

Data Security and Privacy

DSE 3258

**L1 – L2 -Introduction
to Security: OSI
Security Architecture
and Security attacks,
Services and
mechanisms, network
security models**

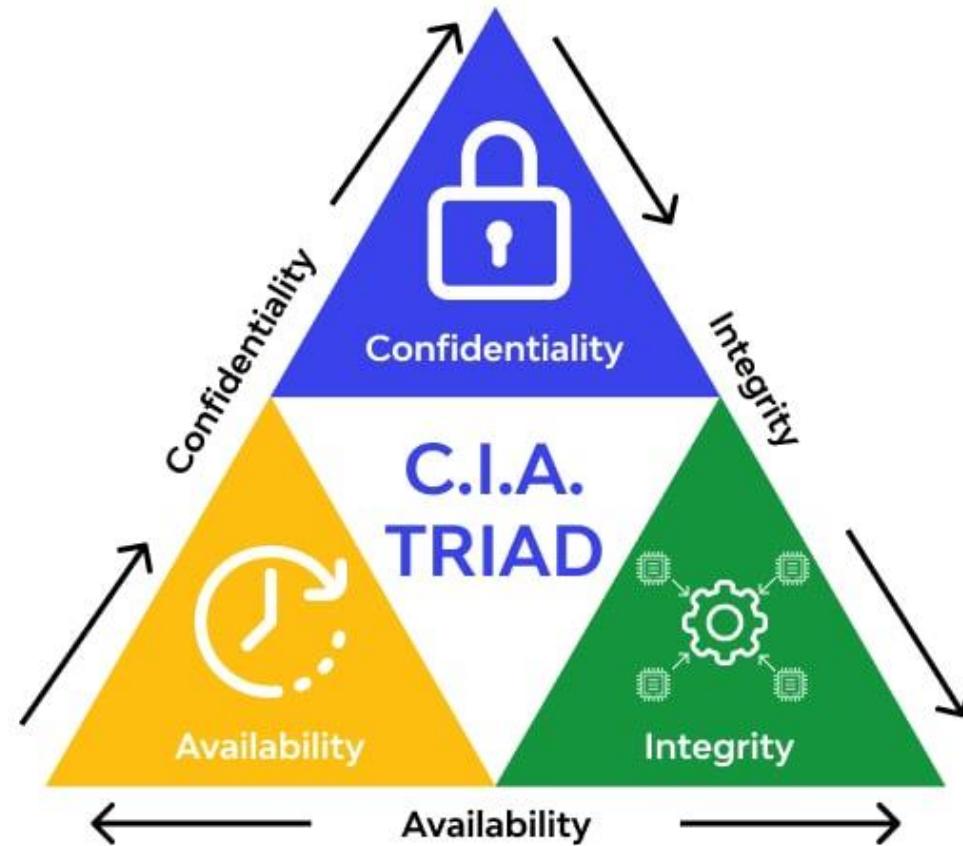
Introduction

- **What is computer security?**

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (including hardware, software, firmware, information/data, and telecommunications).

NIST Computer Security Handbook [NIST95]

Three key objectives of Computer Security



Three key objectives of Computer Security

Confidentiality:

- ✓ **Data confidentiality:** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
- ✓ **Privacy:** Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed

Integrity:

- ✓ **Data integrity:** Assures that information (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner.
- ✓ **System integrity:** Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Availability:

Assures that systems work promptly and service is not denied to authorized users.

Additional Objectives

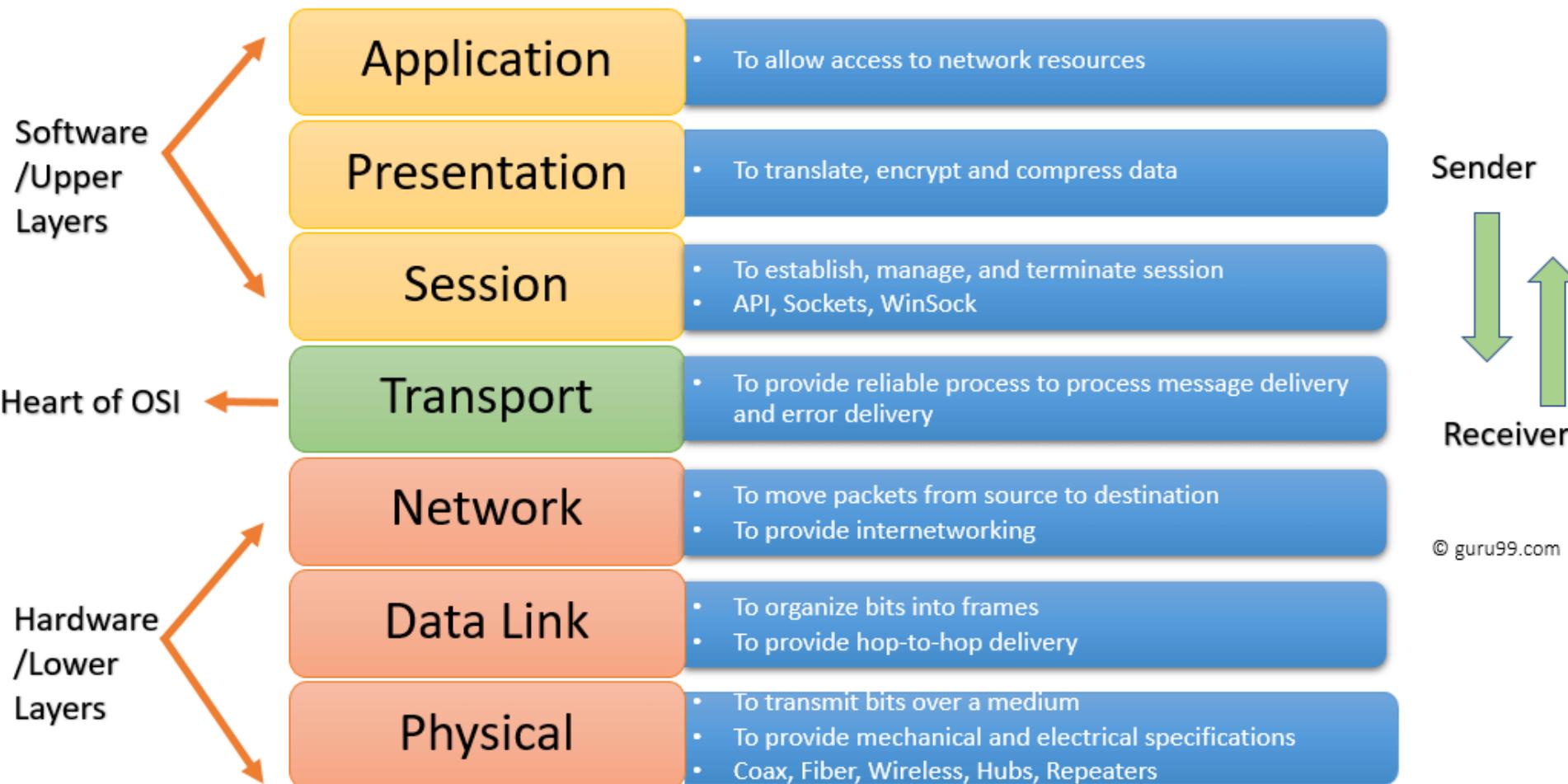
Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.

An *authentication* is a process that ensures and confirms a user's identity or role that someone has.

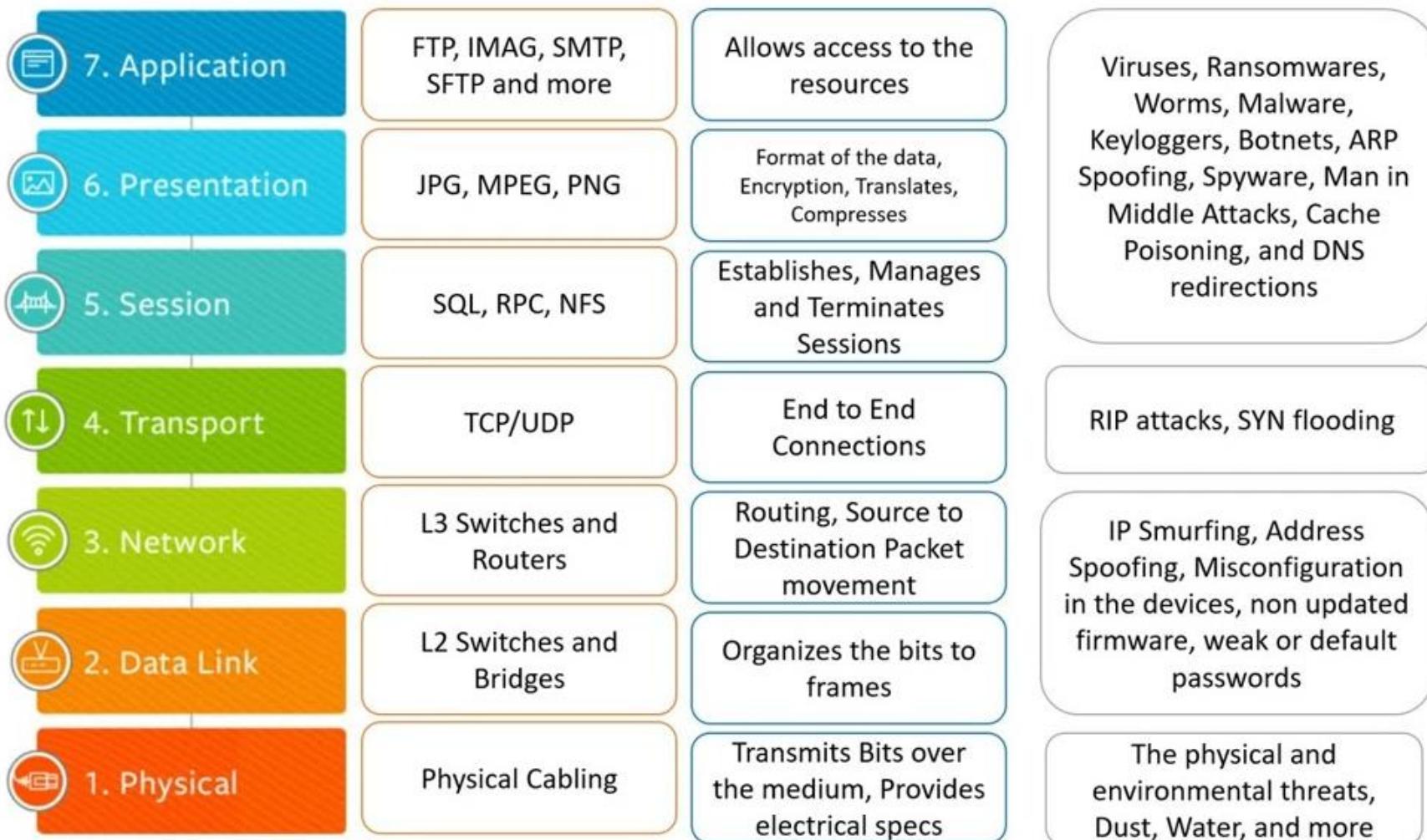
Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention etc.

It means that every individual who works with an information system should have specific responsibilities for information assurance. The tasks for which a individual is responsible are part of the overall information security plan and can be readily measurable by a person who has managerial responsibility for information assurance.

The OSI Security Architecture



The OSI Security Architecture

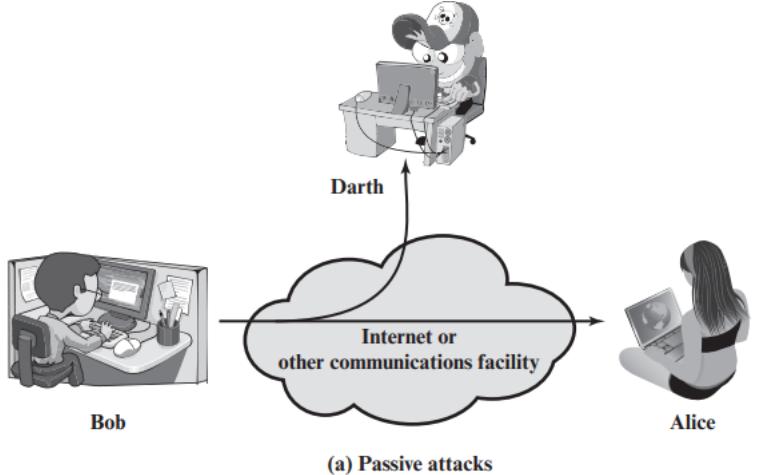


The OSI Security Architecture

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.
- **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

Security attack

- **Passive Attack**



- ✓ Passive attack attempts to learn or make use of information from the system but does not affect system resources.
- ✓ Two types : *Release of message contents* and *Traffic analysis*.

Release of message contents :

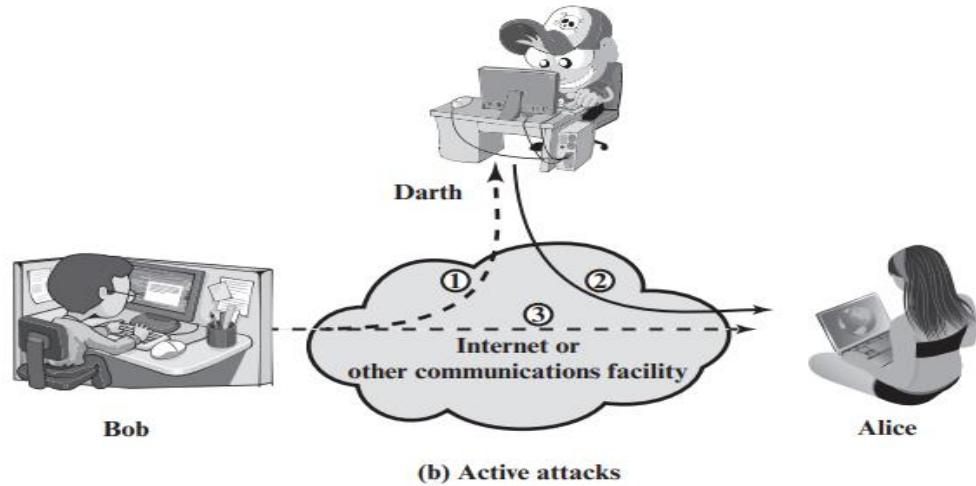
- Happens when confidential user data are released publicly over the network.
- Attacker is monitoring unprotected communication mediums and intercepting them. such as unencrypted data, emails, telephone calls, etc., which results in lost data confidentiality.

Traffic analysis:

- Is the process of intercepting and examining messages in order to deduce information from patterns in communication.
- We need a way of masking (encryption) information, so that the attacker even if captured the message could not extract any information from the message.
- If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place

Security attack

- **Active Attack**

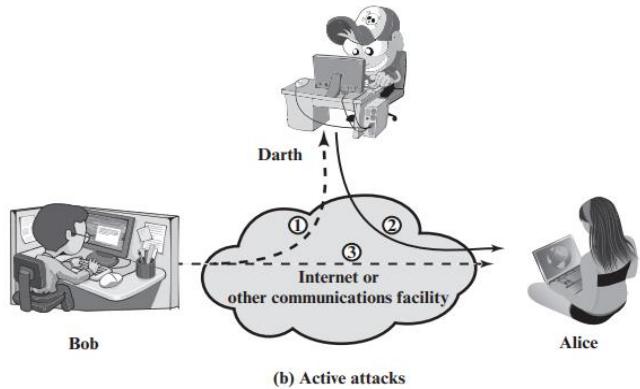


- ✓ An active attack attempts to alter system resources or affect their operation.
- ✓ Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: **masquerade, replay, modification of messages, and denial of service.**

Masquerade

- Takes place when one entity pretends to be a different entity (path 2 of Figure is active).
- A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Security attack



Replay:

Involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (paths 1, 2, and 3 active).

Modification of messages

Means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (paths 1 and 2 active).

Denial of service

Prevents or inhibits the normal use or management of communications facilities (path 3 active). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

Active attack	Passive attack
In active attacks, the attacker intercepts the connection and efforts to modify the message's content.	In passive attacks, the attacker observes the messages, then copy and save them and can use it for malicious purposes.
In an active attack, the attacker modifies the actual information.	In passive attacks, information remains unchanged.
In active attacks, the victim gets notified about the attack.	Unlike active attacks, in passive attacks, victims do not get informed about the attack.
The damage done with active attacks can be harmful to the system and its resources.	The passive attacks do not harm the system.
In active attacks, the system resources can be changed.	In passive attacks, the system resources remain unchanged.
They are dangerous for the integrity and availability of the message.	They can be dangerous for confidentiality of the message.
Active attacks involve Masquerade, Modification of message, Repudiation, Replay, and Denial of service.	It involves traffic analysis, the release of a message.

Security Service

- A service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.
- Security services implement security policies and are implemented by security mechanisms.
 - **Authentication**
 - **Access Control**
 - **Data Confidentiality**
 - **Nonrepudiation**
 - **Data Integrity**
 - **Availability Service**

Security Service

Authentication

- The authentication service is concerned with assuring that a communication is authentic.
- In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from.
- In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, **at the time of connection initiation**, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be.
- Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

Security Service

Authentication

Two specific authentication services are defined in X.800:

- **Peer Entity Authentication**

Two entities are considered peers if they implement the same protocol in different systems; for example two TCP modules in two communicating systems. Peer entity authentication is provided for use at the establishment of, or at times during the data transfer phase of, a connection.

It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection

- **Data-Origin Authentication**

It does not provide protection against the duplication or modification of data units.

AUTHENTICATION

The assurance that the communicating entity is the one that it claims to be.

Peer Entity Authentication

Used in association with a logical connection to provide confidence in the identity of the entities connected.

Data-Origin Authentication

In a connectionless transfer, provides assurance that the source of received data is as claimed.

Security Service

Access Control

In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links.

To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

ACCESS CONTROL

The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

Security Service

Data Confidentiality

Confidentiality is the protection of transmitted data from passive attacks.

With respect to the content of a data transmission, several levels of protection can be identified.

The broadest service protects all user data transmitted between two users over a period of time. For example, when a TCP connection is set up between two systems, this broad protection prevents the release of any user data transmitted over the TCP connection.

Narrower forms of this service can also be defined, including the protection of a single message or even specific fields within a message.

The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility

Security Service

Data Confidentiality

DATA CONFIDENTIALITY

The protection of data from unauthorized disclosure.

Connection Confidentiality

The protection of all user data on a connection.

Connectionless Confidentiality

The protection of all user data in a single data block.

Selective-Field Confidentiality

The confidentiality of selected fields within the user data on a connection or in a single data block.

Traffic-Flow Confidentiality

The protection of the information that might be derived from observation of traffic flows.

Security Service

Data Integrity

- It can apply to a stream of messages, a single message, or selected fields within a message.
- A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays.
- The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service.
- On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.
- We can make a distinction between service with and without recovery.
- If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation. Alternatively, there are mechanisms available to recover from the loss of integrity of data.

Security Service

Data Integrity

DATA INTEGRITY

The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

Connection Integrity with Recovery

Provides for the integrity of all user data on a Connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.

Connection Integrity without Recovery

As above, but provides only detection without recovery.

Selective-Field Connection Integrity

Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.

Connectionless Integrity

Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

Selective-Field Connectionless Integrity

Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

Security Service

Nonrepudiation

- Nonrepudiation prevents either sender or receiver from denying a transmitted message.
- Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

NONREPUDIATION

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

Nonrepudiation, Origin

Proof that the message was sent by the specified party.

Nonrepudiation, Destination

Proof that the message was received by the specified party.

Security Service

Availability Service

- X.800 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them)
- X.800 treats availability as a property to be associated with various security services.
- An availability service is one that protects a system to ensure its availability.
- This service addresses the security concerns raised by denial-of-service attacks.
- It depends on proper management and control of system resources and thus depends on access control service and other security services.

Security Mechanism

- The mechanisms are divided into those that are implemented in a specific protocol layer, and those that are not specific to any particular protocol layer or security service.

Specific Security Mechanisms

- **Encipherment**
- **Digital Signature**
- **Access Control**
- **Data Integrity**
- **Authentication Exchange**
- **Traffic Padding**
- **Routing Control**
- **Notarization**

Pervasive Security Mechanisms

- **Trusted Functionality**
- **Security Label**
- **Event Detection**
- **Security Audit Trail**
- **Security Recovery**

Security Mechanism

Specific Security Mechanisms: May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

- **Encipherment**

- The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

- **Digital Signature**

- Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

- **Access Control**

- A variety of mechanisms that enforce access rights to resources. A variety of mechanisms used to assure the integrity of a data unit or stream of data units

Security Mechanism

Specific Security Mechanisms:

- **Authentication Exchange**
 - A mechanism intended to ensure the identity of an entity by means of information exchange.
- **Traffic Padding**
 - The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.
- **Routing Control**
 - Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
- **Notarization**
 - The use of a trusted third party to assure certain properties of a data exchange.

Security Mechanism

Pervasive Security Mechanisms: Mechanisms that are not specific to any particular OSI security service or protocol layer.

Trusted Functionality

That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

Security Label

The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

Event Detection

Detection of security-relevant events.

Security Audit Trail

Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

Security Recovery

Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions

Table 1.4, based on one in X.800, indicates the relationship between security services and security mechanisms.

Table 1.4 Relationship Between Security Services and Mechanisms

SERVICE	MECHANISM							
	Encipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y					Y		
Traffic flow confidentiality	Y				Y	Y		
Data integrity	Y	Y		Y				
Nonrepudiation		Y		Y			Y	
Availability				Y	Y			

Model for Network Security

- A message is to be transferred from one party to another across some sort of Internet service.
- The two parties, who are the principals in this transaction, must cooperate for the exchange to take place.
- A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.
- Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on.
- All the techniques for providing security have two components:
 - **A security-related transformation** on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.
 - **Some secret information shared** by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception

Model for Network Security

- A message is to be transferred from one party to another across some sort of Internet service.
- The two parties, who are the principals in this transaction, must cooperate for the exchange to take place.
- A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.
- Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on.
- All the techniques for providing security have two components:
 - **A security-related transformation** on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.
 - **Some secret information shared** by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception

Model for Network Security

A trusted third party may be needed to achieve secure transmission.

For example, a third party may be responsible for distributing the secret information to the two principals while keeping it away from any opponent.

Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

Model for Network Security

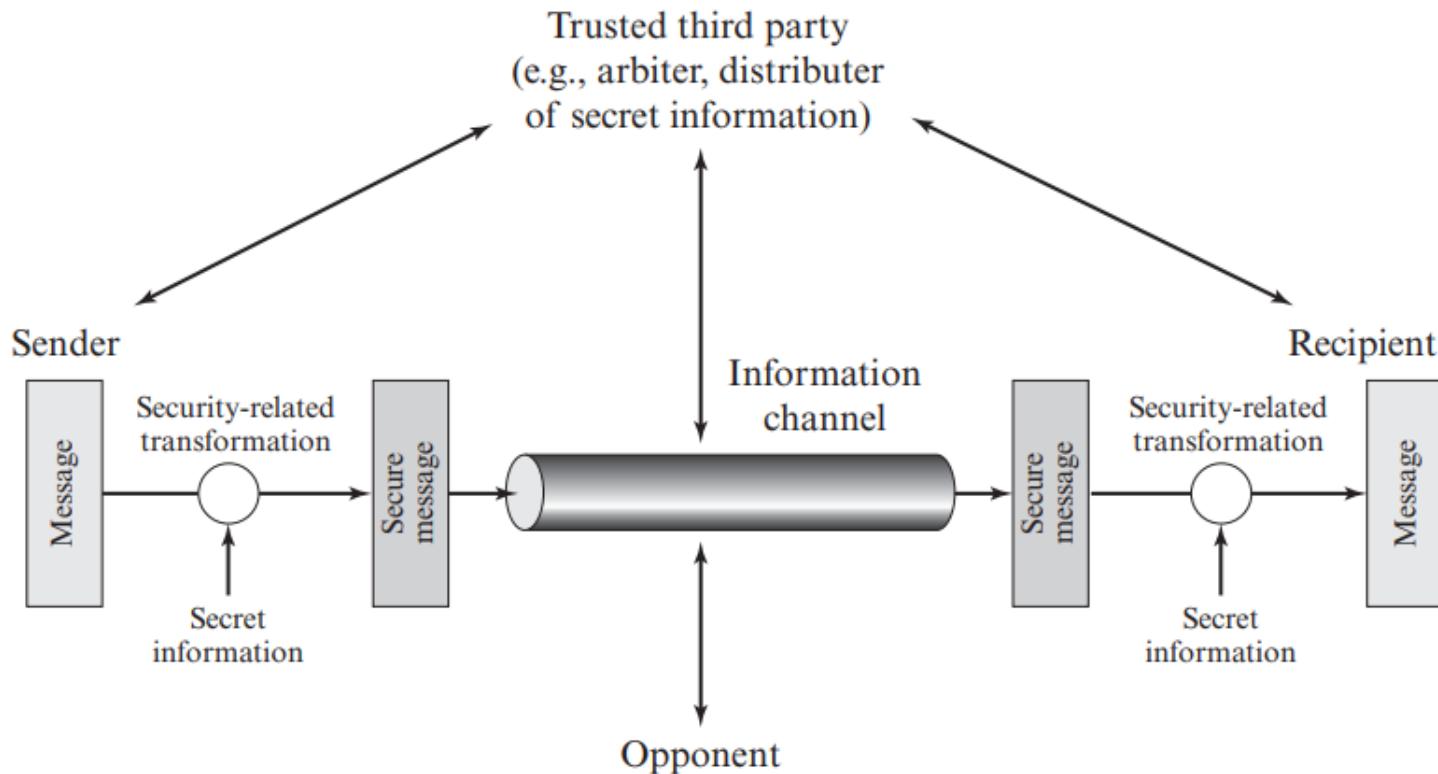


Figure 1.5 Model for Network Security

Model for Network Security

Other possible situations:

- ✓ Hackers attempt to penetrate systems that can be accessed over a network.
- ✓ The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system.
- ✓ The intruder can be a dissatisfied employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial.
- ✓ Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers.

Programs can present two kinds of threats:

- **Information access threats:** Intercept or modify data on behalf of users who should not have access to that data.
- **Service threats:** Exploit service flaws in computers to inhibit use by legitimate users.

Model for Network Security

The security mechanisms needed to cope with unwanted access fall into two broad categories .

Gatekeeper function: It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks.

Once either an unwanted user or unwanted software gains access, the second line of defense consists of a variety of **internal controls** that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

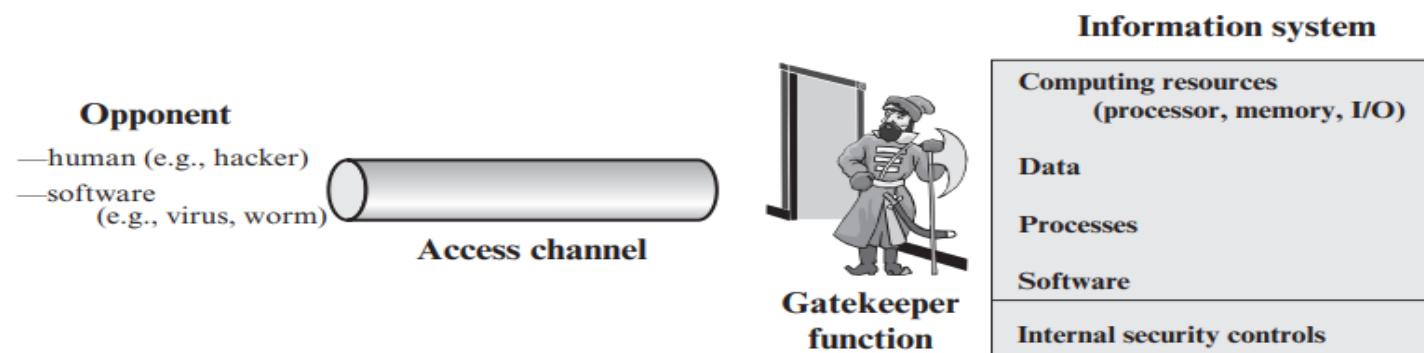


Figure 1.6 Network Access Security Model

Data Security and Privacy

DSE 3258

**L3 -Symmetric Encryption-
Substitution Cipher**
**(Caesar Cipher, Playfair Cipher,
Hill Cipher, Transposition
Techniques)**

Introduction

- An original message is known as the **plaintext**, while the coded message is called the **ciphertext**.
- The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**.
- The many schemes used for encryption constitute the area of study known as **cryptography**.
- Such a scheme is known as a **cryptographic system** or a **cipher**.
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**.
- The areas of cryptography and cryptanalysis together are called **cryptology**.

Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext.

All encryption algorithms are based on two general principles:

substitution: in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element,

transposition: in which elements in the plaintext are rearranged.

Most systems, referred to as *product systems*, involve multiple stages of substitutions and transpositions.

2. The number of keys used.

- If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption.
- If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.

3. The way in which the plaintext is processed.

- A *block cipher* processes the input one block of elements at a time, producing an output block for each input block.
- A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

Symmetric Cipher Model

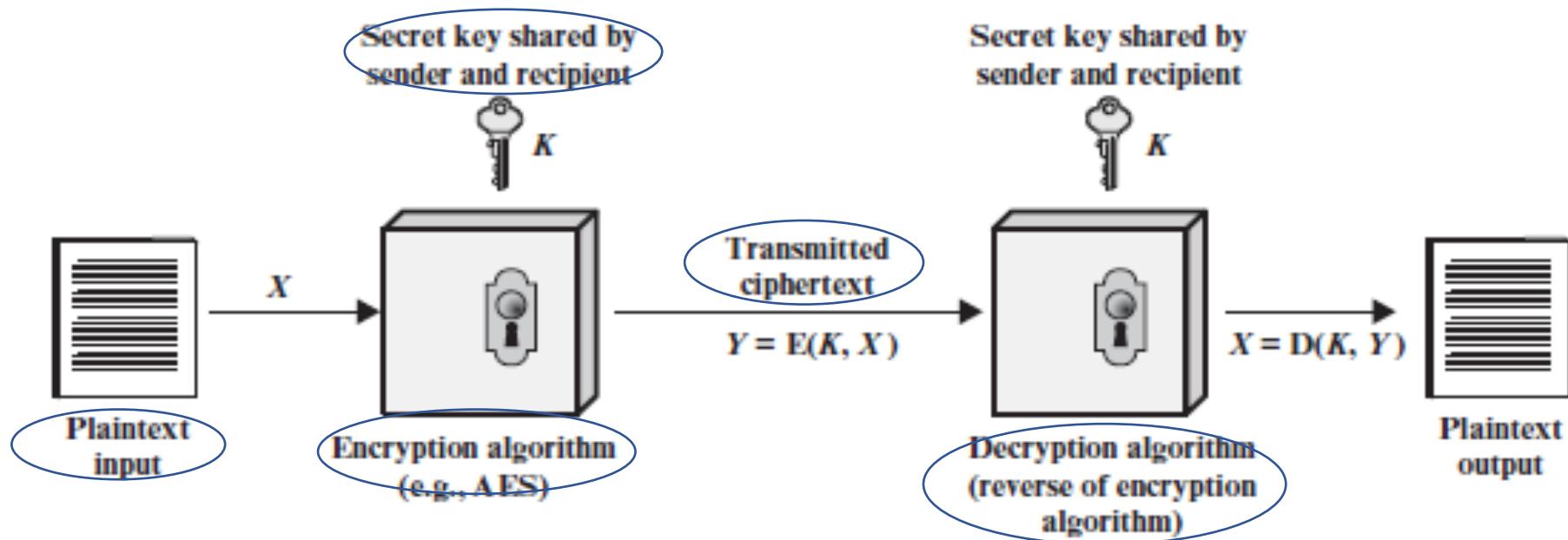


Figure 3.1 Simplified Model of Symmetric Encryption

Symmetric Cipher Model

- **A symmetric encryption scheme has five ingredients :**
 - **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
 - **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
 - **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.
 - **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
 - **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Symmetric Cipher Model

There are two requirements for secure use of conventional encryption:

1. A strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: **The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.**
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

Symmetric Cipher Model

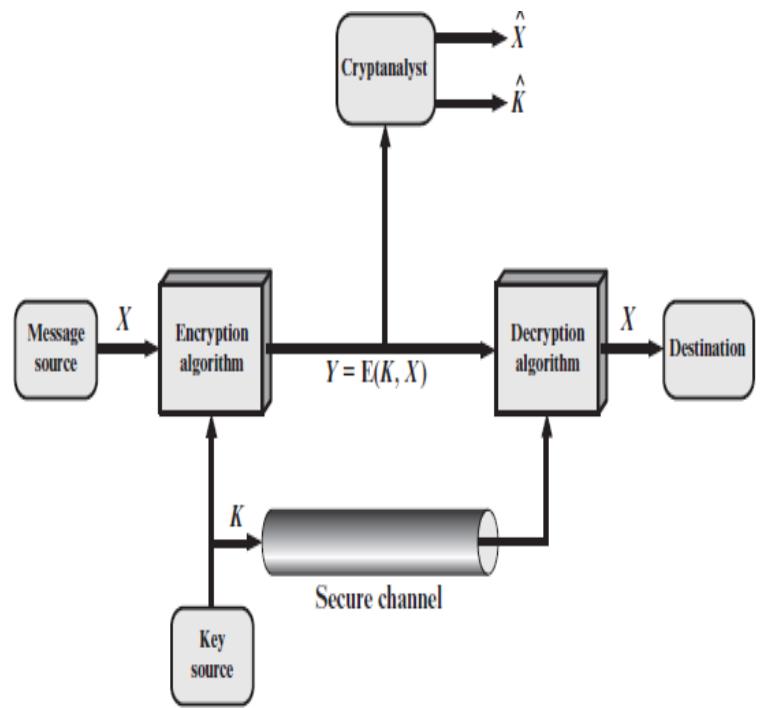


Figure 3.2 Model of Symmetric Cryptosystem

- A source produces a message in plaintext, $X = [X_1, X_2, \dots, X_M]$.
- The M elements of X are letters in some finite alphabet.
- Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet $\{0, 1\}$ is typically used.
- For encryption, a key of the form $K = [K_1, K_2, \dots, K_J]$ is generated.
- If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel.
- Alternatively, a third party could generate the key and securely deliver it to both source and destination.

Symmetric Cipher Model

With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \dots, Y_N]$.

We can write this as $Y = E(K, X)$

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X , with the specific function determined by the value of the key K .

The intended receiver, in possession of the key, is able to invert the transformation:

$X = D(K, Y)$

An opponent, observing Y but not having access to K or X , may attempt to recover X or K or both X and K .

It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover X by generating a plaintext estimate X_n .

Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate K_n .

Cryptanalysis and Brute-Force Attack

- Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:
 - **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
 - **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

Cryptanalysis and Brute-Force Attack

- Summary of various types of **cryptanalytic attacks** based on the amount of information known to the cryptanalyst.

Table 3.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	■ Encryption algorithm ■ Ciphertext
Known Plaintext	■ Encryption algorithm ■ Ciphertext ■ One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	■ Encryption algorithm ■ Ciphertext ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

The analyst may know that certain plaintext patterns will appear in a message

If the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a chosen-plaintext attack is possible.

The attacker has capability to make the victim (who obviously knows the secret key) decrypt any ciphertext and send him back the result. By analysing the chosen ciphertext and the corresponding received plaintext, the intruder tries to guess the secret key which has been used by the victim

- An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there.
- Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:
 - ■ The cost of breaking the cipher exceeds the value of the encrypted information.
 - ■ The time required to break the cipher exceeds the useful lifetime of the information.
- An encryption scheme is said to be **computationally secure** if either of the above two criteria are met.

- A **brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.
- On average, half of all possible keys must be tried to achieve success. That is, if there are X different keys, on average an attacker would discover the actual key after $X/2$ tries.

SUBSTITUTION TECHNIQUES

- A **substitution technique** is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

➤ Caesar Cipher

- The earliest known, and the simplest, use of a substitution cipher was by Julius Caesar.
- The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.
- Caesar Cipher is a method in which each letter in the alphabet is rotated by $(P+K) \bmod 26$ letters as shown:

SUBSTITUTION TECHNIQUES (Caesar Cipher Contd..)

plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

Note that the alphabet is wrapped around, so that the letter following Z is A.

