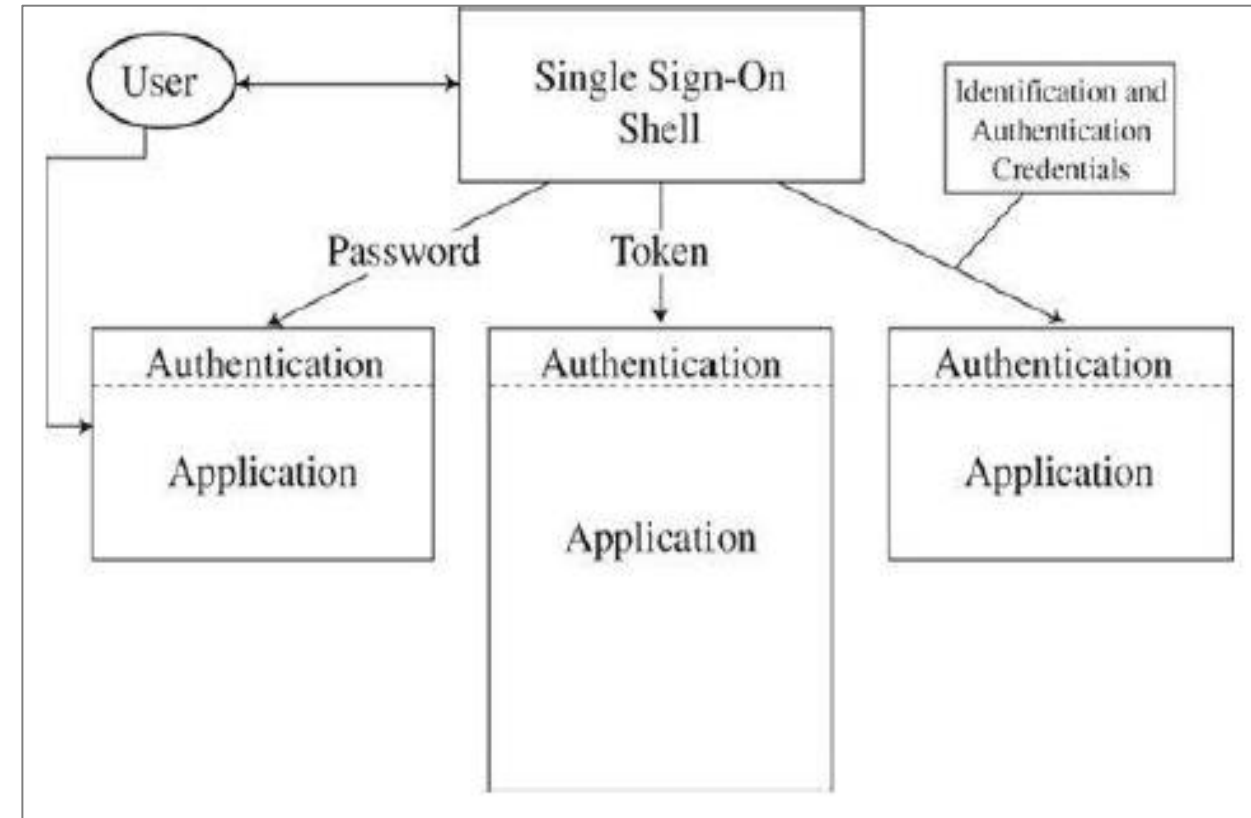
- **Federated identity management** unifies the identification and authentication process for a group of systems. Necessity?
- A federated scheme maintains one profile with one authentication method.
- Separate systems share access to the authenticated identity database.
- Authentication is performed in one place, and separate processes and systems determine that an already authenticated user is to be activated.
- Very similar to a **Single Sign-On** process.
- SSO involves an umbrella procedure where the user logs in once per session.
- The umbrella procedure maintains user identities and authentication codes for all the different processes the user accesses.



# Federated Identity Management vs SSO



Federated Identity Manager



Single Sign-On

# Multifactor Authentication

---

- The single-factor authentication offer advantages and disadvantages.
- Combining authentication information is called **multifactor authentication**.
- As long as the process does not become too onerous, authentication can use two, three, four, or more factors.
- Two-factor authentication.
- From a usability point of view, large values of  $n$  may lead to user frustration and reduced security



# Secure Authentication

---

- Passwords, biometrics, and tokens can all participate in secure authentication.
- No guarantee that an authentication approach will be secure.
- Think about blocking possible attacks and attackers.
- Ways?
- Limiting users to certain workstations or certain times of access can cause complications.
- Security over inconvenience.
- Recognize qualities that distinguish normal, allowed activity.

# Access Control

---

- Limiting **who** can access **what** in **what ways**.
- A **subject** is permitted to access an **object** in a particular **mode**, and only such authorized accesses are allowed.
- Effective separation will keep unauthorized subjects from unauthorized access to objects, but the separation gap must be crossed for authorized subjects and modes.

# Access Policies


---

- A given subject either can or cannot access a particular object in a specified way.
- Before trying to implement access control, an organization needs to take the time to develop a higher-level security policy, which will then drive all the access control rules.
- Effective Policy Implementation
- Tracking
- Granularity
- Access Log
- Limited Privilege


# Effective Policy Implementation

---

- Check every access.
- Enforce least privilege.
- Verify acceptable usage.



The principle of least privilege states that a subject should have access to the smallest number of objects necessary to perform some task.



Ability to access is a yes-or-no decision. But equally important is checking that the activity to be performed on an object is appropriate.

# Tracking

---

- Implementing an appropriate policy is not the end of access administration.
- Administrators need to revisit the access policy to determine whether it is working as it should.
- It must be ensured on timely basis that:
  - Nobody has acquired many no-longer-needed rights.
  - Objects must be suitably split so that individuals can be allowed access to only the pieces they need.

# Granularity

---

- Granularity: the fineness or **specificity** of access control.
- The finer the granularity, the larger number of access control decisions that must be made, so there is a performance penalty.
- Typically a file, a program, or a data space is the smallest unit to which access is controlled.
- Hardware devices, blocks of memory, the space on disk where program code is stored, specific applications, all these are likely objects over which access is controlled.



# Access Log

---

- Audit log: Created and maintained by systems that records which accesses have been permitted.
- Preserved for later analysis.
- Reasons for logging access :
  - Can help plan for new or upgraded equipment, by showing which items have had heavy use.
  - If the system fails, these records can show what accesses were in progress and perhaps help identify the cause of failure.
  - If a user misuses objects, the access log shows exactly which objects the user did access.
  - In the event of an external compromise, the audit log may help identify how the assailant gained access and which data items were accessed.

# Limited Privilege

---

- **Limited privilege** is the act of restraining users and processes so that any harm they can do is not catastrophic.
- Not all users are ethical or even competent and not all processes function as intended.
- Limited privilege is a management concept, not a technical control.

# Implementing Access Control

---

- Access control is often performed by the operating system. Hindrances?
- Current hardware design limits some operating system designs, specially in access control. Justify.
- OS does not usually see inside files or data objects. So it cannot perform row- or element-level access control within a database.
- OS cannot easily differentiate among kinds of network traffic.
- In these cases, the operating system defers to a database manager or a network appliance in implementing some access control aspects.

# Reference Monitor

---

- Access control depends on a combination of hardware and software.
- **Reference monitor:** access control that is always invoked, tamperproof, and verifiable (assuredly correct).
- It could be embedded in an application (to control the application's objects), part of the operating system (for system-managed objects) or part of an appliance.
- To have an effective reference monitor, effective and efficient means to translate policies into action needs to be considered.
- The **representation of a policy in binary data** determines the **efficiency** and **effectiveness** of the mediation.

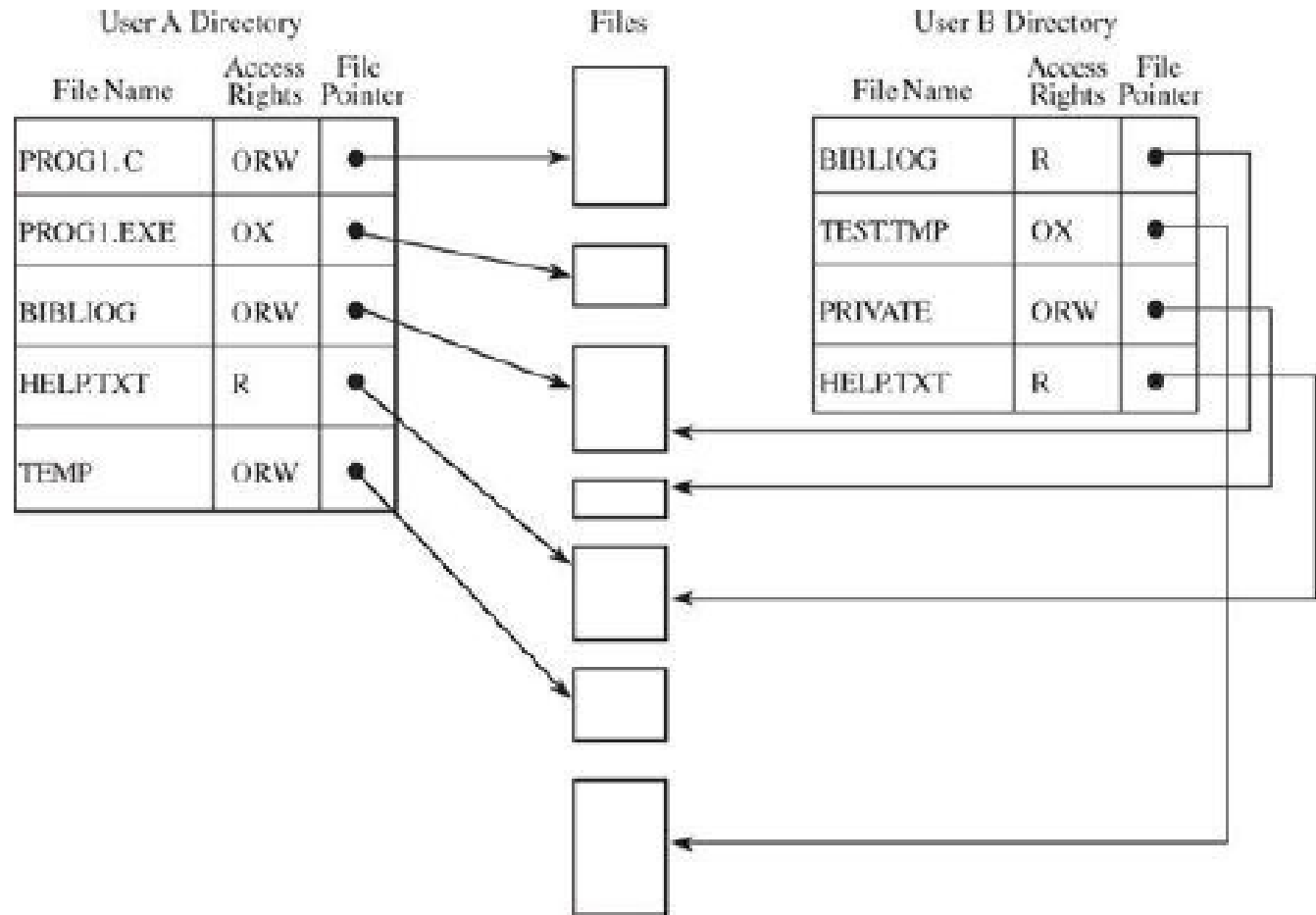
# Implementing Access Control

---

- Models to maintain access rights (implemented by the reference monitor):
  - Access Control Directory
  - Access Control Matrix
  - Access Control List
  - Privilege List
  - Capability

# Access Control Directory

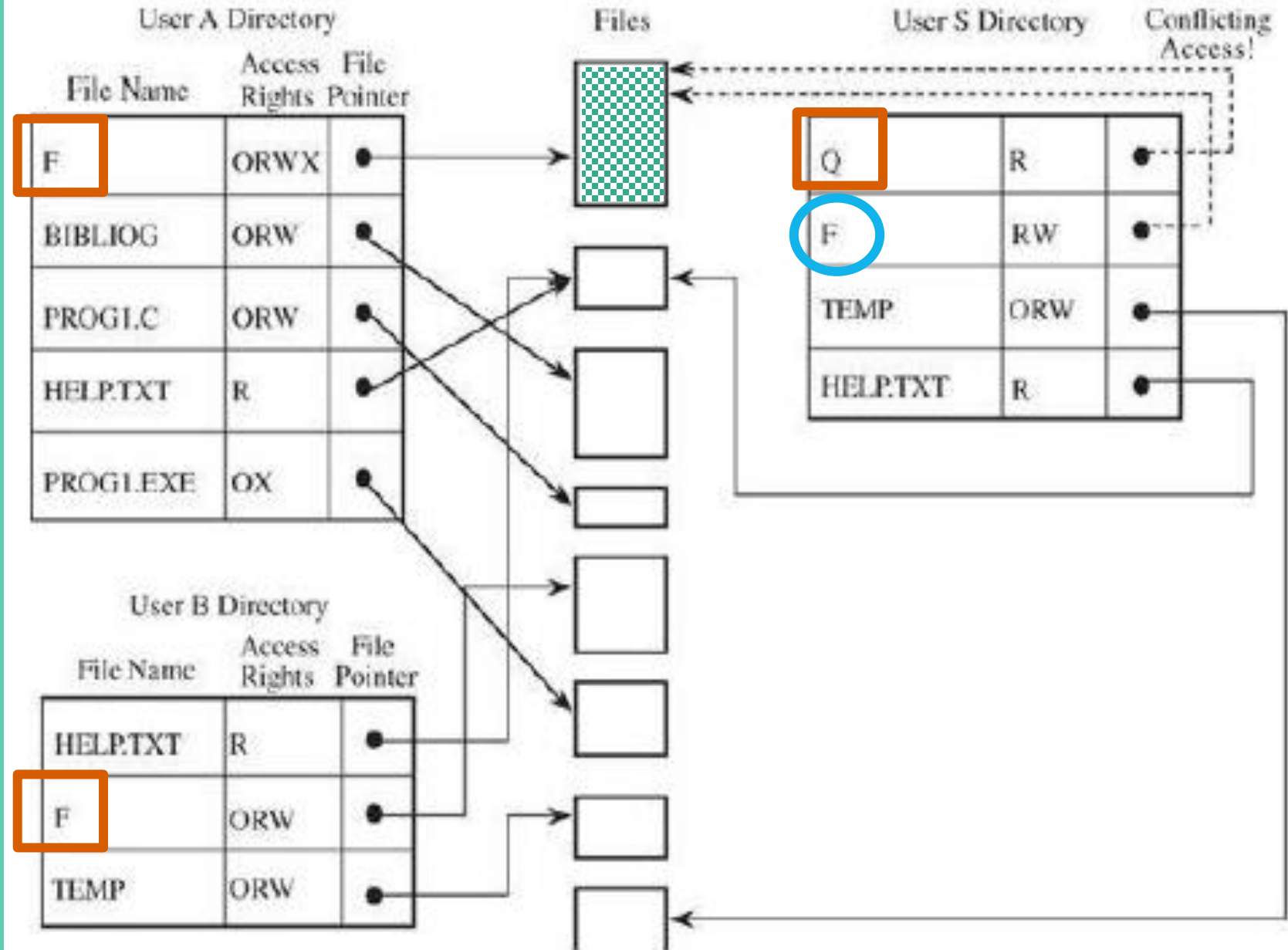
- Every file has a unique **owner** who possesses “control” access rights and to revoke access.
- No user can be allowed to **write in the file directory**, because that would be a way to forge access to a file.
- Therefore, the operating system must maintain all file directories, under commands from the owners of files.
- Easy to implement because it uses one list per user.