We can define the transformation by listing all possibilities, as follows:

plain: a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Let us assign a numerical equivalent to each letter:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Then the algorithm can be expressed as follows. For each plaintext letter p , substitute the ciphertext letter C :²

$$C = E(3, p) = (p + 3) \bmod 26$$

A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(k, p) = (p + k) \bmod 26$$



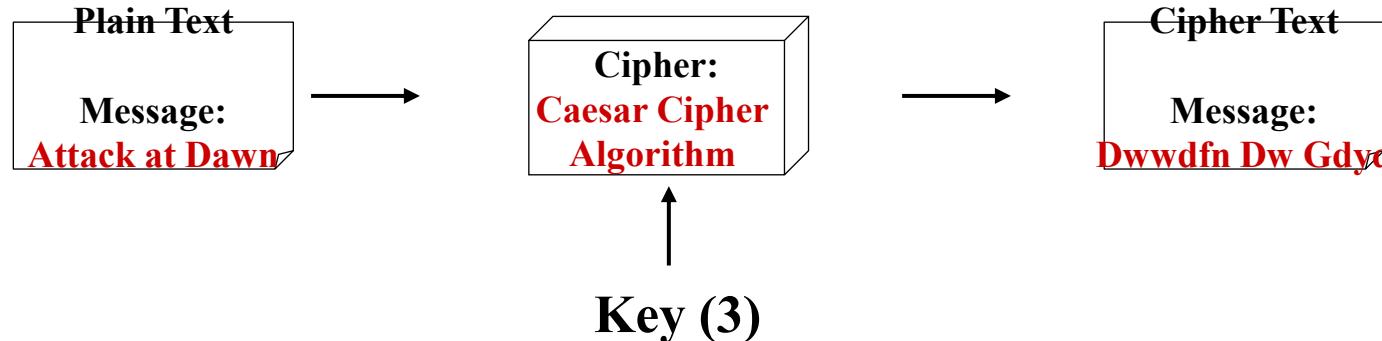
SUBSTITUTION TECHNIQUES (Caesar Cipher Contd..)

where k takes on a value in the range 1 to 25. The decryption algorithm is simply

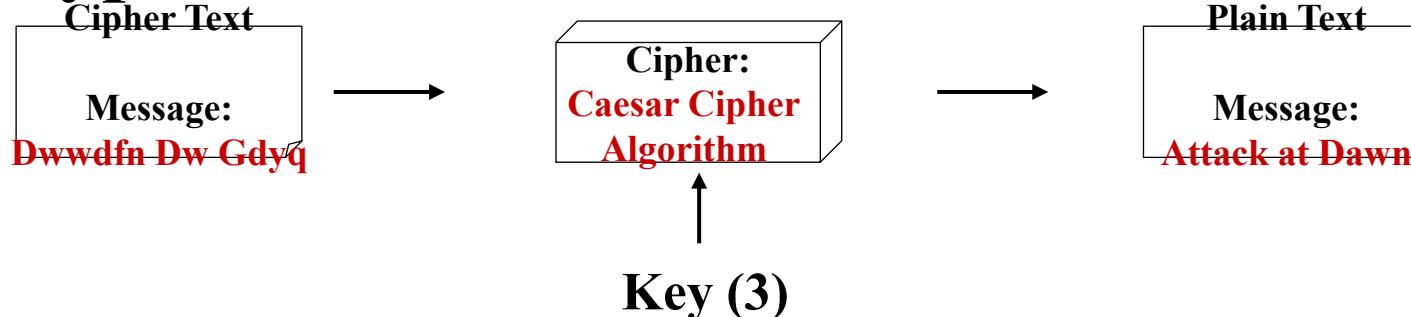
$$p = D(k, C) = (C - k) \bmod 26$$



Encryption



Decryption



SUBSTITUTION TECHNIQUES (**Caesar Cipher Contd..**)

Advantages:

1. It is very easy to implement.
2. This method is the simplest method of cryptography.
3. Only one short key is used in its entire process.
4. If a system does not use complex coding techniques, it is the best method for it.
5. It requires only a few computing resources.

Disadvantages:

1. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.
2. It provides very little security.
3. By looking at the pattern of letters in it, the entire message can be decrypted

SUBSTITUTION TECHNIQUES (Caesar Cipher Contd..)

- If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys.

KEY	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrhcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepec	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbd
20	vnnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

Figure 3.3 Brute-Force Cryptanalysis of Caesar Cipher

SUBSTITUTION TECHNIQUES

➤ Monoalphabetic Cipher

- Caesar's cipher has only 25 keys which were easy to brute-forced.
- But it could increase the security if the key space increased.
- This was done using arbitrary substitution --- by permutating the alphabet.
- A permutation is a finite set of elements S in ordered sequence of all the elements of S with each element appended exactly once.
- Example: $S = \{a, b, c\}$
can generate → {abc, acb, bac, bca, cab, cba}

Technique used→

Mixing the
alphabets as
key

SUBSTITUTION TECHNIQUES (Monoalphabetic Cipher Contd..)

- Example:
 - PT: Data Science
 - Key: Manipal

IMTMSNDPJNP

A	B	C	D	E	F	G	H	I	J	K	L	M	N
m	A	n	i	p	l	b	c	d	e	f	g	h	j

O	P	Q	R	S	T	U	V	W	X	Y	Z		
k	o	q	r	s	t	u	v	w	x	y	z		

SUBSTITUTION TECHNIQUES (Monoalphabetic Cipher Contd..)

- Unfortunately, monoalphabetic substitution ciphers are also subject to a letter frequency analysis...
- If the cryptanalyst knows the nature of the plaintext (e.g., non-compressed English text), then the analyst can exploit the regularities of the language.

SUBSTITUTION TECHNIQUES

➤ Playfair Cipher

- The best known multiple letter encryption cipher is the playfair, which treats digrams in the plaintext as single units and translates these units into cipher text digrams.
- The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword.
- Let the keyword be “monarchy”. The matrix is constructed by filling in the letters of the keyword(minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetical order.
- The letters I and J count as one letter.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

▪ **Plaintext is encrypted two letters at a time, according to the following rules:**

1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.
2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.
3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.
4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

Construction of Diagrams for Plain text

Ex 1: PT = attack

at	ta	ck
----	----	----

Ex 2: PT = manipal

ma	ni	pa	lx
----	----	----	----

Ex 3: PT = balloon

ba	lx	lo	on
----	----	----	----

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

Construction of 5x5 Keyword matrix

MONARCHY

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

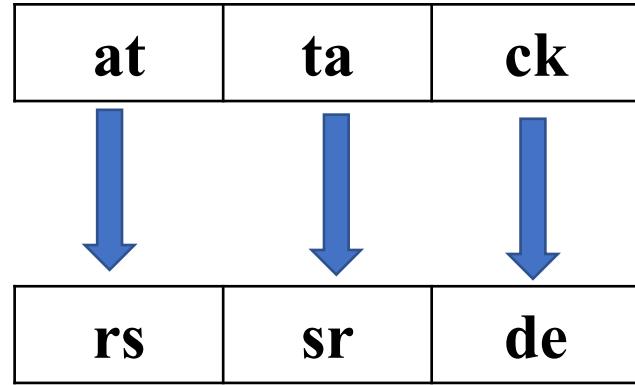
MANIPAL

M	A	N	I/J	P
L	B	C	D	E
F	G	H	K	O
Q	R	S	T	U
V	W	X	Y	Z

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

PT = attack
KW = MONARCHY

Diagram=



M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

Encryption process

PT = balloon

KW = MONARCHY

Diagram=

ba	lx	lo	on
----	----	----	----



M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

I/JB	SU	PM	NA
------	----	----	----

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

- The Playfair cipher is a great advance over simple monoalphabetic ciphers.
- For one thing, whereas there are only 26 letters, there are $26 * 26 = 676$ diagrams, so the identification of individual diagrams is more difficult.
- The relative frequencies of individual letters exhibit a much greater range than that of diagrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time considered unbreakable.
- Despite this level of confidence in its security, the Playfair cipher is relatively easy to break, because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

- **Rules for Decryption**
 1. Two ciphertext letters in the same row of the matrix are each replaced by the letter to the left, with the last element of the row circularly following the first.
 2. Two ciphertext letters that fall in the same column of the matrix are replaced by the letters above, with the bottom element of the column circularly following the top.
 3. Otherwise, each ciphertext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other ciphertext letter.

SUBSTITUTION TECHNIQUES (Playfair Cipher Contd..)

- Find ciphertext for following :

Assume “communication” is the plaintext and “computer” is the encryption key.

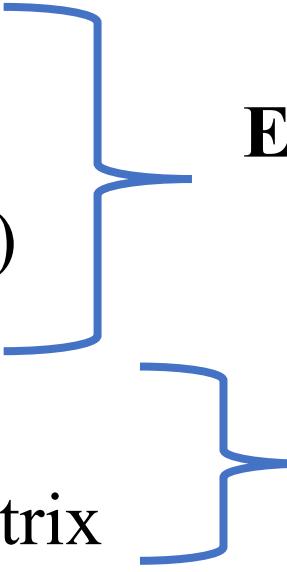
SUBSTITUTION TECHNIQUES

Hill cipher:

- Can encipher **multiple letters** at a time
- No letter gets same cipher value after encryption
 - Ex: $E(A) = Z$
 $E(A) = F$
- **Developed by Lester Hill in 1929**
- Can encrypt a group of letters.
 - Ex: 2, 3 or more letters at a time (**diagraph, trigraph or polygraph**)
- Key is a squared matrix.
 - **2x2 ---- two letters can be encrypted at a time**
 - **3x3 --- three letters can be encrypted at a time**

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

Required math

- Linear algebra
 - Matrix multiplication
 - Arithmetic modulo ($\text{mod } 26$)
 - Square matrix
 - Determinant of matrix
 - Multiplicative inverse of matrix
- 
- Encryption**
- Decryption**

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

- **Hill Algorithm**

$$C = E(K, P) = P K \text{ mod } 26$$

$$\begin{aligned}P = D(K, C) &= C K^{-1} \text{ mod } 26 \\&= P K K^{-1} \text{ mod } 26\end{aligned}$$



Matrix Multiplication

Note:

C and P are row vector of length 3.

K is a 3 x 3 matrix of encryption key

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

THE HILL ALGORITHM This encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters. The substitution is determined by m linear equations in which each character is assigned a numerical value ($a = 0, b = 1, \dots, z = 25$). For $m = 3$, the system can be described as

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3) \bmod 26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3) \bmod 26$$

$$c_3 = (k_{13}p_1 + k_{23}p_2 + k_{33}p_3) \bmod 26$$

This can be expressed in terms of row vectors and matrices:⁶

$$(c_1 \ c_2 \ c_3) = (p_1 \ p_2 \ p_3) \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

or

$$\mathbf{C} = \mathbf{PK} \bmod 26$$

where C and P are row vectors of length 3 representing the plaintext and ciphertext, and K is a 3 * 3 matrix representing the encryption key. Operations are performed mod 26.

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

Example:

PT = “paymoremoney”

$$K = \begin{matrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{matrix}$$

Assume A = 0, Z = 25

Note:

If last pair
is two letter
odd filled
(letter X)

p	a	y	m	o	r	e	m	e	n	e	y
15	0	24	12	14	17	4	12	14	13	4	24

Note: as key is 3x3, it default that PT is grouped to 3 letters

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

- The first three letters of the plaintext are represented by the vector $(15 \ 0 \ 24)$. Then $(15 \ 0 \ 24)K = (303 \ 303 \ 531) \text{ mod } 26 = (17 \ 17 \ 11) = \text{RRL}$.

$$\text{PT (p a y)} = \text{PT (15 0 24)}$$

$$C_1 \ C_2 \ C_3 = \underbrace{(15 \ 0 \ 24)}_{= \begin{array}{l} 15 \times 17 + 0 \times 21 + 24 \times 2 \\ 15 \times 17 + 0 \times 18 + 24 \times 2 \\ 15 \times 5 + 0 \times 21 + 24 \times 19 \end{array}} \underbrace{\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}}_{\text{mod } 26} \text{ mod } 26$$

PT = PAY MORE MONEY

CT = RRLMWBKASPDH

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

Decryption

$$\mathbf{P} = \mathbf{D}(\mathbf{K}, \mathbf{C}) = \mathbf{C} \mathbf{K}^{-1} \bmod 26$$

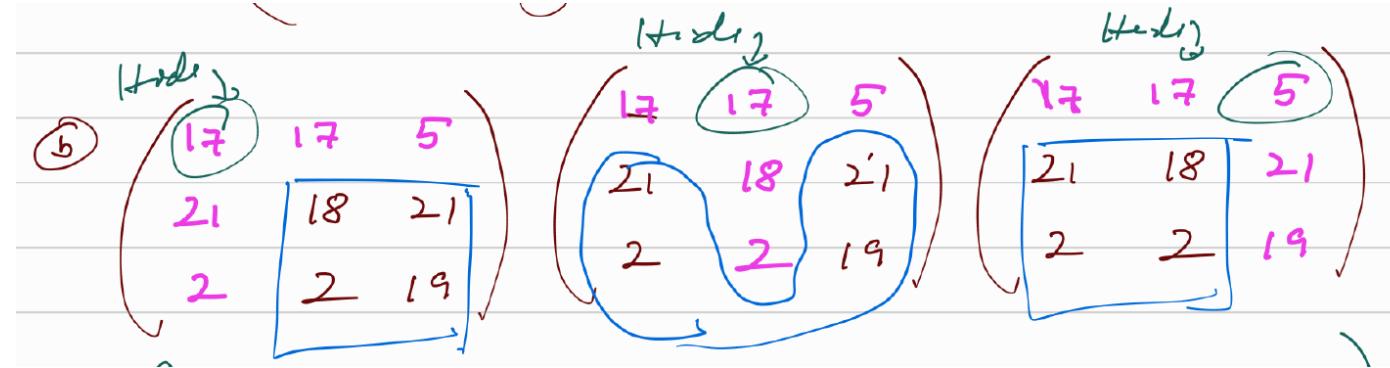
$$\mathbf{K}^{-1} = \frac{1}{\text{Det K}} \times \text{Adj K}$$

Determinant

Adjoint

$$\text{Det K} = \text{Det} \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \bmod 26$$

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)



$$= (17(18 \times 14 - 2 \times 21) - 17(19 \times 21 - 2 \times 21) + 5(21 \times 2 - 18 \times 2)) \pmod{26}$$

$$= 17(300) - 17(352) + 5(6) \pmod{26}$$

$$= 5100 - 6069 + 30 \pmod{26}$$

$$= -939 \pmod{26}$$

16 -ve value
in mod.

$$= -3 \pmod{26} \Rightarrow 26 + (-3)$$

Det = 23

Example (contd..)

$$\text{Adj K} = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

$$\begin{aligned}
 &= \left[\begin{array}{c} f \begin{vmatrix} 18 & 21 \\ 2 & 19 \end{vmatrix} - \begin{vmatrix} 21 & 21 \\ 2 & 9 \end{vmatrix} + \begin{vmatrix} 21 & 18 \\ 2 & 2 \end{vmatrix} \\ - \begin{vmatrix} 17 & 5 \\ 2 & 19 \end{vmatrix} + \begin{vmatrix} 17 & 5 \\ 2 & 19 \end{vmatrix} - \begin{vmatrix} 17 & 17 \\ 2 & 2 \end{vmatrix} \\ + \begin{vmatrix} 17 & 5 \\ 18 & 21 \end{vmatrix} - \begin{vmatrix} 17 & 5 \\ 21 & 21 \end{vmatrix} + \begin{vmatrix} 17 & 17 \\ 21 & 18 \end{vmatrix} \end{array} \right]^T
 \end{aligned}$$

$$= \begin{bmatrix} +300 & -147 & +6 \\ -313 & +313 & -0 \\ +267 & -252 & +(-51) \end{bmatrix}^T$$

↓

$$= \begin{bmatrix} 300 & -147 & 6 \\ -313 & 313 & 0 \\ 267 & -252 & -51 \end{bmatrix}^T \bmod 26$$

$$= \begin{bmatrix} 14 & 7 & 6 \\ 25 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}^T \rightarrow = \begin{bmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{bmatrix}$$

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

Example (contd..)

$$\text{Ad}(K) = \begin{bmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{bmatrix} \pmod{26}$$

$$K^{-1} = \frac{1}{23} \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \pmod{26}$$

$$K^{-1} = 23^{-1} \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \pmod{26}.$$

multiplicative inverse of 23.

$$23^{-1} \pmod{26} = [17 \pmod{26}]$$

SUBSTITUTION TECHNIQUES (Hill cipher Cont..)

Example (contd..)

$$K^{-1} = 17 \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \text{ mod } 26$$

$$K^{-1} = \begin{pmatrix} 238 & 425 & 119 \\ 119 & 17 & 136 \\ 102 & 0 & 17 \end{pmatrix} \text{ mod } 26$$

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \text{ mod } 26$$

Example (contd..)

CT = RRL MWB KAS PDH

$$P_1 P_2 P_3 = (RRL) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \text{ mod } 26$$

(i)

$$= (17 17 14) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \text{ mod } 26$$

$$= (587 \quad 442 \quad 544) \text{ mod } 26$$

$$\boxed{\text{PT}} = \boxed{(15 \quad 0 \quad 24)} \text{ mod } 26$$

② MWB $\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \text{ mod } 26$

= $(402 \ 482 \ 329) \text{ mod } 26$

= 12 14 17

= m or

③ KAS $(10 \ 0 \ 18)$
↓

4 12 14
E M O

④ PDEH $(15 \ 3 \ 7)$
↓

13 4 24
N E U

Transposition Techniques

- Re-arrangement of letters based on permutation key.
- No replacement of letter
- Two variants:
 - Rail fence technique
 - Row transposition cipher.

Transposition Techniques (contd..)

Rail fence technique

- Here the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows to form a ciphertext.
- The number of columns in rail fence cipher remains equal to the length of plain-text message.

PT: “meet me after the toga party” with a rail fence of depth 2, we write the following:

¹ m e m a t r h t g p r y
e t e f e t e o a a t

CT: MEMATRHTGPRYETEFETEOAAT

Note: used for short messages as easy to break

Transposition Techniques (contd..)

For example, let's consider the **plaintext** "This is a secret message".

Plaintext T H I S I S A S E C R E T M E S S A G E

To encode this message we will first write over two lines (the "rails of the fence") as follows:

Rail Fence	T	I		I		A		E		R		T		E		S		G	
Encoding	H	S	S	S	S	C	C	E	E	M	M	S	S	A	A	E	E		

Note that all white spaces have been removed from the plain text.

The **ciphertext** is then read off by writing the top row first, followed by the bottom row:

Ciphertext T I I A E R T E S G H S S S C E M S A E

Transposition Techniques (contd..)

Key = 3

Plaintext	T H I S I S A S E C R E T M E S S A G E																																													
Rail Fence Encoding	<table border="1"><tr><td>T</td><td></td><td></td><td>I</td><td></td><td></td><td>E</td><td></td><td>T</td><td></td><td>S</td><td></td><td>A</td><td></td><td>E</td></tr><tr><td></td><td>H</td><td>S</td><td>S</td><td>S</td><td>C</td><td>E</td><td>M</td><td>S</td><td>A</td><td></td><td></td><td></td><td></td><td>E</td></tr><tr><td></td><td>I</td><td></td><td></td><td>A</td><td>R</td><td></td><td>E</td><td></td><td></td><td></td><td></td><td>G</td><td></td><td></td></tr></table>	T			I			E		T		S		A		E		H	S	S	S	C	E	M	S	A					E		I			A	R		E					G		
T			I			E		T		S		A		E																																
	H	S	S	S	C	E	M	S	A					E																																
	I			A	R		E					G																																		
key = 3																																														
Ciphertext	T I E T S H S S S C E M S A E I A R E G																																													
A Rail Fence Cipher with 3 "rails" (Key = 3)																																														

Transposition Techniques (contd..)

Key = 4

Plaintext	T H I S I S A S E C R E T M E S S A G E																																																																				
<i>Rail Fence Encoding</i>																																																																					
<i>key = 4</i>																																																																					
	<table border="1"><tr><td>T</td><td></td><td></td><td></td><td></td><td>A</td><td></td><td></td><td></td><td></td><td>T</td><td></td><td></td><td></td><td></td><td>G</td><td></td></tr><tr><td></td><td>H</td><td></td><td></td><td></td><td>S</td><td>S</td><td></td><td></td><td>E</td><td>M</td><td></td><td></td><td>A</td><td></td><td>E</td><td></td></tr><tr><td></td><td>I</td><td>I</td><td></td><td></td><td>E</td><td>R</td><td></td><td></td><td>E</td><td>S</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>S</td><td></td><td></td><td></td><td>C</td><td></td><td></td><td></td><td>S</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	T					A					T					G			H				S	S			E	M			A		E			I	I			E	R			E	S								S				C				S							
T					A					T					G																																																						
	H				S	S			E	M			A		E																																																						
	I	I			E	R			E	S																																																											
	S				C				S																																																												
Ciphertext	T A T G H S S E M A E I I E R E S S C S																																																																				
A Rail Fence Cipher with 4 "rails" (Key = 4)																																																																					

Transposition Techniques (contd..)

Row Transposition Cipher

- A more complex scheme is to write the message in a rectangle
- The plaintext is written in a rectangle **row-by-row** matrix.
- The cipher is read **column-by-column** with permutation order mentioned by **key**.
- Key is value between 0 to 9.
 - Ex: 4 5 3 2 1
 4 2 1 3
- If key is a word then convert it to numeric value by assigning least letter as 1 and greatest letter as n.
 - Ex: C R Y P T O
 1 4 6 3 5 2 ----- key

Transposition Techniques (contd..)

- Example

Key:	4 3 1 2 5 6 7
Plaintext:	a t t a c k p o s t p o n e d u n t i l t w o a m x y z
Ciphertext:	TTNAAPMTSUOAODWCOIXKNLYPETZ

Key:	4 3 1 2 5 6 7
Input:	t t n a a p t m t s u o a o d w c o i x k n l y p e t z
Output:	NSCYAUOPTTWLTMDNAOIEPAXTOKZ

**Double
Transposition**

Transposition Techniques (contd..)

- Example

Key:	4 3 1 2 5 6 7
Plaintext:	a t t a c k p o s t p o n e d u n t i l t w o a m x y z
Ciphertext:	TTNAAPMTSUOAODWCOIXKNLYPETZ

Key:	4 3 1 2 5 6 7
Input:	t t n a a p t m t s u o a o d w c o i x k n l y p e t z
Output:	NSCYAUOPTTWLTMDNAOIEPAXTOKZ

**Double
Transposition**

Assignment

- Plain text : " laser beams can be modulated to carry more intelligence than radio"
- Key is: 6 3 4 1 2 5 7

1	2	3	4	5	6	7
L	A	S	E	R	B	E
A	M	S	C	A	N	B
E	M	O	D	U	L	A
T	E	D	T	O	C	A
R	R	Y	M	O	R	E
I	N	T	E	L	L	I
G	E	N	C	E	T	H
A	N	R	A	D	I	O

- " laser beams can be modulated to carry more intelligence than radio"

- Solution is :
- “bselare nscamab lodemua cdtteoa rymrrroe lteinli tncg eeh iraando”

6	3	4	1	2	5	7
B	S	E	L	A	R	E
N	S	C	A	M	A	B
L	O	D	E	M	U	A
C	D	T	T	E	O	A
R	Y	M	R	R	O	E
L	T	E	I	N	L	I
T	N	C	G	E	E	H
I	R	A	A	N	D	O

Data Security and Privacy

DSE 3258

L4 -BLOCK CIPHER

Stream cipher and Block Cipher

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time
- However, the keystream must be provided to both users in advance via some independent and secure channel. This introduces insurmountable logistical problems if the intended data traffic is very large.
- Accordingly, for practical reasons, the bit-stream generator must be implemented as an algorithmic procedure, so that the cryptographic bit stream can be produced by both users. In this approach ,the bit-stream generator is a key-controlled algorithm and must produce a bit stream that is cryptographically strong..
- A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Typically, a block size of 64 or 128 bits is used. As with a stream cipher, the two users share a symmetric encryption key

Stream cipher and Block Cipher

Confusion and Diffusion

Two basic building blocks for any cryptographic system

Confusion:

- Making the relationship between encryption key and the cipher text as complex as possible.
- Relationship between PT and CT is obscured.
- That is given CT no information about PT, Key, encryption algorithm.
- Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key. This is achieved by the use of a complex substitution algorithm.
- Ex: substitution

Diffusion:

- Making each plaintext bit affect as many CT bits as possible
- One bit change in PT has significant change in CT. This is equivalent to having each ciphertext digit be affected by many plaintext digits
- An example of diffusion is to encrypt a message $M = m_1, m_2, m_3, \dots, c$ of characters with an averaging operation:

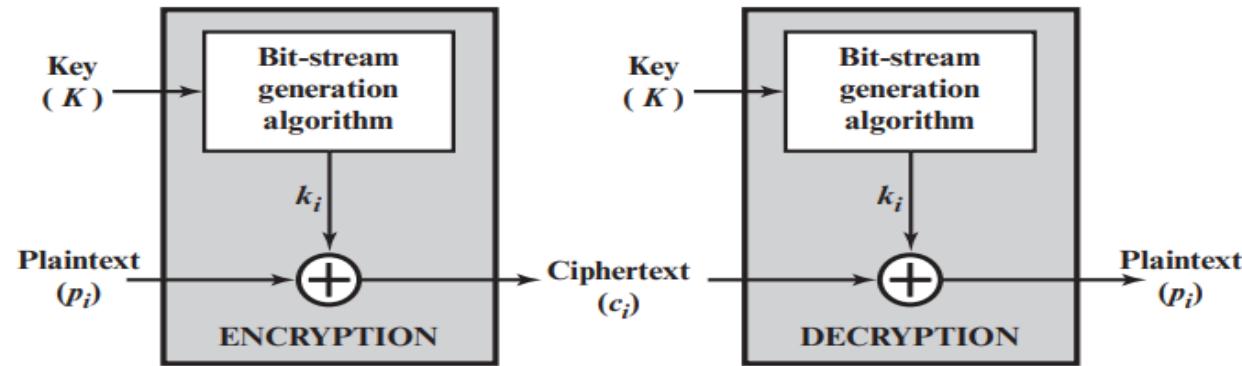
$$y_n = \left(\sum_{i=1}^k m_{n+i} \right) \bmod 26$$

- Ex: permutation or transposition

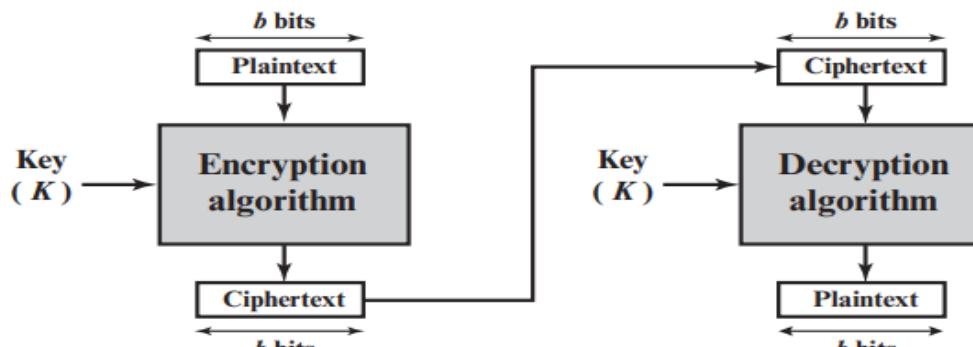
Stream cipher and Block Cipher

Block Cipher	Stream Cipher
Chunk of PT is taking as input and converted to CT	Bit-by-bit PT is converted to CT
PT size is 64 bits or more (multiple of 64)	PT size is 8 bits
Simple operations	Complex operations
Uses confusion and diffusion	Uses only confusion
Decipher is hard	Decipher is easy
Block cipher works on transposition techniques like rail-fence technique, columnar transposition technique, etc.	While stream cipher works on substitution techniques like Caesar cipher, polygram substitution cipher, etc.
Block cipher is slow as compared to a stream cipher.	While stream cipher is fast in comparison to block cipher.

Stream cipher and Block Cipher



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Figure 4.1 Stream Cipher and Block Cipher

Block Cipher

Plaintext and ciphertext consists of fixed sized blocks

Ciphertext obtained from plaintext by iterating a **round function**

Input to round function consists of key and the output of previous round

Feistel Cipher Structure

- Feistel proposed [FEIS73] block cipher by utilizing the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers.
- The essence of the approach is to develop a block cipher with a key length of k bits and a block length of n bits, allowing a total of 2^k possible transformations.
- Feistel proposed the use of a cipher that alternates substitutions and permutations, where these terms are defined as follows:
 - **Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
 - **Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

Feistel Cipher Structure

- Plain text is divided into two equal halves and processed through the algorithm.
- **Feistel cipher** refers to a type of block cipher design, not a specific cipher

Encryption:

- Split plaintext block into left and right halves: Plaintext = (L_0, R_0)
- For each round $i=1,2,\dots,n$, compute

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

where F is **round function** and K_i is **subkey**

- Ciphertext = (L_n, R_n)

Feistel Cipher

Decryption:

Ciphertext = (L_n, R_n)

- For each round $i=n, n-1, \dots, 1$, compute

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$

where F is round function and K_i is subkey

- Plaintext = (L_0, R_0)
- Formula “works” for any function F
- But only secure for certain functions F

Feistel Cipher Structure

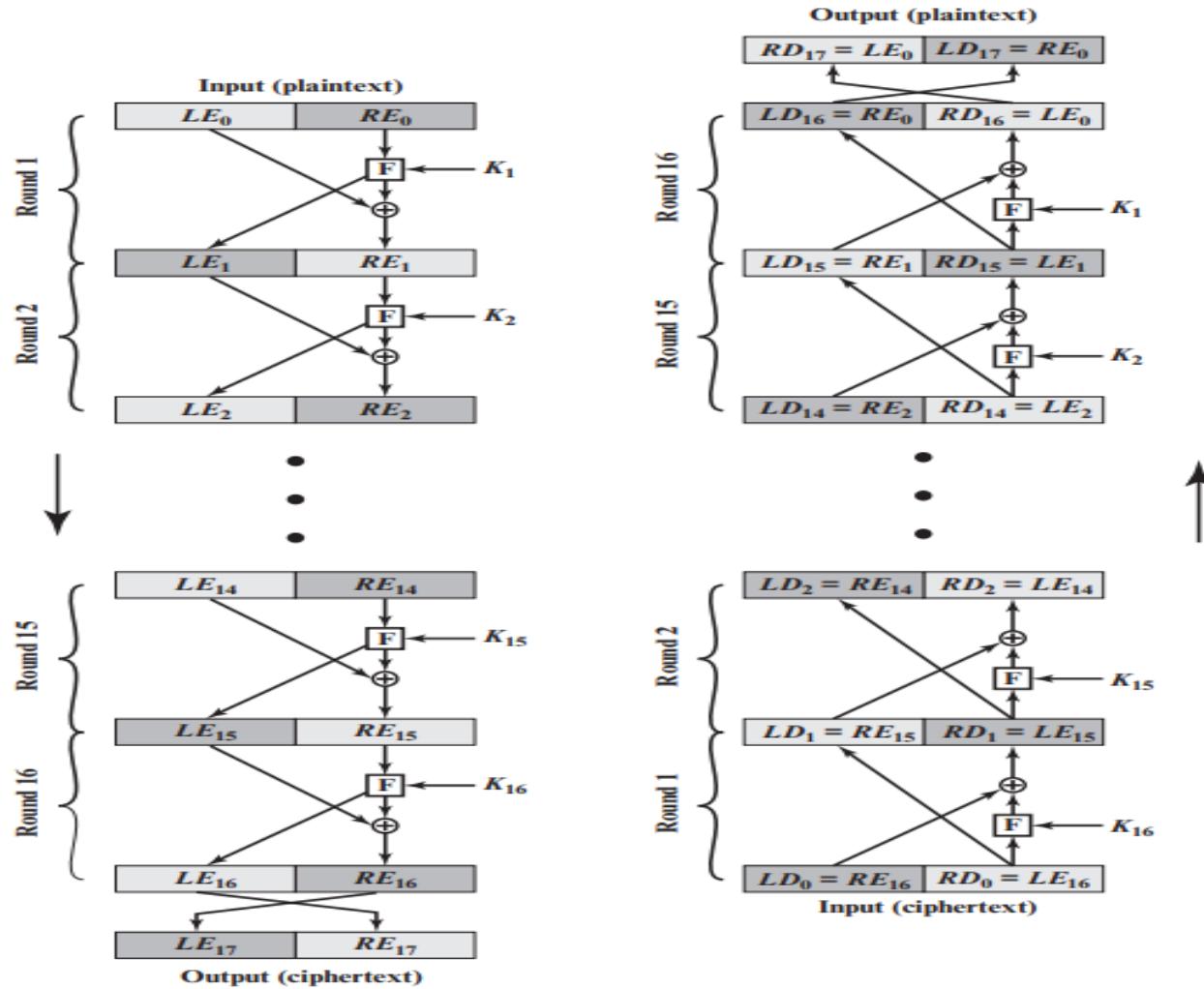
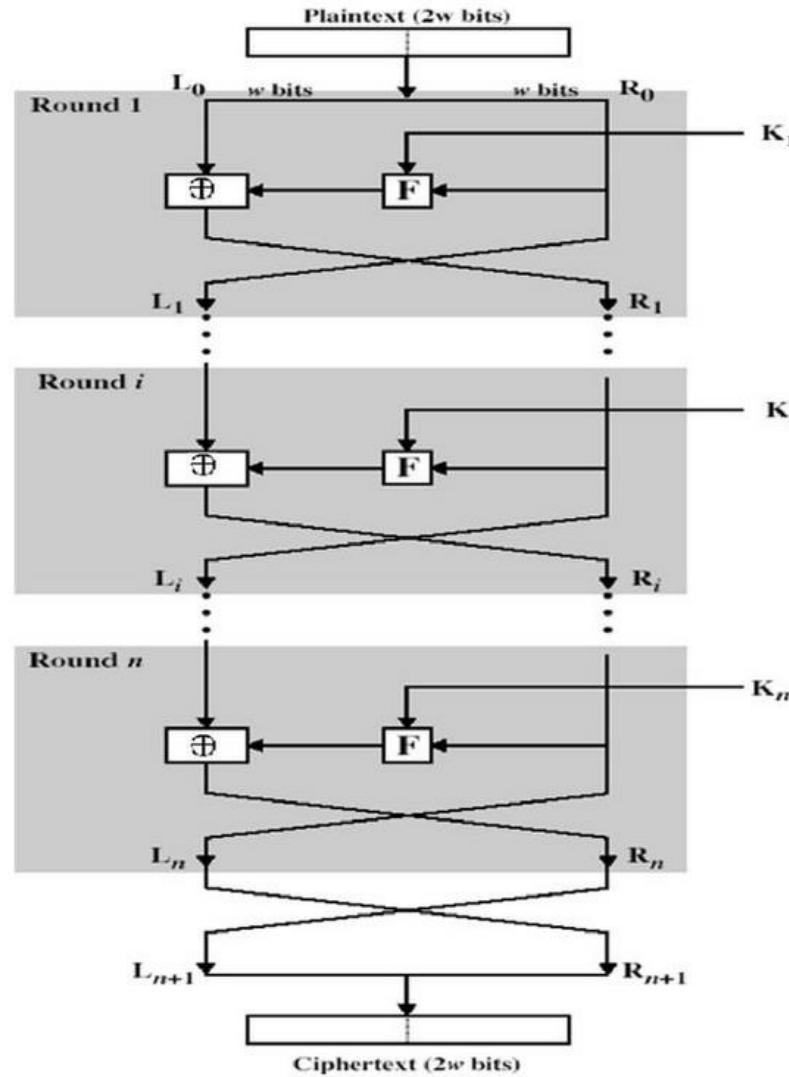


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

Feistel Cipher Structure



Feistel Cipher Structure

A Feistel network depends on the choice of the following parameters and design features:

- **Block size:** Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion. Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.
- **Key Size:** Larger key size means greater security but may decrease encryption/ decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
- **Number of rounds:** multiple rounds offer increasing security. Typical is 16 rounds as a single round provide inadequate security.
- **Subkey generation algorithm:** greater complexity will lead to greater difficulty of cryptanalysis.
- **Round function F** - greater complexity generally means greater resistance to cryptanalysis.

Feistel Cipher Structure

There are two other considerations in the design of a Feistel cipher:

- **Fast software encryption/decryption:** In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.
- **Ease of analysis:** Although we would like to make our algorithm as difficult as possible to cryptanalyze, there is great benefit in making the algorithm easy to analyze. That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength.



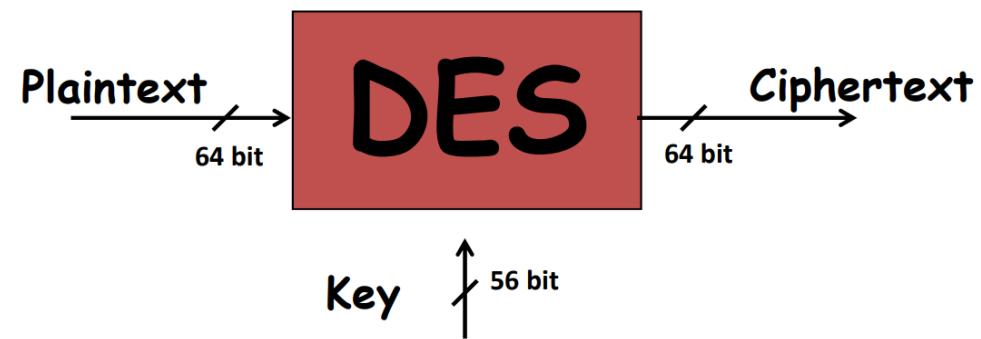
DATA ENCRYPTION STANDARD (DES)

- Symmetric block cipher.
- most widely used block cipher in world
- adopted in 1977 by National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST)
- encrypts 64-bit data using 56-bit key
- has widespread use
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption

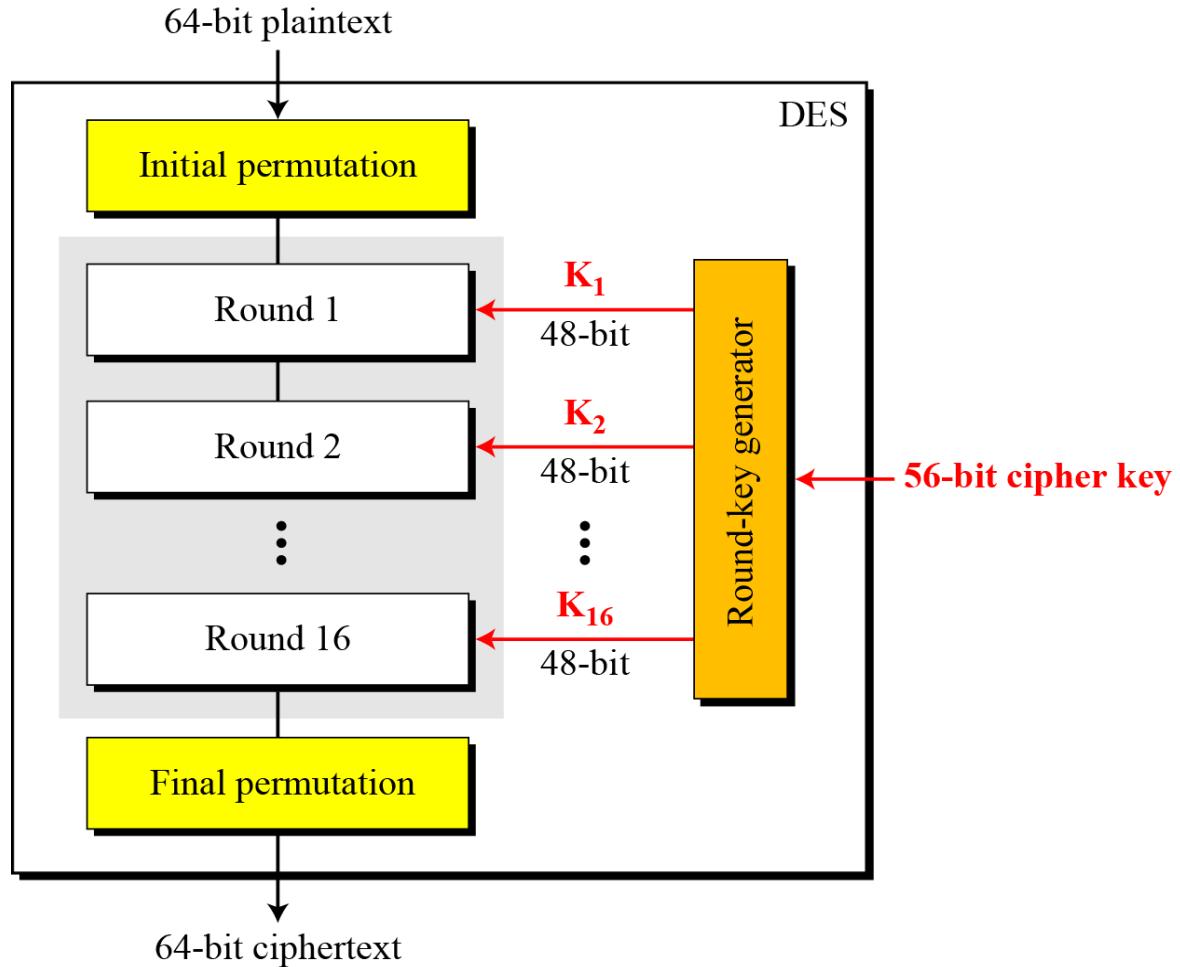
DATA ENCRYPTION STANDARD (DES)

- Follows Feistel structure.
- Block size = 64 bits of plain text
- No of rounds = 16 rounds
- Key size = 56 bits
- No of subkeys = 16 subkeys (16 rounds)
- Sub key size = 48 bits
- Cipher text size = 64 bits

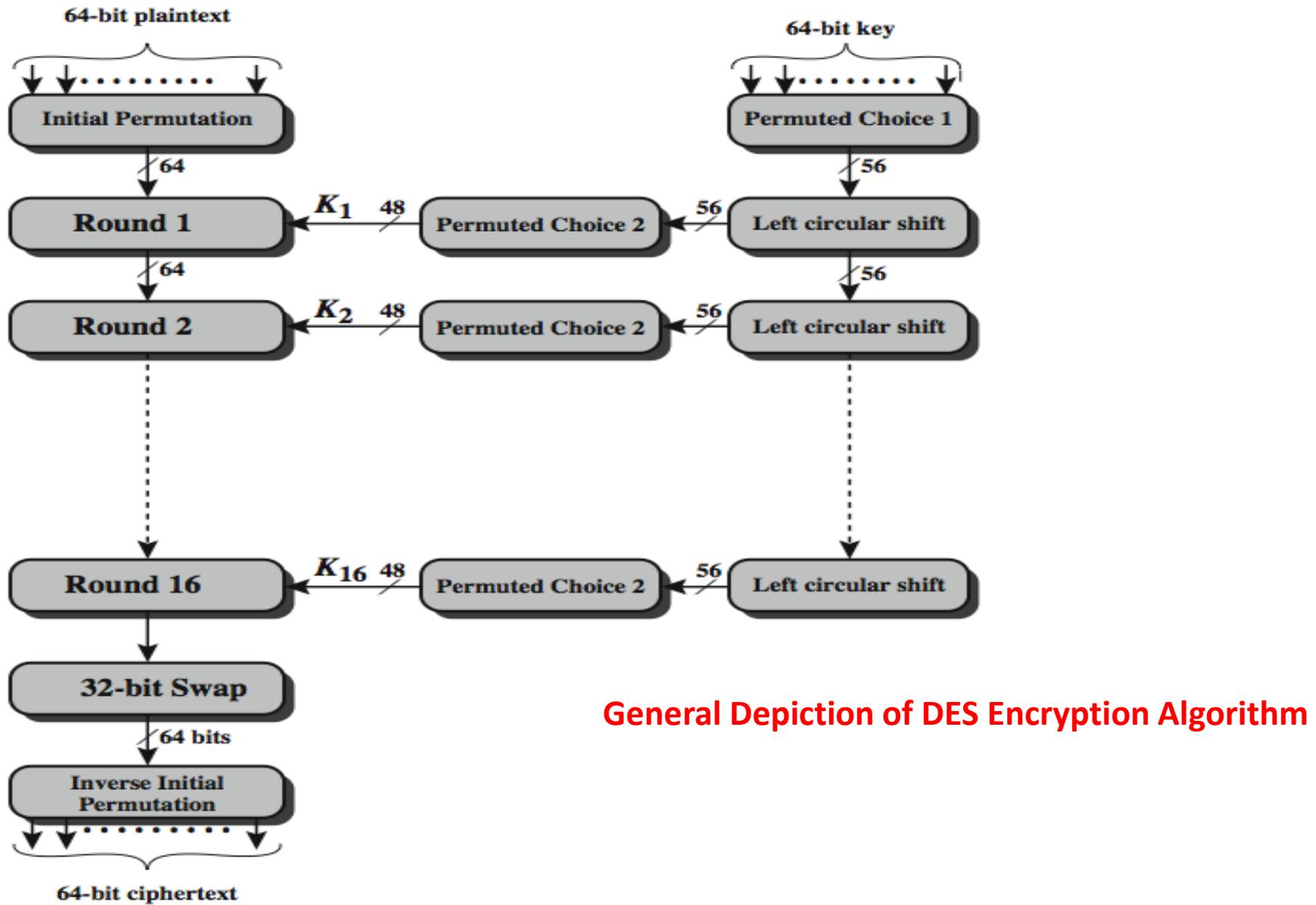
the function expects a 64-bit key as input. However, only 56 of these bits are ever used; the other 8 bits can be used as parity bits or simply set arbitrarily.



DATA ENCRYPTION STANDARD (DES)

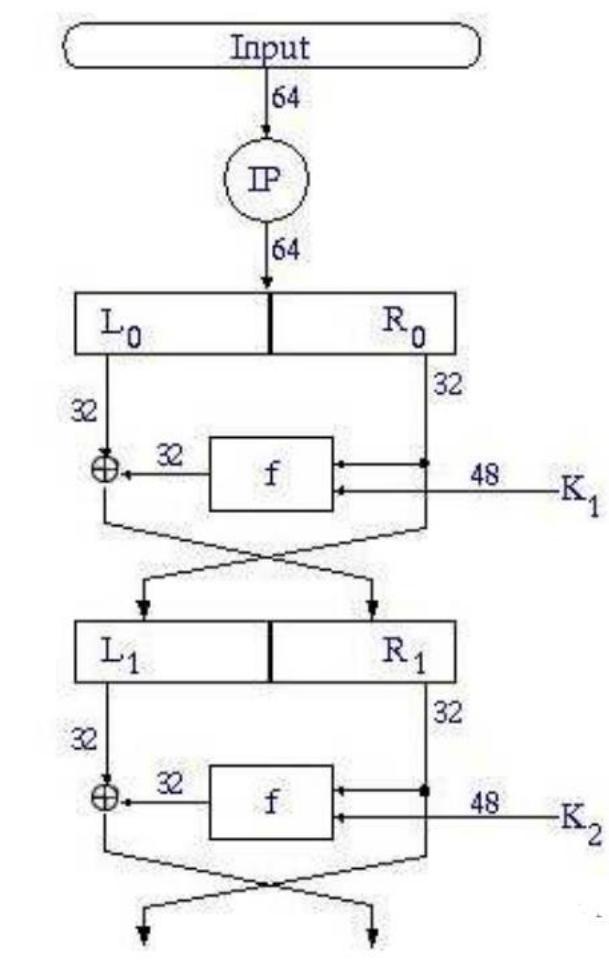


DATA ENCRYPTION STANDARD (DES)



DATA ENCRYPTION STANDARD (DES)

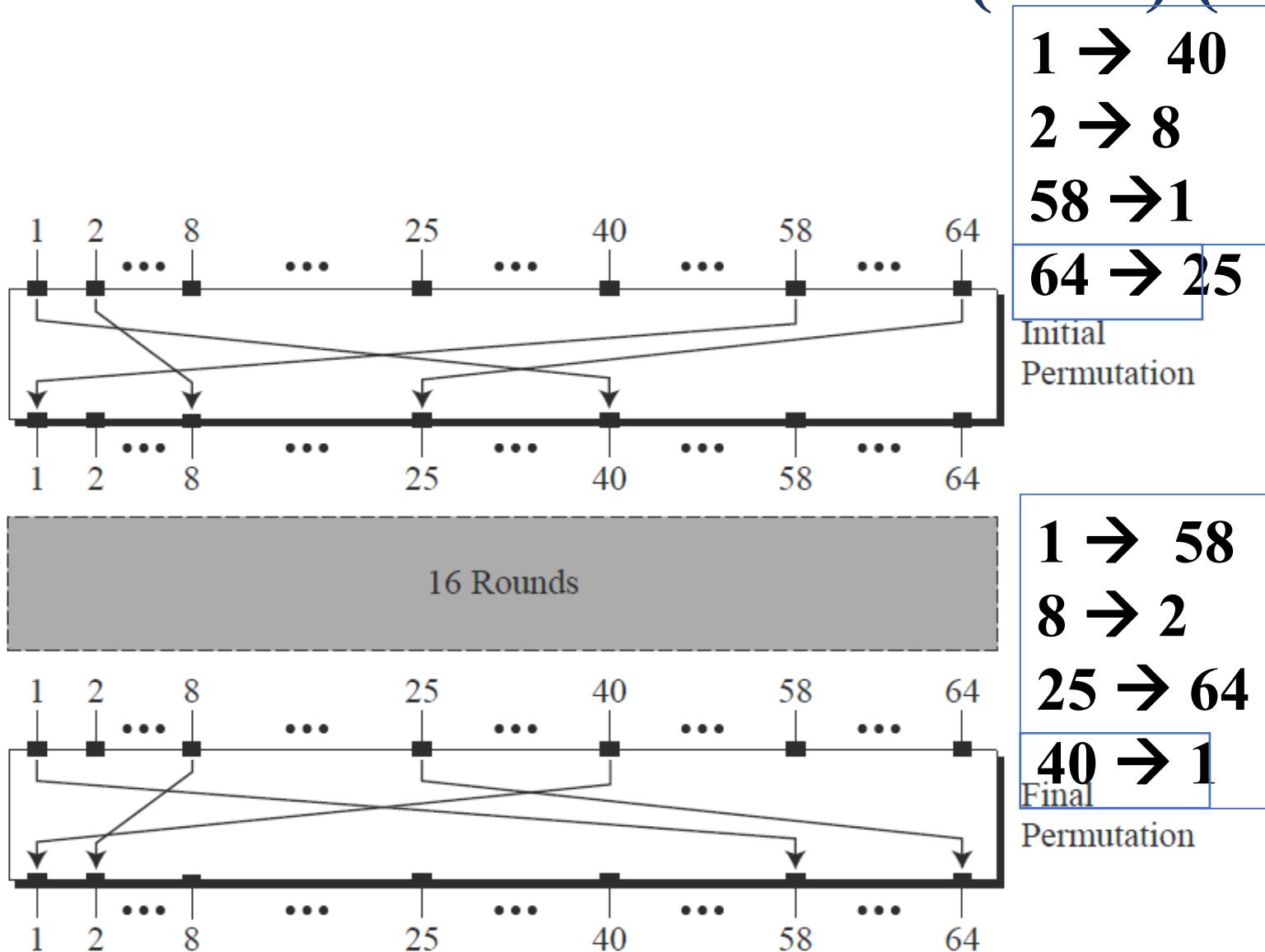
- $IP(x) = L_0 R_0$
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $y = IP^{-1}(R_{16} L_{16})$



DATA ENCRYPTION STANDARD (DES)

- First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.
- This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.
- The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key.
- The left and right halves of the output are swapped to produce the preoutput.
- Finally, the preoutput is passed through a permutation [IP-1] that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.
- With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

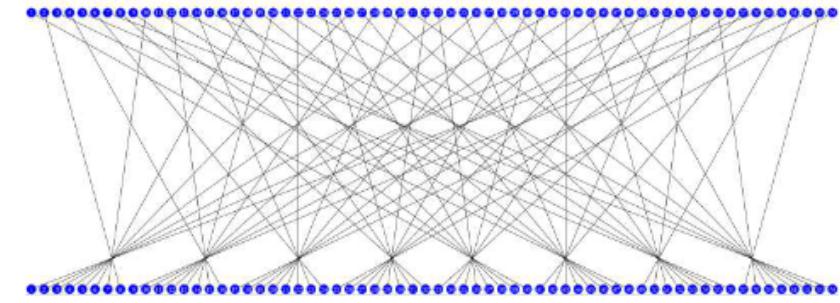
DATA ENCRYPTION STANDARD (DES) (contd..)



DATA ENCRYPTION STANDARD (DES) (contd..)

Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



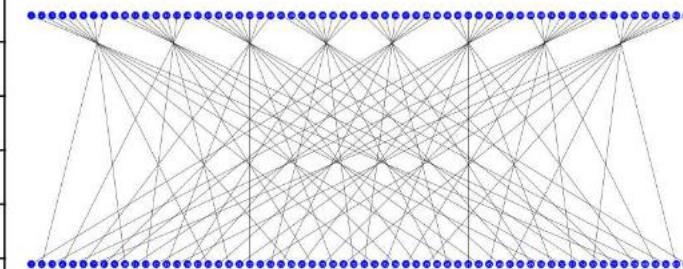
This table specifies the input permutation on a 64-bit block.

- The meaning is as follows:
- the first bit of the output is taken from the 58th bit of the input; the second bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input.
- This information is presented as a table for ease of presentation:
- it is a vector, not a matrix.

DATA ENCRYPTION STANDARD (DES) (contd..)

Final Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



The final permutation is the inverse of the initial permutation; the table is interpreted similarly.

- That is, the output of the Final Permutation has bit 40 of the preoutput block as its first bit, bit 8 as its second bit, and so on, until bit 25 of the preoutput block is the last bit of the output

DATA ENCRYPTION STANDARD (DES) (contd..)

Example 1

- Example:
 - Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0002 0000 0000 0001

- The input has only two 1s (bit 15 and bit 64); the output must also have only two 1s (the nature of straight permutation).
- Using Table, we can find the output related to these two bits.
- Bit 15 in the input becomes bit 63 in the output.
- Bit 64 in the input becomes bit 25 in the output.
- So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is

0x0000 0080 0000 0002

0: 0000 (bits 1-4)
0: 0000 (bits 5-8)
0: 0000 (bits 9-12)
2: 0010 <--bit 15 is 1 (bits 13-16)

0: 0000 (bits 49-52)
0: 0000 (bits 53-56)
0: 0000 (bits 57-60)
1: 0001 <--bit 64 is 1 (bits 61-64)

DATA ENCRYPTION STANDARD (DES) (contd..)

Example 2

Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Solution

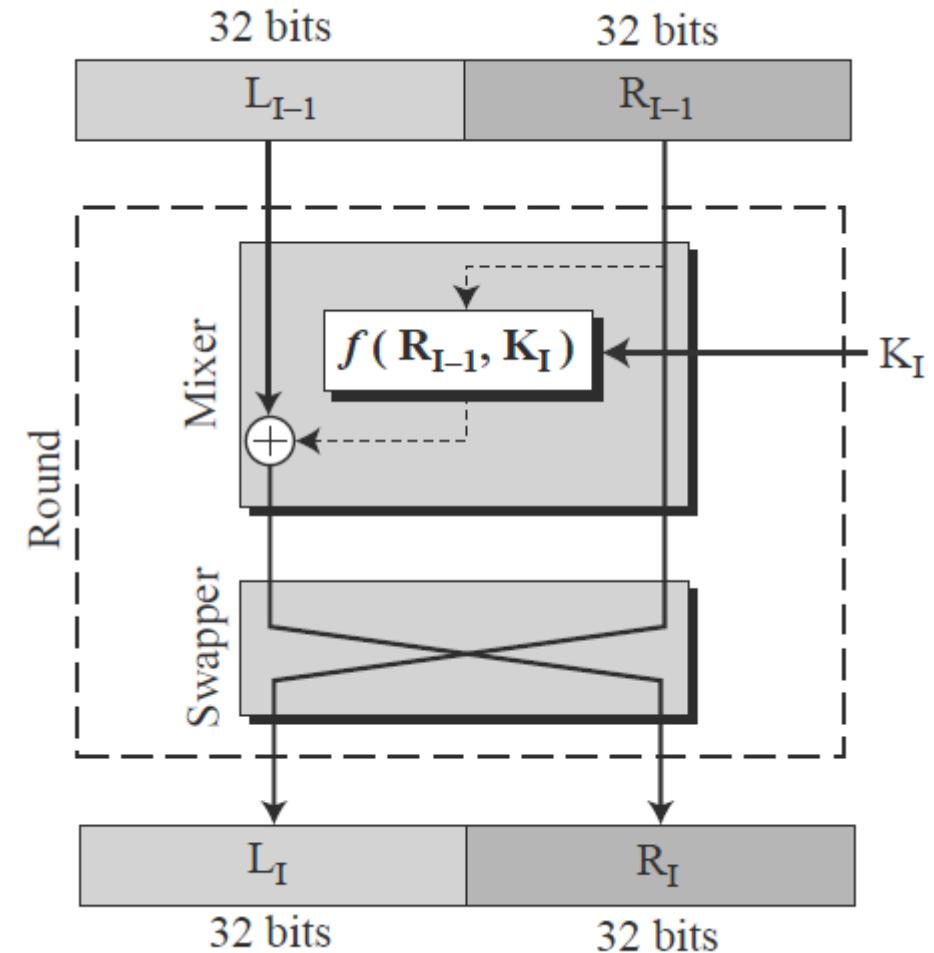
Only bit 25 and bit 64 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

DESRounds (contd..)

DES Rounds:

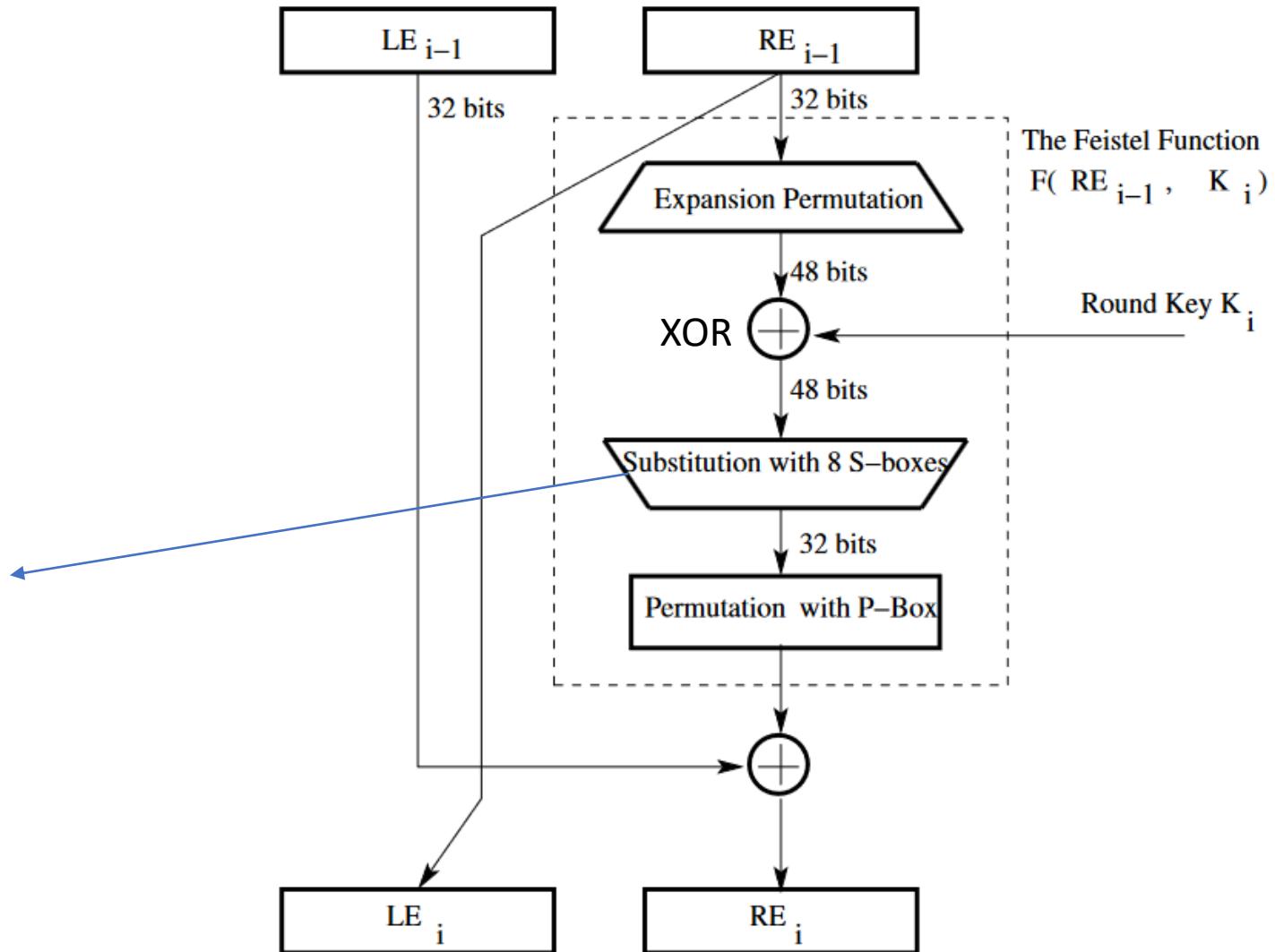
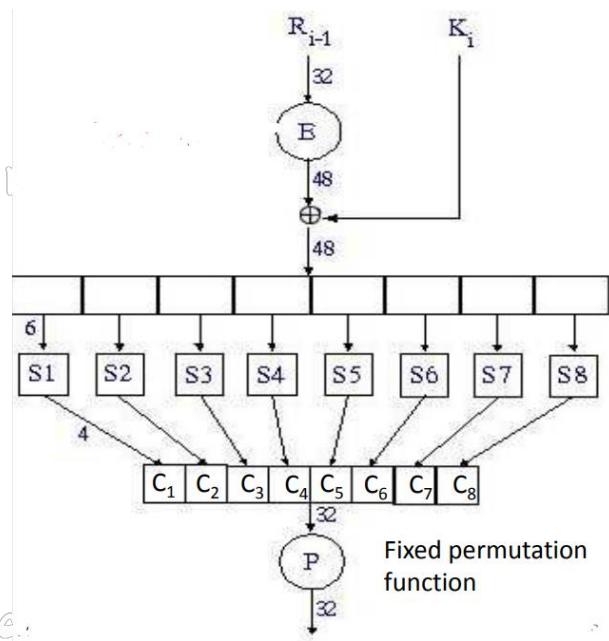
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$



A round in DES (encryption site)

DES Rounds (contd..)

DES “ $f(\bullet)$ ” Function:



DESRounds (contd..)

DES “f(•)” Function:

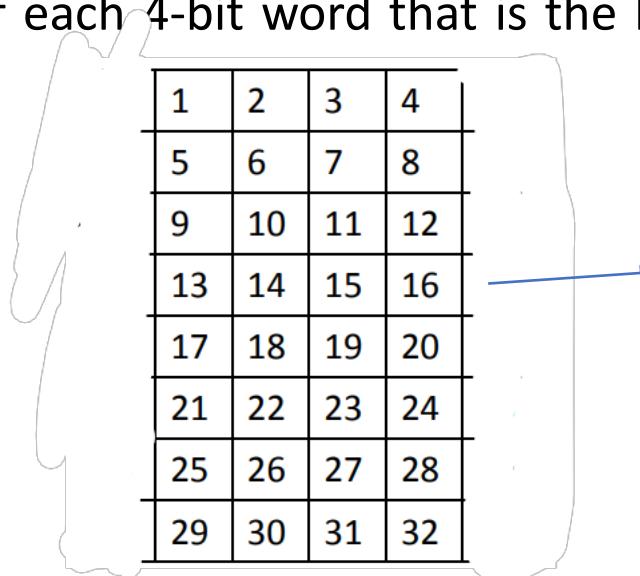
Expansion Function (E):

E is an expansion function which takes a block of 32 bits as input and produces a block of 48 bits as output

E-step entails the following:

- first divide the 32-bit block into eight 4-bit words
- attach an additional bit on the left to each 4-bit word that is the last bit of the previous 4-bit word
- attach an additional bit to the right of each 4-bit word that is the beginning bit of the next 4-bit word.

Note that what gets prefixed to the first 4-bit block is the last bit of the last 4-bit block. By the same token, what gets appended to the last 4-bit block is the first bit of the first 4-bit block



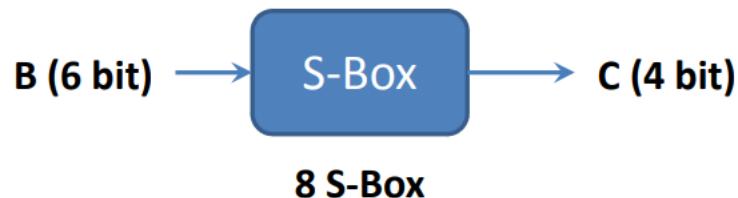
DES Rounds (contd..)

DES “ $f(\bullet)$ ” Function:

The S-Box for the Substitution Step in Each Round

S-box is to introduce diffusion in the generation of the output from the input.

The 48-bit input word is divided into eight 6-bit words and each 6-bit word fed into a separate S-box. Each S-box produces a 4-bit output. Therefore, the 8 S-boxes together generate a 32-bit output.



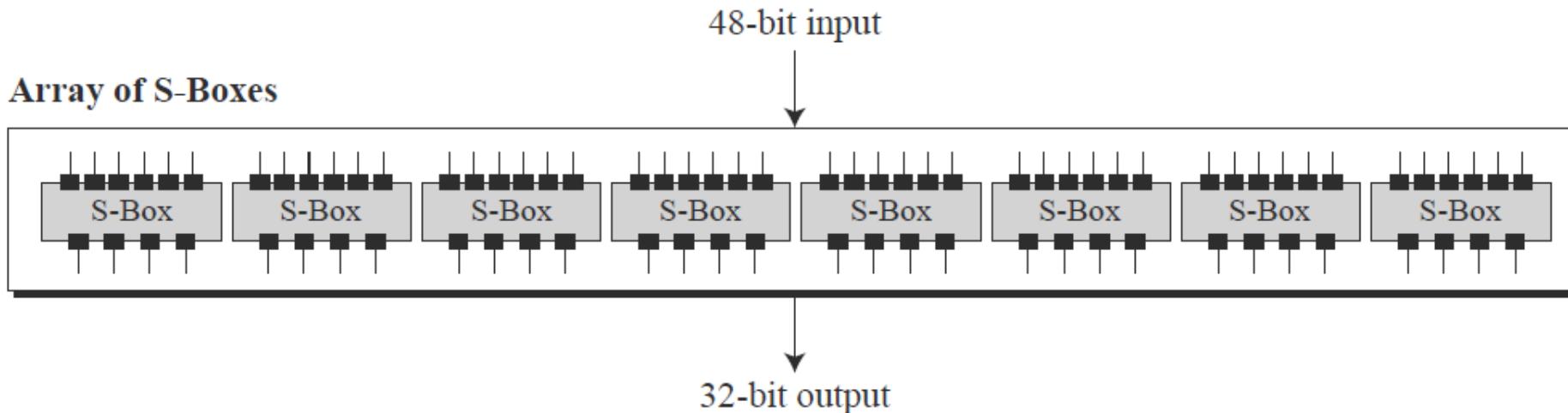
Each of the eight S-boxes consists of a 4×16 table lookup for an output 4-bit word.

- $S = \text{matrix } 4 \times 16$, values from 0 to 15
- B (6 bit long) = $b_1 b_2 b_3 b_4 b_5 b_6$
 - $b_1 b_6$ $\rightarrow r = \text{row of the matrix (2 bits: 0,1,2,3)}$
 - $b_2 b_3 b_4 b_5$ $\rightarrow c = \text{column of the matrix (4 bits: 0,1,...15)}$
- C (4 bit long) = Binary representation of $S(r, c)$

DES Rounds (contd..)

DES “ $f(\bullet)$ ” Function:

There are 8 S-boxes



Each row in all eight tables is a random permutation of the 16 integers, 0 through 15, and no two permutations are the same in all of the eight tables taken together.

DES Rounds (contd..)

DES “f(•)” Function(Feistel Function):

Example S-boxes

Row #	S ₁	1	2	3	...	7								15	Column #	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S(i, j) < 16, can be represented with 4 bits

Example: B = 101111

b₁b₆ = 11 = row 3

b₂b₃b₄b₅ = 0111 = column 7



C=7=0111

Assignment: B=011011, C=?

The 4×16 substitution table for S-box S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S-box S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S-box S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S-box S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S-box S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S-box S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S-box S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S-box S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Example

The input to S-box 1 is 100011. What is the output?

Solution If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

The input to S-box 8 is 000000. What is the output?

Solution If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

DATA ENCRYPTION STANDARD (DES) (contd..)

DES Rounds:

The P-Box Permutation in the Feistel Function

- The last operation in the Feistel function is a permutation with a “Permutation with P-Box”---- 32-bit input and a 32-bit output.
- The input/output relationship for this operation is shown in Table and follows the same general rule as previous tables.

P-Box Permutation							
15	6	19	20	28	11	27	16
0	14	22	25	4	17	30	9
1	7	23	13	31	26	2	8
18	12	29	5	21	10	3	24

This permutation ‘table’ says that the 0th output bit will be the 15th bit of the input, the 1st output bit the 6th bit of the input, and so on, for all of the 32 bits of the output that are obtained from the 32 bits of the input.

DES Rounds (contd..)

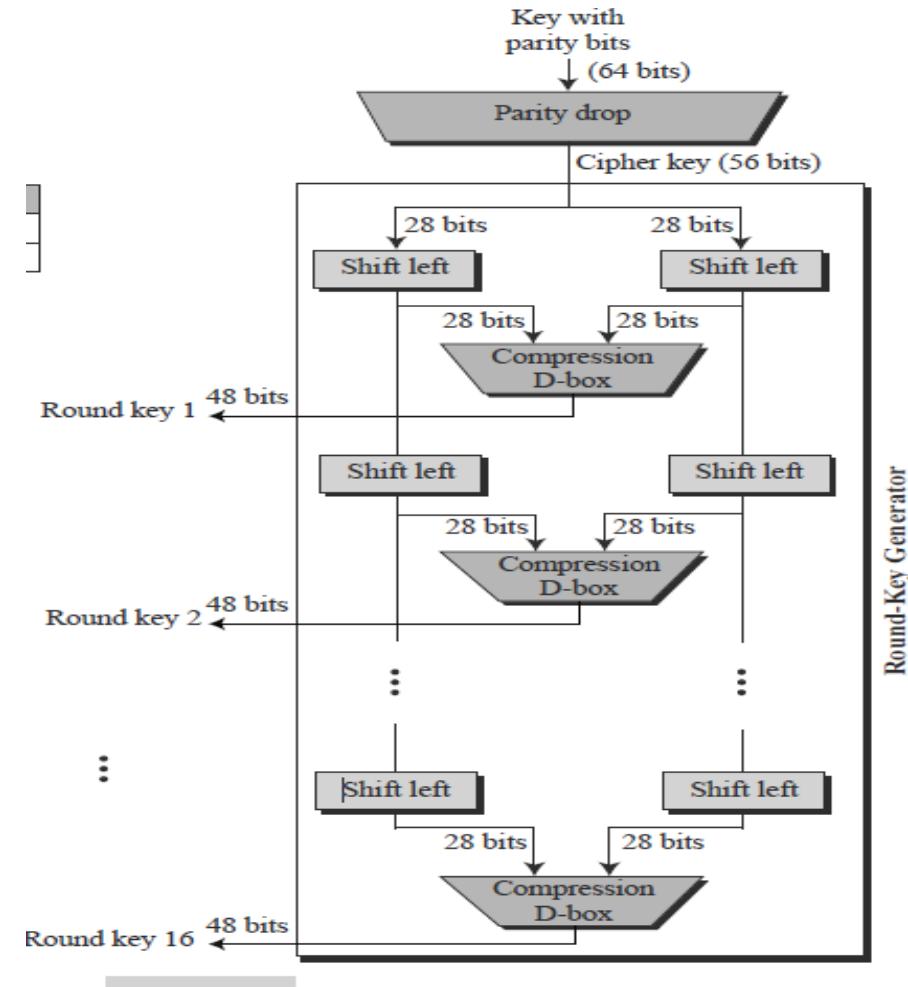
The DES Key Schedule: Generating the Round Keys

The **round-key generator** creates sixteen 48-bit keys out of a 56-bit cipher key.

However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

At the beginning of each round, we divide the 56 relevant key bits into two 28 bit halves and circularly shift to the left each half by one or two bits, depending on the round, as shown in the table

Shifting	
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



Round-Key Generator

DES Rounds (contd..)

The DES Key Schedule: Generating the Round Keys

- The 56-bit encryption key is represented by 8 bytes, with the last bit (the least significant bit) of each byte used as a parity bit.
- The relevant 56 bits are subject to a permutation at the beginning before any round keys are generated. This is referred to as **Key Permutation 1 (PC1)**
- At the beginning of each round, we divide the 56 relevant key bits into two 28 bit halves and circularly shift to the left each half by one or two bits, depending on the round,
- For generating the round key, we join together the two halves and apply a 56 bit to 48 bit contracting permutation this is referred to as **Key Permutation 2 (PC2)**, The resulting 48 bits constitute round key.
- The contraction permutation in **Key Permutation 2**, along with the one-bit or two-bit rotation of the two key halves prior to each round, is meant to ensure that each bit of the original encryption key is used in roughly 14 of the 16 rounds.

DES Rounds (contd..)

The DES Key Schedule: Generating the Round Keys

Parity-check bits (namely, bits 8,16, 4,32,40,48,56,64) are not chosen, they do not appear in PC-1



14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32



PC-2 selects the 48-bit subkey for each round from the 56-bit key-schedule state

Left							
57	49	41	33	25	17	9	
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
Right							
63	55	47	39	31	23	15	
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

The permutation order for the bits is given by reading the entries shown from the upper left corner to the lower right corner.

This permutation tells us that the 0th bit of the output will be the 57th bit of the input (in a 64 bit representation of the 56-bit encryption key), the 1st bit of the output the 49th bit of the input, and so on

THE STRENGTH OF DES

1. The Use of 56-Bit Keys

- With a key length of 56 bits, there are 256 possible keys, which is approximately 7.2×10^{16} keys. Thus, on the face of it, a brute-force attack appears impractical.
- Assuming that, on average, half the key space has to be searched, a single machine

2. Timing attack

- The authors conclude that DES appears to be fairly resistant to a successful timing attack but suggest some avenues to explore.
- A timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts. A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.

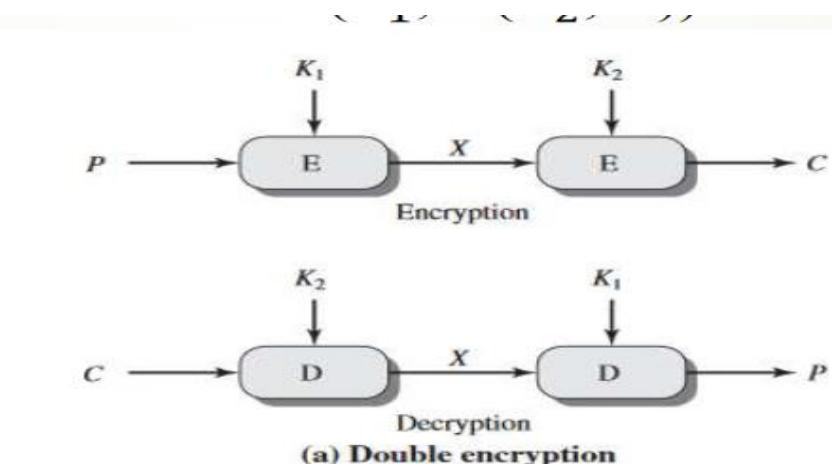
THE STRENGTH OF DES

3. The Nature of the DES Algorithm

Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm. The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent.