# Access Control Directory

## Difficulties:

- List becomes too large if many shared objects are accessible to all users.
- Revocation of access and **propagation of access rights**.
- **Pseudonyms** can lead to multiple permissions that are not necessarily consistent.



# Access Control Matrix

	Bibliog	Temp	F	Help .txt	C_ Comp	Linker	Clock	Printer
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R			R	X	X	R	W
USER S	RW		R	R	X	X	R	W
USER T			R	X	X	X	R	W
SYS MGR				RW	OX	OX	ORW	O
USER SVCS				O	X	X	R	W

Access control matrix is sparse.

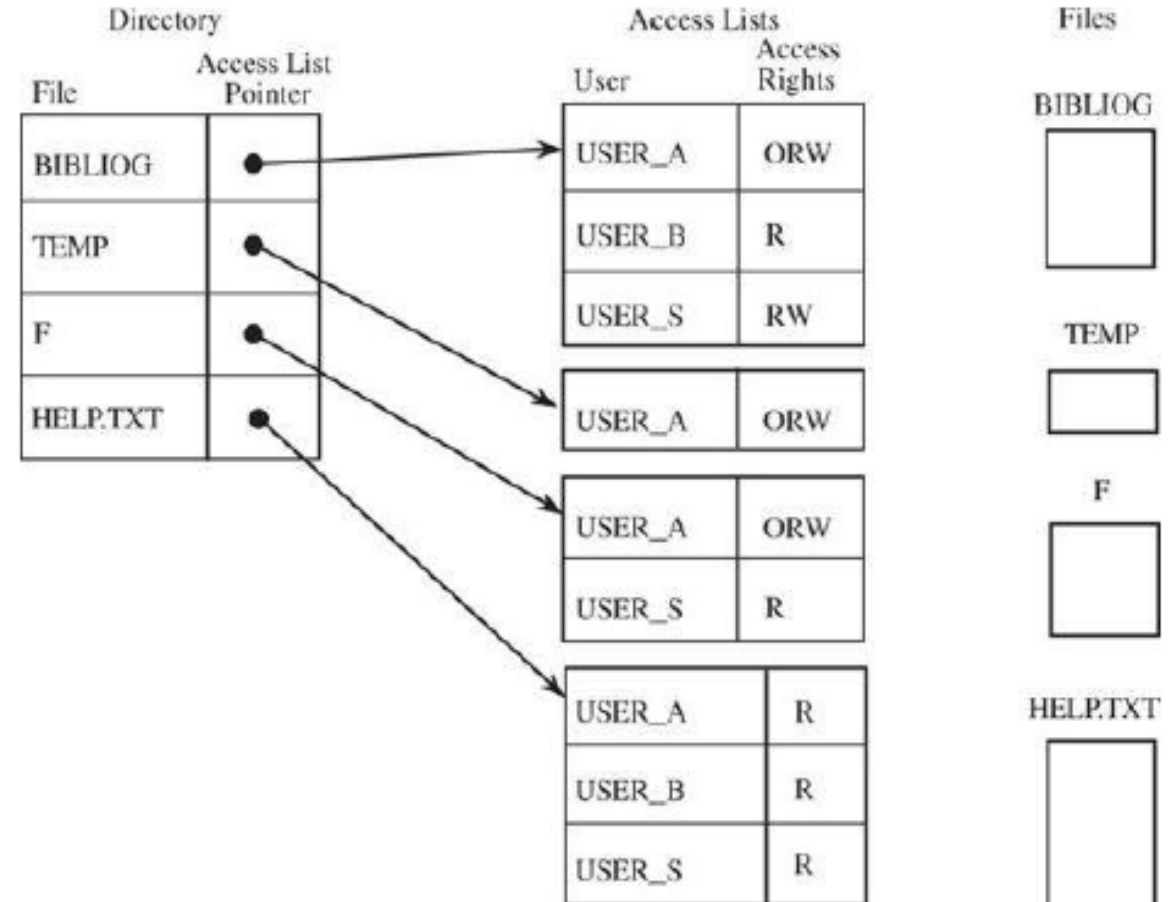
Subject	Object	Right
USER A	Bibliog	ORW
USER B	Bibliog	R
USER S	Bibliog	RW
USER A	Temp	ORW
USER A	F	ORW
USER S	F	R

<Subject, Object, Rights>



# Access Control List

- The list shows all subjects who should have access to the object and what their access is.
- There is one access control list **per object**; a directory is created for each subject.
- Advantages:
- Can include default rights.



# Privilege List

---

- A privilege list, sometimes called a directory, is a row of the access matrix, showing all those privileges or access rights for a given subject.
- Ease of revocation.
- If a user is removed from the system, the privilege list shows all objects to which the user has access so that those rights can be removed from the object.

# Capability

---

- A capability is an **unforgeable** token that gives the possessor certain rights to an object.
- A capability is just one access control triple of a subject, object, and right.
- A user can create completely new objects and can define types of accesses previously unknown to the system.
- One possible access right to an object is transfer or **propagate**.
- Example: User passing access rights to another user.
- Concept of Domain (collection of objects to which the process has access).
- Example: When a process executes, it operates in a domain or local name space.

# Procedure-Oriented Access Control

---

- A procedure that controls access to objects.
- The procedure forms a capsule around the object.
- Procedures can perform actions specific to a particular object in implementing access control.
- Example: Table of valid users in OS. addUser(), deleteUser(), verifyUser().
- Implements the principle of **information hiding** because the means of implementing an object are known only to the object's control procedure.
- Inefficient: no simple, fast access checking.

# Role-Based Access Control

---

- Some users (such as administrators) should have significant privileges, while others (such as regular users or guests) must have lower privileges.
- Role-based access control lets us associate privileges with groups.
- Administering security is easier if we can control access by job demands, not by person.
- Access control keeps up with a person who changes responsibilities.
- System administrator does not have to choose the appropriate access control settings for someone.

GOODMORNING

jrrgpruqlqj  
gpqdnqroknh  
fznlk kmeww d  
haqee cqwob h

Caesar Cipher  
Vigenère Cipher (Key : abc)  
Enigma M3 : UKW B  
Enigma I : UKW A

# Cryptography

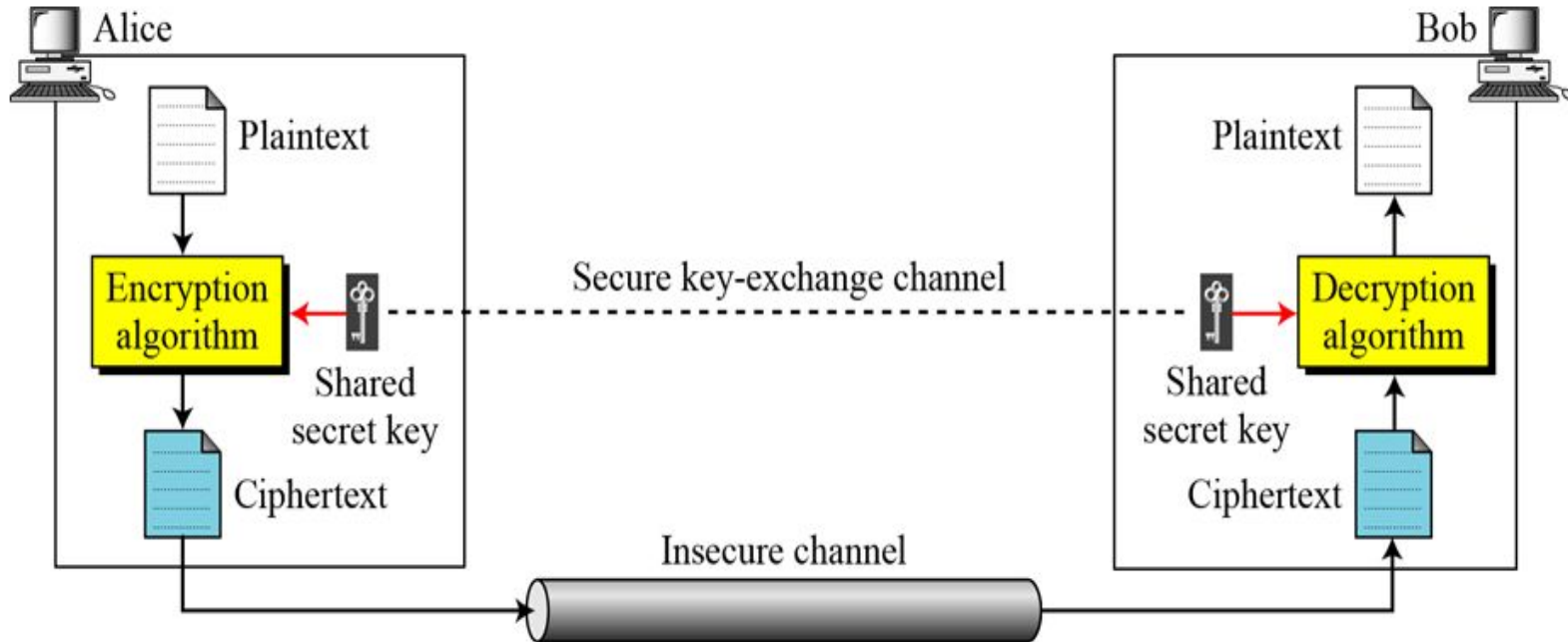
---

- **Crypto**graphy conceals data against unauthorized access.
- Well-disguised data cannot easily be read, modified, or fabricated.
- Cryptography, though used extensively in context of sending secret messages, also involves protecting any digital object for access only by authorized people.
- Probable exploitations:
  - Interception
  - Interruption
  - Modification
  - Fabrication

# Cryptography: Basic Idea

Entities:

- Message, M.
- Sender, S.
- Recipient, R.
- Transmission medium, T  
(Anybody S entrusts the message with).
- Interceptor or Intruder, O.





# Terminology

- Encryption

- Decryption

- Cryptosystem

- Plaintext

- Ciphertext

- Algorithms/Ciphers

- Key

Process of **encoding** a message so that its meaning is not obvious.

Reverse process of transforming an encrypted message back into its normal, original form.

Encode/Decode  
Encipher/Decipher

A system for encryption and decryption.

# Terminology

- Encryption

- Decryption

- Cryptosystem

- Plaintext

- Ciphertext

- Algorithms/Ciphers

- Key

The original form of a message.

The encrypted form of a message.

The encryption and decryption rules.

A device used by ciphers for encryption/decryption.

$$C = E (K_E, P)$$

$$P = D (K_D, C)$$

$$P = D (K_D , E(K_E, P))$$

# Terminology

•Cryptanalysis

•Cryptanalyst

•Cryptology

•Work Factor

Cryptanalysis is the investigation of systems, ciphertext, and ciphers in order to reveal the hidden meaning or details of the system itself.

Studies encryption and encrypted messages, hoping to find the hidden meanings.  
A cryptanalyst might work defensively, probing codes and ciphers to see if they are solid enough to protect data adequately.

## **Cryptographer vs Cryptanalyst?**

- A cryptanalyst's chore is to **break** an encryption.
- An analyst works with a variety of information: encrypted messages, known encryption algorithms, intercepted plaintext, data items known or suspected to be in a ciphertext message, mathematical or statistical tools and techniques, and properties of languages, as well as plenty of ingenuity and luck.

# Terminology

---

- Cryptanalysis
- Cryptanalyst
- Cryptology
- Work Factor

Cryptology is the research into and study of encryption and decryption; it includes both cryptography and cryptanalysis.

Amount of effort needed to **break** an encryption (or mount a successful attack).

- An encryption algorithm is called **breakable** when, given enough time and data, an analyst can determine the algorithm. Two issues:
  1. In cryptanalysis there are no rules: Any action is fair play. Cryptanalysts make use of ingenuity.
  2. Estimates of breakability are based on current technology.

# Terminology: Symmetric and Asymmetric Encryption

---

$$\begin{aligned}C &= E(K_E, P) \\P &= D(K_D, C) \\P &= D(K_D, E(K_E, P))\end{aligned}$$

- *Encryption Technique* =  $\begin{cases} \text{Symmetric,} & K_E = K_D \\ \text{Asymmetric,} & K_E \neq K_D \end{cases}$
- Symmetric encryption: one key encrypts and decrypts.
- Asymmetric encryption: one key encrypts, a different key decrypts.
- Keyless Ciphers: An encryption scheme that does not require the use of a key.
- Significance of a key?