Double DES

- DES uses a 56-bit key, this raised concerns about brute force attacks.
- One proposed solution: double DES.
- Apply DES twice using two keys, K1 and K2.
- – Encryption: $C = E_{K_2} [E_{K_1} [P]]$
- – Decryption: $P = D_{K_2} [D_{K_1} [C]]$
- This leads to a $2 \times 56 = 112$ bit key, so it is more secure than DES



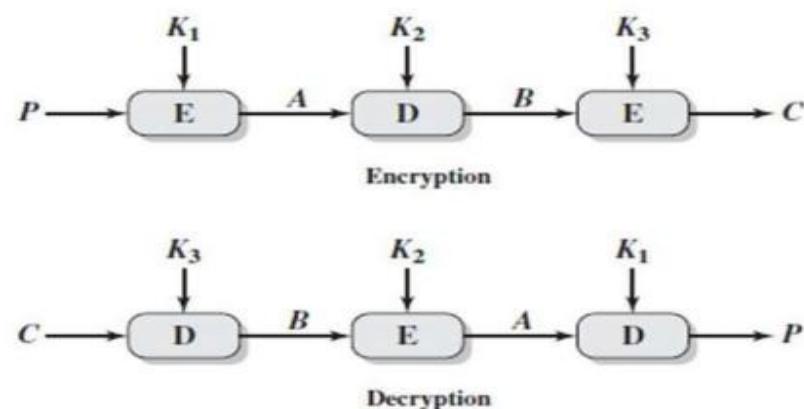
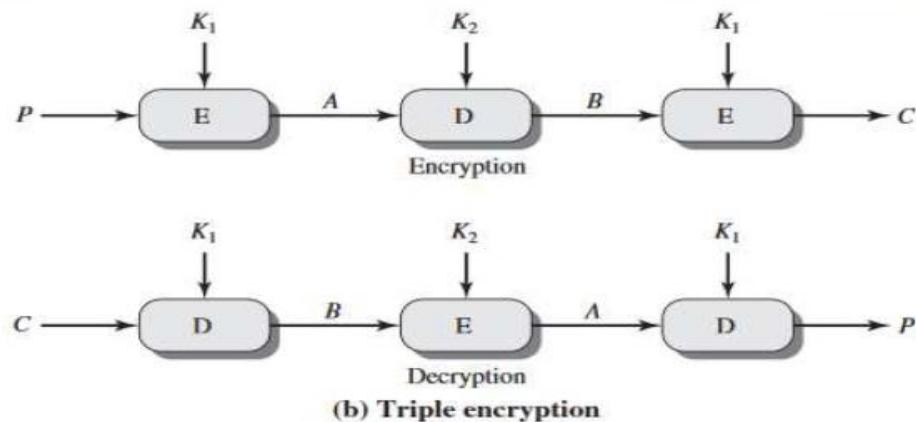
Triple DES

$$C = E(K_3, D(K_2, E(K_1, P)))$$

The encryption-decryption process is as follows –

- Encrypt the plaintext blocks using single DES with key K₁.
- Now decrypt the output of step 1 using single DES with key K₂.
- Finally, encrypt the output of step 2 using single DES with key K₃.
- The output of step 3 is the ciphertext.
- Decryption of a ciphertext is a reverse process. User first decrypt using K₃, then encrypt with K₂, and finally decrypt with K₁.

Triple DES



Triple DES with 2-key

The function follows an encrypt-decrypt-encrypt (EDE) sequence.

$$C = E(K_1, D(K_2, E(K_1, P)))$$

$$P = D(K_1, E(K_2, D(K_1, C)))$$

Triple DES with 3-key

3-key 3DES has an effective key length of 168 bits and is defined as

$$C = E(K_3, D(K_2, E(K_1, P)))$$

$$P = D(K_1, E(K_2, D(K_3, C)))$$

Example (DES Round key generation)

- Generate round-1 key using the key in Hex K=0x133457799BBCDFF1
- Given data: PC1 and PC2=

Left							
57	49	41	33	25	17	9	
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
Right							
63	55	47	39	31	23	15	
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

- Solution:
- Step1: Convert Hex number to 64 bit binary format
- 00010011 00110100 01010111 01111001 10011011 10111100
11011111 11110001

Example:

- Step2: Remove parity bits and apply PC1

Hex digits	Bit position	bits	bits	bits	bits	bi s	bits	bits	bits
13	1-2-3-4-5-6-7-8	0	0	0	1	0	0	1	1
34	9-10-11-12-13-14-15-16	0	0	1	1	0	1	0	0
57	17-18-19-20-21-22-23-24	0	1	0	1	0	1	1	1
79	25-26-27-28-29-30-31-32	0	1	1	1	1	0	0	1
9B	33-34-35-36-37-38-39-40	1	0	0	1	1	0	1	1
BC	41-42-43-44-45-46-47-48	1	0	1	1	1	0	0	0
DF	49-50-51-52-53-54-55-56	1	1	0	1	1	1	1	1
F1	57-58-59-60-61-62-63-64	1	1	1	1	0	0	0	1

- Parity bits
- Apply PC1: we get 56 bit key permutation
- 1111000 0110011 0010101 0101111 0101010 1010001 1001111 0001111

Left							
57	49	41	33	25	17	9	
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
Right							
63	55	47	39	31	23	15	
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	1
0	0	1	0	1	0	1	1
0	1	0	1	1	1	1	1
0	1	0	1	0	1	0	0
1	0	1	0	0	0	0	1
1	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1

Example

- **Step3:** Split this key into left and right halves, C_0 and D_0 , where each half has 28 bits.
- $C_0 = 1111000\ 0110011\ 0010101\ 0101111$
 $D_0 = 0101010\ 1011001\ 1001111\ 0001111$
- **Step4:** Left shift → Round 1 i-shift
- $C_0 = 111000\ 0110011\ 0010101\ 0101111$
 $D_0 = 101010\ 1011001\ 1001111\ 0001110$

Step5: Apply PC2

1	1	1	0	0	0	0	0
1	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	0	1	0	1	0	1	1
0	1	0	0	0	1	1	1
0	0	1	1	1	1	0	
0	0	1	1	1	1	1	0

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

1-7
8-14
15-21
22-28
29-35
36-42
43-49
50-56

1	1	1	0	0	0	0
1	1	0	0	1	1	0
0	1	0	1	0	1	0
1	0	1	1	1	1	1
1	0	1	0	1	0	1
0	1	0	0	0	1	1
0	0	1	1	1	1	0
0	0	1	1	1	1	0



0	0	0	1	1	0	1	1
0	0	0	0	0	0	1	0
1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	1	0

$K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001$
 $110010 \rightarrow 48 \text{ bit key}$

$$C_0 = 111100001100110010101010101111$$

$$D_0 = 0101010101100110011110001111$$

$$C_1 = 1110000110011001010101011111$$

$$D_1 = 1010101011001100111100011110$$

$$C_2 = 1100001100110010101010111111$$

$$D_2 = 0101010110011001111000111101$$

$$C_3 = 000011001100101010101111111$$

$$D_3 = 0101011001100111100011110101$$

$$C_4 = 001100110010101010111111100$$

$$D_4 = 0101100110011110001111010101$$

$$C_5 = 1100110010101010111111110000$$

$$D_5 = 0110011001111000111101010101$$

$$C_6 = 0011001010101011111111000011$$

$$D_6 = 1001100111100011110101010101$$

$$C_7 = 110010101010111111100001100$$

$$D_7 = 0110011110001111010101010110$$

$$C_8 = 001010101011111110000110011$$

$$D_8 = 1001111000111101010101011001$$

$$C_9 = 010101010111111100001100110$$

$$D_9 = 0011110001111010101010110011$$

$$C_{10} = 010101011111110000110011001$$

$$D_{10} = 1111000111101010101011001100$$

$$C_{11} = 010101111111000011001100101$$

$$D_{11} = 1100011110101010101100110011$$

$$C_{12} = 010111111100001100110010101$$

$$D_{12} = 0001111010101010110011001111$$

$$C_{13} = 011111110000110011001010101$$

$$D_{13} = 0111101010101011001100111100$$

$$C_{14} = 111111000011001100101010101$$

$$D_{14} = 1110101010101100110011110001$$

$$C_{15} = 1111100001100110010101010111$$

$$D_{15} = 1010101010110011001111000111$$

$$C_{16} = 1111000011001100101010101111$$

$$D_{16} = 0101010101100110011110001111$$

$$K_2 = 011110\ 011010\ 111011\ 011001\ 110110\ 111100\ 100111\ 100101$$

$$K_3 = 010101\ 011111\ 110010\ 001010\ 010000\ 101100\ 111110\ 011001$$

$$K_4 = 011100\ 101010\ 110111\ 010110\ 110110\ 110011\ 010100\ 011101$$

$$K_5 = 011111\ 001110\ 110000\ 000111\ 111010\ 110101\ 001110\ 101000$$

$$K_6 = 011000\ 111010\ 010100\ 111110\ 010100\ 000111\ 101100\ 101111$$

$$K_7 = 111011\ 001000\ 010010\ 110111\ 111101\ 100001\ 100010\ 111100$$

$$K_8 = 111101\ 111000\ 101000\ 111010\ 110000\ 010011\ 101111\ 111011$$

$$K_9 = 111000\ 001101\ 101111\ 101011\ 111011\ 011110\ 011110\ 000001$$

$$K_{10} = 101100\ 011111\ 001101\ 000111\ 101110\ 100100\ 011001\ 001111$$

$$K_{11} = 001000\ 010101\ 111111\ 010011\ 110111\ 101101\ 001110\ 000110$$

$$K_{12} = 011101\ 010111\ 000111\ 110101\ 100101\ 000110\ 011111\ 101001$$

$$K_{13} = 100101\ 111100\ 010111\ 010001\ 111110\ 101011\ 101001\ 000001$$

$$K_{14} = 010111\ 110100\ 001110\ 110111\ 111100\ 101110\ 011100\ 111010$$

$$K_{15} = 101111\ 111001\ 000110\ 001101\ 001111\ 010011\ 111100\ 001010$$

$$K_{16} = 110010\ 110011\ 110110\ 001011\ 000011\ 100001\ 011111\ 110101$$

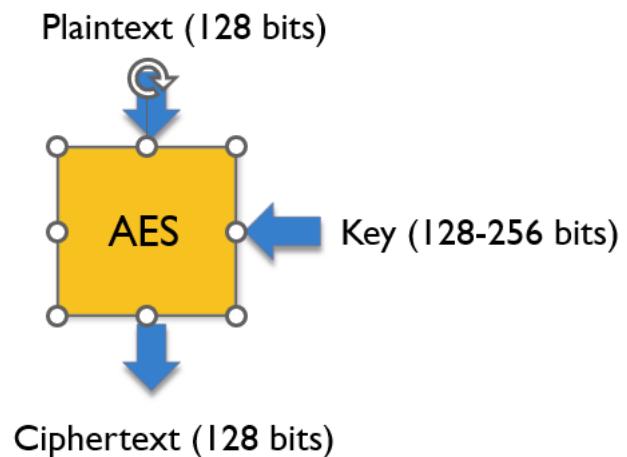
Data Security and Privacy

DSE 3258

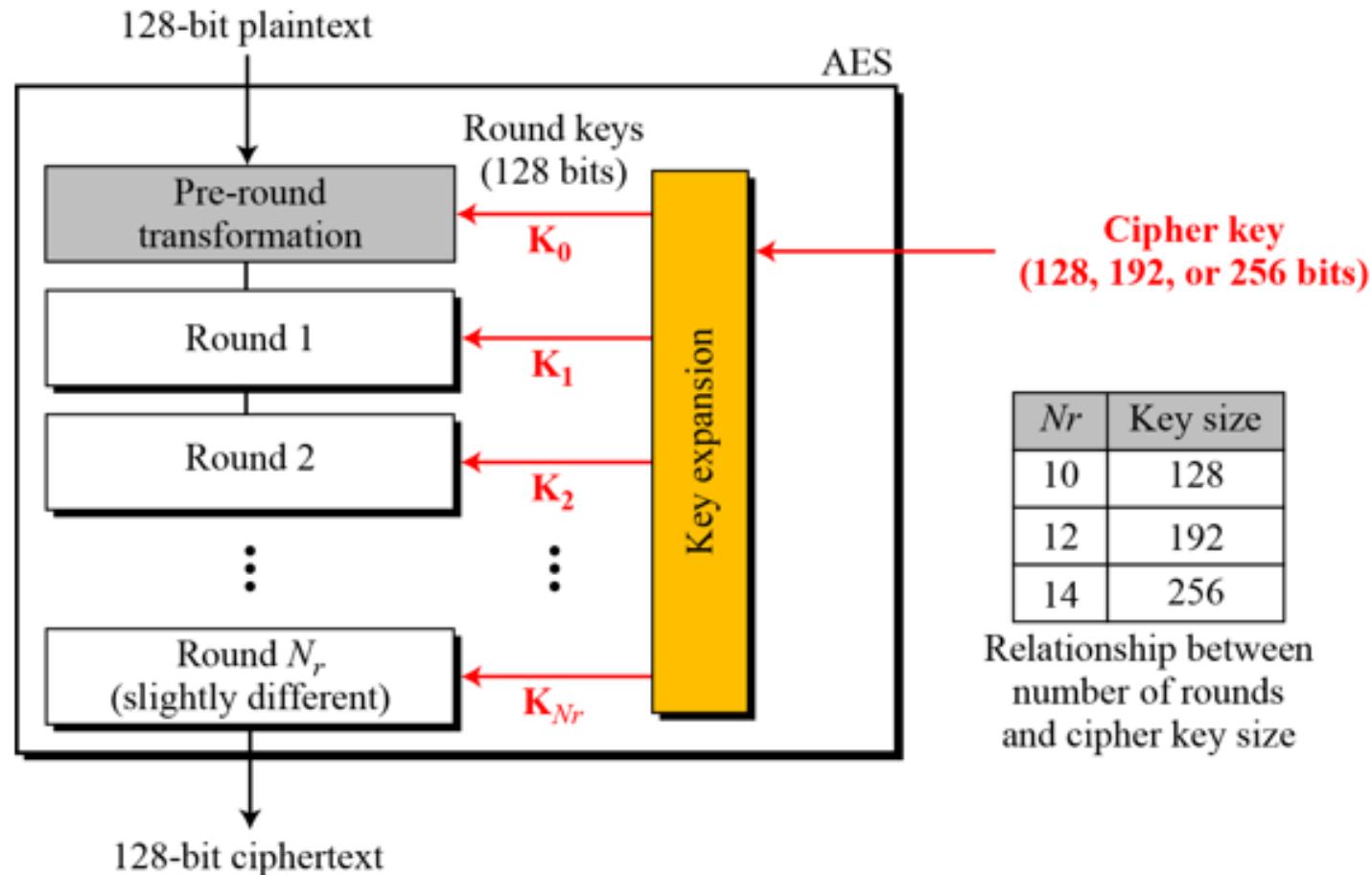
L5 -BLOCK CIPHER
**Advanced Encryption Standard
(AES)**

Introduction

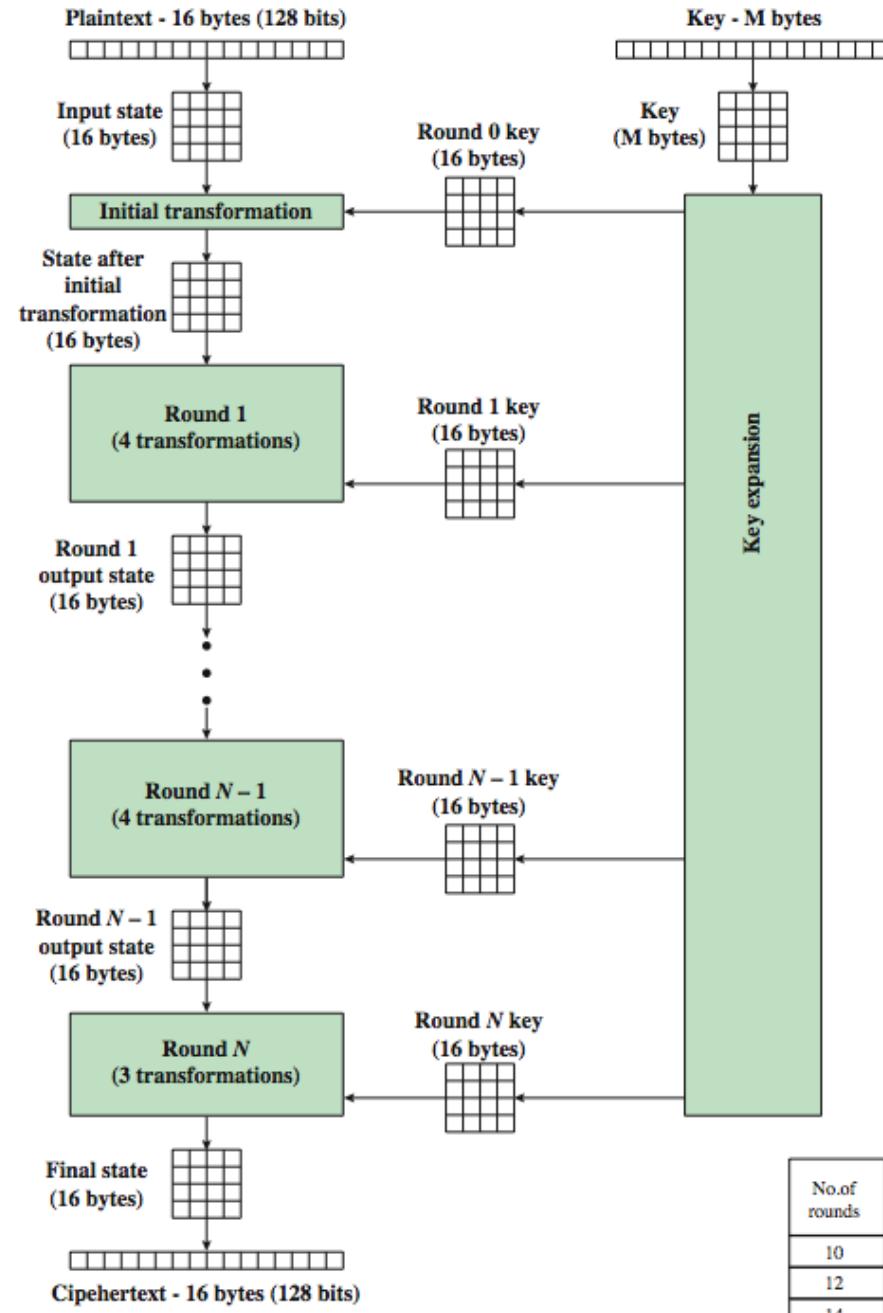
- Advanced Encryption Standard
- Symmetric block cipher
- Designed by Rijmen-Daemen in Belgium
- has 128/192/256 bit keys, 128 bit data
- The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length
- an iterative rather than Feistel cipher
 - processes data as block of 4 columns of 4 bytes
 - operates on entire data block in every round
- Designed to have:
 - resistance against known attacks
 - speed and code compactness on many CPUs
 - design simplicity



AES Encryption Process



AES Encryption Process (Detailed View)



No.of rounds	Key Length (bytes)
10	16
12	24
14	32

High Level Description

Key Expansion

- Round keys are derived from the cipher key using Rijndael's key schedule

Initial Round

- AddRoundKey : Each byte of the state is combined with the round key using bitwise xor

Rounds

- SubBytes : non-linear substitution step
- ShiftRows : transposition step
- MixColumns : mixing operation of each column.
- AddRoundKey

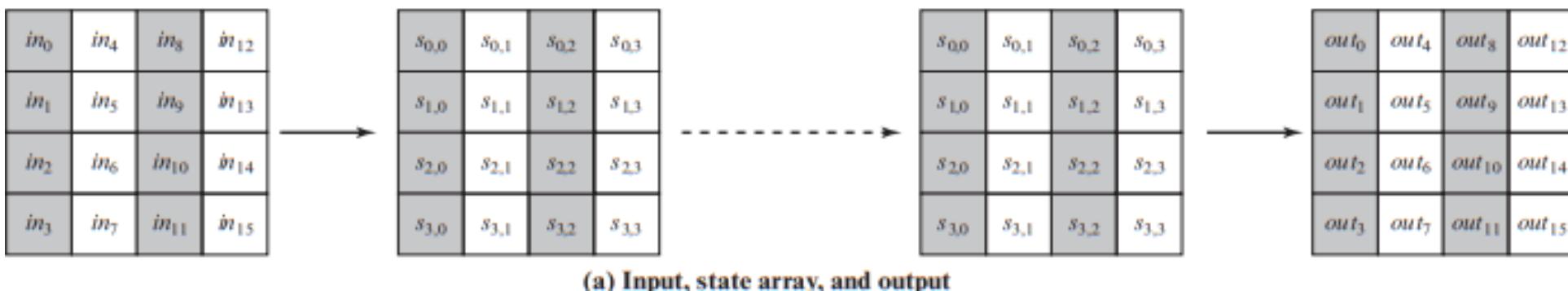
Final Round

- SubBytes
- ShiftRows
- AddRoundKey

No MixColumns

AES Encryption Process (Cont..)

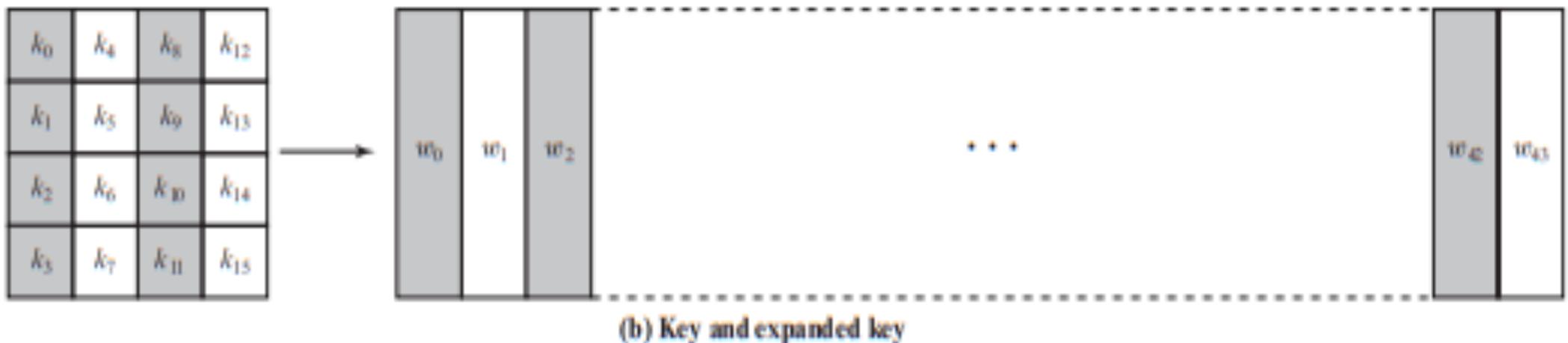
- The input to the encryption and decryption algorithms is a single 128-bit block which is depicted as a $4 * 4$ square matrix of bytes.
- Input block is copied into the **State** array, which is modified at each stage of encryption or decryption. At the final stage, **State** is copied to an output matrix.



- The key is also depicted as a square matrix of bytes. Later, this key is expanded into an array of key schedule words.

AES Encryption Process (Cont..)

- The key is also depicted as a square matrix of bytes. Later, this key is expanded into an array of key schedule words.
- For 128 bit key, each word is four bytes, and the total key schedule is 44 words for the 128-bit key.



AES Encryption Process (Cont..)

- The ordering of bytes within a matrix is by column.
- So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the in matrix, the second four bytes occupy the second column, and so on.
- Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix.
- The cipher consists of N rounds, where the number of rounds depends on the key length.
- The first N - 1 rounds consist of four distinct transformation functions: **SubBytes**, **ShiftRows**, **MixColumns**, and **AddRoundKey**.
- The final round contains only three transformations, and there is a initial single transformation (**AddRoundKey**) before the first round, which can be considered Round 0.
- Each transformation takes one or more 4 * 4 matrices as input and produces a 4 * 4 matrix as output.
- The key expansion function generates N + 1 round keys, each of which is a distinct 4 * 4 matrix. Each round key serves as one of the inputs to the **AddRoundKey transformation** in each round.

AES Encryption Process (Cont..)

Table 6.1 AES Parameters

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

AES Detailed Structure

1. One noteworthy feature of this structure is that it is not a Feistel structure. AES processes the entire data block as a single matrix during each round using substitutions and permutation.
2. The key that is provided as input is expanded into an array of forty-four 32-bit words, $w[i]$. Four distinct words (128 bits) serve as a round key for each round.
3. Four different stages are used, one of permutation and three of substitution:
 - a. Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block.
 - b. ShiftRows: A simple permutation.
 - c. MixColumns: A substitution that makes use of matrix multiply of groups
 - d. AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key
4. For both encryption and decryption, the cipher begins with an **AddRoundKey** stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages.
5. Only the **AddRoundKey** stage makes use of the key. For this reason, the cipher begins and ends with an **AddRoundKey** stage.

AES Detailed Structure

6. The **AddRoundKey** stage is, in effect, a form of Vernam cipher and by itself would not be formidable. The other three stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key..
7. Each stage is easily reversible. For the **Substitute Byte**, **ShiftRows**, and **MixColumns** stages, an **inverse function** is used in the decryption algorithm.
8. As with most block ciphers, the decryption algorithm makes use of the expanded key in reverse order. However, the decryption algorithm is not identical to the encryption algorithm.
10. The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required to make the cipher reversible

AES Inner Workings of a Round

- The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages.
- This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of it's counterpart in the encryption algorithm. The four stages are as follows:
 - 1. Substitute bytes
 - 2. Shift rows
 - 3. Mix Columns
 - 4. Add Round Key

AES Inner Workings of a Round

- **Substitute Bytes**
- The forward substitute byte transformation, called SubBytes.
- It is a table lookup using a 16×16 matrix of byte values called an s-box.
- This matrix consists of all the possible combinations of an 8 bit sequence ($2^8 = 16 \times 16 = 256$). However, the s-box is not just a random permutation of these values and there is a well defined creating the s-box tables.
- Again the matrix that gets operated upon throughout the encryption is known as **state**.
- In a round each byte is mapped into a new byte in the following way: the leftmost nibble of the byte is used to specify a particular row of the s-box and the rightmost nibble specifies a column. For example, the byte {95} (curly brackets represent hex values) selects row 9 column 5 which turns out to contain the value {2A}.
- This is then used to update the state matrix.
- The Inverse substitute byte transformation (known as InvSubBytes) makes use of an inverse s-box. In this case what is desired is to select the value {2A} and get the value {95}.

AES Inner Workings of a Round

- Substitute Bytes

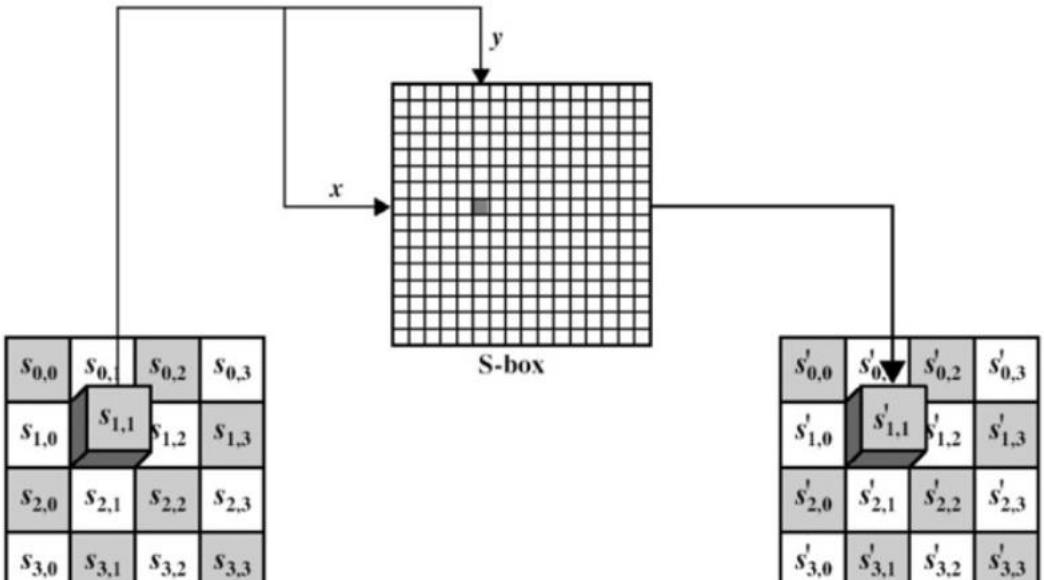


Figure 7.3: Substitute Bytes Stage of the AES algorithm.

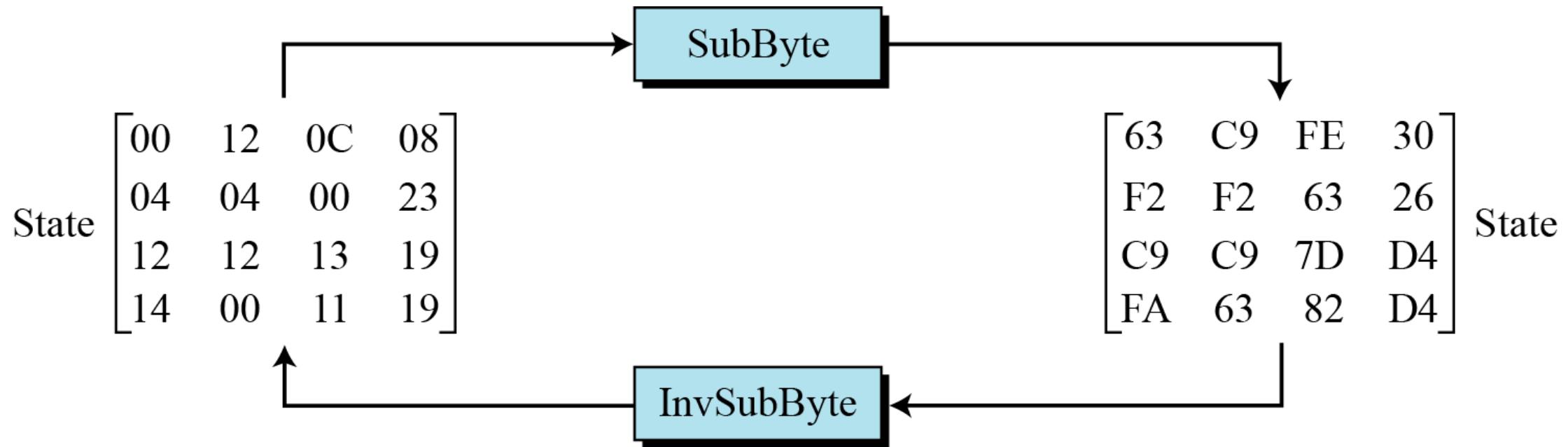
$b_0 b_1 b_2 b_3, b_4 b_5 b_6 b_7$
↓ ↓
Row index Column index.
S-box.

SubBytes Table

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

InvSubBytes Table

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D



AES Inner Workings of a Round

- **Shift Row Transformation**

It works as follow:

- The first row of state is not altered.
- The second row is shifted 1 bytes to the left in a circular manner.
- The third row is shifted 2 bytes to the left in a circular manner.
- The fourth row is shifted 3 bytes to the left in a circular manner

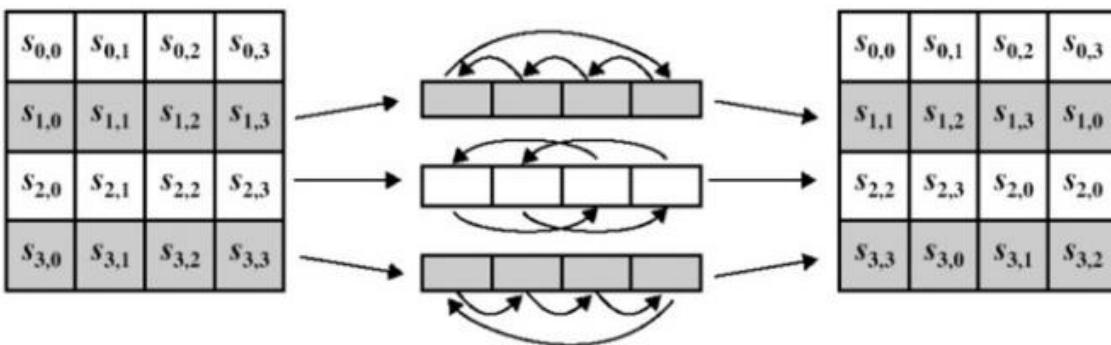
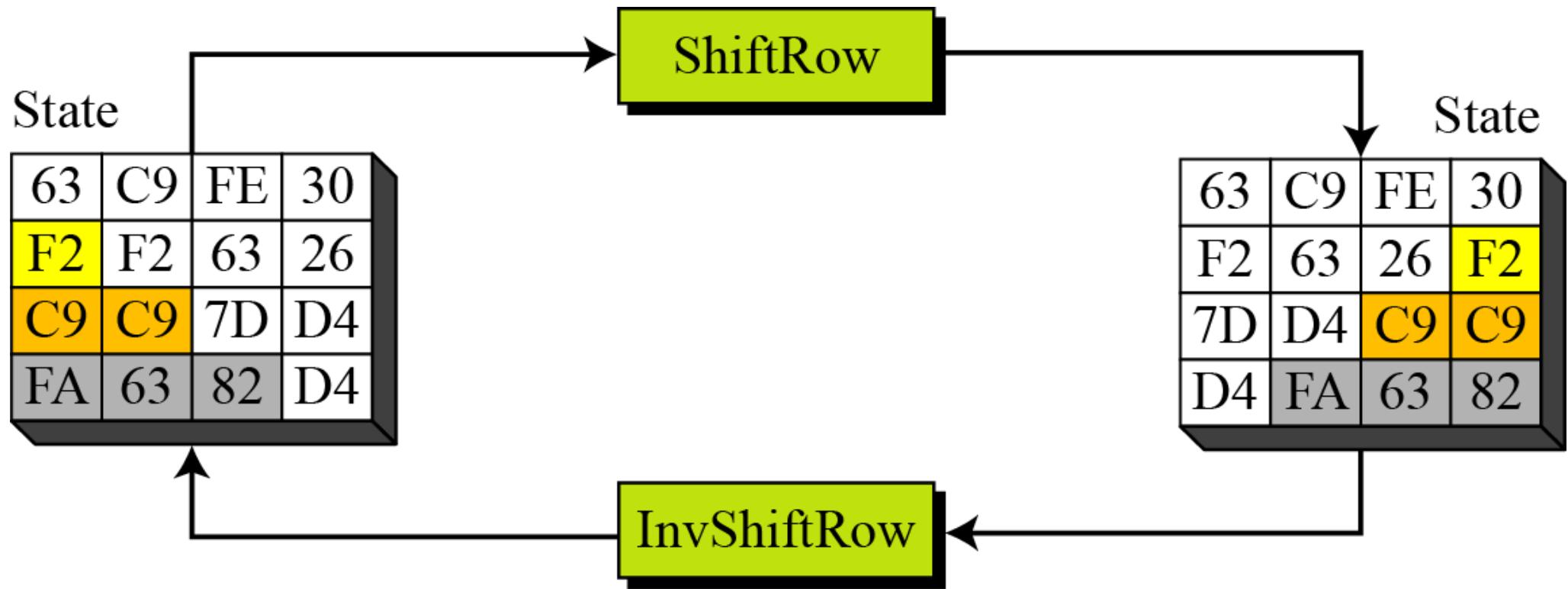


Figure 7.5: ShiftRows stage.

The **Inverse Shift Rows** transformation (known as InvShiftRows) performs these circular shifts in the opposite direction for each of the last three rows (the first row was unaltered to begin with)



AES Inner Workings of a Round

- **Mix Column Transformation**

This stage (known as MixColumn) is basically a substitution but it makes use of arithmetic of $GF(2^8)$. (Galois Field)

Each column is operated on individually.

Each byte of a column is mapped into a new value that is a **function of all four bytes** in the column.

Effectively a matrix multiplication in $GF(2^8)$ using prime poly $m(x) = x^8 + x^4 + x^3 + x + 1$

The transformation can be determined by the following matrix multiplication on state:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Each element of the product matrix is the sum of products of elements of one row and one column.

In this case the individual additions and multiplications are performed in $GF(2^8)$.

AES Inner Workings of a Round

• Mix Column Transformation

The MixColumns transformation of a single column j ($0 \leq j \leq 3$) of state can be expressed as:

$$\begin{aligned}s'_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\s'_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\s'_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})\end{aligned}$$

where \bullet denotes multiplication over the finite field $\text{GF}(2^8)$.

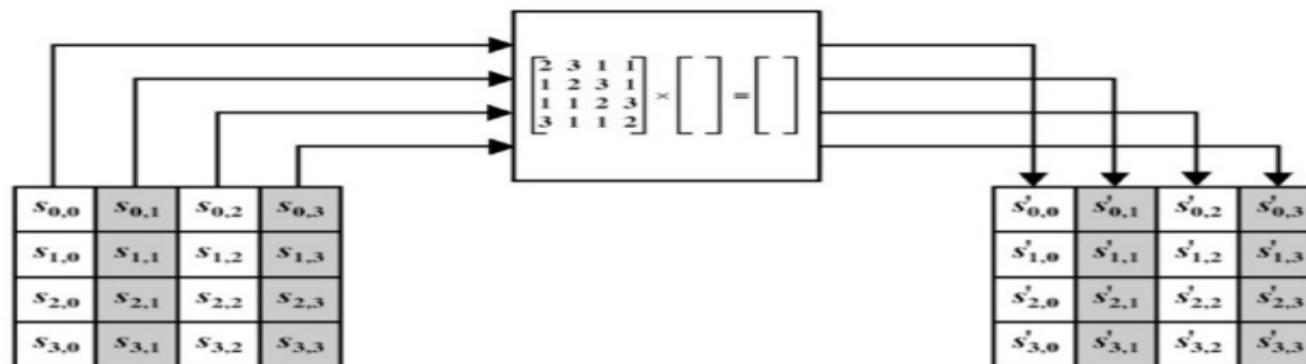


Figure 7.6: MixColumns stage.