# Terminology: Symmetric and Asymmetric Encryption

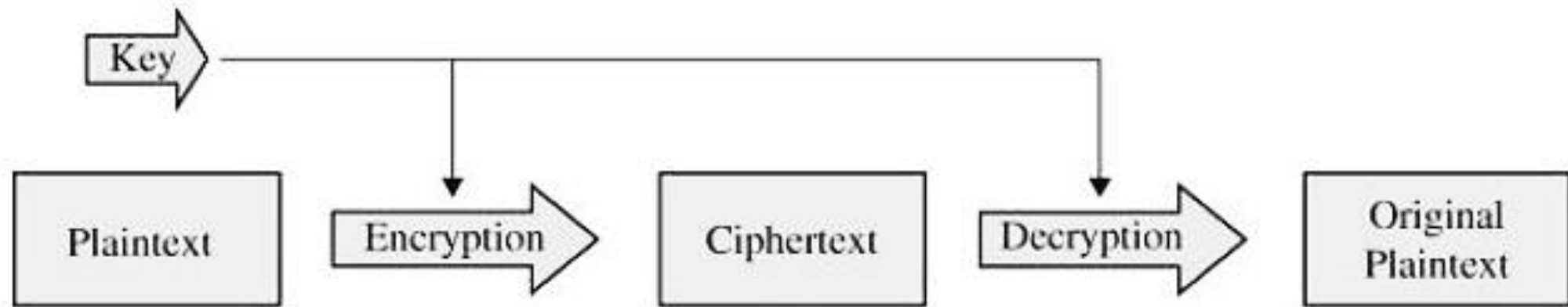
---

- Symmetric Encryption : single-key or secret key encryption.
- The symmetric systems provide a two-way channel to their users: A and B share a secret key, and they can both encrypt information to send to the other as well as decrypt information from the other.
- If the key remains secret, the system also provides **authenticity**. How?
- Advantages and Disadvantages?
- Key Distribution. Key Management.
- 'n' users who want to communicate in pairs need  $n * (n - 1)/2$  keys.
- The number of keys needed increases at a rate proportional to the square of the number of users!

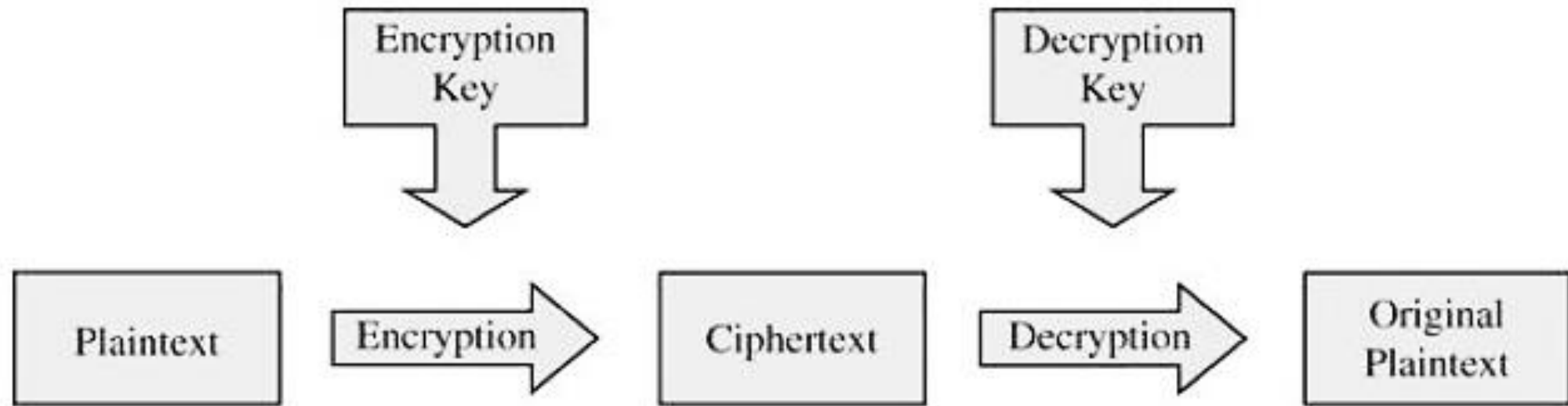
# Terminology: Symmetric and Asymmetric Encryption

---

- Asymmetric Encryption : Public key encryption.
- Precisely matched pairs of keys. The keys are produced together, or one is derived mathematically from the other.
- Key management involves storing, safeguarding, and activating keys.
- Asymmetric systems excel at key management.



(a) Symmetric Cryptosystem

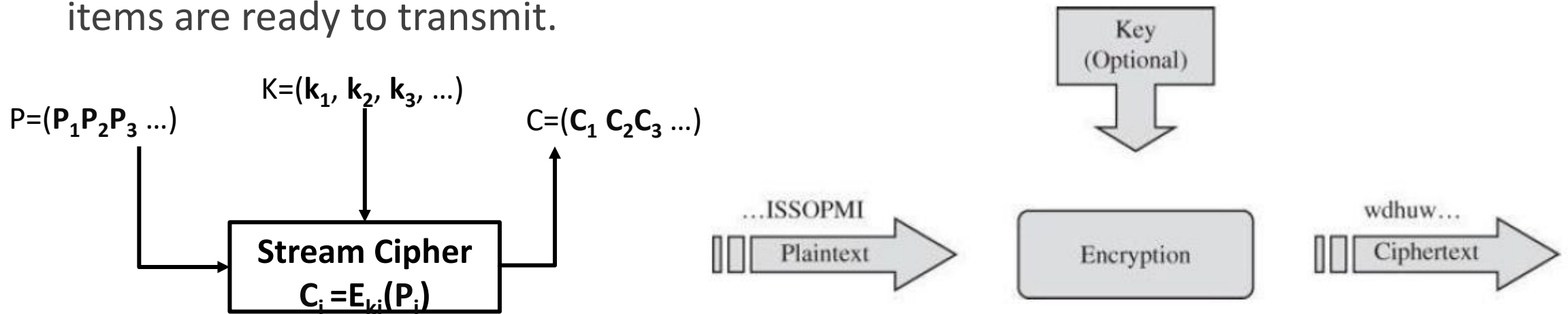


(b) Asymmetric Cryptosystem



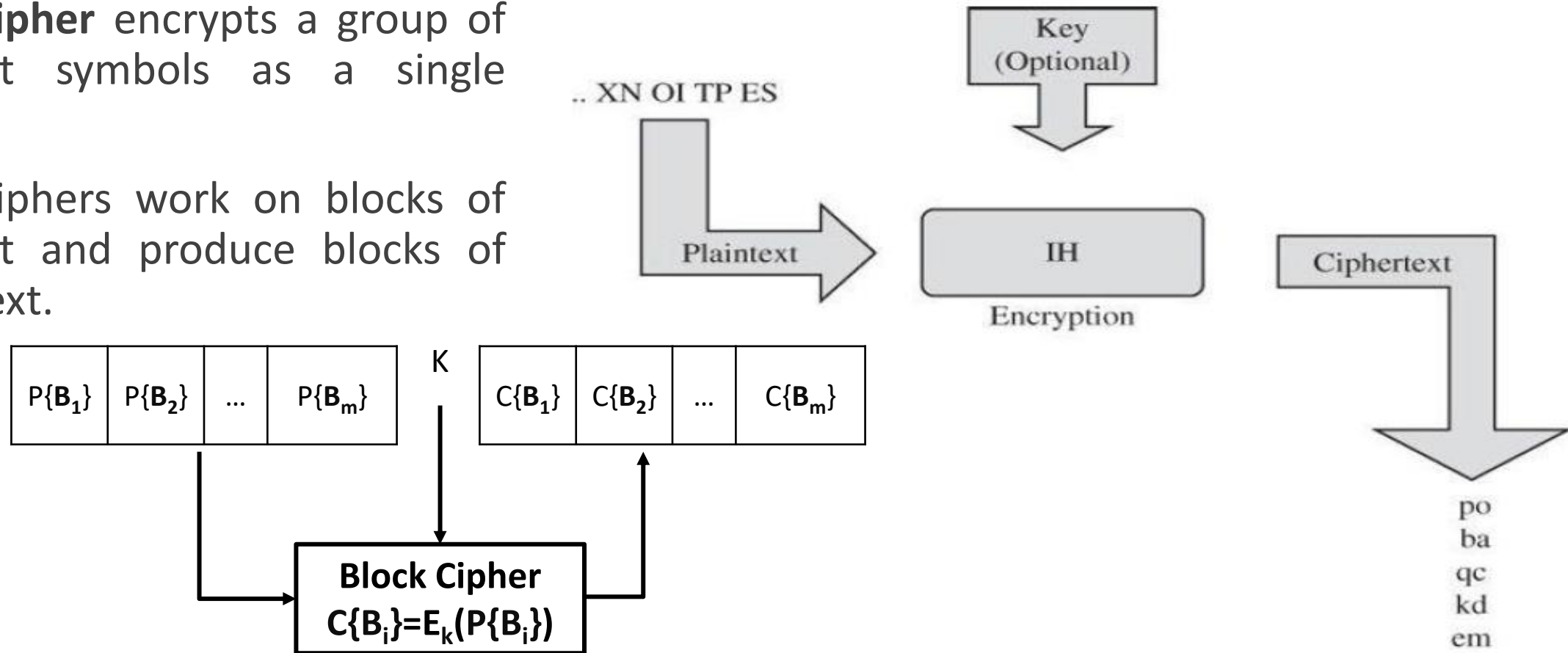
# Stream and Block Ciphers

- Depends on the nature of the data to be concealed.
- **Stream encryption:** Each bit (or perhaps each byte) of the data stream is encrypted separately.
- The input symbols are transformed one at a time.
- Advantage of a stream cipher: can be applied immediately to whatever data items are ready to transmit.



# Stream and Block Ciphers

- **Block cipher** encrypts a group of plaintext symbols as a single block.
- Block ciphers work on blocks of plaintext and produce blocks of ciphertext.



# Stream and Block Ciphers

	Stream	Block
Advantages	<ul style="list-style-type: none"><li>• Speed of transmission</li><li>• Low Error Propagation</li></ul>	<ul style="list-style-type: none"><li>• High Diffusion</li><li>• Immunity to insertion of symbol</li></ul>
Disadvantages	<ul style="list-style-type: none"><li>• Low Diffusion</li><li>• Susceptible to malicious insertions and modifications</li></ul>	<ul style="list-style-type: none"><li>• Slowness of encryption</li><li>• Padding</li><li>• Error Propagation</li></ul>
Examples	Additive Ciphers Shift Cipher Caesar Cipher Playfair Cipher Vigenere Cipher	DES AES

# Data Encryption Standard (DES)

---

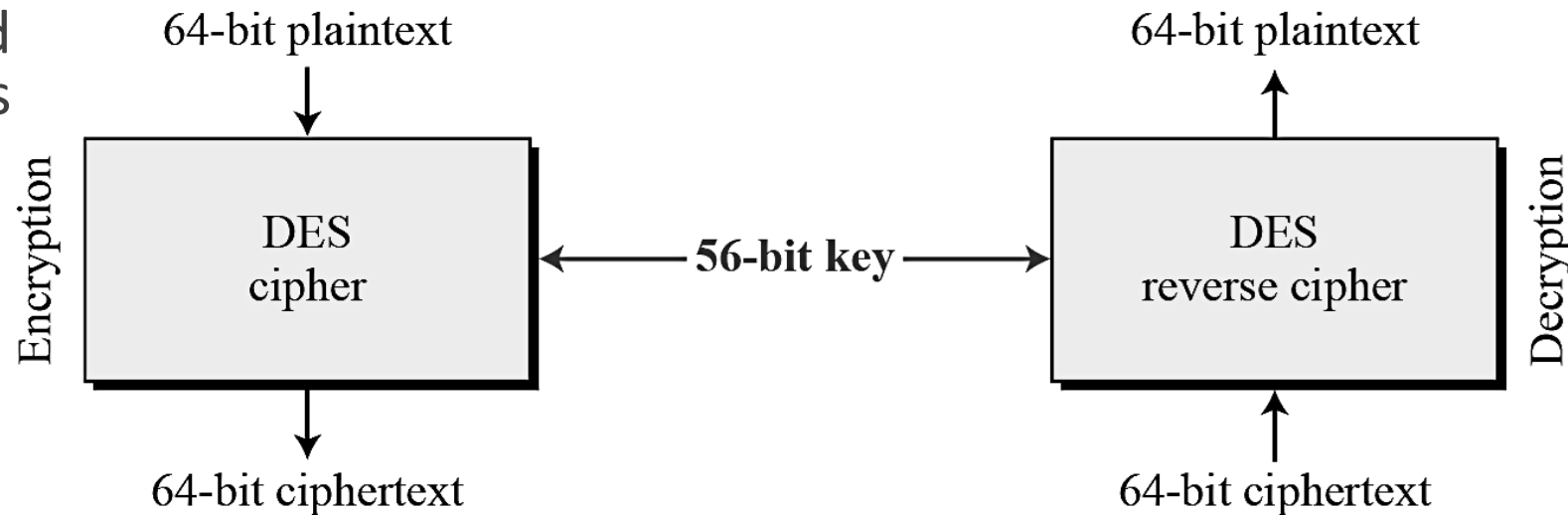
- The DES algorithm was developed in the 1970s by IBM for the U.S. NIST.
- DES is a careful and complex combination of two fundamental building blocks of encryption: **substitution** and **transposition**.
- DESign Overview:

<b>Cipher Type</b>	Feistel, Block
<b>Block Length</b>	64 bits
<b>Key Length</b>	56 bits
<b>Number of Rounds</b>	16
<b>Size of subkey (round key)</b>	48 bits
<b>Cipher Design Components</b>	S-box, P-box