AES Inner Workings of a Round

MixColumn and InvMixColumn

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \xleftarrow{\text{Inverse}} \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

The **inverse mix column transformation**, called InvMixColumns, is defined by the following matrix multiplication:

$$\begin{bmatrix} \text{OE} & \text{OB} & \text{OD} & \text{09} \\ \text{09} & \text{0E} & \text{OB} & \text{0D} \\ \text{0D} & \text{09} & \text{0E} & \text{OB} \\ \text{OB} & \text{0D} & \text{09} & \text{0E} \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \quad (6.5)$$

Example

$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$	<table border="1"> <tr><td>87</td><td>F2</td><td>4D</td><td>97</td></tr> <tr><td>6E</td><td>4C</td><td>90</td><td>EC</td></tr> <tr><td>46</td><td>E7</td><td>4A</td><td>C3</td></tr> <tr><td>A6</td><td>8C</td><td>D8</td><td>95</td></tr> </table>	87	F2	4D	97	6E	4C	90	EC	46	E7	4A	C3	A6	8C	D8	95	\rightarrow	<table border="1"> <tr><td>47</td><td>40</td><td>A3</td><td>4C</td></tr> <tr><td>37</td><td>D4</td><td>70</td><td>9F</td></tr> <tr><td>94</td><td>E4</td><td>3A</td><td>42</td></tr> <tr><td>ED</td><td>A5</td><td>A6</td><td>BC</td></tr> </table>	47	40	A3	4C	37	D4	70	9F	94	E4	3A	42	ED	A5	A6	BC
87	F2	4D	97																																
6E	4C	90	EC																																
46	E7	4A	C3																																
A6	8C	D8	95																																
47	40	A3	4C																																
37	D4	70	9F																																
94	E4	3A	42																																
ED	A5	A6	BC																																

$$(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

$$\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\}$$

$$\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\}$$

$$(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\}$$

For the first equation, we have $\{02\} \cdot \{87\} = (0000\ 1110) \oplus (0001\ 1011) = (0001\ 0101)$ and $\{03\} \cdot \{6E\} = \{6E\} \oplus (\{02\} \cdot \{6E\}) = (0110\ 1110) \oplus (1101\ 1100) = (1011\ 0010)$. Then,

$$\{02\} \cdot \{87\} = 0001\ 0101$$

$$\{03\} \cdot \{6E\} = 1011\ 0010$$

$$\{46\} = 0100\ 0110$$

$$\{A6\} = \underline{1010\ 0110}$$

$$0100\ 0111 = \{47\}$$

$$\begin{bmatrix} 0_2 & 0_3 & 0_1 & 0_1 \\ 0_1 & 0_2 & 0_3 & 0_1 \\ 0_1 & 0_1 & 0_2 & 0_3 \\ 0_3 & 0_1 & 0_1 & 0_2 \end{bmatrix} \times \underbrace{\begin{bmatrix} S_{00} \\ S_{01} \\ S_{02} \\ S_{03} \end{bmatrix}}_{W_1} = \begin{bmatrix} S_{00}' & S_{10}' & S_{20}' & S_{30}' \\ S_{01}' & S_{11}' & S_{21}' & S_{31}' \\ S_{02}' & S_{12}' & S_{22}' & S_{32}' \\ S_{03}' & S_{13}' & S_{23}' & S_{33}' \end{bmatrix}$$

$$(0_2 \cdot S_{00}) \Downarrow \oplus (0_3 \cdot S_{01}) \Downarrow \oplus (0_1 \cdot S_{02}) \oplus (0_1 \cdot S_{03})$$

\therefore Galois Field based multiplication
 $GF(2^8)$



$$1 = 1$$

$$2 = \alpha$$

$$3 = \alpha + 1$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 87 \\ 6E \\ 46 \\ A6 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$$

$$(02 \cdot 87) \oplus (03 \cdot 6E) \oplus (01 \cdot 46) \oplus \underline{(01 \cdot A6)}$$

$$\begin{aligned} (02 \cdot 87) &= D(x^{1000} 0111) \\ &= D(x^7 + x^2 + x' + 1) \\ &= \underline{\underline{D(x^8 + x^3 + D(x^2 + D(x))}}} \end{aligned}$$

$\overline{x^8 \rightarrow \text{not possible}}$
as we have
 $x^0 + x^7$
 $b^2 + s$

So x^8 should be reduced.

By factorizing we get

$$x^8 = x^4 + x^3 + x + 1$$

so replace x^8 \uparrow

$$x^4 + \cancel{x^3} + \cancel{x} + 1 + \cancel{x^2} + \cancel{x}$$

$$= x^4 + x^2 + 1$$

$$= 00010101$$

$$\begin{aligned}
 (63 \cdot 6E) &= (x+1)(0110 \ 1110) \\
 &= (x+1)(x^6 + x^5 + x^3 + x^2 + x) \\
 &= x^7 + x^6 + x^4 + x^3 + x^2 + \\
 &\quad \cancel{x^6} + x^5 + \cancel{x^3} + \cancel{x^2} + x \\
 &= x^7 + x^5 + x^4 + x \\
 \Rightarrow &= \underline{\underline{10110010}}
 \end{aligned}$$

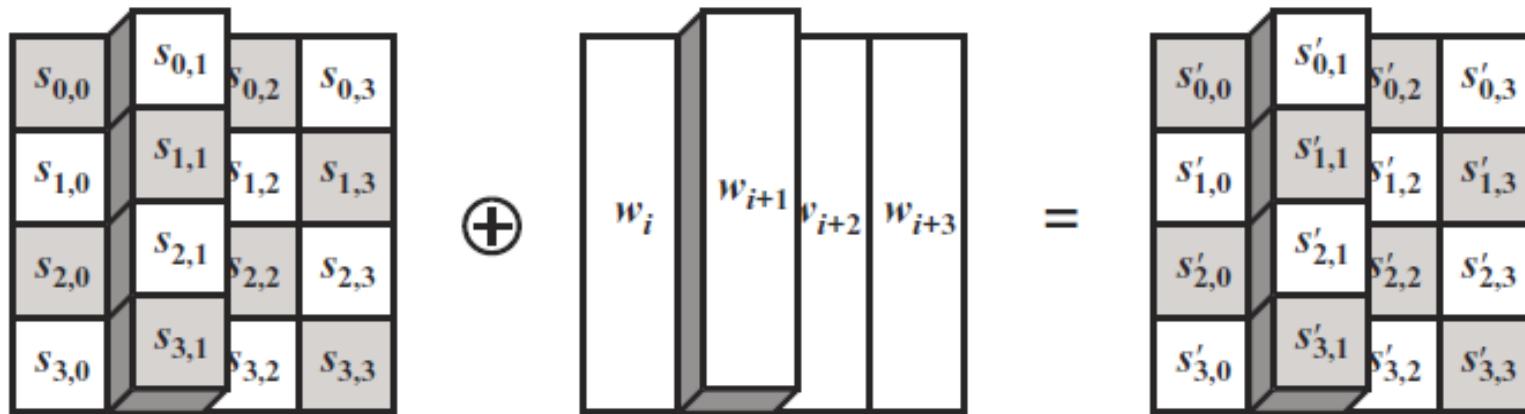
$$\begin{aligned}
 (01 \cdot 46) &= (1)(0100 \ 0110) \\
 (01 \cdot A6) &= (1)(1010 \ 0110)
 \end{aligned}$$

$$\begin{array}{r}
 00010101 \\
 10110010 \\
 \oplus 01000110 \\
 10100110 \\
 \hline
 \underline{\underline{0100}} \quad \underline{\underline{0110}} \\
 4 \quad \Rightarrow \\
 \hline
 \end{array}$$

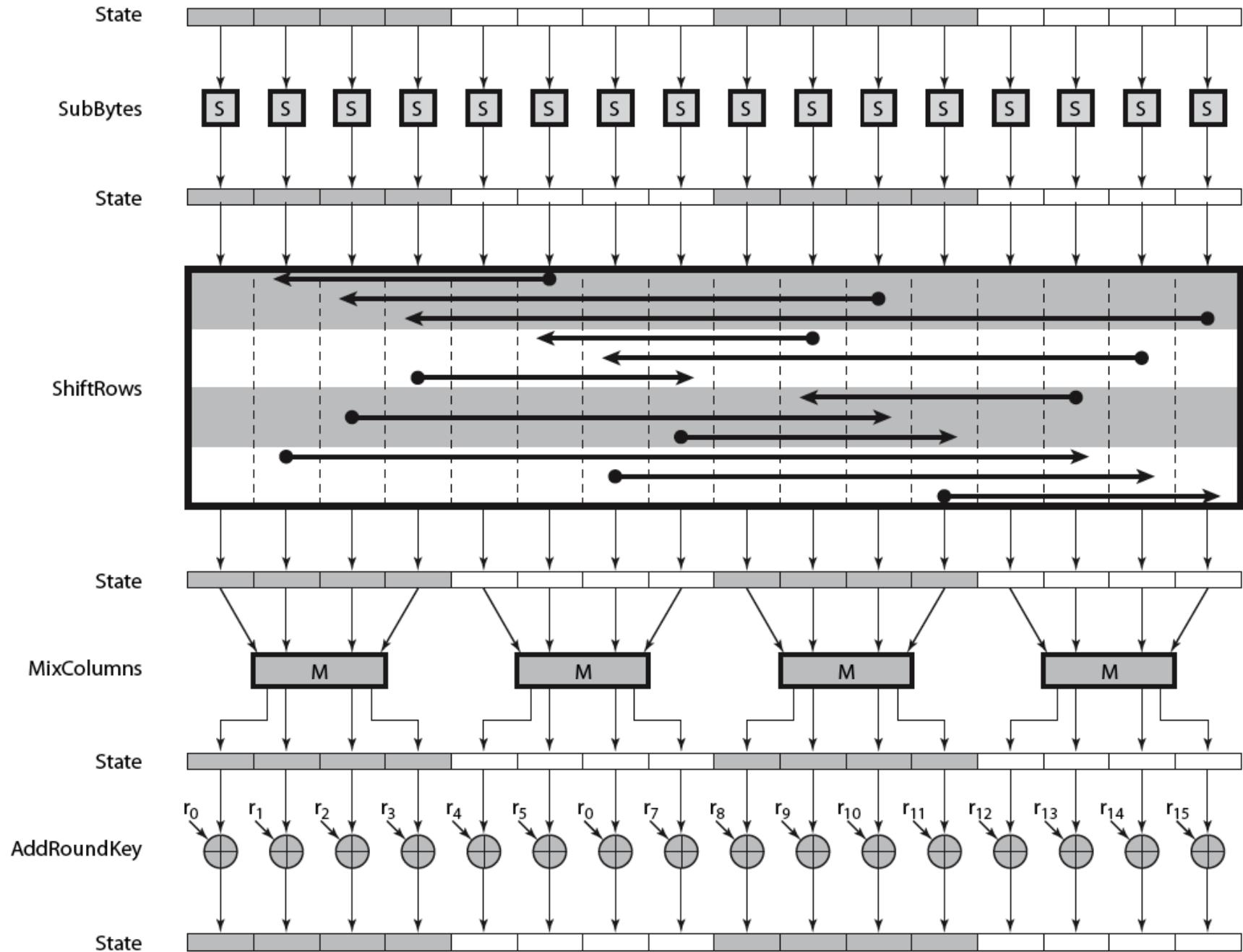
$$\begin{array}{lllll}
 (\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} & \oplus \{A6\} & = & \{47\} \\
 \{87\} & \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} & = & \{37\} \\
 \{87\} & \oplus \{6E\} & \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) & = & \{94\} \\
 (\{03\} \cdot \{87\}) \oplus \{6E\} & \oplus \{46\} & \oplus (\{02\} \cdot \{A6\}) & = & \{ED\}
 \end{array}$$

Add Round Key Transformation

- In this stage (known as AddRoundKey) the 128 bits of state are bitwise XORed with the 128 bits of the round key. The operation is viewed as a column-wise operation between the 4 bytes of a state column and one word of the round key.



(b) Add round key transformation



Add Round Key Transformation

AES Key Expansion

Table 7.3 Words for each round

Round	Words			
Pre-round	w_0	w_1	w_2	w_3
1	w_4	w_5	w_6	w_7
2	w_8	w_9	w_{10}	w_{11}
...	...			
N_r	w_{4N_r}	w_{4N_r+1}	w_{4N_r+2}	w_{4N_r+3}

Table 7.4 RCon constants

Round	Constant (RCon)	Round	Constant (RCon)
1	(<u>01</u> 00 00 00) ₁₆	6	(<u>20</u> 00 00 00) ₁₆
2	(<u>02</u> 00 00 00) ₁₆	7	(<u>40</u> 00 00 00) ₁₆
3	(<u>04</u> 00 00 00) ₁₆	8	(<u>80</u> 00 00 00) ₁₆
4	(<u>08</u> 00 00 00) ₁₆	9	(<u>1B</u> 00 00 00) ₁₆
5	(<u>10</u> 00 00 00) ₁₆	10	(<u>36</u> 00 00 00) ₁₆

The key expansion was designed to be resistant to known cryptanalytic attacks. The inclusion of a round-dependent round constant eliminates the symmetry, or similarity, between the way in which round keys are generated in different rounds.

AES Key Expansion

- The key is copied into the first four words of the expanded key.
- The remainder of the expanded key is filled in four words at a time.
- Each added word $w[i]$ depends on the immediately preceding word, $w[i - 1]$, and the word four positions back $w[i - 4]$.
- In three out of four cases, a simple XOR is used.
- For a word whose position in the w array is a multiple of 4, a more complex function is used.
- The function g consists of the following subfunctions:
 1. **RotWord** performs a one-byte circular left shift on a word. This means that an input word $[b_0, b_1, b_2, b_3]$ is transformed into $[b_1, b_2, b_3, b_0]$.
 2. **SubWord** performs a byte substitution on each byte of its input word, using the s-box described earlier.
 3. The result of steps 1 and 2 is XORed with round constant, $Rcon[j]$

Add Round Key Transformation

AES Key Expansion

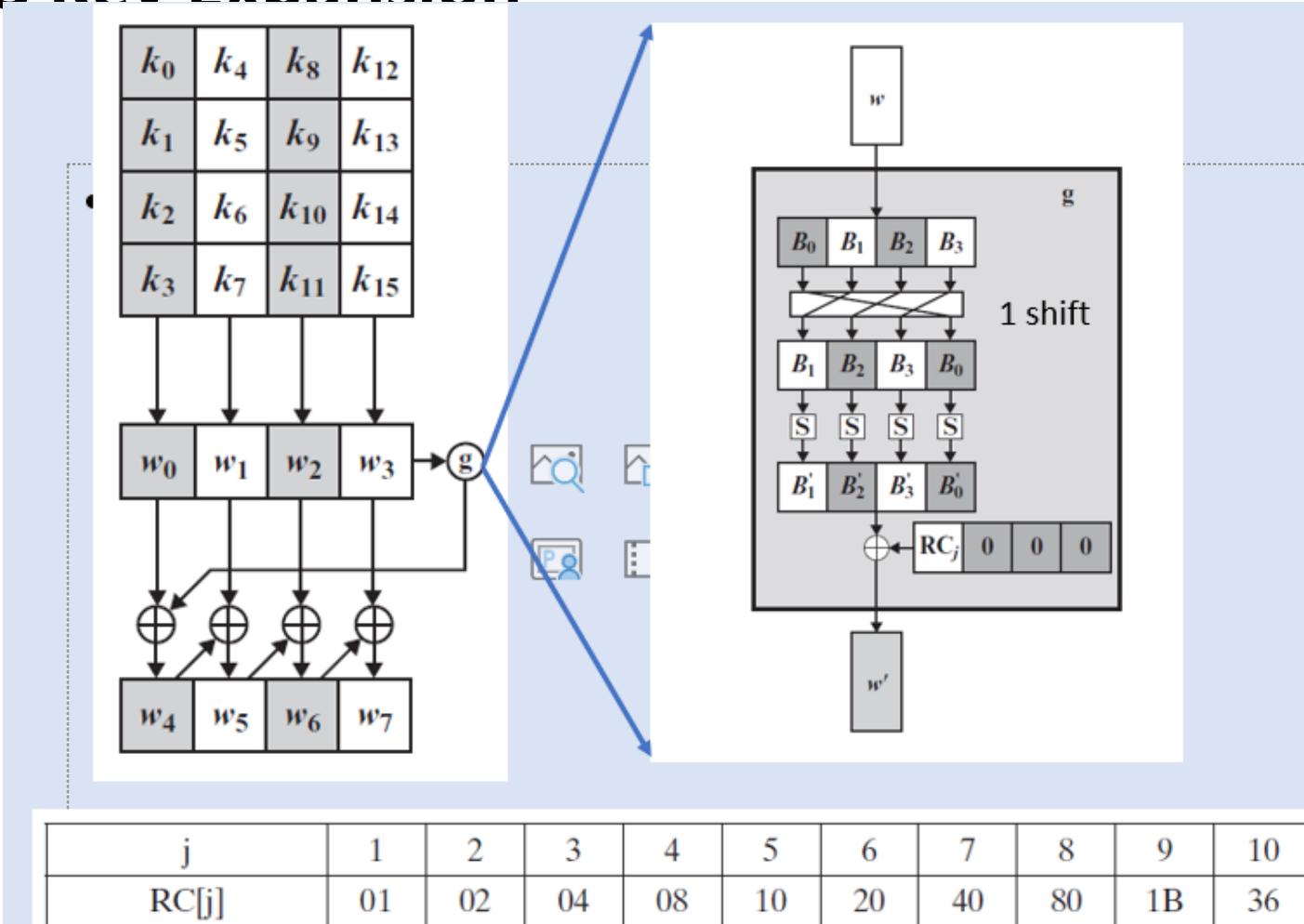


Table 7.4 RCon constants

Round	Constant (RCon)	Round	Constant (RCon)
1	(01 00 00 00) ₁₆	6	(20 00 00 00) ₁₆
2	(02 00 00 00) ₁₆	7	(40 00 00 00) ₁₆
3	(04 00 00 00) ₁₆	8	(80 00 00 00) ₁₆
4	(08 00 00 00) ₁₆	9	(1B 00 00 00) ₁₆
5	(10 00 00 00) ₁₆	10	(36 00 00 00) ₁₆

Key Words	Auxiliary Function
w0 = 0f 15 71 c9 w1 = 47 d9 e8 59 w2 = 0c b7 ad w3 = af 7f 67 98	RotWord(w3)= 7f 67 98 af = x1 SubWord(x1)= d2 85 46 79 = y1 Rcon(1)= 01 00 00 00 y1 ⊕ Rcon(1)= d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0 w5 = w4 ⊕ w1 = 9b 49 df e9 w6 = w5 ⊕ w2 = 97 fe 72 3f w7 = w6 ⊕ w3 = 38 81 15 a7	RotWord(w7)= 81 15 a7 38 = x2 SubWord(x4)= 0c 59 5c 07 = y2 Rcon(2)= 02 00 00 00 y2 ⊕ Rcon(2)= 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7 w9 = w8 ⊕ w5 = 49 80 b4 5e w10 = w9 ⊕ w6 = de 7e c6 61 w11 = w10 ⊕ w7 = e6 ff d3 c6	RotWord(w11)= ff d3 c6 e6 = x3 SubWord(x2)= 16 66 b4 8e = y3 Rcon(3)= 04 00 00 00 y3 ⊕ Rcon(3)= 12 66 b4 8e = z3
w12 = w8 ⊕ z3 = c0 af df 39 w13 = w12 ⊕ w9 = 89 2f 6b 67 w14 = w13 ⊕ w10 = 57 51 ad 06 w15 = w14 ⊕ w11 = b1 ae 7e c0	RotWord(w15)= ae 7e c0 b1 = x4 SubWord(x3)= e4 f3 ba c8 = y4 Rcon(4)= 08 00 00 00 y4 ⊕ Rcon(4)= ec f3 ba c8 = 4
w16 = w12 ⊕ z4 = 2c 5c 65 f1 w17 = w16 ⊕ w13 = a5 73 0e 96 w18 = w17 ⊕ w14 = f2 22 a3 90 w19 = w18 ⊕ w15 = 43 8c dd 50	RotWord(w19)= 8c dd 50 43 = x5 SubWord(x4)= 64 c1 53 1a = y5 Rcon(5)= 10 00 00 00 y5 ⊕ Rcon(5)= 74 c1 53 1a = z5

w20 = w16 ⊕ z5 = 58 9d 36 eb w21 = w20 ⊕ w17 = fd ee 38 7d w22 = w21 ⊕ w18 = 0f cc 9b ed w23 = w22 ⊕ w19 = 4c 40 46 bd	RotWord(w23)= 40 46 bd 4c = x6 SubWord(x5)= 09 5a 7a 29 = y6 Rcon(6)= 20 00 00 00 y6 ⊕ Rcon(6)= 29 5a 7a 29 = z6
w24 = w20 ⊕ z6 = 71 c7 4c c2 w25 = w24 ⊕ w21 = 8c 29 74 bf w26 = w25 ⊕ w22 = 83 e5 ef 52 w27 = w26 ⊕ w23 = cf a5 a9 ef	RotWord(w27)= a5 a9 ef cf = x7 SubWord(x6)= 06 d3 df 8a = y7 Rcon(7)= 40 00 00 00 y7 ⊕ Rcon(7)= 46 d3 df 8a = z7
w28 = w24 ⊕ z7 = 37 14 93 48 w29 = w28 ⊕ w25 = bb 3d e7 f7 w30 = w29 ⊕ w26 = 38 d8 08 a5 w31 = w30 ⊕ w27 = f7 7d a1 4a	RotWord(w31)= 7d a1 4a f7 = x8 SubWord(x7)= ff 32 d6 68 = y8 Rcon(8)= 80 00 00 00 y8 ⊕ Rcon(8)= 7f 32 d6 68 = z8
w32 = w28 ⊕ z8 = 48 26 45 20 w33 = w32 ⊕ w29 = f3 1b a2 d7 w34 = w33 ⊕ w30 = cb c3 aa 72 w35 = w34 ⊕ w32 = 3c be 0b 38	RotWord(w35)= be 0b 38 3c = x9 SubWord(x8)= ae 2b 07 eb = y9 Rcon(9)= 1B 00 00 00 y9 ⊕ Rcon(9)= b5 2b 07 eb = z9
w36 = w32 ⊕ z9 = fd 0d 42 cb w37 = w36 ⊕ w33 = 0e 16 e0 1c w38 = w37 ⊕ w34 = c5 d5 4a 6e w39 = w38 ⊕ w35 = f9 6b 41 56	RotWord(w39)= 6b 41 56 f9 = x10 SubWord(x9)= 7f 83 b1 99 = y10 Rcon(10)= 36 00 00 00 y10 ⊕ Rcon(10)= 49 83 b1 99 = z10
w40 = w36 ⊕ z10 = b4 8e f3 52 w41 = w40 ⊕ w37 = ba 98 13 4e w42 = w41 ⊕ w38 = 7f 4d 59 20 w43 = w42 ⊕ w39 = 86 26 18 76	

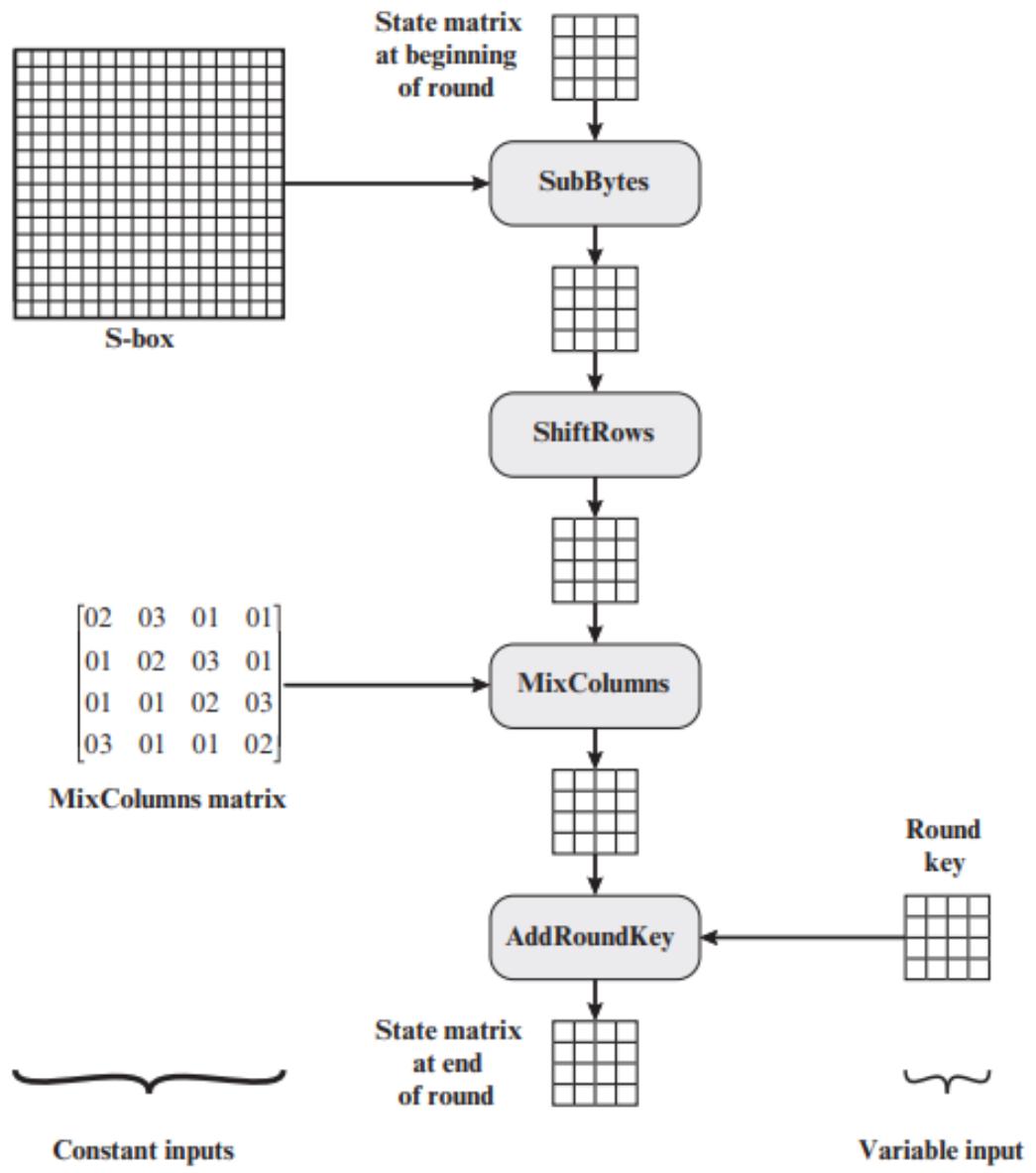


Figure 6.8 Inputs for Single AES Round

AES Example - Input (128 bit key and message)

- Key in English: **Thats my Kung Fu** (16 ASCII characters, 1 byte each)

Translation into Hex:

T	h	a	t	s		m	y		K	u	n	g		F	u
54	68	61	74	73	20	6D	79	20	4B	75	6E	67	20	46	75

Key in Hex (128 bits): **54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75**

- **The first Roundkey:**

AES Example - Input (128 bit key and message)

- **The first Roundkey:**

- Key in Hex (128 bits): 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- $w[0] = (54, 68, 61, 74), w[1] = (73, 20, 6D, 79), w[2] = (20, 4B, 75, 6E), w[3] = (67, 20, 46, 75)$
- $g(w[3]):$
 - circular byte left shift of $w[3]$: (20, 46, 75, 67)
 - Byte Substitution (S-Box): ($B7, 5A, 9D, 85$)
 - Adding round constant (01, 00, 00, 00) gives: $g(w[3]) = (B6, 5A, 9D, 85)$
- $w[4] = w[0] \oplus g(w[3]) = (E2, 32, FC, F1):$

↓
 10110111
 0000 0001 XOR
10110110

B 6

	y																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0	
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15	
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75	
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84	
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF	
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8	
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2	
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73	
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB	
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79	
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08	
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A	
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E	
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF	
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16	

- $w[4] = w[0] \oplus g(w[3]) = (E2, 32, FC, F1)$:

0101 0100	0110 1000	0110 0001	0111 0100
1011 0110	0101 1010	1001 1101	1000 0101
1110 0010	0011 0010	1111 1100	1111 0001
E2	32	FC	F1

- $w[5] = w[4] \oplus w[1] = (91, 12, 91, 88)$, $w[6] = w[5] \oplus w[2] = (B1, 59, E4, E6)$,
 $w[7] = w[6] \oplus w[3] = (D6, 79, A2, 93)$
- first roundkey: **E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93**

- Round 0: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- Round 1: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
- Round 2: 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA

Plaintext in English: Two One Nine Two (16 ASCII characters, 1 byte each)

Translation into Hex:

T	w	o		O	n	e		N	i	n	e		T	w	o
54	77	6F	20	4F	6E	65	20	4E	69	6E	65	20	54	77	6F

AES Example - Add Roundkey, Round 0

- State Matrix and Roundkey No.0 Matrix:

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \quad \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix}$$

- XOR the corresponding entries, e.g., $69 \oplus 4B = 22$

$$\begin{array}{r} 0110\ 1001 \\ 0100\ 1011 \\ \hline 0010\ 0010 \end{array}$$

- the new State Matrix is

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

AES Example - Round 1, Substitution Bytes

- current State Matrix is

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

- substitute each entry (byte) of current state matrix by corresponding entry in AES S-Box
- for instance: byte 6E is substituted by entry of S-Box in row 6 and column E, i.e., by 9F
- this leads to new State Matrix

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

- this non-linear layer is for resistance to differential and linear cryptanalysis attacks

AES Example - Round 1, Shift Row

- the current State Matrix is

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

- four rows are shifted cyclically to the left by offsets of 0,1,2, and 3
- the new State Matrix is

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

- this linear mixing step causes diffusion of the bits over multiple rounds

AES Example - Round 1, Mix Column

- Mix Column multiplies fixed matrix against current State Matrix:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

- entry BA is result of $(02 \bullet 63) \oplus (03 \bullet 2F) \oplus (01 \bullet AF) \oplus (01 \bullet A2)$:
 - $02 \bullet 63 = 00000010 \bullet 01100011 = 11000110$
 - $03 \bullet 2F = (02 \bullet 2F) \oplus 2F = (00000010 \bullet 00101111) \oplus 00101111 = 01110001$
 - $01 \bullet AF = AF = 10101111$ and $01 \bullet A2 = A2 = 10100010$
 - hence

$$\begin{array}{r} 11000110 \\ 01110001 \\ 10101111 \\ 10100010 \\ \hline 10111010 \end{array}$$

AES Example - Add Roundkey, Round 1

- State Matrix and Roundkey No.1 Matrix:

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \quad \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix}$$

- XOR yields new State Matrix

$$\begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

- AES output after Round 1: 58 47 08 8B 15 B6 1C BA 59 D4 E2 E8 CD 39 DF CE

- Continue with next 9 rounds: (in 10th round no mixcolumn operation)

Additional Materials

So the 2^3 polynomials of $GF(2^3)$ can therefore be represented by the bit strings:

0	\Rightarrow	000
1	\Rightarrow	001
x	\Rightarrow	010
x^2	\Rightarrow	100
$x + 1$	\Rightarrow	011
$x^2 + 1$	\Rightarrow	101
$x^2 + x$	\Rightarrow	110
$x^2 + x + 1$	\Rightarrow	111

- Given any n at all, exactly the same approach can be used to come up with 2^n bit patterns, each pattern consisting of n bits,

<i>Number</i>	<i>Binary</i>	<i>GF(2⁸) Polynomial</i>	<i>Simplified</i>
0	0	0	0
1	1	1	1
2	10	1x+0	x
3	11	1x+1	x+1
4	100	1x ² +0x+0	x ²
5	101	1x ² +0x+1	x ² +1
8	1000	1x ³ +0x ² +0x+0	x ³
16	10000	1x ⁴ +0x ³ +0x ² +0x+0	x ⁴
21	10101	1x ⁴ +0x ³ +1x ² +0x+1	x ⁴ +x ² +1

Data Security and Privacy

DSE 3258

L6 –Asymmetric Ciphers Public-Key Cryptosystems

Public-Key Cryptosystems

Terminology Related to Asymmetric Encryption:

Asymmetric Keys:

Two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Public Key Certificate:

A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.

Public Key (Asymmetric) Cryptographic Algorithm:

A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.

Public Key Infrastructure (PKI):

A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.

Public-Key Cryptosystems

Algorithms rely on one key for encryption and a different but related key for decryption.

Important characteristics

- It is computationally infeasible to determine the decryption key with only knowledge of the cryptographic algorithm and the encryption key.
- Either of the two related keys can be used for encryption, with the other used for decryption.
(exhibited in RSA)

Ingredients

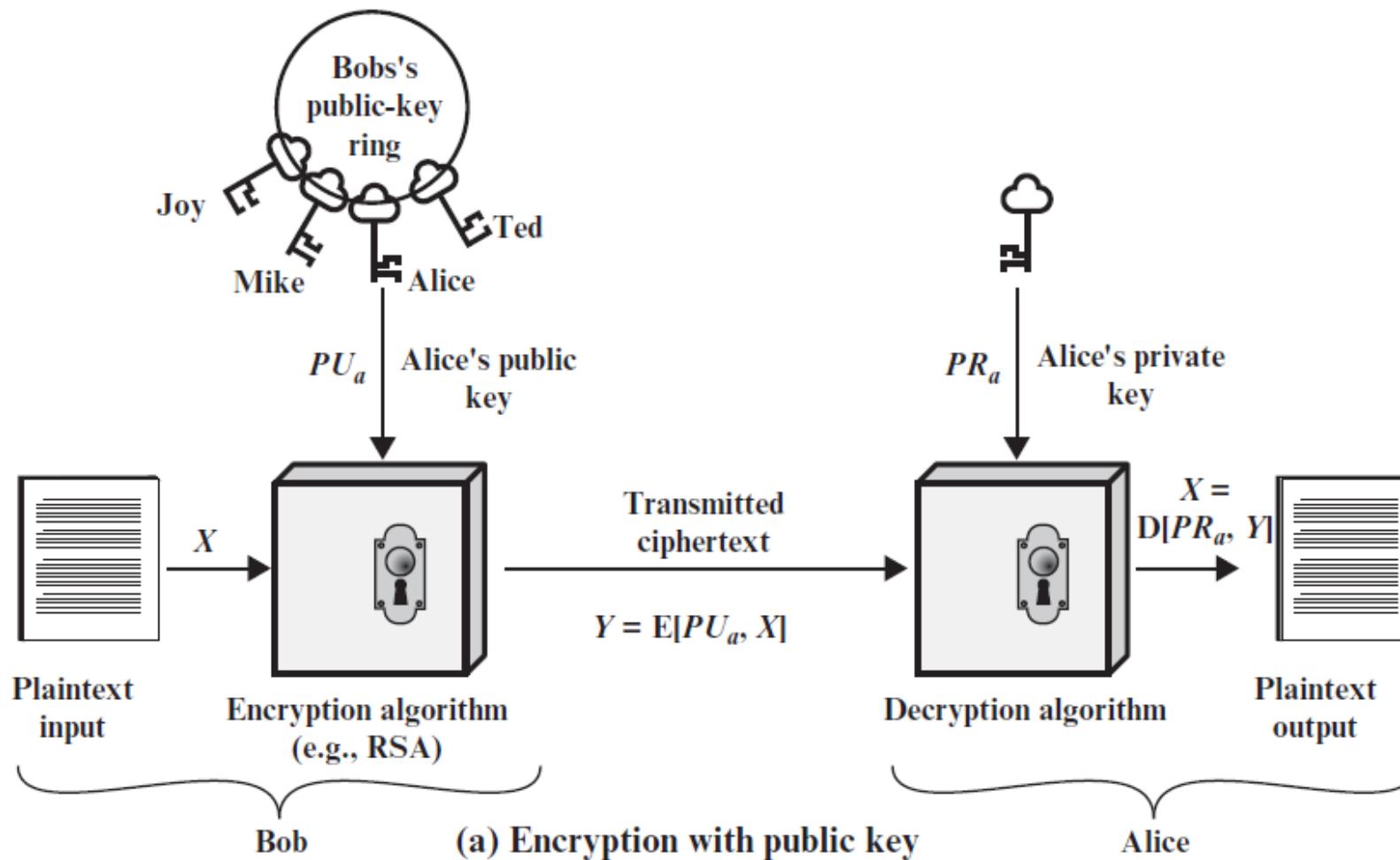
- Public-key encryption scheme has six ingredients
 - **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
 - **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.
 - **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
 - **Ciphertext:** This is the encrypted message produced as output. It depends on the plaintext and the key.
 - For a given message, two different keys will produce two different ciphertexts.
 - **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. Each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

Encryption with Public key



Encryption with Private key

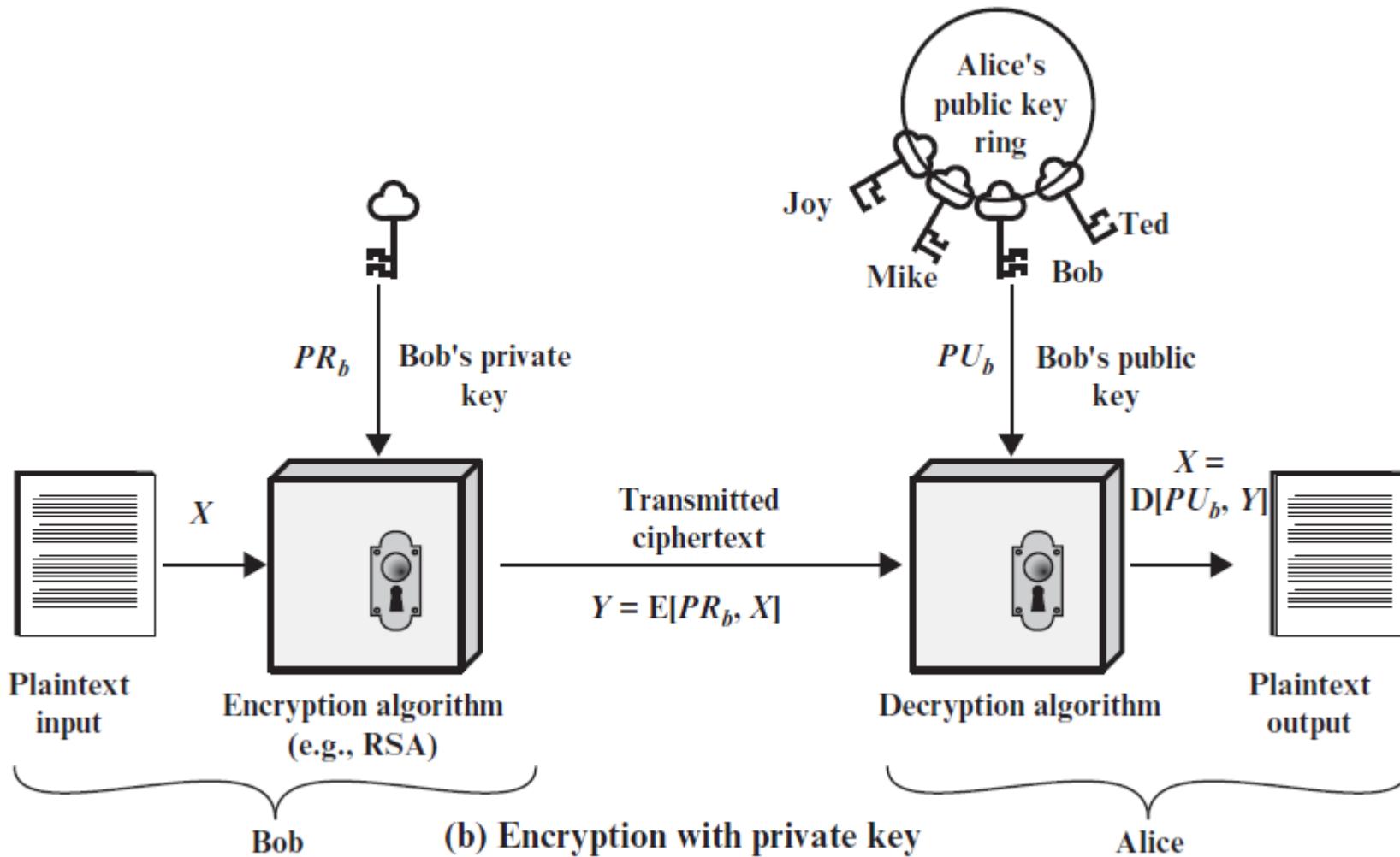


Table 9.2 Conventional and Public-Key Encryption

Conventional Encryption	Public-Key Encryption
<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. The same algorithm with the same key is used for encryption and decryption.2. The sender and receiver must share the algorithm and the key. <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. The key must be kept secret.2. It must be impossible or at least impractical to decipher a message if the key is kept secret.3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption.2. The sender and receiver must each have one of the matched pair of keys (not the same one). <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. One of the two keys must be kept secret.2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret.3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Applications of PKC

1. *Encryption / Decryption*

Sender encrypts a message with the recipient's public key

2. *Digital signature*

sender signs a message with private key

3. *Key exchange*

two sides cooperate to exchange a session key

Requirements for public key cryptography

1. Pair of keys (public key KU_b , private key KR_b)

2. Easy to encrypt the message $C=E_{KU_b}(M)$

3. Easy to decrypt the ciphertext

$$M = D_{KR_b}(C) = D_{KR_b}[E_{KU_b}(M)]$$

4. Knowing KU_b , it is infeasible to determine KR_b

5. Knowing C & KU_b , it is infeasible to determine M

6. Either of 2 keys can be used for encryption

$$M = D_{KR_b}[E_{KU_b}(M)] = D_{KU_b}[E_{KR_b}(M)]$$

- An adversary, observing Y and having access to PU_b, but not having access to PR_b or X, must attempt to recover X and/or PR_b.
- It is assumed that the adversary does have knowledge of the encryption (E) and decryption (D) algorithms.
- If the adversary is interested only in this particular message, then the focus of effort is to recover X by generating a plaintext estimate X_n.
- Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PR_b by generating an estimate PR_{n b}.