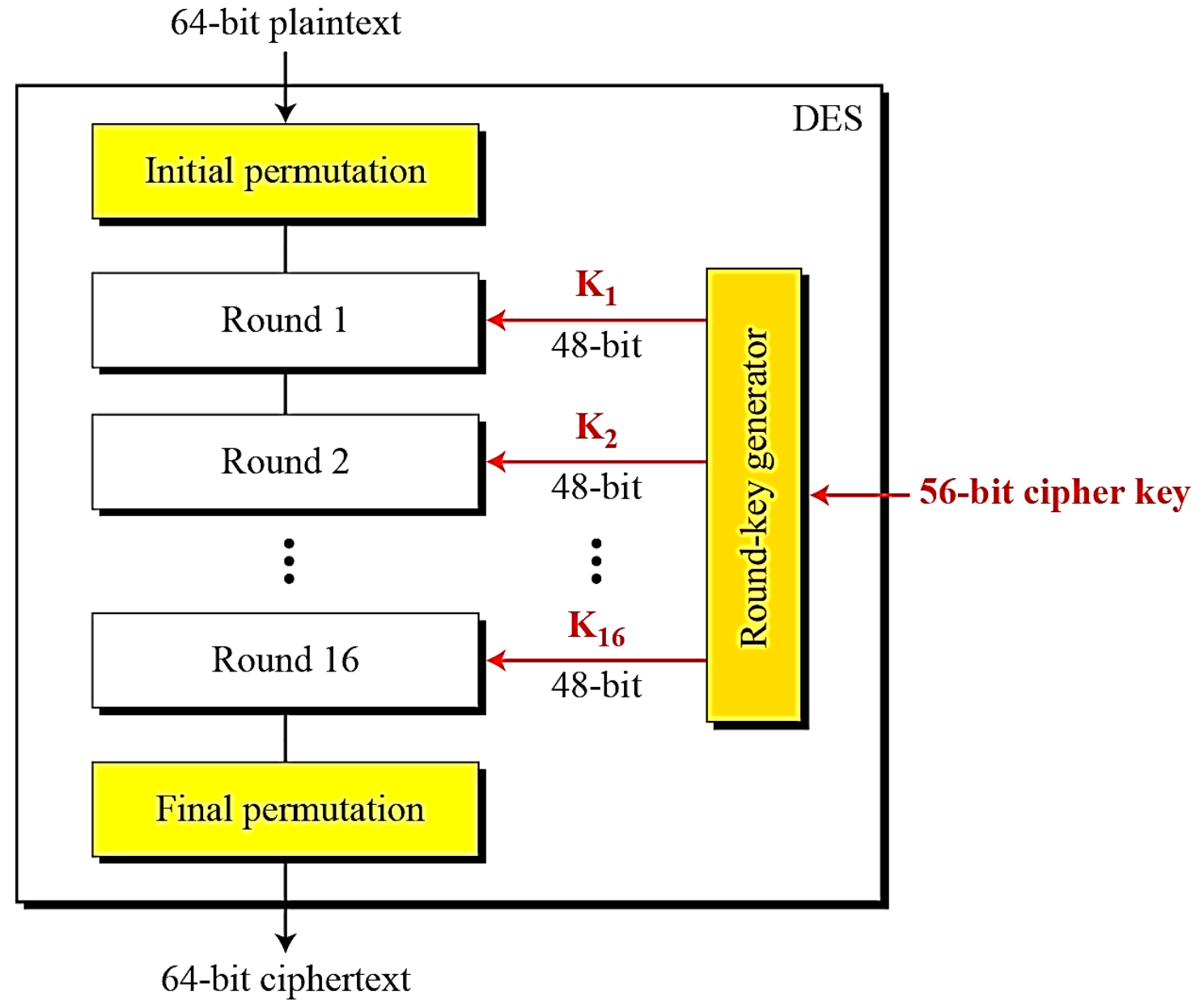
# Data Encryption Standard (DES)

---

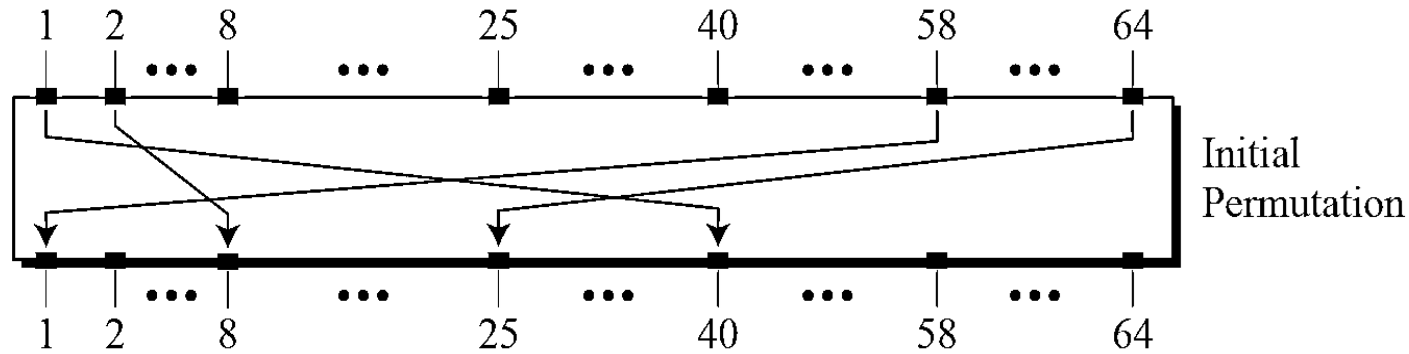
- Suitable for implementation in software on most current computers : uses only standard **arithmetic** and **logical** operations on binary data up to 64 bits long.
- Involves 16 iterations.
- Each iteration employs:
  - a substitution step,
  - a permutation step, and
  - a key transformation.



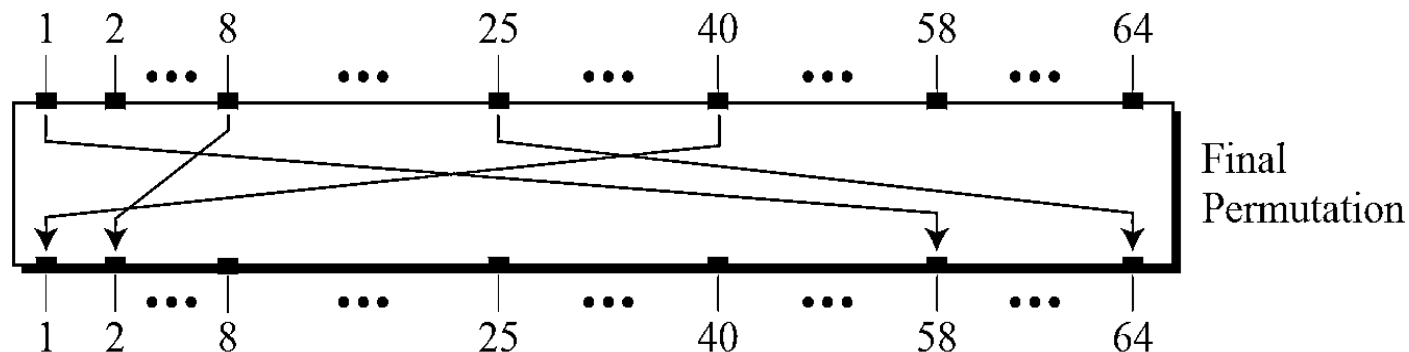
# DES Structure



# Initial and Final Permutations



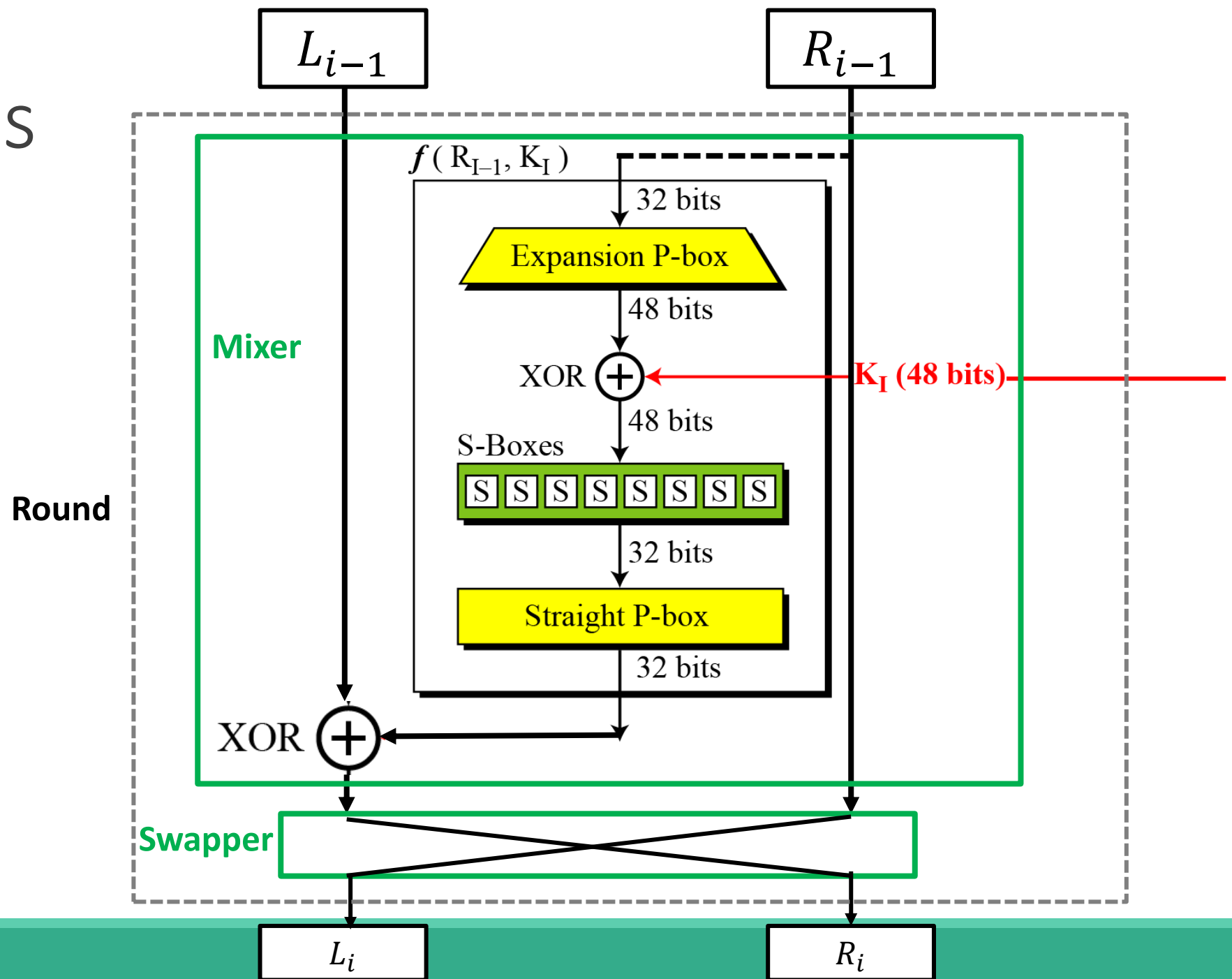
**16 Rounds**



Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Final Permutation							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

# Rounds





# Double DES

---

- How double encryption works?
- Take two keys,  $k_1$  and  $k_2$ , and perform two encryptions, one on top of the other:  $E(k_2, E(k_1, m))$ .
- Two encryptions with different 56-bit keys are equivalent in work factor to one encryption with a 57-bit key.
- Some 56-bit DES keys have been derived in just days.
- Double DES adds essentially no more security.

# Triple DES

---

- Triple DES procedure :  $C = E(k_3, E(k_2, E(k_1, m)))$ .
- This process gives a strength roughly equivalent to a 112-bit key.
- Different variation: encrypt–decrypt–encrypt.
- $C = E(k_1, D(k_2, E(k_1, m)))$ .
- Advantage?
- One algorithm can produce results for both conventional single-key DES and the more secure two-key method.
- This two-key, three-step version is rated at only about 80 bits.

# Data Encryption Standard (DES)

---

Form	Operation	Properties	Strength
<b>DES</b>	Encrypt with one key	56-bit key	Inadequate for high-security applications by today's computing capabilities
<b>Double DES</b>	Encrypt with first key; then encrypt result with second key	Two 56-bit keys	Only doubles strength of 56-bit key version
<b>Two-key triple DES</b>	Encrypt with first key, then encrypt (or decrypt) result with second key, then encrypt result with first key (E-D-E)	Two 56-bit keys	Gives strength equivalent to about 80-bit key (about 16 million times as strong as 56-bit version)
<b>Three-key triple DES</b>	Encrypt with first key, then encrypt or decrypt result with second key, then encrypt result with third key (E-E-E)	Three 56-bit keys	Gives strength equivalent to about 112-bit key about 72 quintillion ( $72 \times 10^{15}$ ) times as strong as 56-bit version

# Security of DES

---

- Brute Force
- Differential Cryptanalysis
- Linear Cryptanalysis
- MIM Attack

## Is DES insecure?

- Assume that single-key DES can be broken in one hour.
- The simple double-key version could then be broken in two hours?
- But  $2^{80}/2^{56} = 2^{24} = \text{over } 16,700,000$ .
- It would take 16 million hours, nearly 2,000 years, to defeat a two-key triple DES encryption, and considerably longer for the three-key version.

- Conspiracy Theorists view!
- In 1997, researchers using a network of over 3,500 machines in parallel were able to infer a DES key in four months' work.
- In 1998 for approximately \$200,000 U.S. researchers built a special "DES cracker" machine that could find a DES key in approximately four days, a result later improved to a few hours.

# Advanced Encryption Standard (AES)

---

## A Little Perspective

- Search for AES :1997
- Requirements: Open algorithm, 128-bit blocks, 3 key sizes:128, 192, 256 bits
- Proposals submitted: 21
- Accepted: 15
- Finalists: MARS, RC6, Rijndael, Serpent, Twofish
- Rijndael by John Daemen and Vincent Rijmen selected as AES in October 2000.
- AES published as FIPS 197 in December 2001.

# Evaluation Criteria

---

- Security
  - Resistance to cryptanalysis, soundness of math, randomness of output.
- Cost
  - Computational efficiency (speed)
  - Memory requirements
- Algorithm / Implementation Characteristics
  - Flexibility, hardware and software suitability, algorithm simplicity

# AES Overview

---

<b>Cipher Type</b>	Non-Feistel, Block
<b>Block Length</b>	128 bits
<b>Key Length</b>	128, 192 or 256 bits
<b>Number of Rounds</b>	10, 12 or 14
<b>Size of subkey( round key)</b>	128 bits
<b>Number of round Keys</b>	$N_r + 1$
<b>Data Units</b>	bits, bytes, words, blocks and <b>States</b>
<b>Operations Used</b>	substitution, transposition, the shift, exclusive OR, and addition operations.

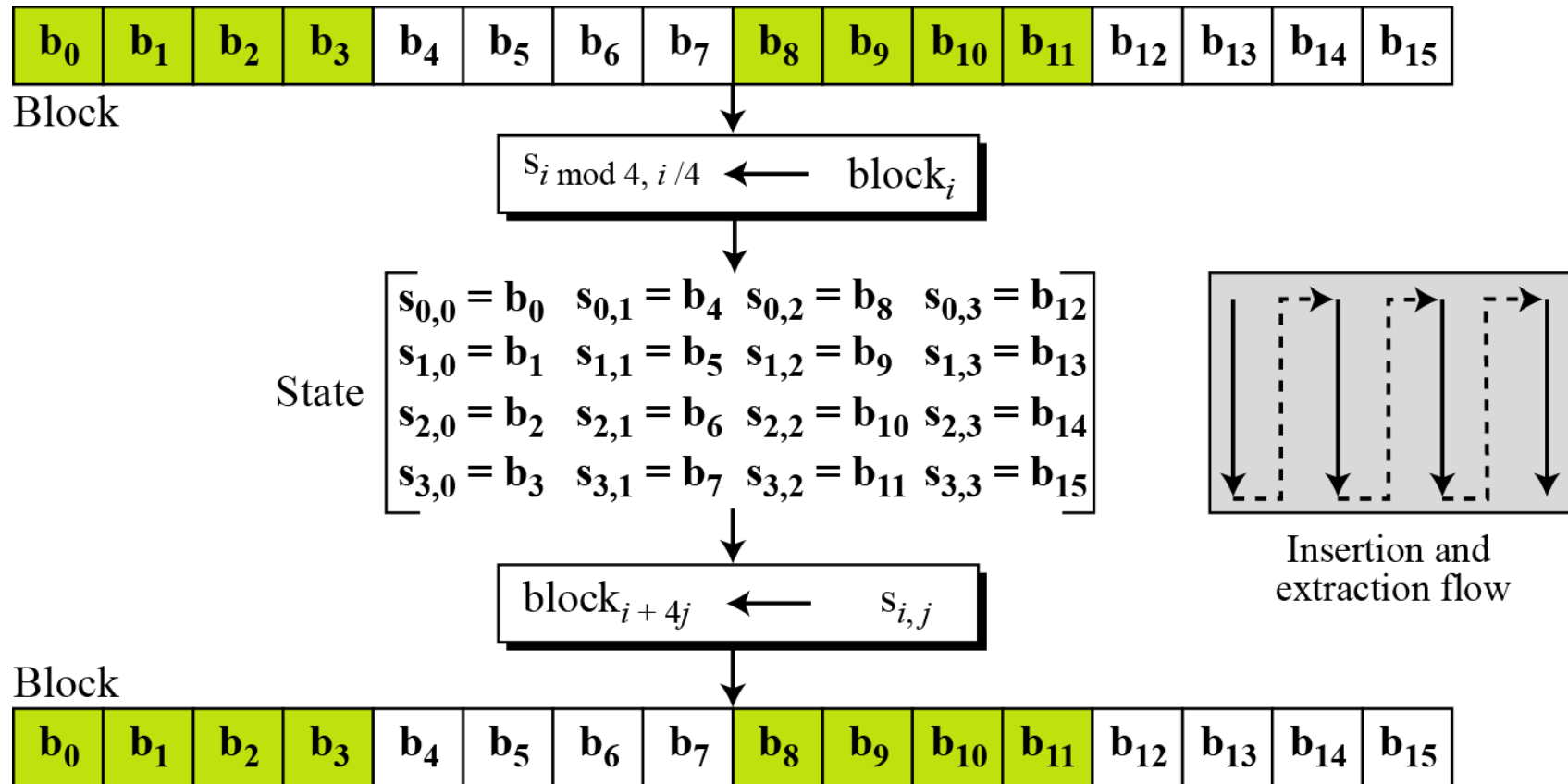
# AES: Components and Methods

---

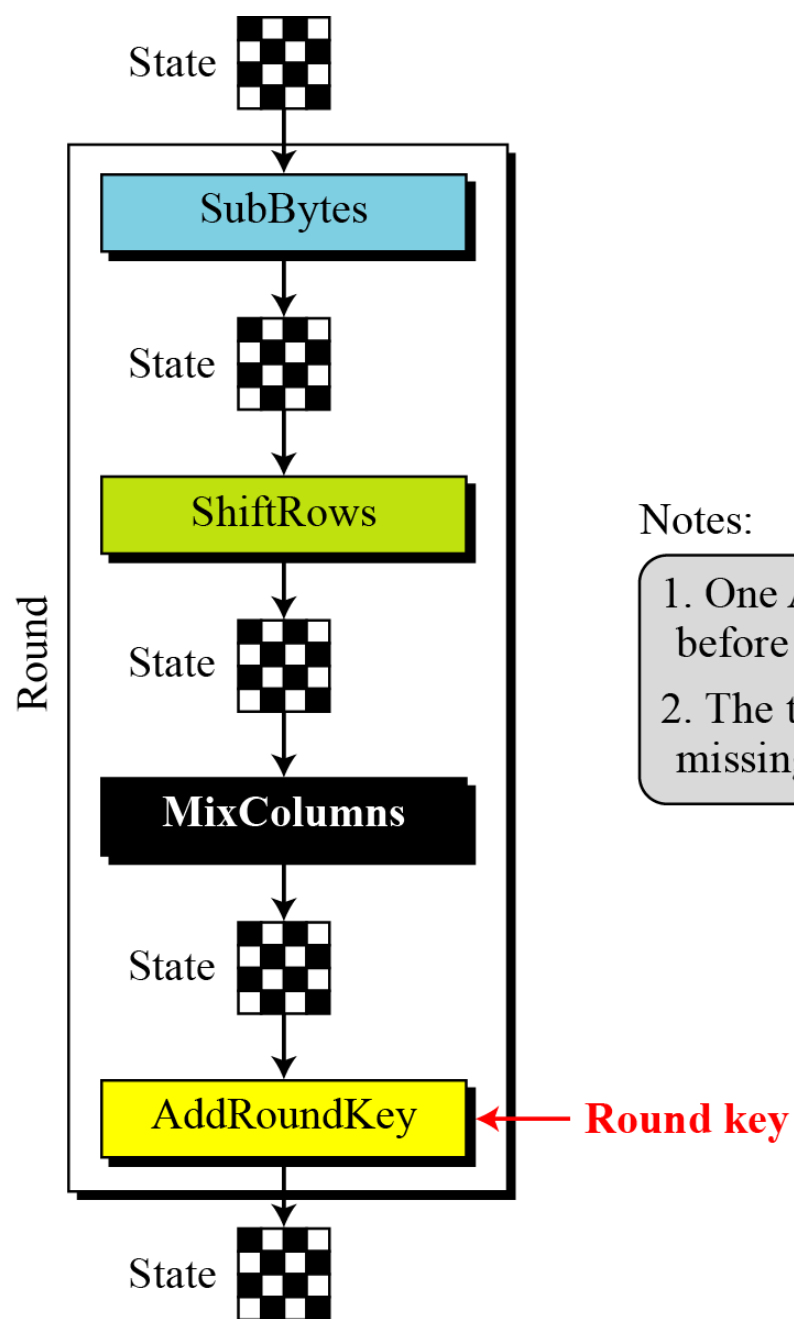
- Convert bytes to state arrays.
- Transformations (and their inverses)
  - SubBytes → Substitution
  - ShiftRows → Permutation
  - MixColumns
  - AddRoundKey
- Key Expansion



# Bytes to State



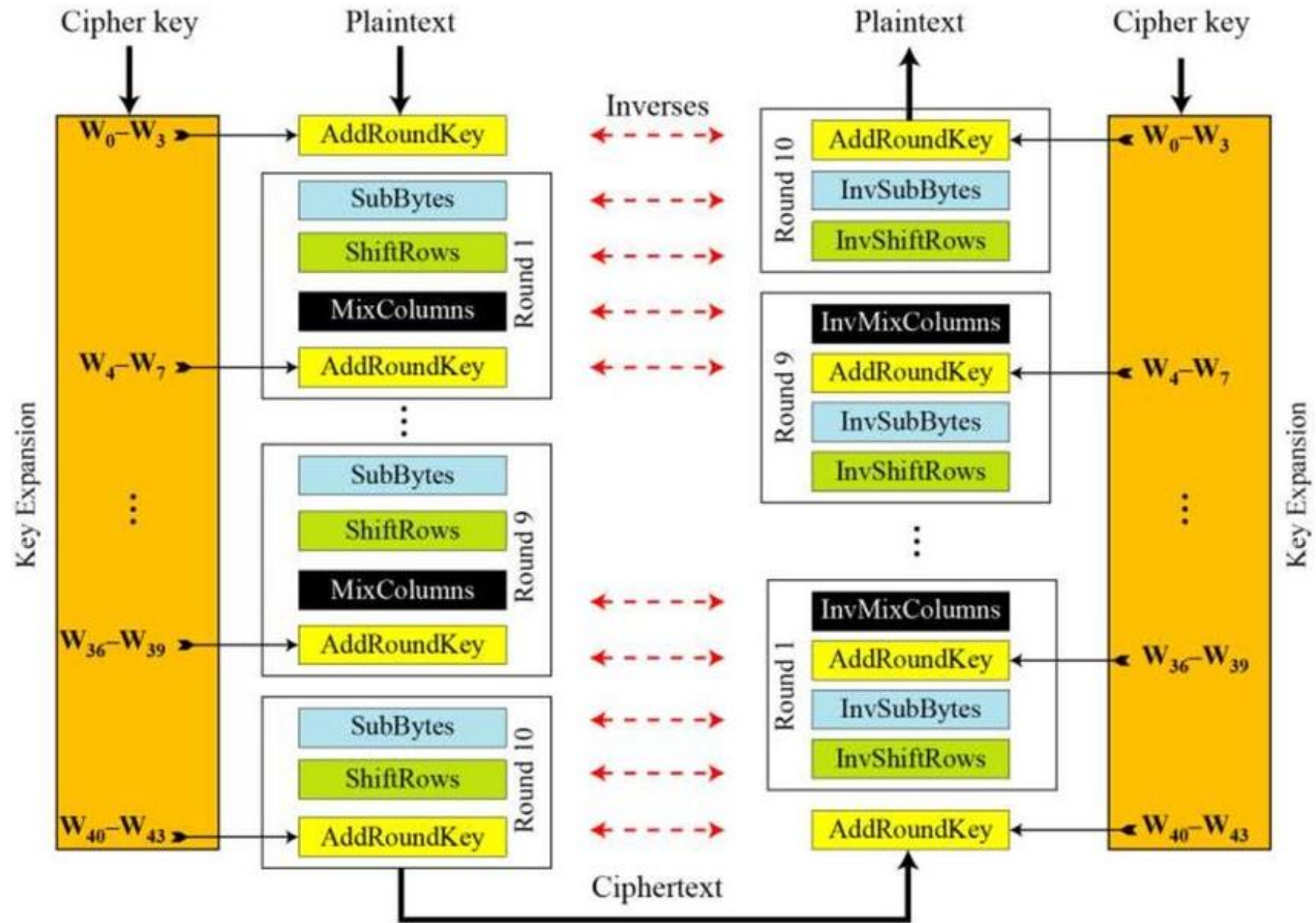
# AES: A Round



Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

# The AES Cipher : Original Design



# AES versus DES

	DES	AES
Date designed	1976	1999
Block size	64 bits	128 bits
Key length	56 bits (effective length); up to 112 bits with multiple keys	128, 192, 256 (and possibly more) bits
Operations	16 rounds	10, 12, 14 (depending on key length); can be increased
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but open public comments and criticisms invited
Source	IBM, enhanced by NSA	Independent Dutch cryptographers

# Public Key Cryptography (PKC)

---

- Need for Public Key Cryptography?
- Key Distribution. With a conventional symmetric key system, each pair of users needs a separate key. An  $n$ -user system requires  $n * (n - 1)/2$  keys.
- PKC : Invented by Whitfield Diffie and Martin Hellman in 1976.
- In a public key or asymmetric encryption system, each user has two keys: a public key and a private key. (One of the keys does not have to be kept secret.)
- Basis: Allow the key to be divulged but keep the decryption technique secret. How is it accomplished?

# PKC : Characteristics

---

- Each user has two keys: a public key and a private key.
- Each key does only encryption or decryption, but not both.
- The keys operate as inverses, meaning that one key undoes the encryption provided by the other key.

$$P = D(k_{\text{PRIV}}, E(k_{\text{PUB}}, P))$$

- For some public key encryption algorithms,  $P = D(k_{\text{PUB}}, E(k_{\text{PRIV}}, P))$ .
- Any deductions?
- Public and private keys can be applied in either order.
- Decryption function  $D$  can be applied to any argument so that we can **decrypt before we encrypt**.