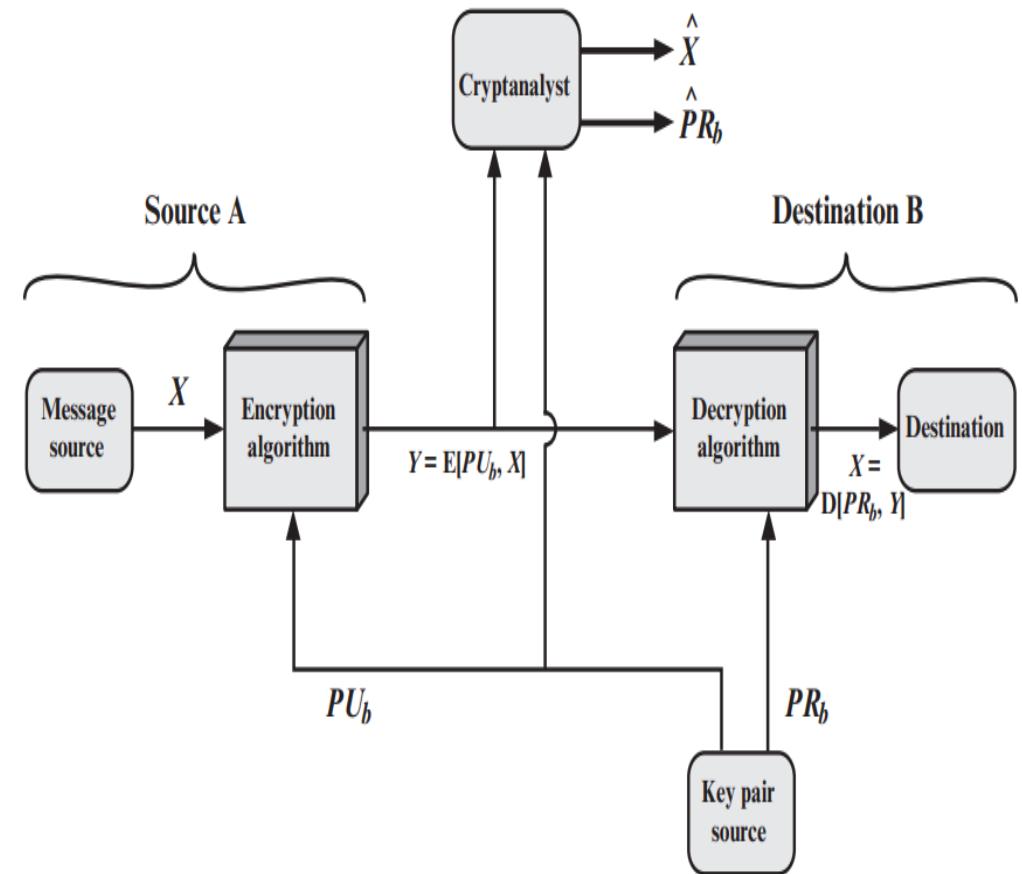


Figure 9.2 Public-Key Cryptosystem: Confidentiality

$$Y = E(PR_a, X)$$

$$X = D(PU_a, Y)$$

- In this case, A prepares a message to B and encrypts it using A's private key before transmitting it.
- B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message.
- Therefore, the entire encrypted message serves as a digital signature.
- In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity

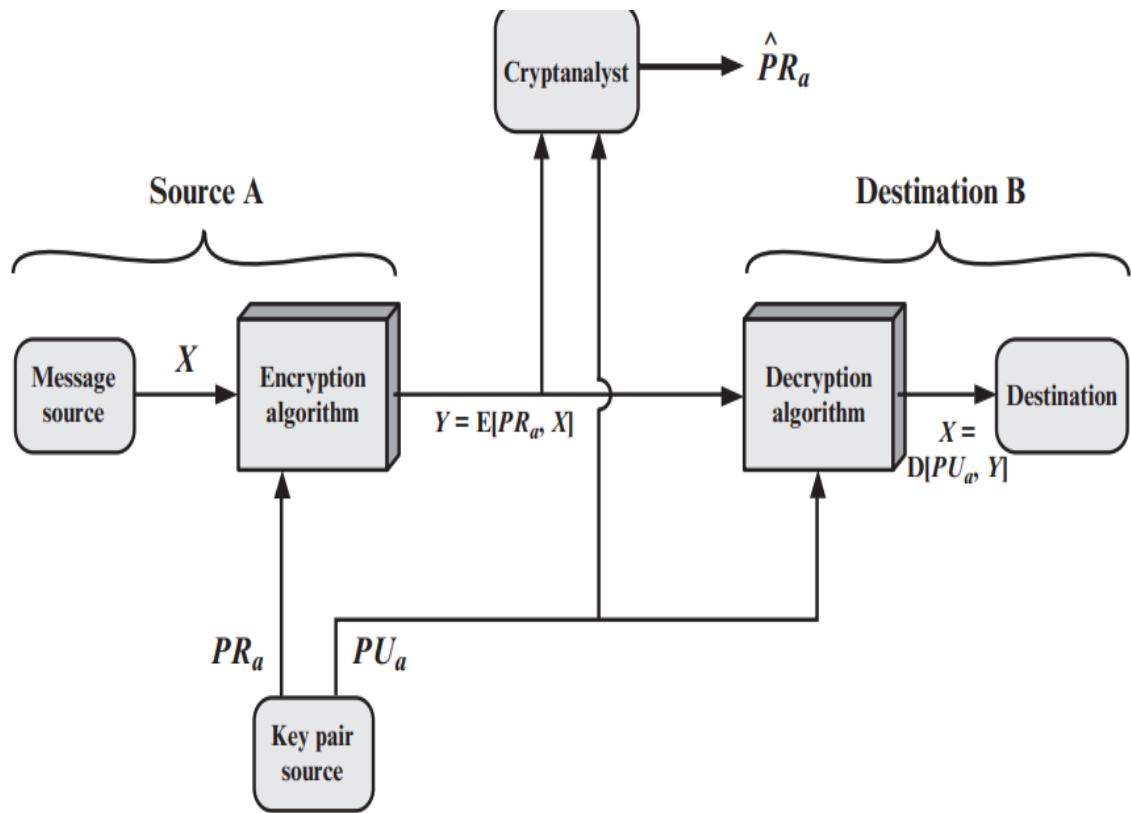


Figure 9.3 Public-Key Cryptosystem: Authentication

- **Drawback:**
- In the preceding scheme, the entire message is encrypted, which, although validating both author and contents, requires a great deal of storage.
- Each document must be kept in plaintext to be used for practical purposes.
- A copy also must be stored in ciphertext so that the origin and contents can be verified in case of a dispute.
- **Solution:**
- A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such a block, called an **authenticator**, must have the property that it is infeasible to change the document without changing the authenticator.

- It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme

$$Z = E(PU_b, E(PR_a, X))$$

$$X = D(PU_a, D(PR_b, Z))$$

- In this case, we begin as before by encrypting a message, using the sender's private key.
- This provides the digital signature. Next, we encrypt again, using the receiver's public key.
- The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus, confidentiality is provided.
- The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

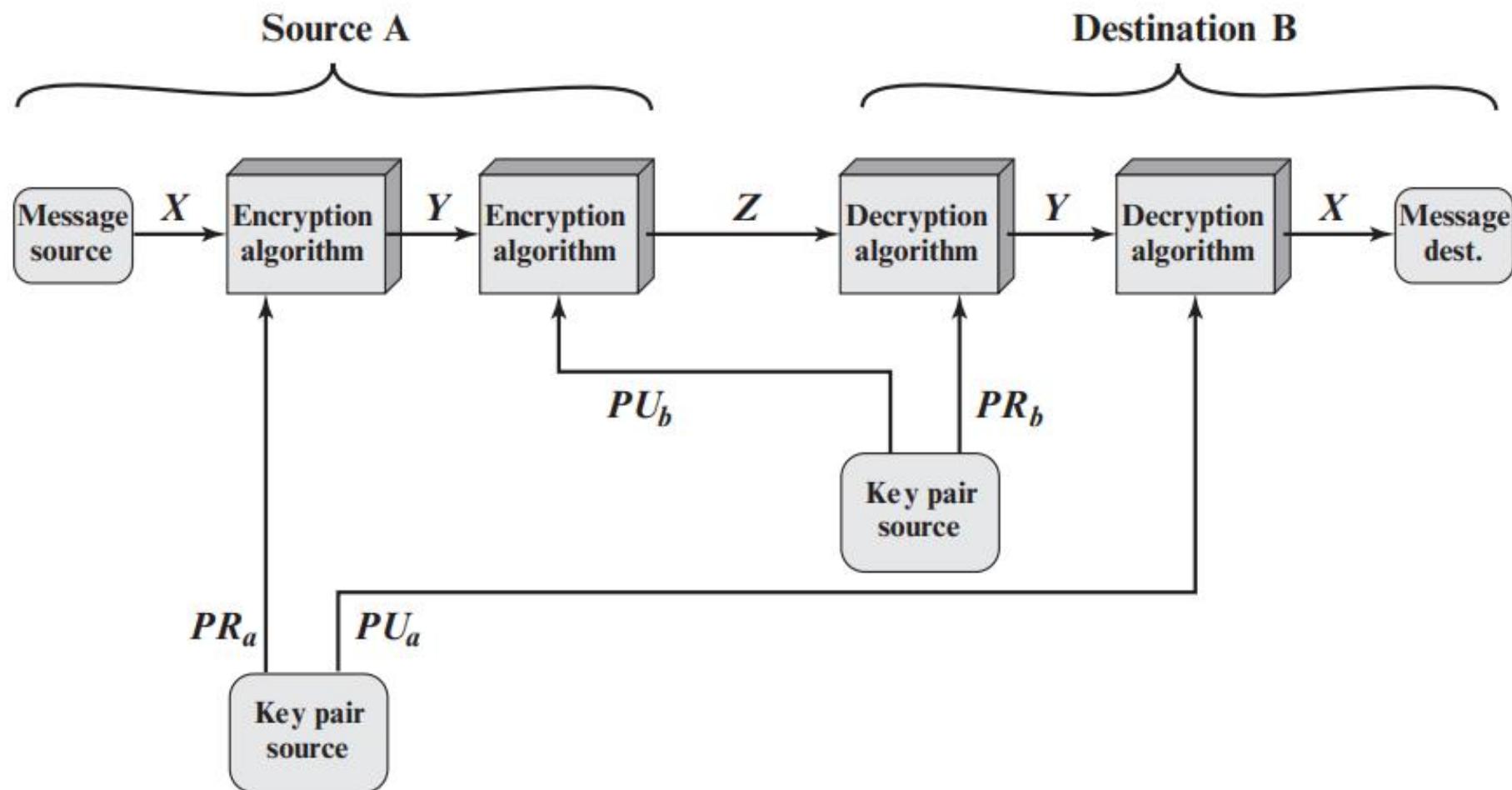


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Applications of PKC

1. *Encryption / Decryption*

The sender encrypts a message with the recipient's public key, and the recipient decrypts the message with the recipient's private key.

2. *Digital signature*

The sender “signs” a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message

3. *Key exchange*

Two sides cooperate to exchange a session key, which is a secret key for symmetric encryption generated for use for a particular transaction (or session) and valid for a short period of time. Several different approaches are possible, involving the private key(s) of one or both parties; t

Table 9.3 Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie–Hellman	No	No	Yes
DSS	No	Yes	No

Requirements for Public-Key Cryptography

1. It is computationally easy for a party B to generate a key pair (public key PUb, private key PRb).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext: $C = E(PUb, M)$
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: $M = D(PRb, C) = D[PRb, E(PUb, M)]$
4. It is computationally infeasible for an adversary, knowing the public key, PUb, to determine the private key, PRb.
5. It is computationally infeasible for an adversary, knowing the public key, PUb, and a ciphertext, C, to recover the original message, M.

A sixth requirement that, although useful, is not necessary for all public-key applications:

6. The two keys can be applied in either order: $M = D[P Ub, E(PRb, M)] = D[PRb, E(P Ub, M)]$

Rivest-Shamir-Adleman (RSA) Algorithm

- The RSA scheme is a cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n .
- A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 2^{1024} .

➤ to encrypt a message M the sender:

- obtains **public key** of recipient PU = { e, n }
- computes: $C = M^e \text{ mod } n$, where $0 \leq M < n$

➤ to decrypt the ciphertext C the owner:

- uses their private key PR = { d, n }
- computes: $M = C^d \text{ mod } n$

➤ note that the message M must be smaller than the modulus n

Rivest-Shamir-Adleman (RSA) Algorithm

- each user generates a public/private key pair by:
- selecting two **large primes at random**: p, q
- computing their system **modulus** $n=p \cdot q$
 - note $\phi(n)=(p-1)(q-1)$ [Euler's totient function]
- selecting at random the **encryption key** e
 - where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
- solve following equation to find **decryption key** d
 - $e \cdot d \equiv 1 \pmod{\phi(n)}$ and $0 \leq d \leq n$
- publish their public encryption key: $PU = \{e, n\}$
- keep secret private decryption key: $PR = \{d, n\}$

RSA Example:

1. Select primes: $p=17$ & $q=11$
2. Calculate $n = pq = 17 \times 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e, 160) = 1$; choose $e=7$
5. Determine d : $de \equiv 1 \pmod{160}$ and $d < 160$ Value is
 $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key PU= $\{7, 187\}$
7. Keep secret private key PR= $\{23, 187\}$

RSA Example:

- sample RSA encryption/decryption is:
- given message $M = 88$ (see that $88 < 187$)
- encryption:

$$C = 88^7 \bmod 187 = 11$$

- decryption:

$$M = 11^{23} \bmod 187 = 88$$

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \\ \times (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59,969,536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894,432 \bmod 187 = 11$$

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \\ \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14,641 \bmod 187 = 55$$

$$11^8 \bmod 187 = 214,358,881 \bmod 187 = 33$$

$$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 \\ = 79,720,245 \bmod 187 = 88$$

Question: P and Q are two prime numbers. P=7, and Q=17. Take public key E=5. If plain text value is 6, then what will be cipher text value according to RSA algorithm? Again calculate plain text value from cipher text.

Solution:

1. Two prime numbers P=7, Q=17

2. $n = P * Q = 17 * 7 = 119$ **n = 119**

3. $\Phi(n) = (P-1) * (Q - 1) = (17 - 1) * (7 - 1) = 16 * 6 = 96$ **$\Phi(n) = 96$**

4. Public key E = 5. **E = 5**

5. Calculate d = 77. $d = ((\Phi(n) * i) + 1) / e$ **d = 77**

$$d = ((96*1)+1) / 5 = 19.4$$

$$d = ((96*2)+1) / 5 = 38.6$$

$$d = ((96*3)+1) / 5 = 57.8$$

$d = ((96*4)+1) / 5 = 77$ **(Stop finding d because getting integer value)**

6. Public key = {e, n} = {5, 119}, private key = {d, n} = {77, 119}.

7. Plain text PT = 6, $CT = PT^E \text{ mod } n = 6^5 \text{ mod } 119 = 41$. **Cipher Text = 41**

8. Cipher text CT = 41, $PT = CT^d \text{ mod } n = 41^{77} \text{ mod } 119 = 6$. **Plain Text = 6**

Key Generation by Alice

Select p, q

p and q both prime, $p \neq q$

Calculate $n = p \times q$

Calculate $\phi(n) = (p - 1)(q - 1)$

Select integer e

$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$

Calculate d

$d \equiv e^{-1} \pmod{\phi(n)}$

Public key

$PU = \{e, n\}$

Private key

$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:

$M < n$

Ciphertext:

$C = M^e \pmod{n}$

Decryption by Alice with Alice's Public Key

Ciphertext:

C

Plaintext:

$M = C^d \pmod{n}$

Figure 9.5 The RSA Algorithm

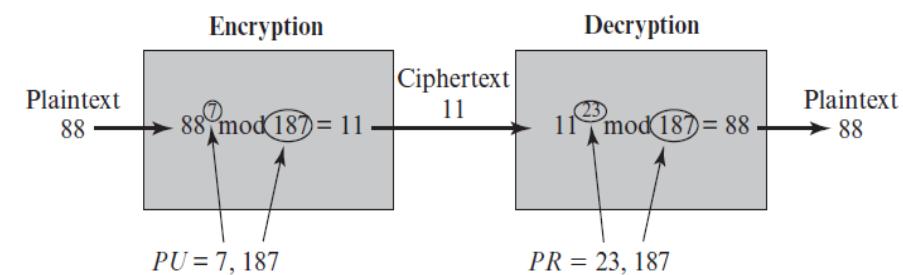


Figure 9.6 Example of RSA Algorithm

SOLVE

- 9.3 In a public-key system using RSA, you intercept the ciphertext $C = 20$ sent to a user whose public key is $e = 13, n = 77$. What is the plaintext M ?

Data Security and Privacy

DSE 3258

L7 –Asymmetric Ciphers

Public-Key Cryptosystems

Diffie- Hellman Key Exchange

Diffie-Hellman Key Exchange

- It is a protocol that enables two users to establish a secret key using a public-key scheme based on discrete logarithms. **The protocol is secure only if the authenticity of the two participants can be established.**
 - The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages.
 - The algorithm itself is limited to the exchange of secret values.
 - The Diffie–Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.
-
- This algorithm facilitates the exchange of secret key without actually transmitting it.

Diffie-Hellman Key Exchange

- Primitive Root:
- A primitive root of a prime number p is defined as one whose powers modulo generate all the integers from 1 to $p-1$. That is, if a is a primitive root of the prime number p , then the numbers

$$a \bmod p, a^2 \bmod p, \dots, a^{p-1} \bmod p$$

- are distinct and consist of the integers from 1 through $p - 1$ in some permutation.
- For any integer b and a primitive root a of prime number p , we can find a unique exponent i such that

$$b \equiv a^i \pmod{p} \quad \text{where } 0 \leq i \leq (p - 1)$$

- The exponent i is referred to as the **discrete logarithm** of b for the base a , mod p

Diffie-Hellman Key Exchange

- Primitive roots of value q
- Example:

$$a = 3$$

$$q = 7$$

To say **a** is primitive to **q**:

$$3^1 \bmod 7 = 3$$

$$3^2 \bmod 7 = 2$$

$$3^3 \bmod 7 = 6$$

$$3^4 \bmod 7 = 4$$

$$3^5 \bmod 7 = 5$$

$$3^6 \bmod 7 = 1$$

Diffie-Hellman Key Exchange

- Diffie-Hellman Key Exchange Algorithm:
 - For this scheme, there are two publicly known numbers:
 - a prime number q and an integer α that is a primitive root of q .
 - Suppose the users **A** and **B** wish to exchange a key **K**.
 - User **A** selects a random integer $X_A < q$ and computes Y_A .
 - Similarly, user **B** independently selects a random integer $X_B < q$ and computes Y_B .
 - Each side keeps the **X** value private and makes the **Y** value available publicly to the other side. The whole algorithm can be summarized as follows:

Diffie-Hellman Key Exchange

Global Public Elements

q prime number

α $\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A $X_A < q$

Calculate public Y_A $Y_A = \alpha^{X_A} \text{ mod } q$

User B Key Generation

Select private X_B $X_B < q$

Calculate public Y_B $Y_B = \alpha^{X_B} \text{ mod } q$

Calculation of Secret Key by User A

$$K = (Y_B)^{X_A} \text{ mod } q$$

Calculation of Secret Key by User B

$$K = (Y_A)^{X_B} \text{ mod } q$$

Diffie-Hellman Key Exchange Algorithm

Diffie-Hellman Key Exchange

- Key Exchange Protocol

Scenario using Diffie-Hellman:



Alice



Bob

Alice and Bob share a prime number q and an integer α , such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \text{ mod } q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \text{ mod } q$

Alice and Bob share a prime number q and an integer α , such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \text{ mod } q$

Bob receives Alice's public key Y_A in plaintext

Bob calculates shared secret key $K = (Y_A)^{X_B} \text{ mod } q$



Diffie-Hellman Key Exchange

- **Diffie-Hellman Key Exchange Example:**
- Key exchange is based on the use of the prime number $q=353$ and a primitive root of 353, in this case $\alpha=3$. A and B select secret keys $X_A=97$, $X_B=233$ respectively. Each computes its public key:
 - A computes $Y_A = 3^{97} \text{ mod } 353 = 40$.
 - B computes $Y_B = 3^{233} \text{ mod } 353 = 248$.
- After they exchange public keys, each can compute the common secret key:
 - A computes $K = (Y_B)^{X_A} \text{ mod } 353 = 248^{97} \text{ mod } 353 = 160$.
 - B computes $K = (Y_A)^{X_B} \text{ mod } 353 = 40^{233} \text{ mod } 353 = 160$.

DH Example

$q=11 \quad \alpha=3$

$X_A = 5$

$Y_A = 3^5 \text{ mod } 11 = 1$

$X_B = 3$

$Y_B = 3^3 \text{ mod } 11 = 27 \text{ mod } 11 = 5$

$K_1 = 5^5 \text{ mod } 11 = 1$

$K_2 = 1^3 \text{ mod } 11 = 1$

A & B can share 1

Q1. In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value $q = 17$ and primitive root = 5. If Alice's secret key is 4 and Bob's secret key is 6, what is the secret key they exchanged?

1.16

2.17

3.18

4.19

Q2. In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value $q = 23$ and primitive root = 5. If Alice's secret key is 6 and Bob's secret key is 15, what is the secret key they exchanged?

1.4

2.3

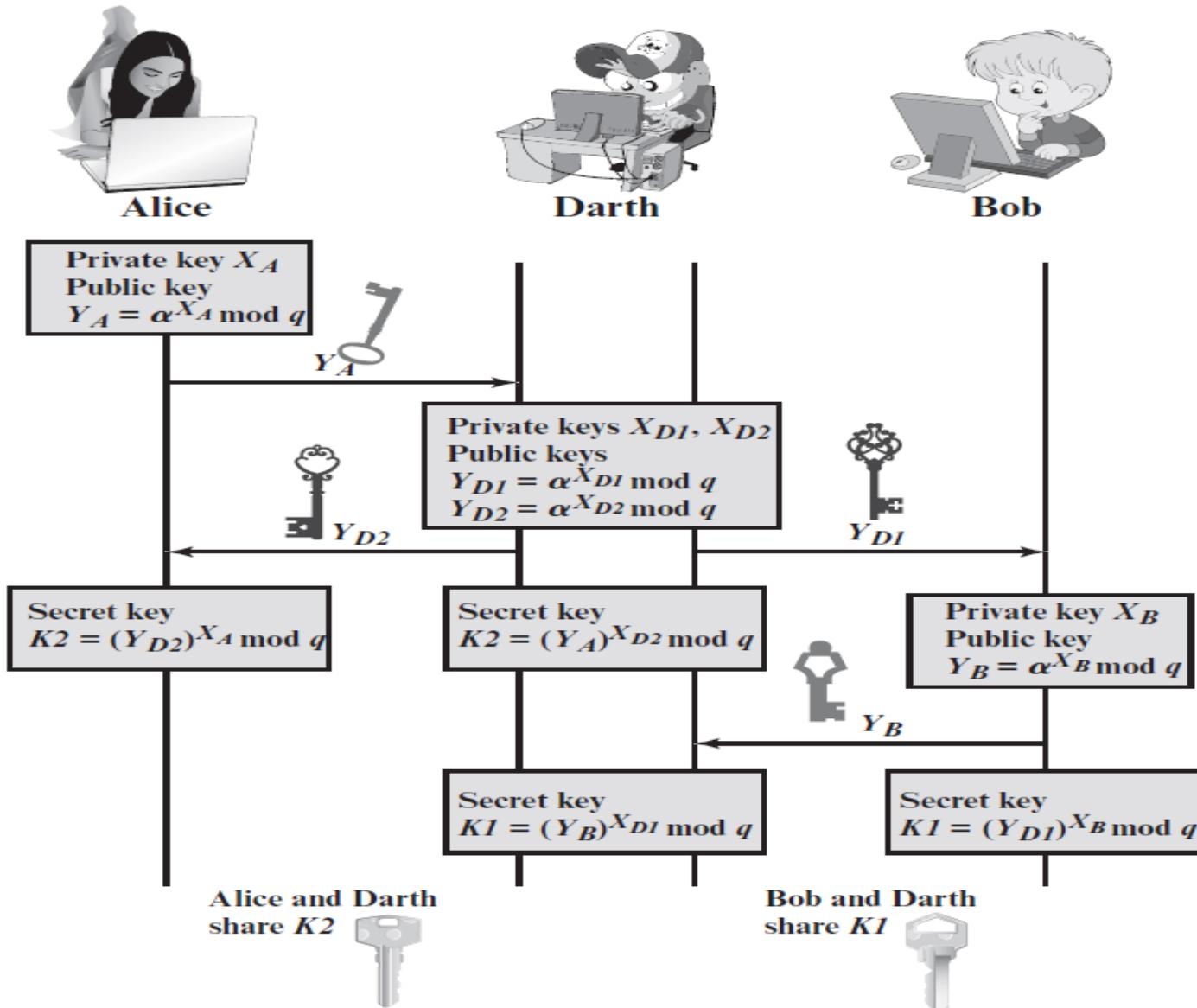
3.2

4.1

Breaking of Diffie-Hellman

- The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack.
- In this attack, an opponent Darth intercepts Alice's public value and sends her own public value to Bob.
- When Bob transmits his public value, Darth substitutes it with her own and sends it to Alice.
- Darth and Alice thus agree on one shared key and Darth and Bob agree on another shared key.
- After this exchange, Darth simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party.
- This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants.

Middle man attack / bucket bridge attack



Middle man attack / bucket bridge attack

Alice	Darth	Bob
$q=11 \quad \alpha=7$	$q=11 \quad \alpha=7$	$q=11 \quad \alpha=7$
$X_A = 3$	$M_{XA}=8 \quad M_{XB}=6$	$X_B = 9$
$Y_A = 7^3 \bmod 11$ = 2	$Y_A = 7^8 \bmod 11$ = 9	$Y_B = 7^9 \bmod 11$ = 8
$Y_B = 4$	$Y_B = 7^6 \bmod 11$ = 4	$Y_A = 9$
$K1 = 4^3 \bmod 11$ = 9	$Y_A = 2 \quad Y_B = 8$ $K1 = 8^8 \bmod 11$ = 5	$K2 = 9^9 \bmod 11$ = 5
	$K2 = 2^6 \bmod 11$ = 9	

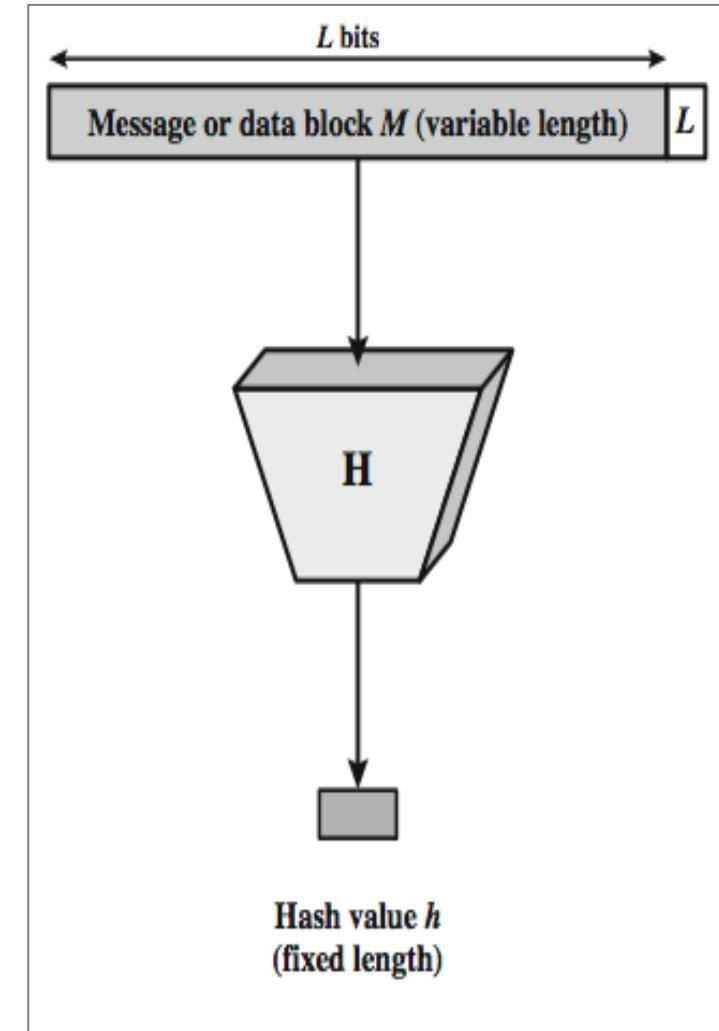
Data Security and Privacy

DSE 3258

L8 –Cryptographic Hash Functions

Hash Functions

- A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value $h = H(M)$.
- A “good” hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random.
- The kind of hash function needed for security applications is referred to as **a *cryptographic hash function***.



Hash Functions (Cont..)

- usually assume hash function is public
- hash used to detect changes to message
- want a cryptographic hash function
 - computationally infeasible to find data mapping to specific hash (**one-way property**)
 - computationally infeasible to find two data to same hash (**collision-free property**)

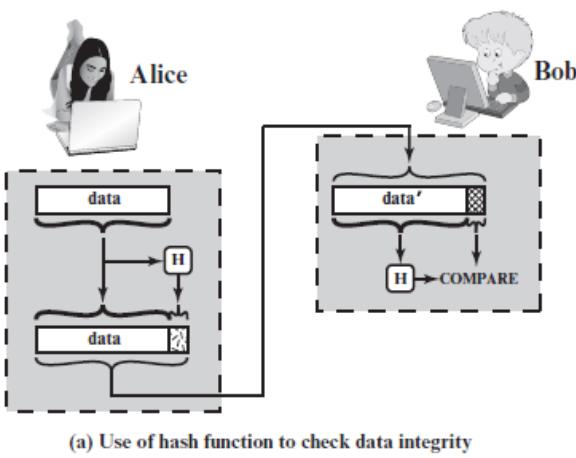
APPLICATIONS OF CRYPTOGRAPHIC HASH FUNCTIONS

• Hash Functions & Message Authentication

- ✓ Message authentication is a mechanism or service used to verify the integrity of a message.
- ✓ Message authentication assures that data received are exactly as sent (i.e., there is no modification, insertion, deletion, or replay).
- ✓ In many cases, there is a requirement that the authentication mechanism assures that purported identity of the sender is valid.
- ✓ When a hash function is used to provide message authentication, the hash function value is often referred to as a **message digest**.

Approach:

- The sender computes a hash value as a function of the bits in the message and transmits both the hash value and the message.
- The receiver performs the same hash calculation on the message bits and compares this value with the incoming hash value.
- If there is a mismatch, the receiver knows that the message (or possibly the hash value) has been altered.



The hash value must be transmitted in a secure fashion. That is, the hash value must be protected so that if an adversary alters or replaces the message, it is not feasible for adversary to also alter the hash value to fool the receiver.