# The Rivest–Shamir–Adelman (RSA) Algorithm

---

- RSA cryptosystem is a public key system.
- Keys used in RSA:  $d$  and  $e$ , for decryption and encryption.
- Either can be chosen as the public key (keys are interchangeable).

$$C = \text{RSA}(P, e).$$

$$P = \text{RSA}(\text{RSA}(P, e), d) = \text{RSA}(\text{RSA}(P, d), e)$$

# RSA Properties

---

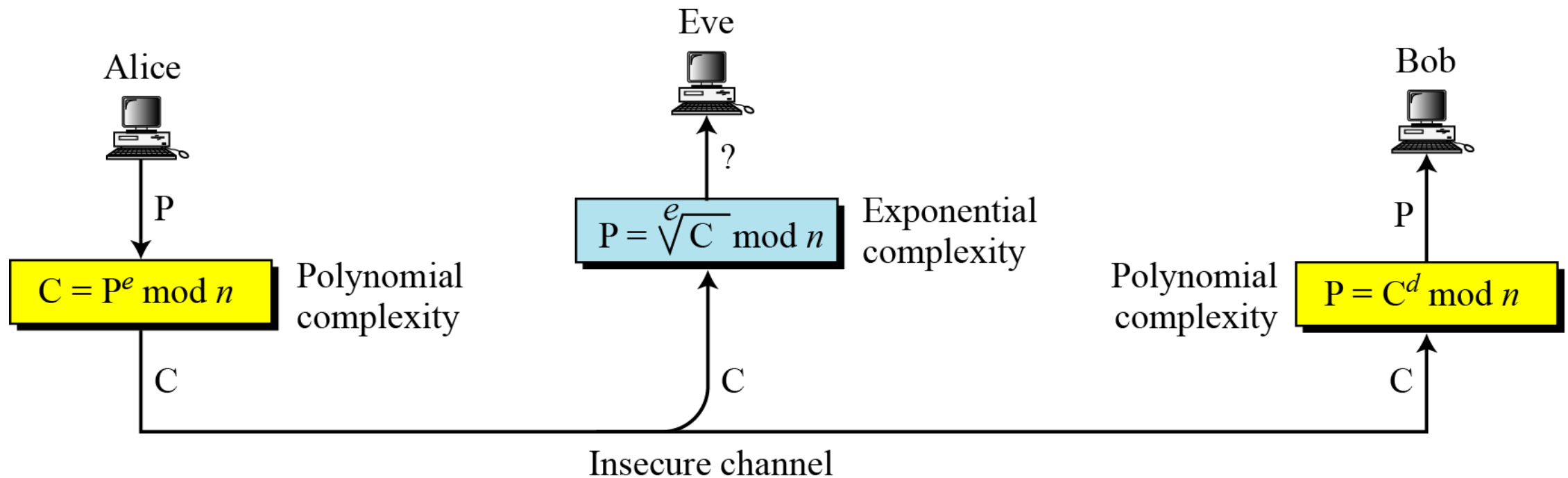
- Keys are long. 256 bits is considered the minimum usable length.
- Slower than DES and AES. Why?
  - Encryption in RSA is done by exponentiation. For DES and AES uses basic operations like substitution and transposition.
  - the time to encrypt increases exponentially as the exponent (key) grows longer.
- The encryption algorithm is based on the underlying problem of factoring large numbers in a finite set called a field.
- RSA encrypts blocks of various sizes.
- Generally reserved for limited uses at which it excels. Applications: Digital Signatures



# RSA Cryptosystem

Key calculation in  
 $\mathbf{G} = \langle \mathbb{Z}_{\phi(n)}^*, \times \rangle$

Select  $p, q$   
 $n = p \times q$   
Select  $e$  and  $d$



# Symmetric vs Asymmetric Cryptography

---

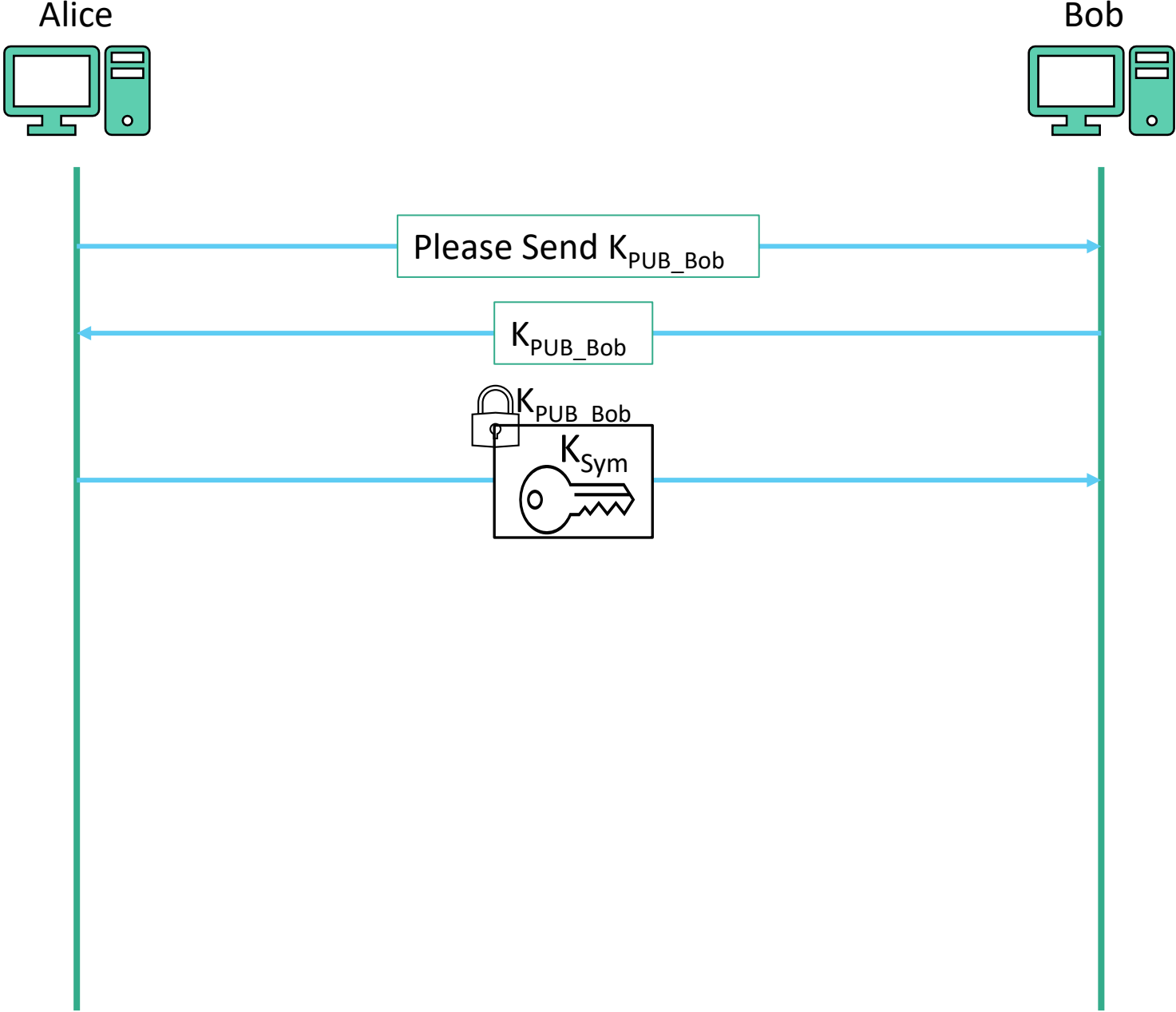
	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	Depends on the algorithm; 56–112 (DES), 128–256 (AES)	Unlimited; typically no less than 256; 1000 to 2000 currently considered desirable for most uses
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

# Public Key Cryptography to Exchange Secret Keys

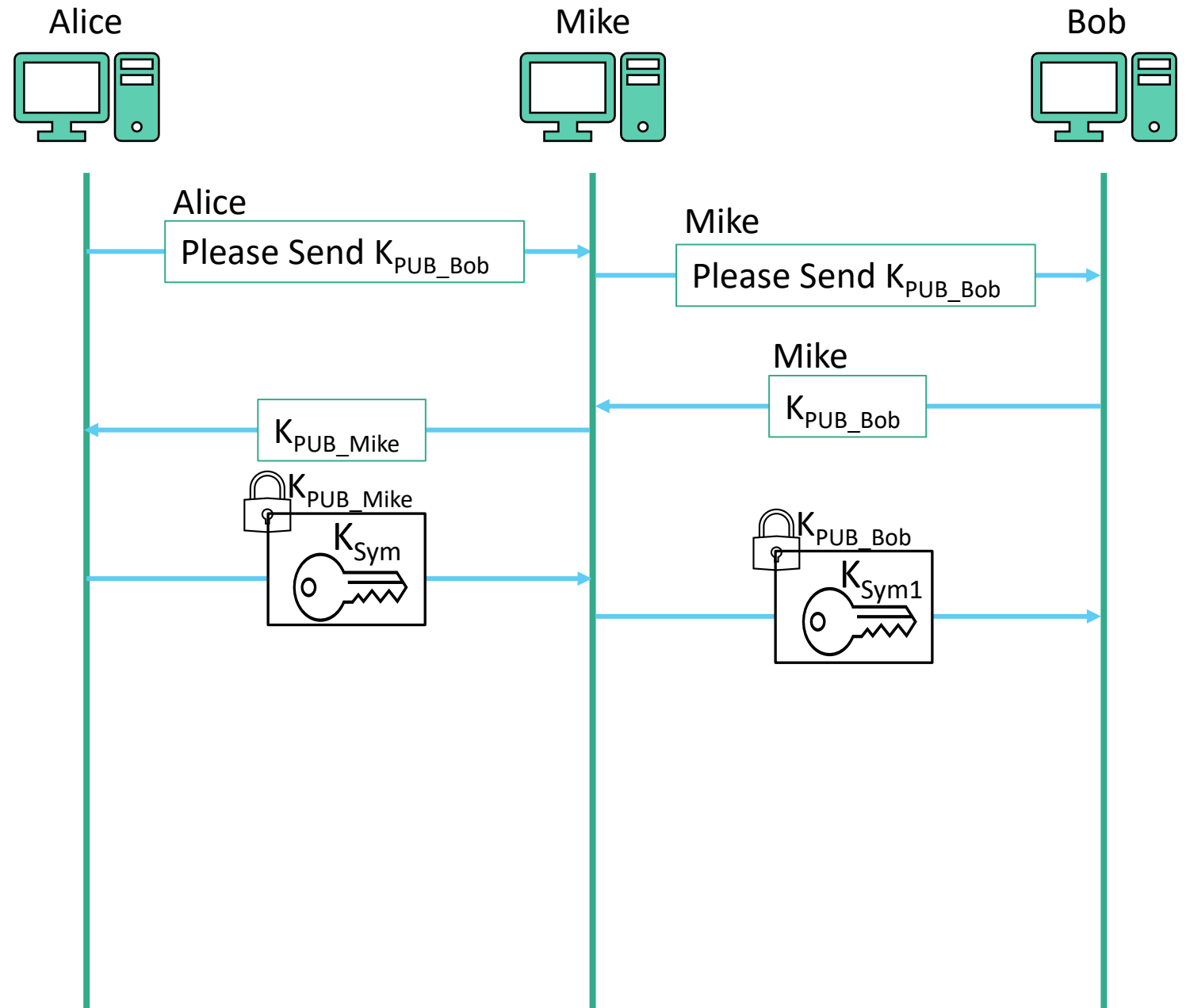
---

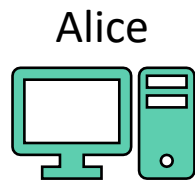
- While transacting with unknown parties, an encrypted means to exchange keys is needed to establish an encrypted session.
- Since asymmetric keys come in pairs, one half of the pair can be exposed without compromising the other half.
- Two key exchange protocols:
  1. Simple Key Exchange Protocol
  2. Revised Key Exchange Protocol

# Simple Key Exchange Protocol

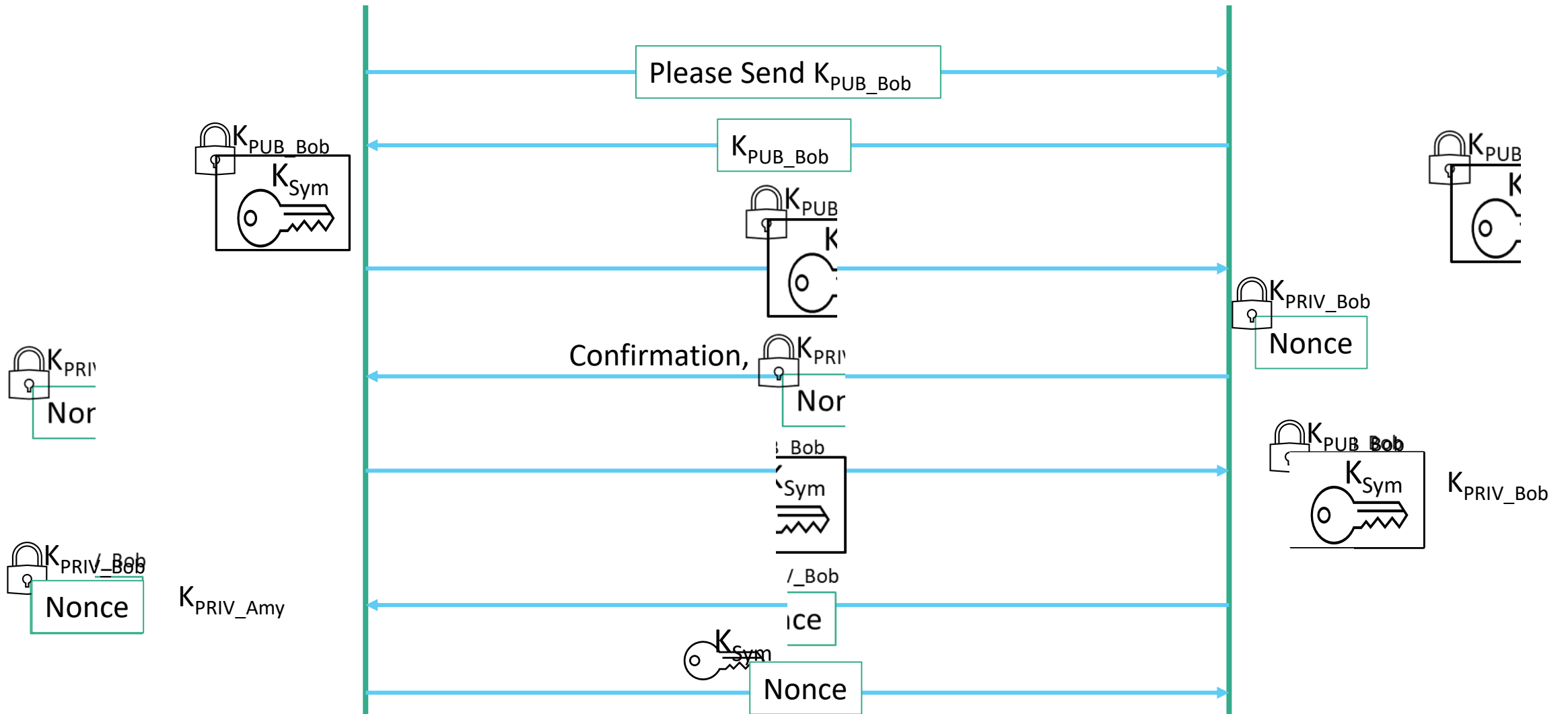
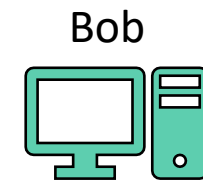


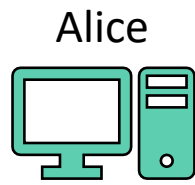
# Man-in-the-Middle Attack



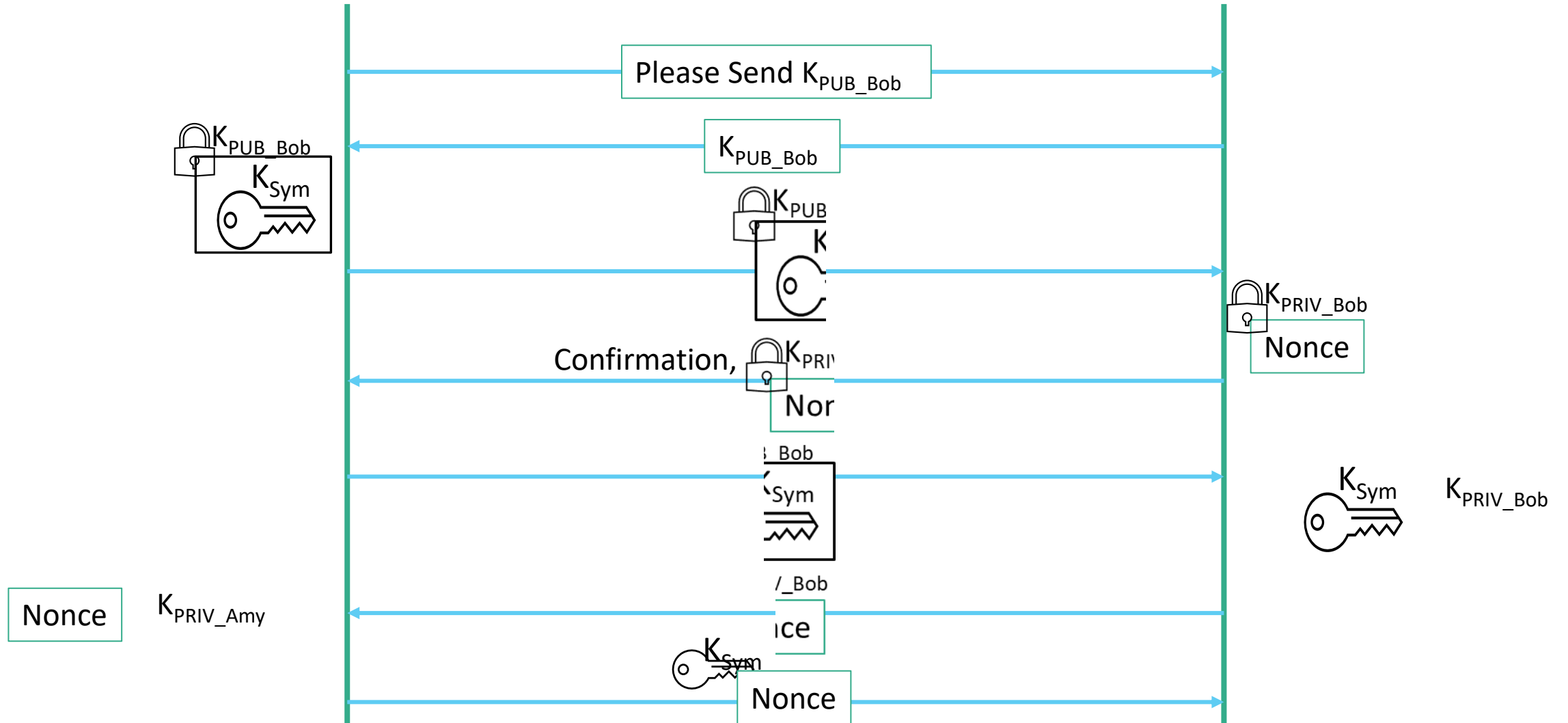
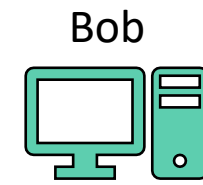


# Revised Key Exchange Protocol





# Revised Key Exchange Protocol



# Security of Revised Key Exchange Protocol

---

- Mike can certainly intercept both public keys in steps 1 and 2 and substitute his own.
- Mike cannot take half the result, decrypt it using his private key, and re-encrypt it under Bob's key.
- Bits cannot be decrypted one by one and reassembled.
- The problem of the person in the middle can be solved in another way:

$$E(k_{\text{PUB-B}}, E(k_{\text{PRIV-A}}, K))$$

- Only Bob can remove the encryption applied with  $k_{\text{PUB-B}}$ , using  $k_{\text{PRIV-B}}$ .
- Bob knows that only Alice could have applied  $k_{\text{PRIV-A}}$  that Bob removes with  $k_{\text{PUB-A}}$ .



# Error Detecting Codes: Concepts

---

- Communications are prone to errors in transmission.
- The need now: Some way to determine that the transmission is complete and intact.
- Error detecting codes: hash codes, message digests, checksums, integrity checks, error detection and correction codes, and redundancy tests.
- Basic purpose: demonstrate that a block of **data has been modified**.
- A **message digest** will (sometimes) signal that content has changed, but it is **less solid at demonstrating no modification**, even though that is what we really want. We

# Error Detecting Codes: Concepts

---

- Three cases can arise with the code value:

1. Value changed
2. No value included
3. No change



Problematic Situation.

Two major uses of cryptographic checksums:

1. Code-tamper protection
2. Message integrity protection in transit

- **Collision:** Two inputs that produce the same output. Why does it happen?

- Message digests are many-to-one functions.

- Hash or checksum or **message digest**: shields a file so that change is detected.

- **Cryptographic Checksum:** A cryptographic function that produces a checksum. It is a **digest function using a cryptographic key** that is presumably known only to the originator and the proper recipient of the data.

# Digital Signature

---

- A **digital signature** is a protocol that produces the same effect as a real signature.
- It is a mark that only the sender can make but that other people can easily recognize as belonging to the sender.
- Properties of Secure Paper-Based Signatures: (Example: Cheques)
  - A cheque is a **tangible** object authorizing a financial transaction.
  - The signature on the check confirms **authenticity**.
  - In the case of an alleged forgery, a third party can be called in **to judge authenticity**.
  - Once a cheque is cashed, it is canceled so that it **cannot be reused**.
  - The paper cheque is **not alterable**.

# Digital Signature: Requirements

---

Four **objectives** of a digital signature:

- Unforgeable

If person S signs message M with signature  $\text{Sig}(S,M)$ , no one else can produce the pair  $[M, \text{Sig}(S,M)]$ .

- Authentic

R can check that the signature is really from S.

To address **authenticity**, we need a structure that binds a user's identity and public key in a **trustworthy** way.

- Unalterable

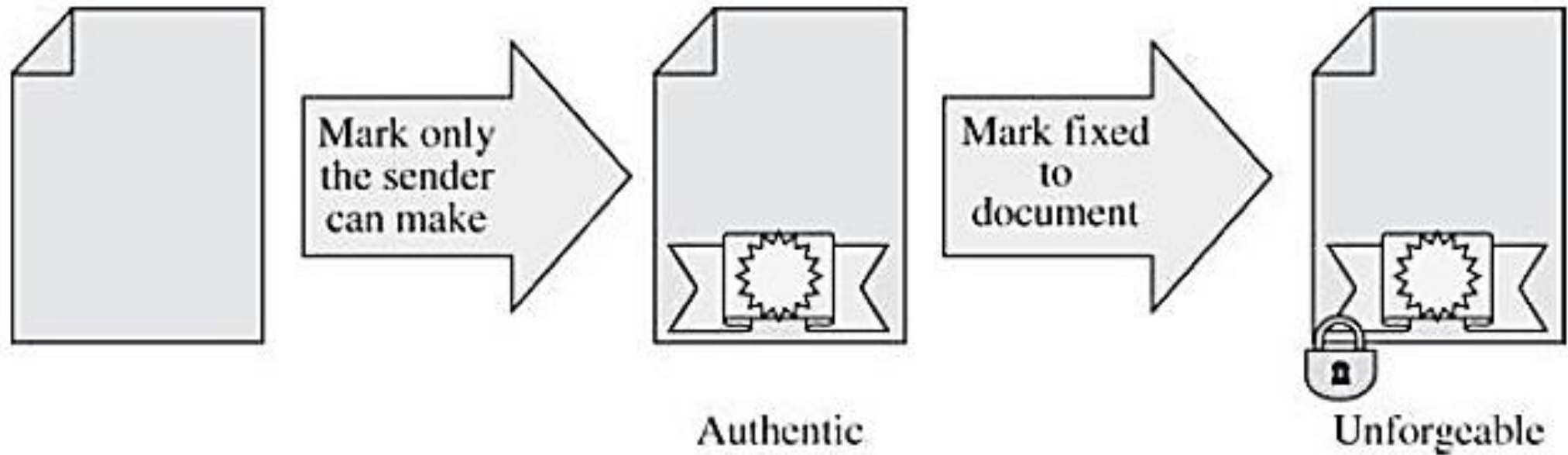
After being transmitted, M cannot be changed by S, R, or an interceptor.

- Not reusable

A previous message presented again will be instantly detected by R.

# Digital Signature: Requirements

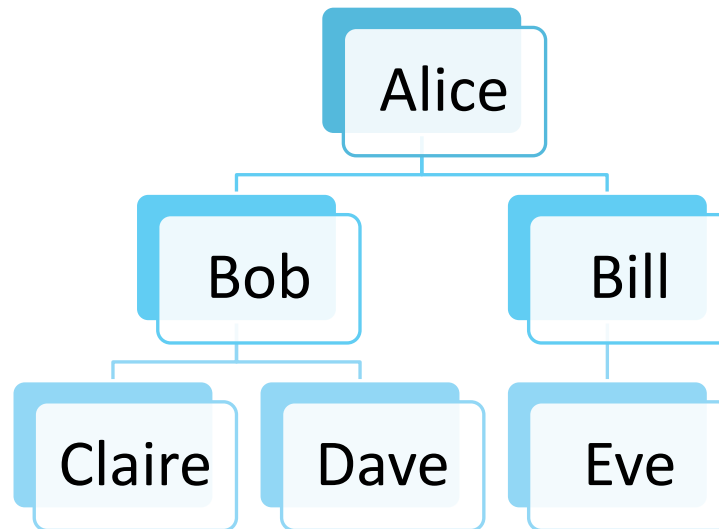
---



# Certificates

---

- A public key and user's identity are bound together in a **certificate**, which is then signed by a certificate authority.
- A **certificate authority** certifies the accuracy of the binding.
- Examples.
- Trust.



# Digital Signature Components

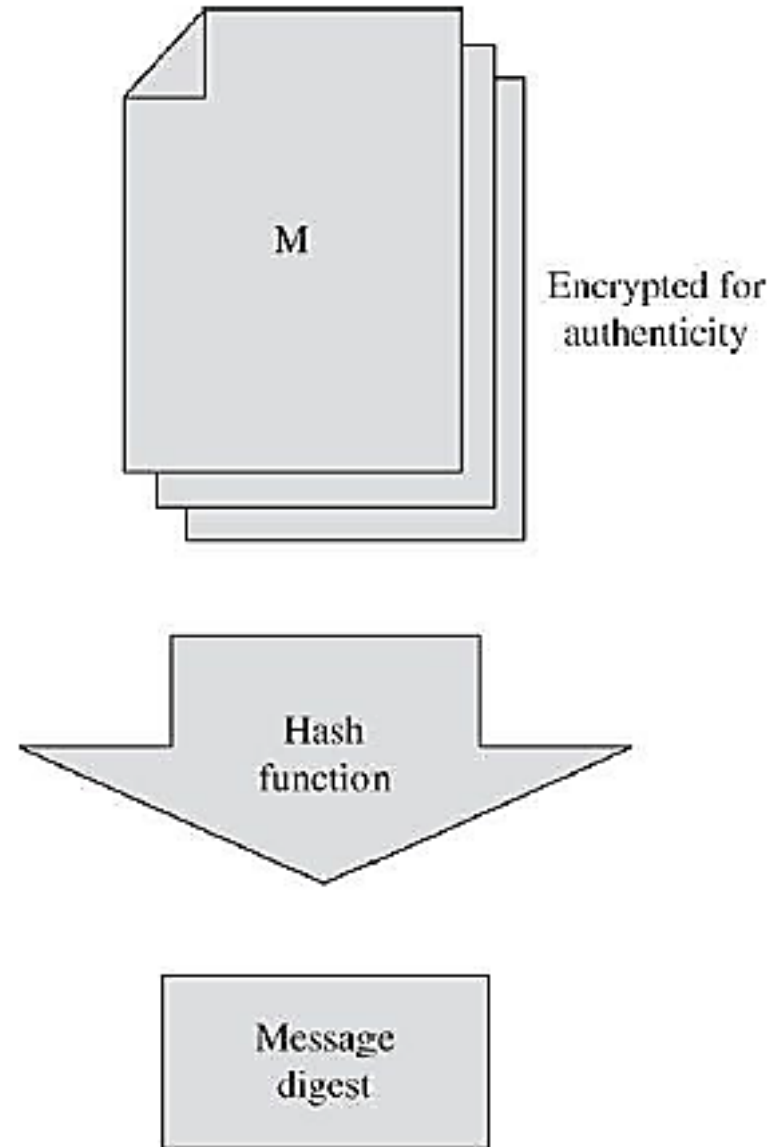
---

Components of a digital signature:

- A file.
- Demonstration that the file has not been altered.
- Indication of who applied the signature.
- Validation that the signature is authentic, that is, that it belongs to the signer.
- Connection of the signature to the file.