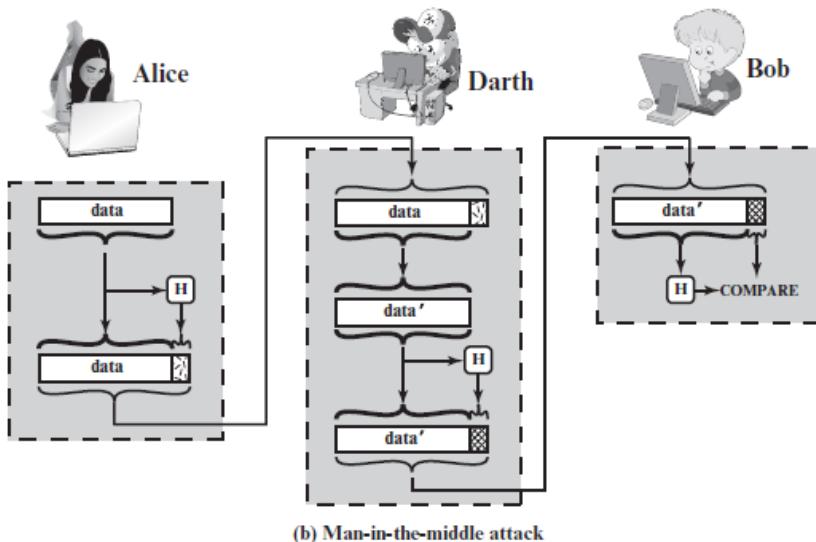
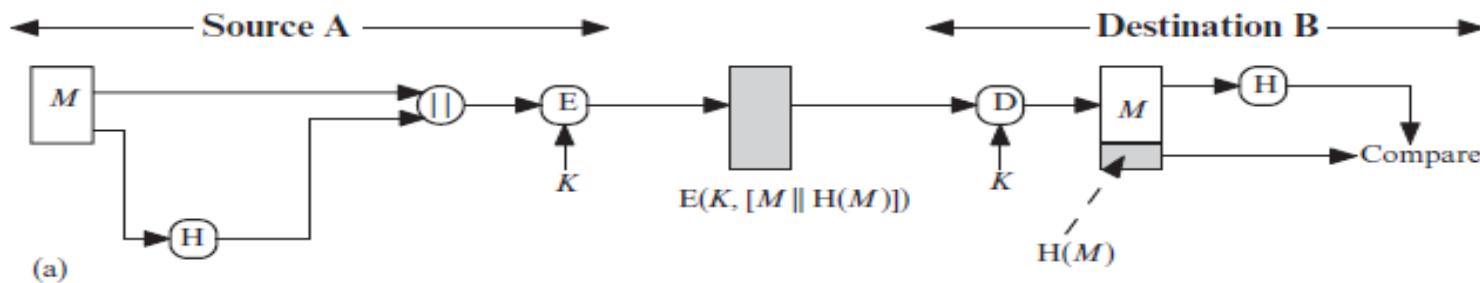
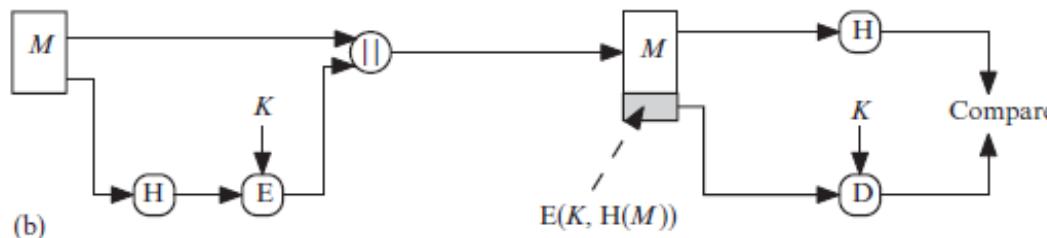


Figure 11.2 Attack Against Hash Function

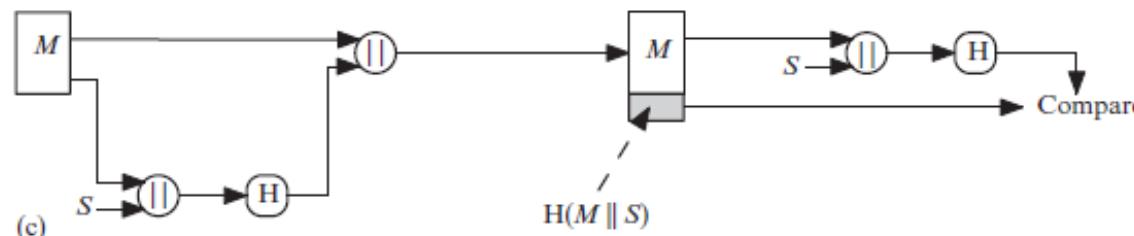
- **Different ways in which a hash code can be used to provide message authentication:**
- a. The message plus concatenated hash code is encrypted using symmetric encryption. Because only A and B share the secret key, the message must have come from A and has not been altered. The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.



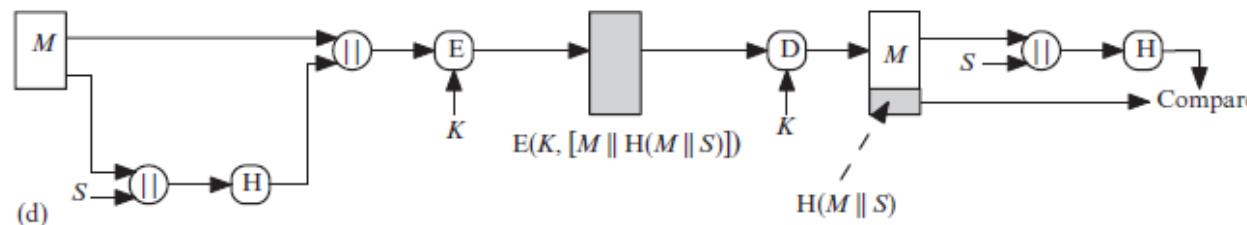
- b. Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.



- **Different ways in which a hash code can be used to provide message authentication:**
- c. It is possible to use a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value S . A computes the hash value over the concatenation of M and S and appends the resulting hash value to M . Because B possesses S , it can recompute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.



- d. Confidentiality can be added to the approach of method (c) by encrypting the entire message plus the hash code.



- More commonly, message authentication is achieved using a **message authentication code (MAC)**, also known as a **keyed hash function**.
- Typically, MACs are used between two parties that share a secret key to authenticate information exchanged between those parties.
- A MAC function takes as input a secret key and a data block and produces a hash value, referred to as the MAC, which is associated with the protected message.
- If the integrity of the message needs to be checked, the MAC function can be applied to the message and the result compared with the associated MAC value.
- An attacker who alters the message will be unable to alter the associated MAC value without knowledge of the secret key.

APPLICATIONS OF CRYPTOGRAPHIC HASH FUNCTIONS

• Hash Functions & Digital Signatures

- The operation of the digital signature is similar to that of the MAC. In the case of the digital signature, the hash value of a message is encrypted with a user's private key.
- Anyone who knows the user's public key can verify the integrity of the message that is associated with the digital signature.
- In this case, an attacker who wishes to alter the message would need to know the user's private key.
- **Different ways:**
 - a. The hash code is encrypted, using public-key encryption with the sender's private key. It also provides a digital signature, because only the sender could have produced the encrypted hash code. In fact, this is the essence of the digital signature technique.
 - b. If confidentiality as well as a digital signature is desired, then the message plus the private-key-encrypted hash code can be encrypted using a symmetric secret key. This is a common technique.

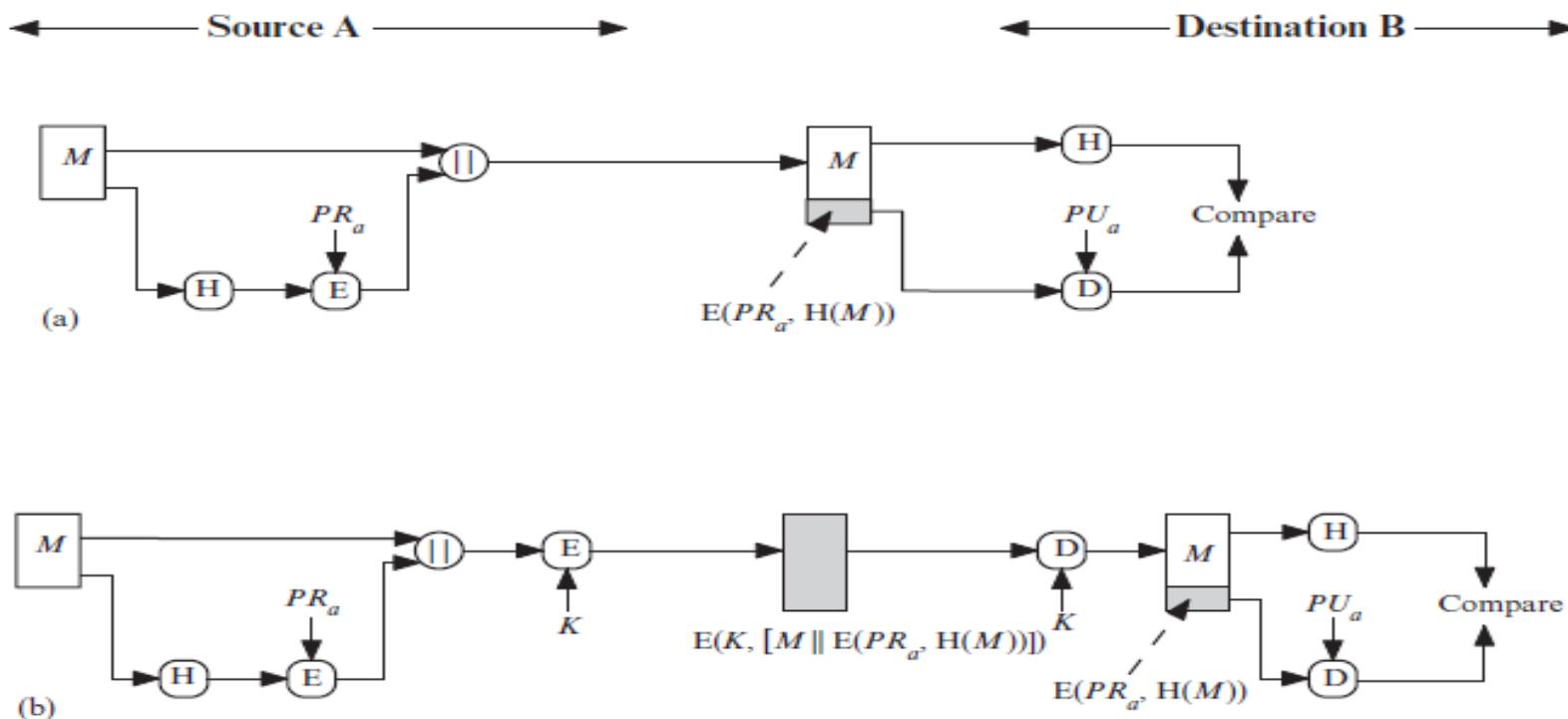


Figure 11.4 Simplified Examples of Digital Signatures

SIMPLE HASH FUNCTIONS

- bit-by-bit exclusive-OR (XOR)

Expressed as:

$$C_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im}$$

where

C_i = i th bit of the hash code, $1 \leq i \leq n$

m = number of n -bit blocks in the input

b_{ij} = i th bit in j th block

\oplus = XOR operation

- This operation produces a simple parity bit for each bit position and is known as a **longitudinal redundancy check**
- the probability that a data error will result in an unchanged hash value is 2^{-n} .
- in most normal text files, the high-order bit of each octet is always zero. So if a 128-bit hash value is used, instead of an effectiveness of 2^{128} , the hash function on this type of data has an effectiveness of 2^{112} .

input: 00111011 11101101 00101000 00101011 01011000 11001110

chunked:

00111011

11101101

00101000

00101011

01011000

11001110

XOR'd columns: 01010011 (output)

SIMPLE HASH FUNCTIONS

Improve

- perform a one-bit circular shift, or rotation , on the hash value after each block is processed.
- 1. Initially set the n-bit hash value to zero.
- 2. Process each successive n-bit block of data as follows:
 - a. Rotate the current hash value to the left by one bit.
 - b. XOR the block into the hash value.
- This has the effect of “randomizing” the input more completely and overcoming
- any regularities that appear in the i

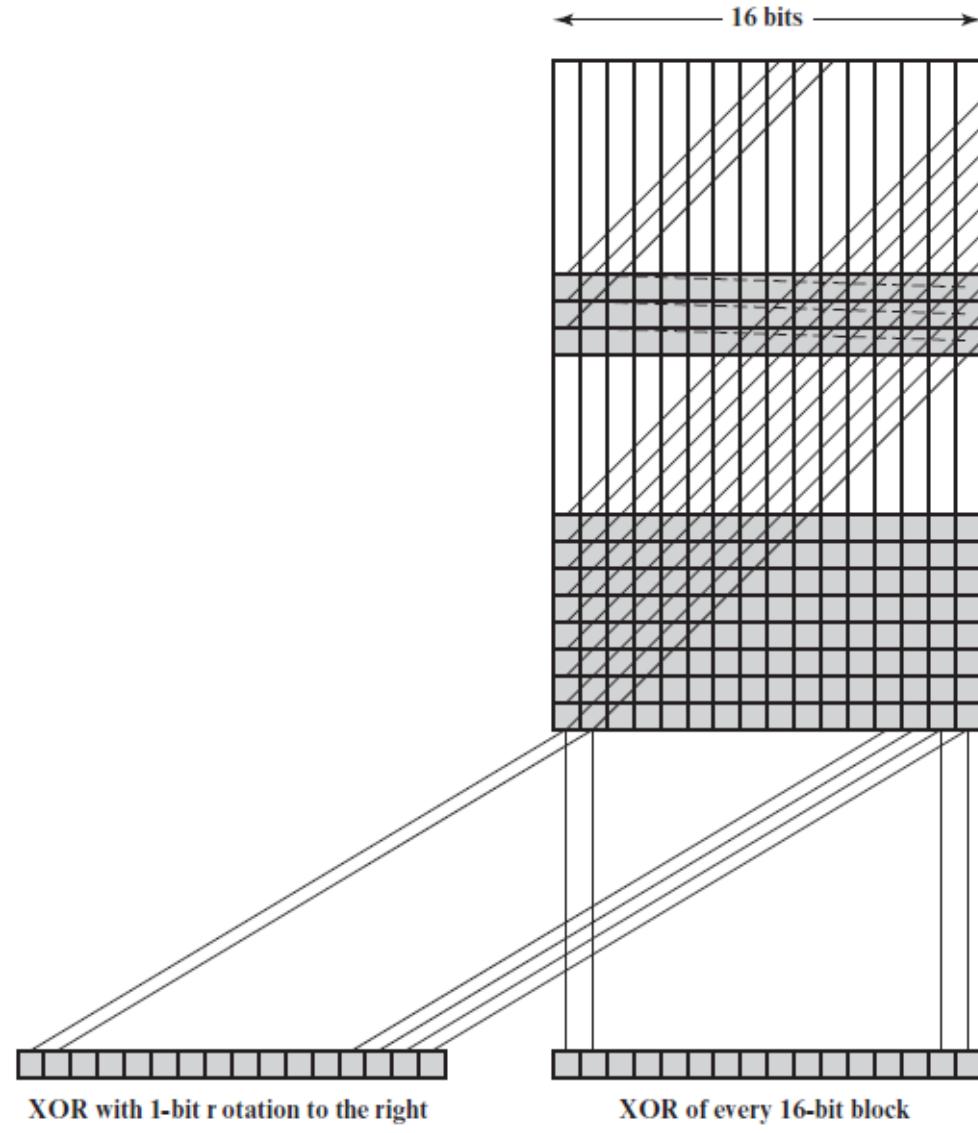


Figure 11.5 Two Simple Hash Functions

Requirements

- For a hash value $h = H(x)$, we say that x is the **preimage** of h . That is, x is a data block whose hash value, using the function H , is h .
- Because H is a many-to-one mapping, for any given hash value h , there will in general be multiple preimages. A **collision** occurs if we have $x \neq y$ and $H(x) = H(y)$
- Suppose the length of the hash code is n bits, and the function H takes as input messages or data blocks of length b bits with $b > n$. Then, the total number of possible messages is 2^b and the total number of possible hash values is 2^n . On average, each hash value corresponds to 2^{b-n} preimages.

Requirements for a Cryptographic Hash Function

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h , it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness.

Table 11.2 Hash Function Resistance Properties Required for Various Data Integrity Applications

	Preimage Resistant	Second Preimage Resistant	Collision Resistant
Hash + digital signature	yes	yes	yes*
Intrusion detection and virus detection		yes	
Hash + symmetric encryption			
One-way password file	yes		
MAC	yes	yes	yes*

*Resistance required if attacker is able to mount a chosen message attack

Secure Hash Algorithm (SHA)

- SHA was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993.
- When weaknesses were discovered in SHA, now known as **SHA-0**, a revised version was issued as FIPS 180-1 in 1995 and is referred to as **SHA-1**.
- The actual standards document is entitled “Secure Hash Standard.”

Secure Hash Algorithm (SHA)

Table 11.3 Comparison of SHA Parameters

Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Note: All sizes are measured in bits.

SHA-512 Logic

- The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512-bit message digest.
- The input is processed in 1024-bit blocks.

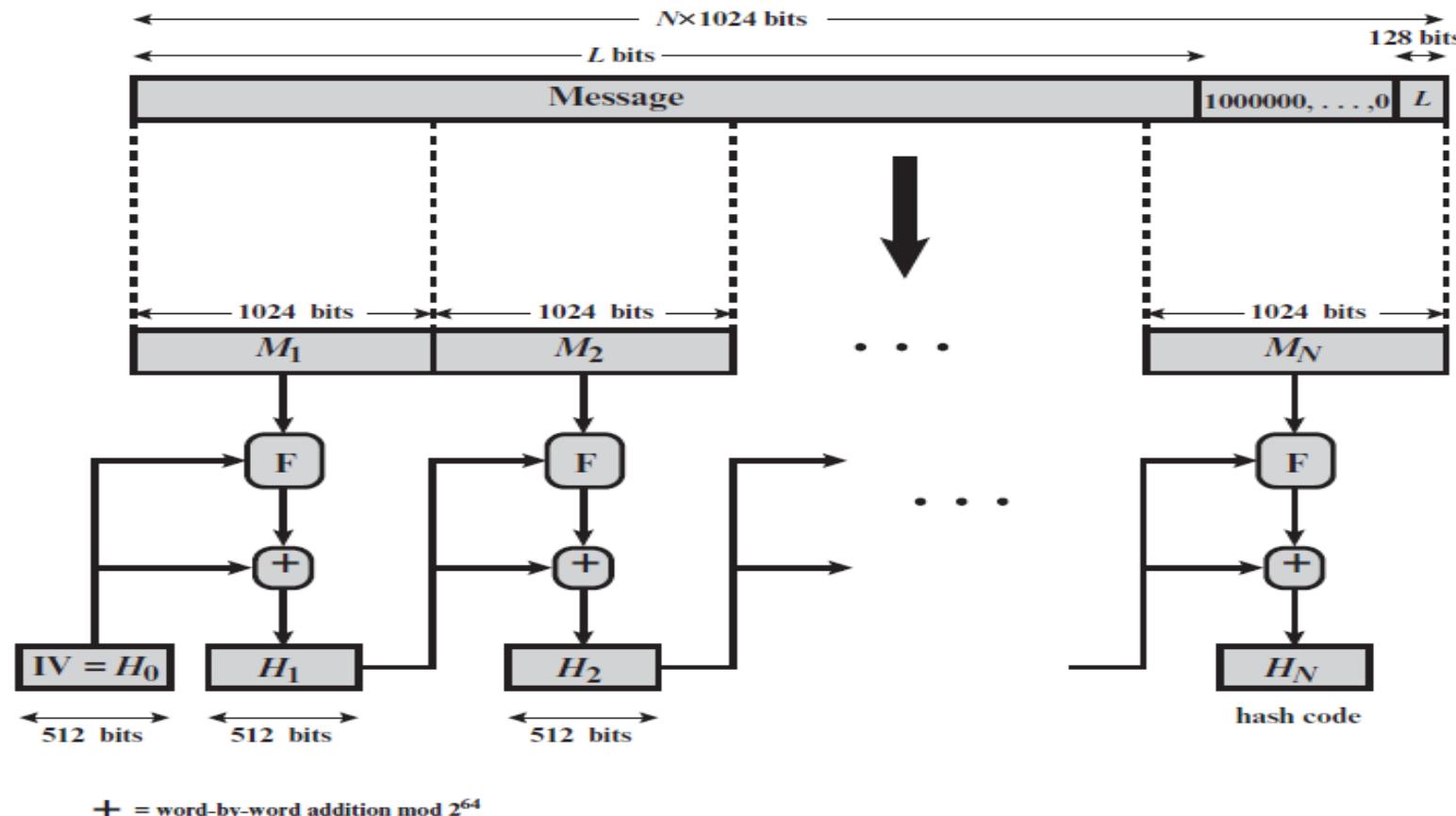


Figure 11.9 Message Digest Generation Using SHA-512

SHA-512 Logic

- **Step 1: Append padding bits.**

The message is padded so that its length is congruent to 896 modulo 1024 Padding is always added, even if the message is already of the desired length. Thus, the number of padding bits is in the range of 1 to 1024. The padding consists of a single 1 bit followed by the necessary number of 0 bits.

- **Step 2 Append length.**

A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer and contains the length of the original message in bits (before the padding). The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length.

- **Step 3 Initialize hash buffer.**

A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers (hexadecimal values):

a = 6A09E667F3BCC908	e = 510E527FADE682D1
b = BB67AE8584CAA73B	f = 9B05688C2B3E6C1F
c = 3C6EF372FE94F82B	g = 1F83D9ABFB41BD6B
d = A54FF53A5F1D36F1	h = 5BE0CD19137E2179

SHA-512 Logic

- **Step 4 Process message in 1024-bit (128-byte) blocks.**

The heart of the algorithm is a module that consists of 80 rounds

- **Step 5 Output.**

After all N 1024-bit blocks have been processed, the output from the N th stage is the 512-bit message digest.

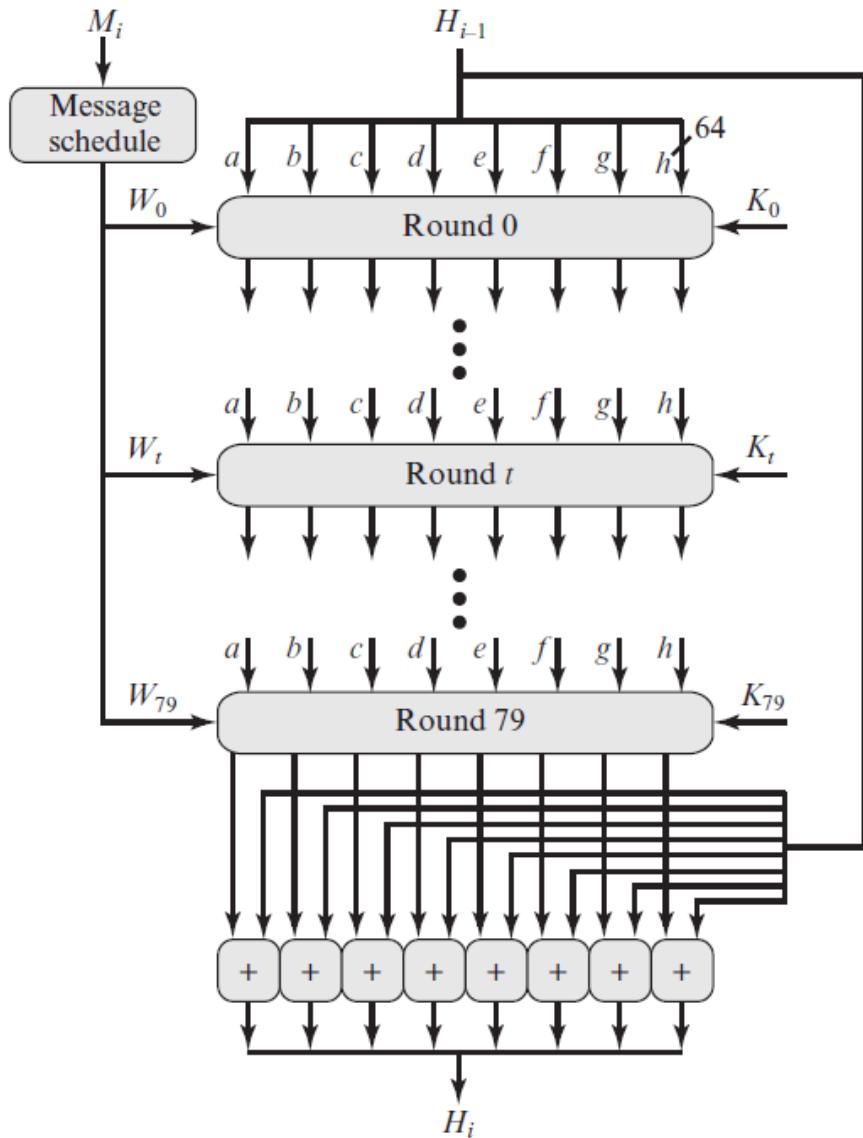


Figure 11.10 SHA-512 Processing of a Single 1024-Bit Block

Each round takes as input the 512-bit buffer value, abcdefgh, and updates the contents of the buffer.

At input to the first round, the buffer has the value of the intermediate hash value, H_{i-1} .

Each round t makes use of a 64-bit value W_t , derived from the current 1024-bit block being (M_i) .

These values are derived using a message schedule.

Each round also makes use of an additive constant K_t , where $0 \dots t \dots 79$ indicates one of the 80 rounds.

The output of the eightieth round is added to the input to the first round (H_{i-1}) to produce H_i .

The addition is done independently for each of the eight words in the buffer with each of the corresponding words in H_{i-1} , using addition modulo 2^{64} .

We can summarize the behavior of SHA-512 as follows:

$$H_0 = \text{IV}$$

$$H_i = \text{SUM}_{64}(H_{i-1}, \text{abcdefg}_i)$$

$$MD = H_N$$

where

IV = initial value of the abcdefgh buffer, defined in step 3

abcdefg_i = the output of the last round of processing of the i th message block

N = the number of blocks in the message (including padding and length fields)

SUM_{64} = addition modulo 2^{64} performed separately on each word of the pair of inputs

MD = final message digest value

SHA-512 Round Function

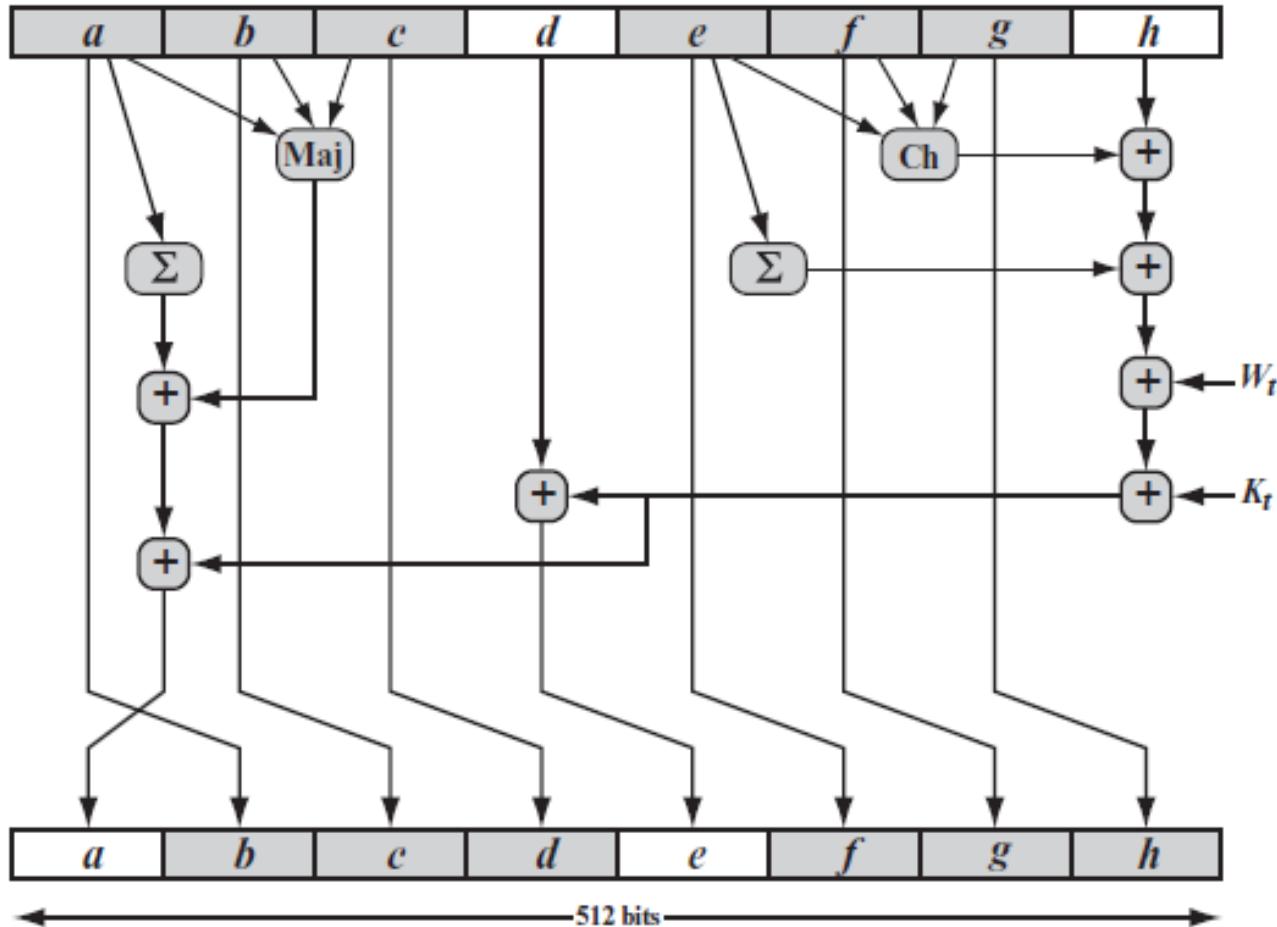


Figure 11.11 Elementary SHA-512 Operation (single round)

SHA-512 Round Function Summarized as follows:

$$T_1 = h + \text{Ch}(e, f, g) + (\sum_1^{512} e) + W_t + K_t$$

$$T_2 = (\sum_0^{512} a) + \text{Maj}(a, b, c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

where

t = step number; $0 \leq t \leq 79$

$\text{Ch}(e, f, g) = (e \text{ AND } f) \oplus (\text{NOT } e \text{ AND } g)$
the conditional function: If e then f else g

$\text{Maj}(a, b, c) = (a \text{ AND } b) \oplus (a \text{ AND } c) \oplus (b \text{ AND } c)$
the function is true only if the majority (two or three) of the arguments are true

$(\sum_0^{512} a) = \text{ROTR}^{28}(a) \oplus \text{ROTR}^{34}(a) \oplus \text{ROTR}^{39}(a)$

$(\sum_1^{512} e) = \text{ROTR}^{14}(e) \oplus \text{ROTR}^{18}(e) \oplus \text{ROTR}^{41}(e)$

$\text{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

SHA-512 Round Function Summarized as follows (Contd..):

W_t = a 64-bit word derived from the current 1024-bit input block

K_t = a 64-bit additive constant

$+$ = addition modulo 2^{64}

Two observations can be made about the round function.

1. Six of the eight words of the output of the round function involve simply permutation (b, c, d, f, g, h) by means of rotation. This is indicated by shading in Figure 11.11.
2. Only two of the output words (a, e) are generated by substitution. Word e is a function of input variables (d, e, f, g, h), as well as the round word W_t and the constant K_t . Word a is a function of all of the input variables except d , as well as the round word W_t and the constant K_t .

How the 64-bit word values W_t are derived from the 1024-bit message. ?

- The first 16 values of W_t are taken directly from the 16 words of the current block. The remaining values are defined as

$$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}$$

where

$$\sigma_0^{512}(x) = \text{ROTR}^1(x) \oplus \text{ROTR}^8(x) \oplus \text{SHR}^7(x)$$

$$\sigma_1^{512}(x) = \text{ROTR}^{19}(x) \oplus \text{ROTR}^{61}(x) \oplus \text{SHR}^6(x)$$

$\text{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

$\text{SHR}^n(x)$ = right shift of the 64-bit argument x by n bits with padding by zeros on the left

$+$ = addition modulo 2^{64}

How the 64-bit word values W_t are derived from the 1024-bit message. ?

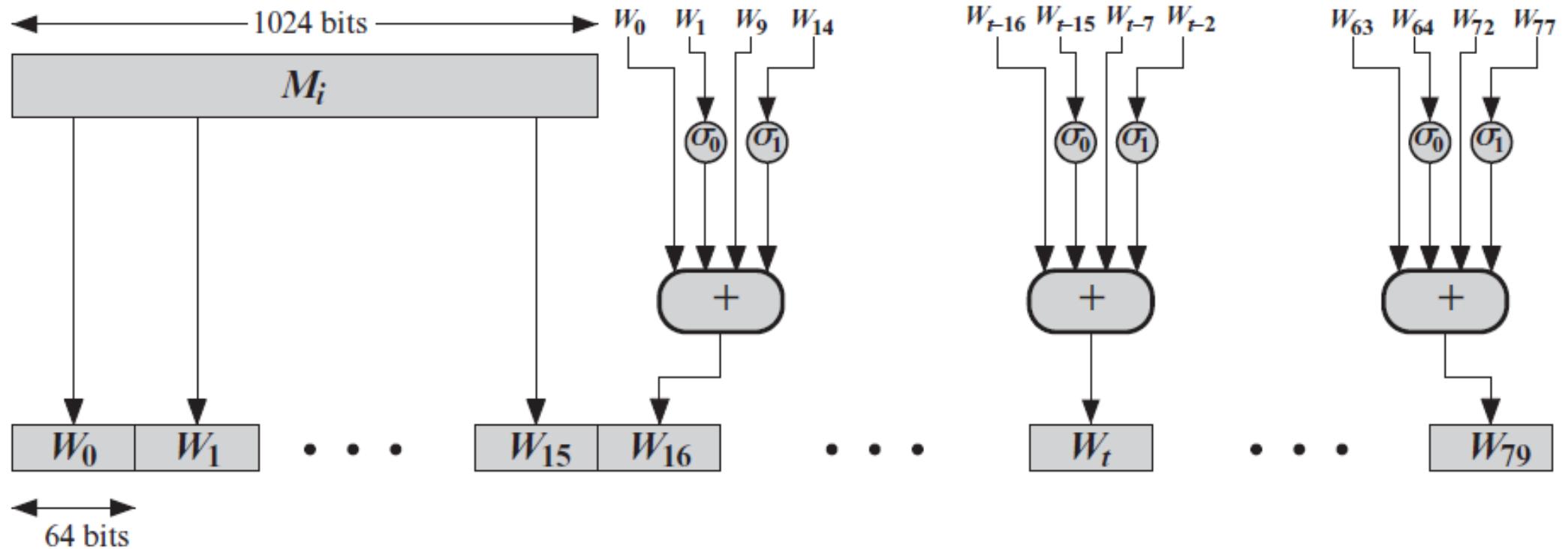


Figure 11.12 Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

Data Security and Privacy

DSE 3258

L10 —Introduction to Data Privacy

Introduction to Data Privacy

- **What is data Privacy?**
- Data privacy, also called information privacy, is an aspect of data protection that addresses the proper storage, access, retention, immutability and security of sensitive data.
- Data privacy is typically associated with the proper handling of personal data or personally identifiable information (PII), such as names, addresses, Social Security numbers and credit card numbers. However, the idea also extends to other valuable or confidential data, including financial data, intellectual property and personal health information.
- Data privacy is not a single concept or approach. Instead, it's a discipline involving rules, practices, guidelines and tools to help organizations establish and maintain required levels of privacy compliance.

Introduction to Data Privacy

As the laws and regulations related to Privacy and Data Protection are constantly changing, it is important to keep abreast of any changes in the law and continually reassess compliance with data privacy and security regulations.

Introduction to Data Privacy

Data privacy issues can arise in response to information from a wide range of sources, such as:

- ▶ Healthcare records
 - ▶ Criminal justice investigations and proceedings
 - ▶ Financial institutions and transactions
 - ▶ Biological traits, such as genetic material
 - ▶ Residence and geographic records
 - ▶ Ethnicity
 - ▶ Privacy breach
-

Introduction to Data Privacy

Data privacy issues can arise in response to information from a wide range of sources, such as:

- ▶ Healthcare records
- ▶ Criminal justice investigations and proceedings
- ▶ Financial institutions and transactions
- ▶ Biological traits, such as genetic material
- ▶ Residence and geographic records
- ▶ Ethnicity
- ▶ Privacy breach

Data Security Vs Privacy

- ▶ Data security is commonly referred to as the confidentiality, availability, and integrity of data.
- ▶ Data privacy is suitably defined as the appropriate use of data.
- ▶ When companies and merchants use data or information that is provided or entrusted to them, the data should be used according to the agreed purposes.
- ▶ Companies need to enact a data security policy for the sole purpose of ensuring data privacy or the privacy of their consumers' information.

Types of privacy attacks

- In privacy-related attacks, the goal of an adversary is to gain knowledge that was not intended to be shared.
- There are several types based on different domains.
- **Linkage Attacks**
 - Linkage attacks use more than one data source and link them together to re-identify individuals or to gain more information to identify individuals. In general, linkage attacks are successful when attackers have an auxiliary data source that connects easily with another dataset.
- **Singling Out Attacks**
 - Singling out attacks work by singling out an individual in a public release and attempting to gather more information about them via the same dataset or via other sources. These attacks can also be performed in reverse, by bringing information about an individual to a released dataset and attempting to deduce whether this person is included and can be identified.

Types of privacy attacks

- **Membership Inference Attack**

- In a membership inference attack, the attacker tries to learn if a person was a member of the training data.

- **Inferring Sensitive Attributes**

- Membership inference attacks can be generalized to describe group attributes of the training data population, called an *attribute privacy attack*. In this case, the attacker wants to learn things about the underlying population and uses the same technique to test theories about types of people who might be represented in the training dataset. This attack reveals sensitive group details.

Types of privacy attacks

- **Identity theft:** This involves stealing someone's personal information, such as their name, date of birth, Social Security number, and financial information, with the intent of assuming their identity or using the information for fraudulent purposes.
- Identity theft is committed in many different ways and its victims are typically left with damage to their credit, finances, and reputation.
- To find such information, they may search the hard drives of stolen or discarded computers; hack into computers or computer networks of organizations and corporations; access computer-based public records; use information-gathering malware to infect computers; browse social networking sites; or use deceptive emails or text messages.

Types of privacy attacks

- **Data breaches:** This occurs when an unauthorized person gains access to sensitive information that is stored in a database or computer system. This can result in the exposure of personal information, such as credit card numbers, passwords, and other sensitive data.
- **Phishing :** Phishing refers to an attempt to steal sensitive information, typically in the form of usernames, passwords, credit card numbers, bank account information or other important data in order to utilize or sell the stolen information. This involves tricking individuals into providing their personal information by posing as a legitimate company or organization. Phishing attacks are often carried out through email, social media, or other forms of communication.

Types of privacy attacks

- **Malware attacks:** This involves infecting a computer or device with malicious software that can steal personal information, monitor user activity, or perform other harmful actions.
- **Eavesdropping:** This involves intercepting and listening to private conversations or communications, such as phone calls or text messages, without the knowledge or consent of the individuals involved
- **Password cracking (also called, password hacking)** is an attack vector that involves hackers attempting to crack or determine a password. Password hacking uses a variety of programmatic techniques and automation using specialized tools.
- **Surveillance:** This involves monitoring an individual's activities, either physically or digitally, without their knowledge or consent. This can include tracking their location, online activity, or other behaviors.