# DS: Hash Code to Detect Changes

Use a secure hash code of the file to compute a message digest and include that hash code in the signature.

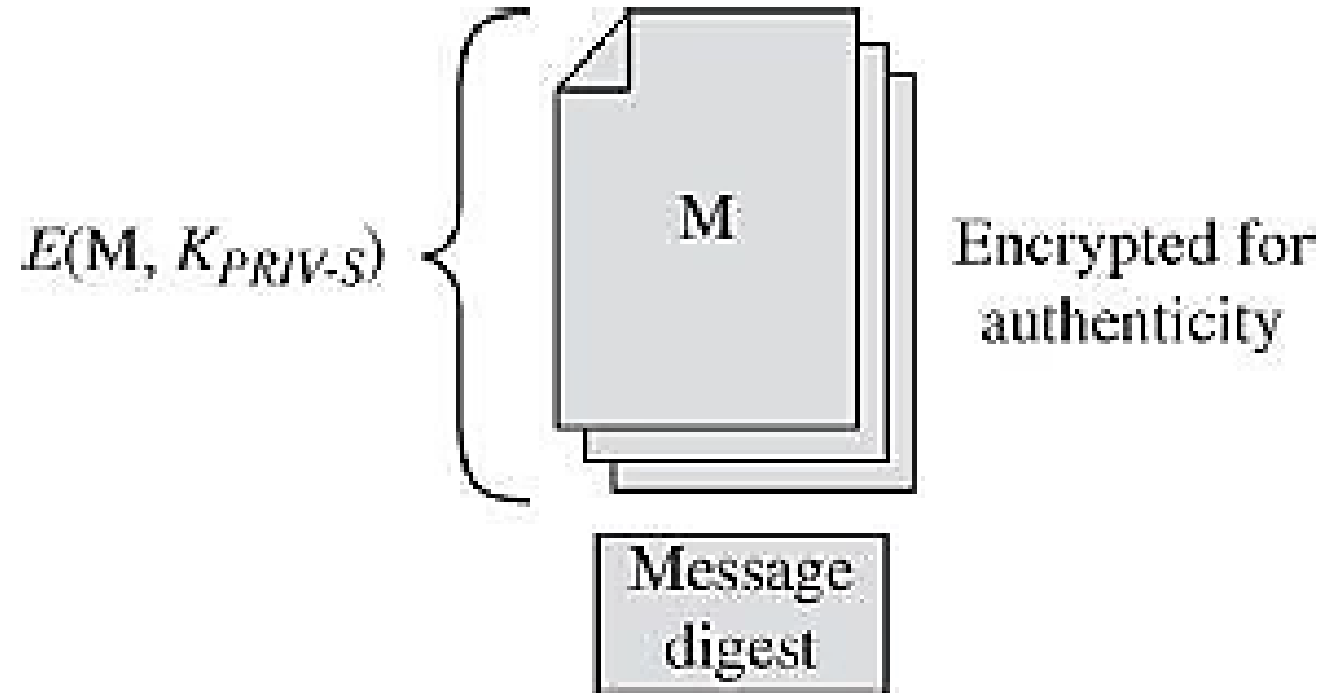




# DS: Encryption to Show Authenticity

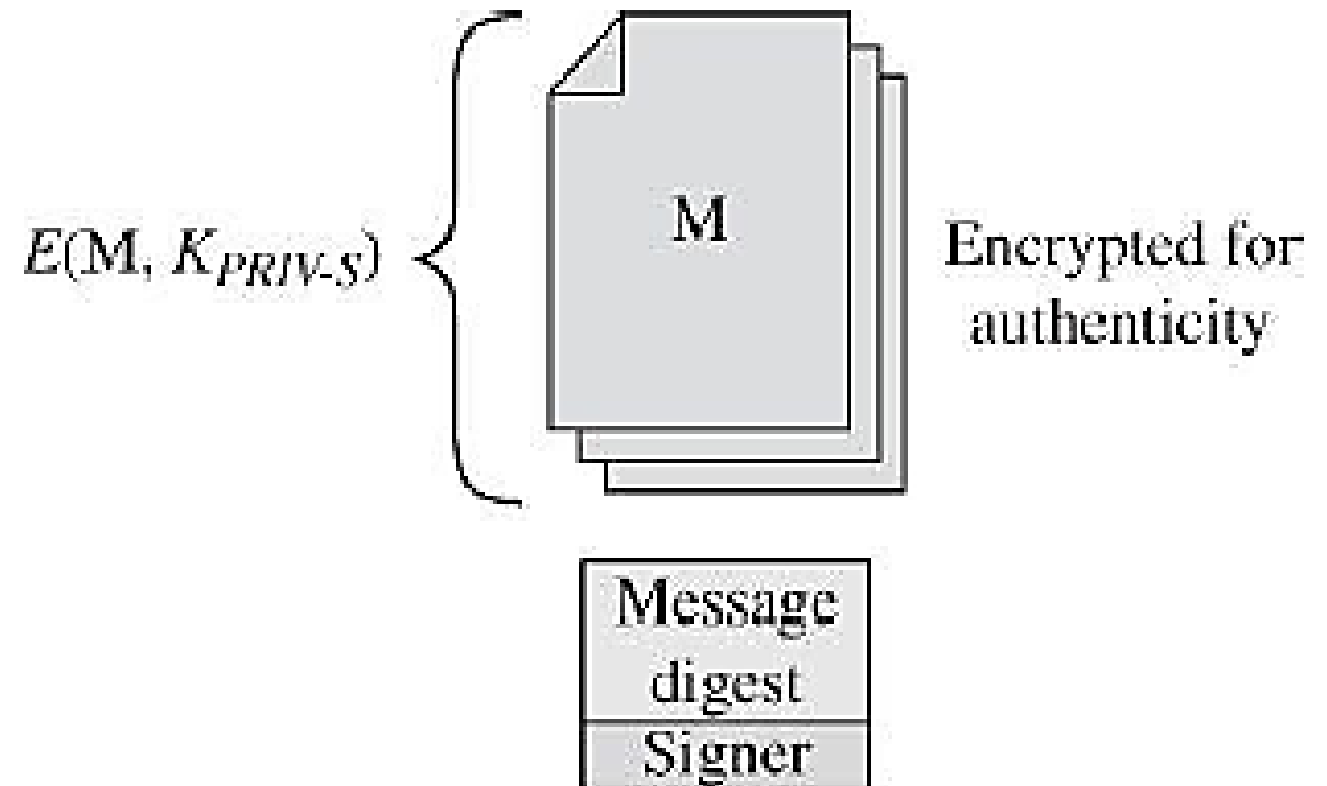
Apply the signer's private encryption key to encrypt the message digest.

Because only the signer knows that key, the signer is the only one who could have applied it.



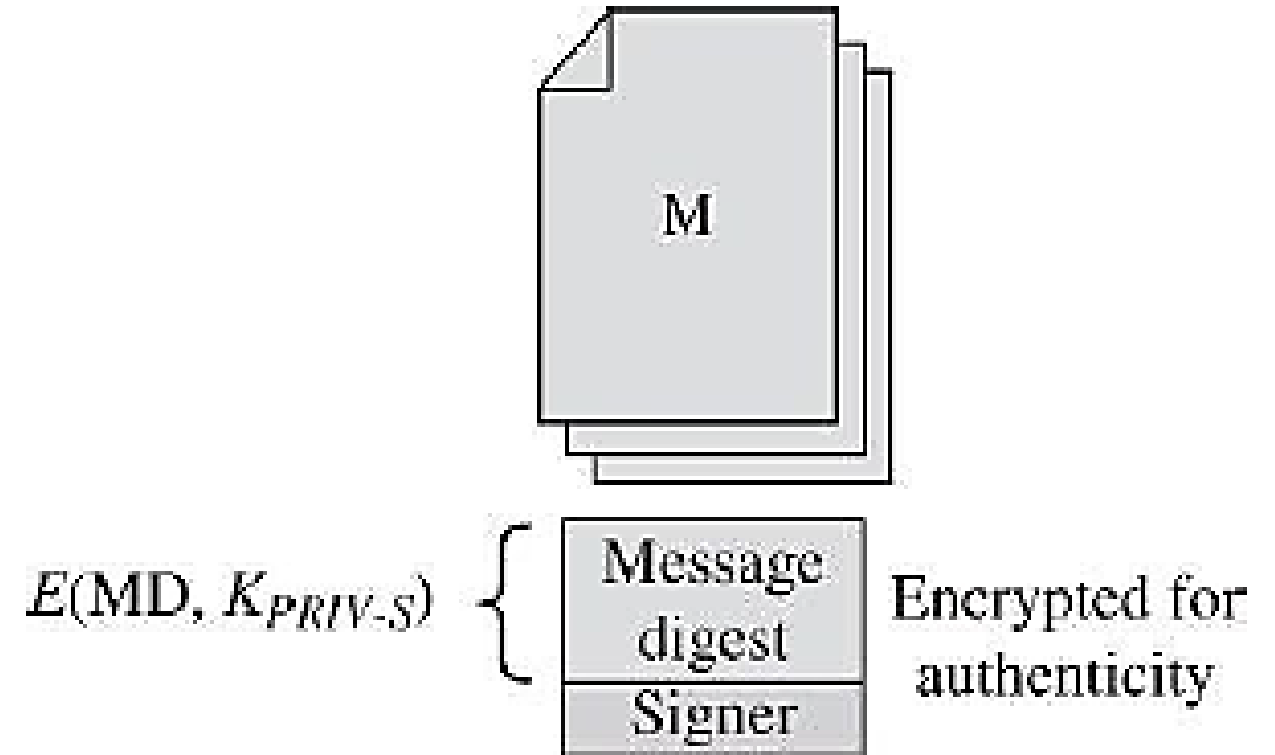
## DS: Indication of Signer

The signer's identity has to be outside the encryption because if it were inside, the identity could not be extracted.

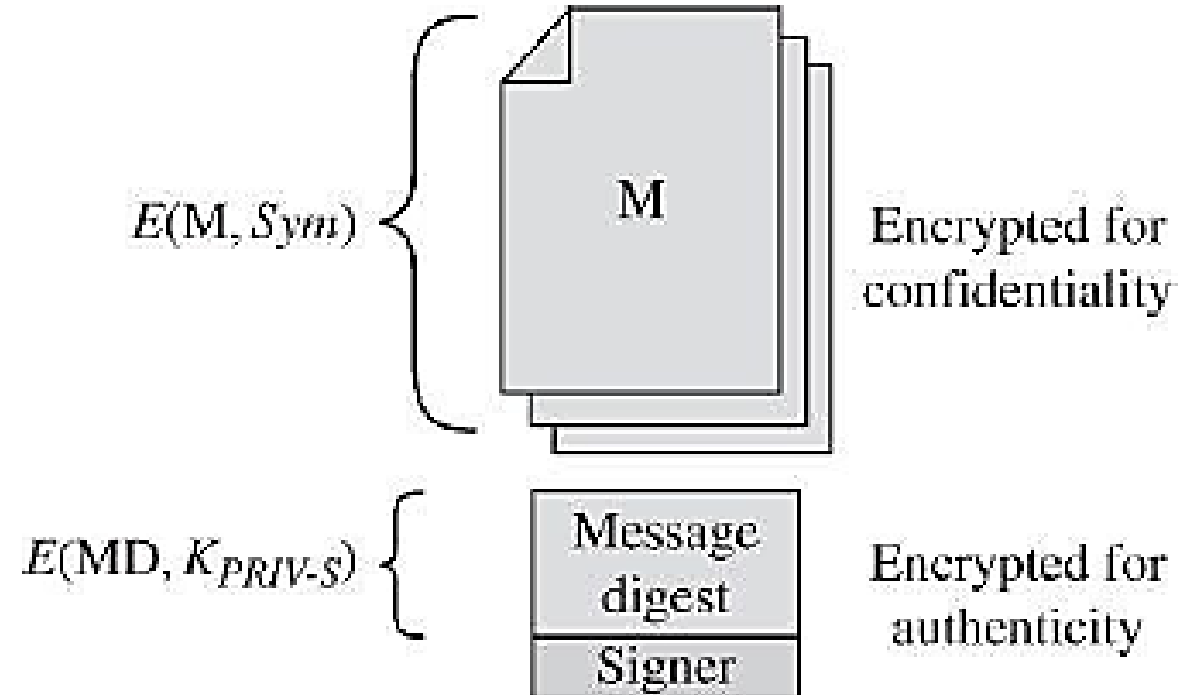


# DS: Asymmetric Encryption Covering the Hash Value

Only Hash value. Is it  
enough?



DS:  
Digitally Signed  
Object Protected  
for Both Integrity  
and  
Confidentiality



# Summary

Prepare a Map.

Tool	Uses
Secret key (symmetric) encryption	Protecting confidentiality and integrity of data at rest or in transit
Public key (asymmetric) encryption	Exchanging (symmetric) encryption keys Signing data to show authenticity and proof of origin
Error detection codes	Detect changes in data
Hash codes and functions (forms of error detection codes)	Detect changes in data
Cryptographic hash functions	Detect changes in data, using a function that only the data owner can compute (so an outsider cannot change both data and the hash code result to conceal the fact of the change)
Error correction codes	Detect and repair errors in data
Digital signatures	Attest to the authenticity of data
Digital certificates	Allow parties to exchange cryptographic keys with confidence of the identities of both parties

# Book

---

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 2.