Data linking and profiling

- **Data profiling** : is the process of examining, analyzing, and creating useful summaries of data. The process yields a high-level overview which aids in the discovery of data quality issues, risks, and overall trends.
- **Data profiling** is the systematic process of determining and recording the characteristics of data sets. We can also think of it as building a metadata catalog that summarizes the essential characteristics.
 - This can include information such as demographic data, buying habits, online activity, and other personal information.
 - Profiling can be used for a variety of purposes, such as marketing, advertising, and personalization of services.

Uses of data profiling

- **Query optimization**
 - ✓ Data profiling provides information on the characteristics of a database, such as rows, columns, average values, and more.
 - ✓ Statistics about each database helps to estimate the query design, considerations, and implementation plan. As a result, we can optimize your queries for better performance.
- **Data integration**
 - ✓ To integrate multiple datasets, we first need to understand the datasets and their relationships. This is crucial to understand how to link datasets, what's the best way to link them, do we need to take into account different conventions such as name or unit of measurement, and so on.
- **Scientific data management**
 - ✓ Before importing raw data into your databases, it's important to understand the nature of that data. That's where data profiling can help. After profiling these datasets, you can develop a plan to extract that data and adopt the appropriate schema.
- **Data analytics**
 - ✓ Any analysis or data mining starts with data profiling. Data profiling gives an initial high-level understanding of the dataset and its characteristics so that you can choose the right algorithms.

Uses of data profiling

- **Project management**

- Project management
 - ✓ Taking data-driven decisions requires a solid understanding of the data you have, and the information you need for the project. With data profiling, you can take stock of your data, its quality, completeness, and credibility. You can also determine whether you have all the data you need to make your project work.

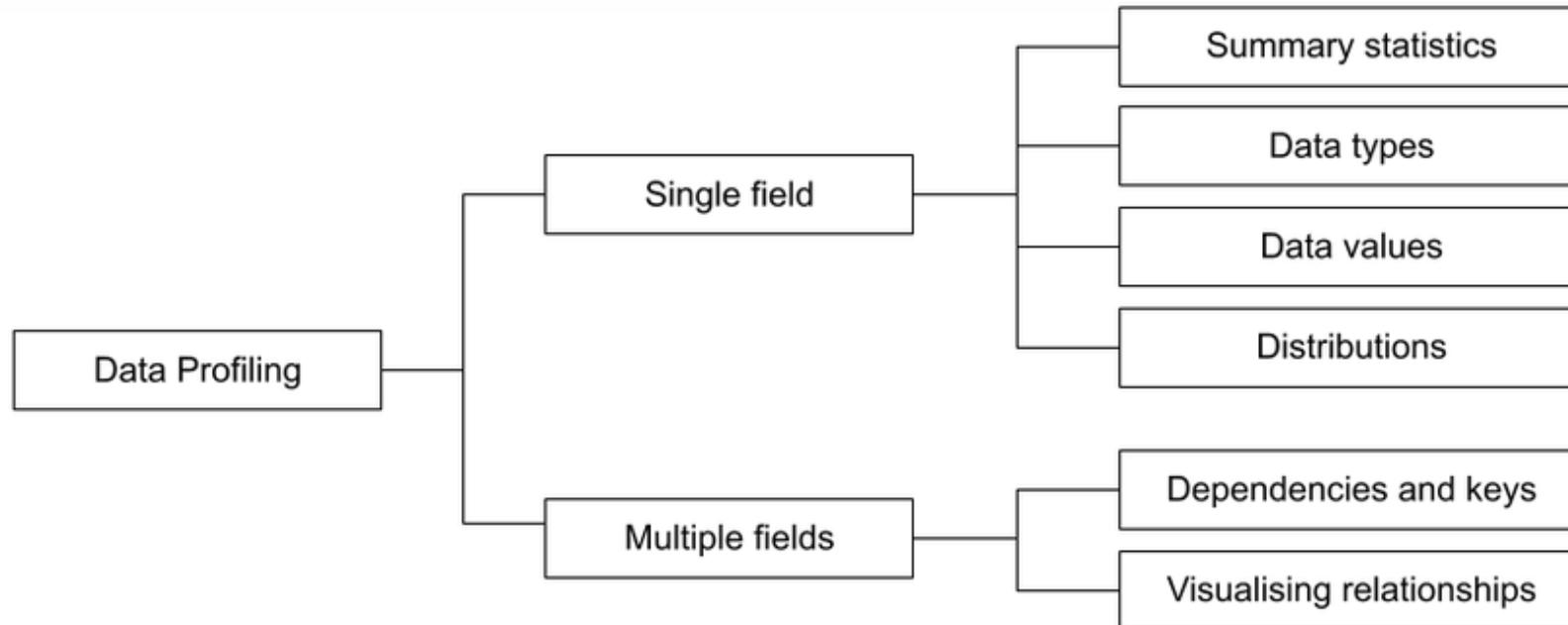
- **Data discovery**

- Data discovery
 - ✓

- Data discovery
 - ✓ Having data available to be used broadly across an organization requires that data be easily accessible, searchable, and understandable. Data profiling can help by enabling you to compile the metadata needed, along with descriptive summaries and metrics for better context.

data profiling: Techniques

- According to Felix Naumann's data profiling can be done using single or multiple fields.



■ Single column profiling

- ✓ Most basic form of data profiling
- ✓ Assumption: All values are of same type
- ✓ Assumption: All values have some common properties
 - ✓ – That are to be discovered
- ✓ Often part of the basic statistics gathered by DBMS

■ Multicolumn profiling

- ✓ Discover joint properties
- ✓ Discover dependencies

Single field profiling is the most basic form of profiling that assumes all fields are of the same type and share common properties. This type of profiling helps you discover:

- **Summary statistics:** This includes count of data and mathematical aggregations such as maximum, minimum, and mean values.
- **Data types:** This involves determining whether the data is categorical, continuous, and exhibits any patterns. Simple data types include strings, numbers, and timestamps, whereas more complex types include XML
- **Data values:** This means identifying the characteristics and patterns in data values. Examples include address fields, cities, ID strings, and more. Profiling data values also helps you assess your data against known business rules.
- **Distributions:** Visualizing data distribution is useful in spotting outliers. For categorical data, you can see counts per category. Meanwhile, for numerical data, you can plot histograms and note characteristics like skewness, presence of outliers etc..

Multi-field profiling explores the relationship between fields to discover:

- **Inclusion dependencies, keys, and functional dependencies:** With profiling, you can find out if the values in one field are a subset of values in other fields.
- **Visualize numerical relationships:** Profiling helps explore the relationships between numerical fields using pair plots, cross-correlation heat maps, or tables of correlations between fields. These visualizations provide a quick overview of the relationships each data set has with other assets.

Data profiling tools and algorithms

IBM InfoSphere Information Analyzer

□ <http://www.ibm.com/software/data/infosphere/information-analyzer/>

■ Oracle Enterprise Data Quality

□ <http://www.oracle.com/us/products/middleware/data-integration/enterprise-data-quality/overview/index.html>

■ Talend Data Quality

□ <http://www.talend.com/products/data-quality>

■ Ataccama DQ Analyzer

□ <http://www.attaccama.com/en/products/dq-analyzer.html>

■ SAP BusinessObjects Data Insight and SAP BusinessObjects Information Steward

□ <http://www.sap.com/germany/solutions/sapbusinessobjects/large/eim/datainsight/index.epx>

□ <http://www.sap.com/germany/solutions/sapbusinessobjects/large/eim/information-steward/index.epx>

■ Informatica Data Explorer

□ <http://www.informatica.com/us/products/data-quality/data-explorer/>

■ Microsoft SQL Server Integration Services Data Profiling Task and Viewer

□ <http://msdn.microsoft.com/en-us/library/bb895310.aspx>

■ Trillium Software Data Profiling

□ <http://www.trilliumsoftware.com/home/products/data-profiling.aspx>

■ CloverETL Data Profiler

□ <http://www.cloveretl.com/products/profiler>

■ OpenRefine

□ <http://www.openrefine.org>

Data linking

- **Data linking** refers to the process of connecting two or more sets of data that were previously thought to be unrelated.
 - Can involve combining data from different sources, such as databases, social media, and online activity, to create a more comprehensive picture of an individual's behavior or preferences.
- Data linking is the process of collating information from different sources in order to create a more valuable and helpful data set. The linking of information about the same person or entity from disparate sources allows, among other things, the construction of a chronological sequence of events. This information is of immense value at the policy level to derive meaningful decisions.

Ways to link data sets

1. Unique identifier

A unique identifier is available on each data set that establishes the links between these data sets. It is also called deterministic or exact linking because the unique identifiers either match completely, or do not at all.

2. Linkage key

- ✓ When a unique identifier is not available, or there isn't enough quality in the data to rely on, another approach is used called linkage key.
- ✓ The linkage key works like a substitute for the unique identifier in this method.
- ✓ This key is created using information like name and address available on both data sets.
- ✓ These linkage keys maintain the privacy of the person or entity as the key is used in place of the name and address.

Ways to link data sets

- **3. Probabilistic linking**
- This is another style of data linking, and it is used when a unique identifier is unavailable. It is based on the probability that the pair of records, taken from one data set, refers to the same entity or person. In this method, advanced data linking software is used to obtain accurate results.
- **4. Statistical linking**
- This technique combines records similar to the entity but not necessarily the same person or organization. This kind of data linking may not give the most accurate results but does provide a pattern or trend from the given information or statistics.

Access control models

- Access control is the act of maintaining building security by strategically controlling who can access your property and when
- Access control is the process of controlling who has access to resources or information within an organization's computer systems.
- Access control mechanisms are designed to ensure that only authorized individuals are able to access resources, while preventing unauthorized access.

Access Control Principles

- **Authentication:** Verification that the credentials of a user or other system entity are valid.
- **Authorization:** The granting of a right or permission to a system entity to access a system resource. This function determines who is trusted for a given purpose.
- **Audit:** An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy and procedures.
- An access control mechanism mediates between a user and system resources, such as applications, files, and databases.

Access Control Principles

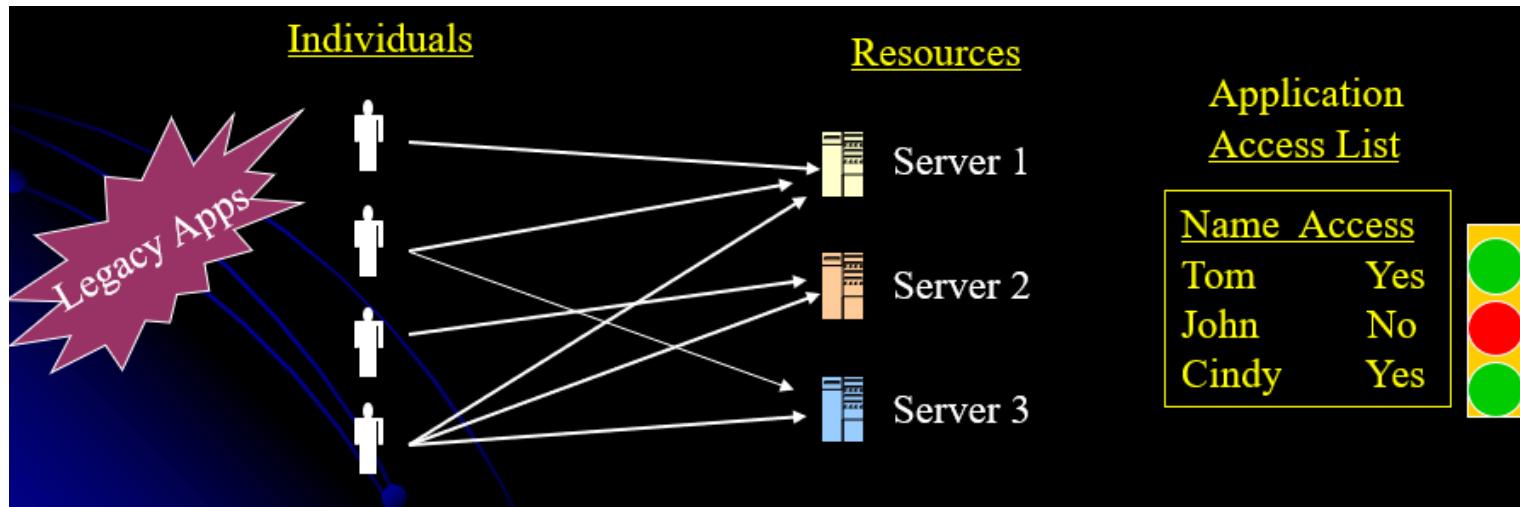
- The system must first authenticate an entity seeking access.
- Typically, the authentication function determines whether the user is permitted to access the system at all. Then the access control function determines if the specific requested access by this user is permitted.
- A security administrator maintains an authorization database that specifies what type of access to which resources is allowed for this user.
- The access control function consults this database to determine whether to grant access. An auditing function monitors and keeps a record of user accesses to system resources.

Access Control Policies

- An access control policy, which can be embodied in an authorization database, dictates what types of access are permitted, under what circumstances, and by whom.
- Based on these we have following different types of access control models.
 1. Discretionary access control (DAC)
 2. Mandatory access control (MAC)
 3. Role-based access control (RBAC)
 4. Rule-based access control (RuBAC)

Discretionary access control (DAC)

- Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do.
- Least restrictive model
- Every object has a owner, who has complete control over the object.
- Allows the owner of the resource to specify which subjects can access which resources
- Access control is at the discretion of the owner
- DAC defines access control policy – That restricts access to files and other system resources based on identity
- This gives DAC two major weaknesses.
 - First, it gives the end-user complete control to set security level settings for other users which could result in users having higher privileges than they're supposed to.
 - Secondly, and worse, the permissions that the end-user has are inherited into other programs they execute. This means the end-user can execute malware without knowing it and the malware could take advantage of the potentially high-level privileges the end-user possesses.

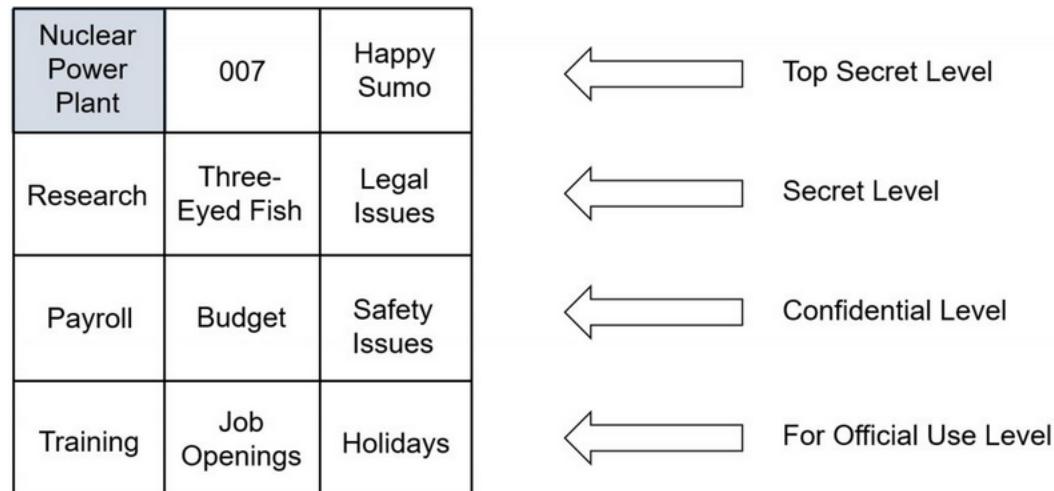


The Mandatory Access Control, or MAC

- Most restrictive access control
- MAC assigns users' access controls strictly according to custodian's desires and end user has no freedom to set any controls
- Two key elements to MAC:
 - labels - every entity is an object (laptops, files, projects, and so on) and assigned classification label (confidential, secret, and top secret) while subjects assigned privilege label (a clearance)
 - levels - hierarchy based on labels is also used, both for objects and subjects (top secret higher level than secret)

The Mandatory Access Control, or MAC

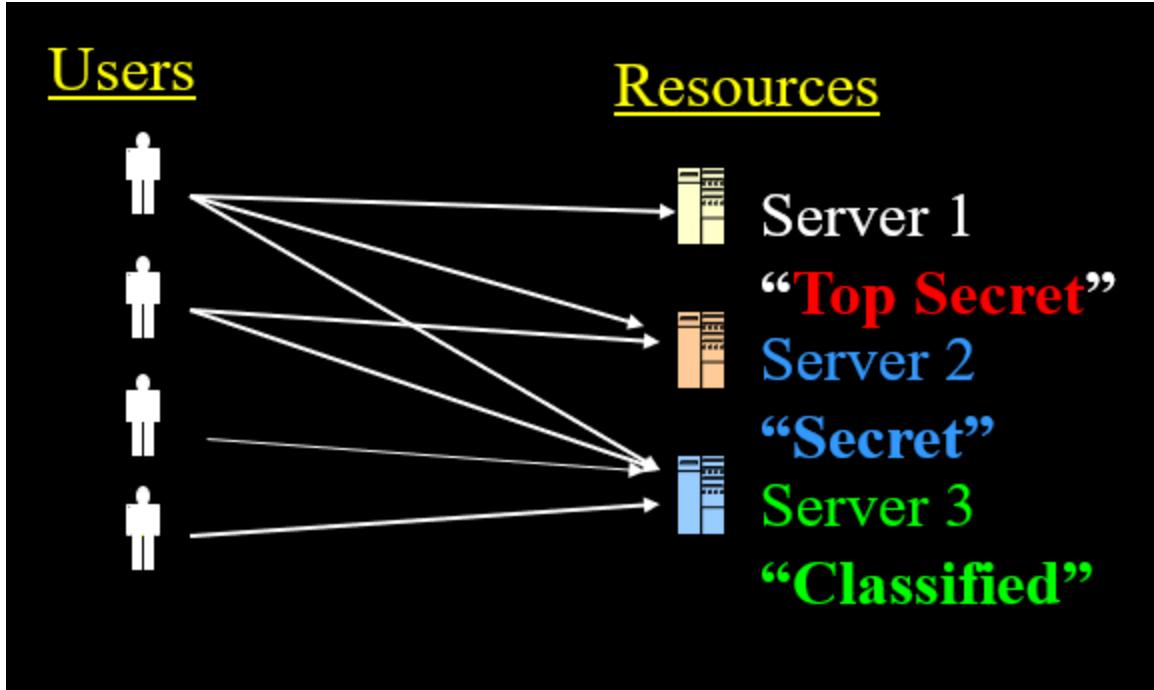
- Major implementations
- lattice model - The MAC model uses different levels of security to classify both the users and the data. These levels are defined in a lattice. The lattice can be a complex relationship between different ordered sets of labels. These labels define the boundaries for the security levels.
- In this type of label-based mandatory access control model, a lattice is used to define the levels of security that an object may have and that a subject may have access to. The subject is only allowed to access an object if the security level of the subject is greater than or equal to that of the object.



MAC model lattice

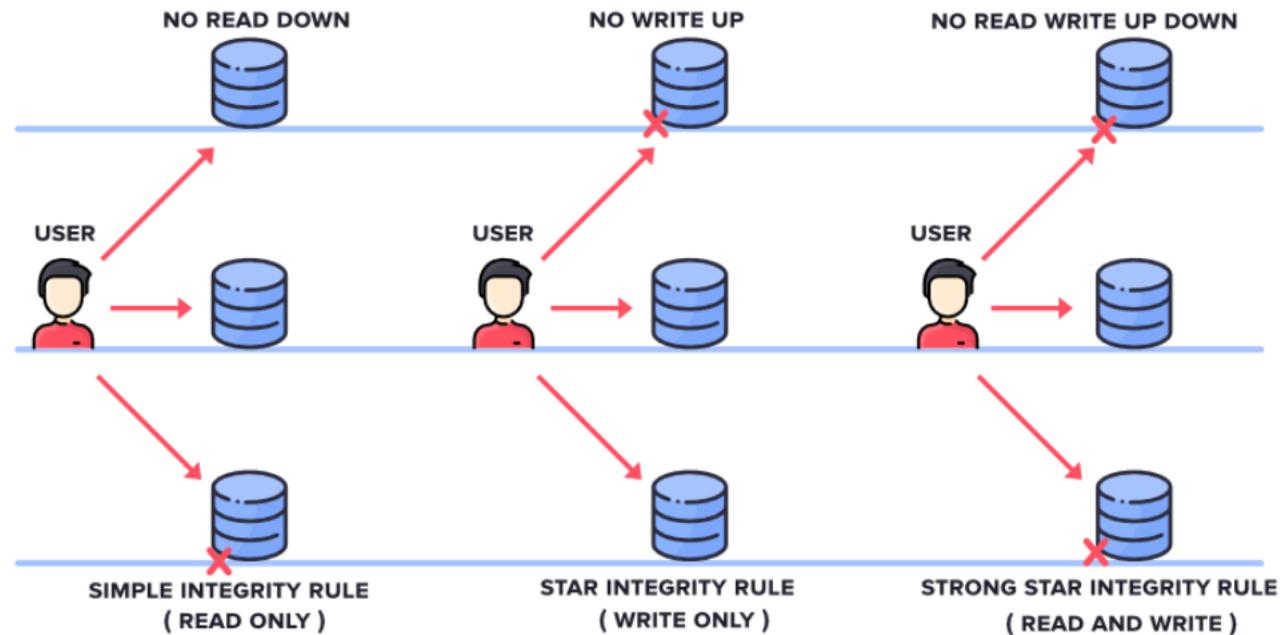
The Mandatory Access Control, or MAC

- Major implementations
 - **biba integrity model** –
 - This is used to maintain the **Integrity** of Security. Here, the classification of Subjects(Users) and Objects(Files) are organized in a non-discretionary fashion, with respect to different layers of secrecy.
- **It has mainly 3 Rules:**
- **SIMPLE INTEGRITY RULE:** Simple Integrity Rule states that the Subject can only **Read** the files on the Same Layer of Secrecy and the Upper Layer of Secrecy but not the Lower Layer of Secrecy, due to which we call this rule as **NO READ DOWN**
- **STAR INTEGRITY RULE:** Star Integrity Rule states that the Subject can only **Write** the files on the Same Layer of Secrecy and the Lower Layer of Secrecy but not the Upper Layer of Secrecy, due to which we call this rule as **NO WRITE-UP**
- **STRONG STAR INTEGRITY RULE**



The Mandatory Access Control, or MAC

- Major implementations
 - **biba integrity model –**

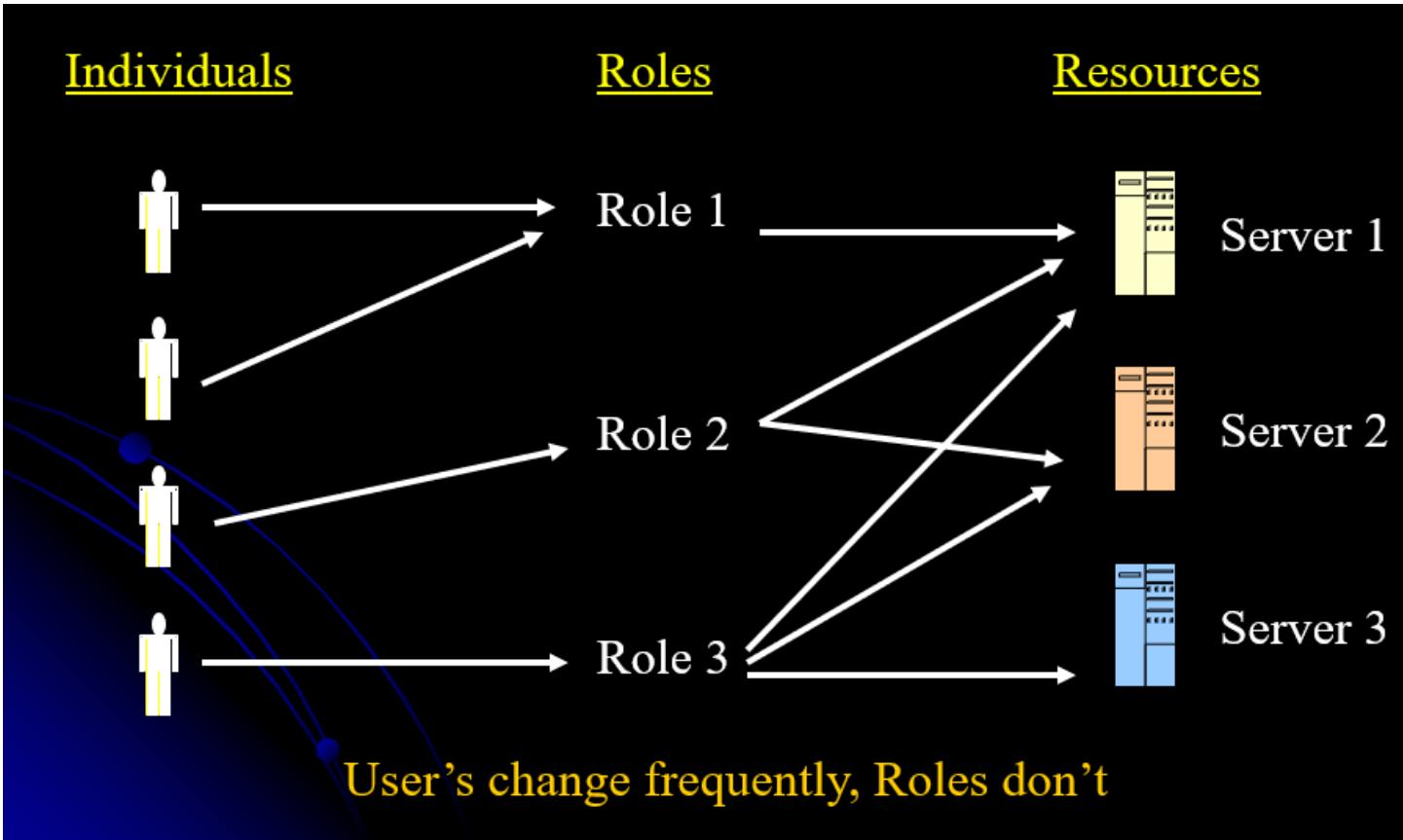


The Mandatory Access Control, or MAC

- Major implementations
 - **bell-lapadula** - similar to lattice model but subjects may not create new object or perform specific functions on lower level objects.
 - On the other hand, is a setup where a user at a higher level (i.e. Top Secret) can only write at that level and no lower (called “write up”), but can also read at lower levels (called “read down”).

Role-based access control (RBAC)

- A user has access to an object based on the assigned role.
 - Roles are defined based on job functions.
 - Permissions are defined based on authorities and responsibilities of a job.
 - Operations on an object are invoked based on the permissions.
-
- Model provides access control based on the position an individual fills in an organization. So, instead of assigning Alice permissions as a security manager, the position of security manager already has permissions assigned to it.
 - In essence, Alice would just need access to the security manager profile.
 - RBAC makes life easier for the system administrator of the organization. The big issue with this access control model is that if Alice requires access to other files, there has to be another way to do it since the roles are only associated with the position; otherwise, security managers from other organizations could possibly get access to files for which they are unauthorized.



Rule-Based Access Control

- The fourth and final access control model is Rule-Based Access Control, also with the acronym RBAC or RB-RBAC.
- Rule-Based Access Control will dynamically assign roles to users based on criteria defined by the custodian or system administrator.
- For example, if someone is only allowed access to files during certain hours of the day, Rule-Based Access Control would be the tool of choice.
- The additional “rules” of Rule-Based Access Control requiring implementation may need to be “programmed” into the network by the custodian or system administrator in the form of code versus “checking the box.”

Privacy policies, their specifications, privacy policy languages

- Privacy policies are statements or documents that detail how an organization collects, uses, and manages personal information that it collects from individuals.
- A privacy policy is a legal document that discloses the way a party gathers, uses, discloses, and manages a customer or client's data. It fulfils a legal requirement to protect a customer or client's privacy.
- Privacy policies are typically posted on an organization's website and are intended to inform individuals about their privacy rights and how their personal information will be used.

Privacy policies, their specifications, privacy policy languages

- Such privacy policy must provide the following:
 1. clearly and easily accessible statements of its practices and policies;
 2. clearly state the type of personal and sensitive personal data or information collected by the business;
 3. purpose of collection and usage of such information;
 4. about disclosure of information including sensitive personal data or information collected; and
 5. Reasonable security practices and procedures adopted by it.

- The **specifications for privacy policies** may vary depending on the organization and the nature of its business:
 - **Information collected:** The policy should identify the types of personal information that the organization collects, such as name, email address, and phone number.
 - **Use of information:** The policy should detail how the organization will use the personal information it collects, such as for marketing purposes or to fulfill a customer order.
 - **Sharing of information:** The policy should outline with whom the organization shares personal information, such as third-party service providers or government agencies.
 - **Security:** The policy should describe the measures the organization takes to protect personal information from unauthorized access or disclosure.
 - **User rights:** The policy should describe the rights of individuals with respect to their personal information, such as the right to access and correct personal information.

- **Elements of a privacy policy:**

The following are the main elements which shall be consisted of a privacy policy, are as follows:

- Consent: The most crucial component of a privacy policy is 'consent'.
- Purpose of information collected.
- Disclosure of information.
- Security practice

Policy language

- There are several privacy policy languages that organizations can use to create their privacy policies, including:
- **Natural language:** This is the most common privacy policy language and is written in plain English or other natural language.
- **Legalese:** This language is more formal and technical and is typically used in legal documents.
- **Machine-readable:** This language is designed for computer programs to read and interpret, and is often used in software applications that collect personal information.

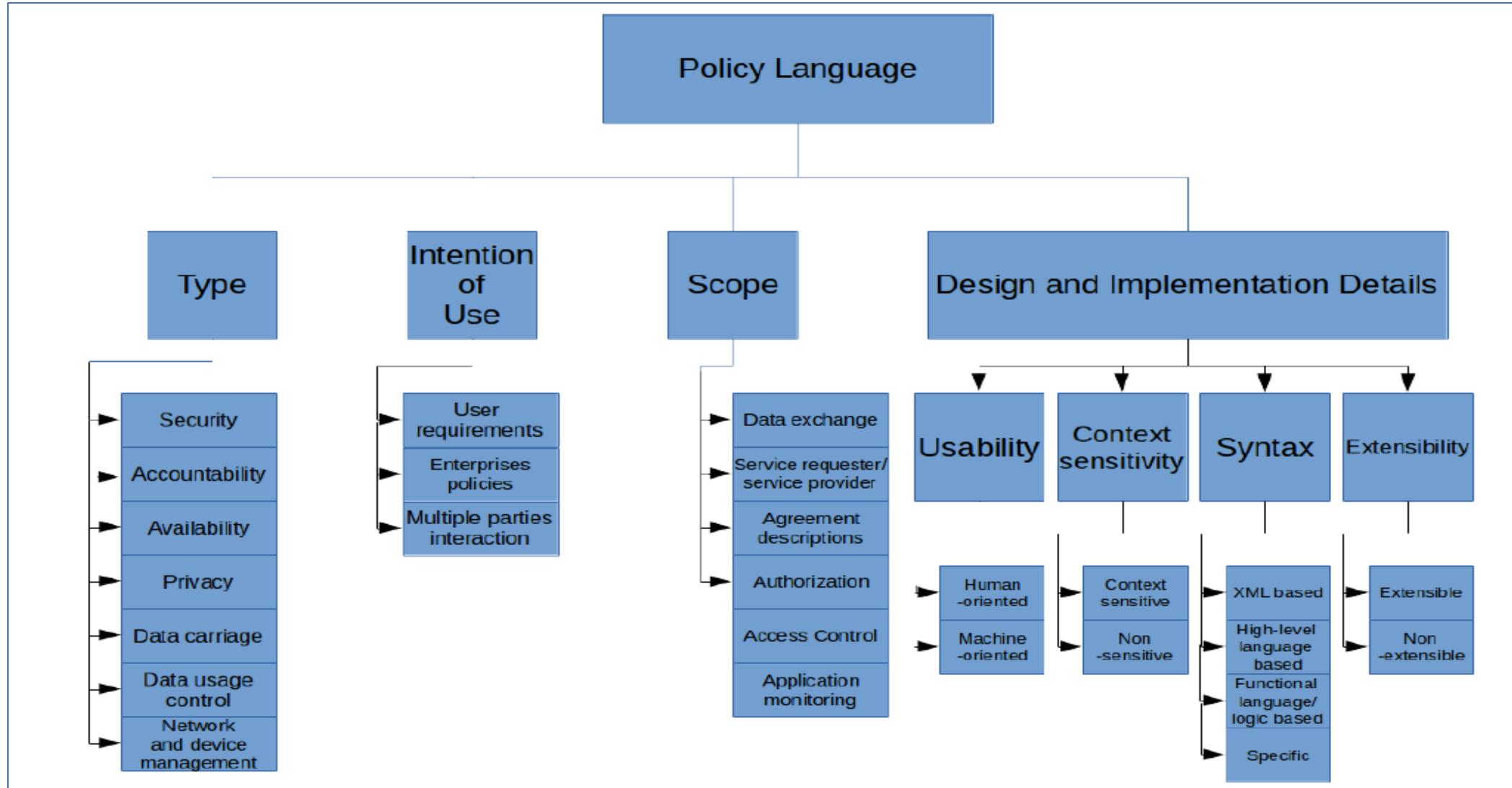
Policy language

Privacy policy languages can help with several stages involved in managing privacy policies (writing, reviewing, testing, approving, issuing, combining, analyzing, modifying, withdrawing, retrieving and enforcing policy).

- Privacy policy languages were designed to express the privacy controls that both organizations and users want to express.
- Most of the privacy policy languages were designed for specific purposes with specific features and characteristics.
- In 1997, the World Wide Web Consortium (W3C) began development of the Platform for Privacy Preferences (P3P) to express website privacy policies in machine-readable format.
- A P3P Preference Exchange Language (APPEL) was also designed by W3C in 1997 to express an individual's privacy preferences, to query the data represented by P3P, and to make decisions accordingly.
- CPExchange (Customer Profile Exchange) was developed in 2000 to facilitate business-to-business communication about privacy policies.

- Later, the industry felt the need for languages to express the internal privacy policies of the organizations themselves.
- With that goal IBM designed the **Enterprise Privacy Authorization Language (EPAL)** in 2003.
- During the same period a consortium of organizations joined to design **the eXtensible Access Control Markup Language (XACML)** for expressing both privacy and security policies in a machine readable format.

Policy language - Categorization Framework



Categorization Framework (Type)

- **Security:** In this subcategory, we categorize all policy languages that allow for the definition of general security-related issues.
- **Accountability:** A language enables for accountability if it contains supporting means, i.e. rules on logging, notification, retention and location of data.
- **Availability:** A language has the scope availability if it enables for stating rules w.r.t the security mean of availability.
- **Privacy:** Languages that focus on privacy-enhancing rules as the confidentiality of person-identifiable information as well as context information and the handling of privacy-relevant data.
- **Data carriage:** Languages that provide a structured format for data carriage as well as data handling relevant information.
- **Data usage control:** Languages that enable for the definition of rules for controlling the data usage.
- **Network and device management:** These languages are mainly designed for the definition of rules that control the management of devices and networks of an organization.

Categorization Framework (Intention of Use)

- **Users requirements:** Some languages are intended to be used for specifying the security or privacy requirements of a user of a system or the owner of data on a system. Rules can be formulated for (privacy-preserving) access control, browsing privacy, user's privacy requirements while sharing PII, or data owners' privacy while sharing collected information with others.
- **Enterprises policies:** In this category we collect languages that are used for the description of how an enterprise implements privacy-related policies.
- **Multiple Parties interaction:** The languages of this category are used for stating privacy requirement of a service requestor on one side, and the abilities of meeting these requirements on the server side.

Categorization Framework (Scope)

- **Data exchange:** These languages are intended to be used for the description of how and under which conditions data is exchanged between different parties.
- **Service requester/service provider:** Languages that are considered as service requester/service provider languages are intended to be **used for stating the conditions a service may be requested**.
- **Agreement descriptions:** **Languages that are considered as Information sharing agreement** languages are intended to be used to specify **under which conditions data is transferred from one party to another in a system**.
- **Authorization:** Languages that are explicitly meant to be used for providing authorization information and rules.
- **Access control:** Languages that are categorized as access control languages are intended to be used for **stating the conditions under which certain defined subjects may access certain defined objects in a system**.
- **Application Monitoring:** These languages provide means of specifying application monitoring rules. This may include design and development, testing, and runtime monitoring.

Categorization Framework (Design and Implementation Details)

- **Usability:** We refer to the human readability and ease of handling of a policy language with usability. On the other hand, some of them are explicitly meant to be usable.
- **Context sensitivity:**
 - Context sensitive
 - Non-sensitive
- **Syntax:** Depending on the use case, the decision of using a certain policy language may be influenced by the syntax the language use.
- **Extensibility:** Some Languages are defined for a specific application. Others are defined for fulfilling certain purposes that can be performed by different applications. This needs a flexibility in the syntax by providing extension points.

Example

- <https://www.cybersecurityeducation.org/about/privacy-policy/#toc9>

Privacy in different domains(Privacy in finance)

-)
- Privacy in finance refers to the protection of individuals' personal and financial information from unauthorized access, use, or disclosure.
 - Financial information can include a wide range of data, such as bank account numbers, credit card information, and investment details, and is often highly sensitive and confidential.

Privacy in different domains(Privacy in finance)

)

- Laws
- There are several laws and regulations that govern privacy in finance, including:
 - **Gramm-Leach-Bliley Act (GLBA):** The GLBA requires financial institutions to develop and implement policies and procedures to protect the privacy and security of consumers' personal financial information.
 - **Fair Credit Reporting Act (FCRA):** The FCRA regulates the collection, dissemination, and use of consumer credit information, including credit reports and scores.
 - **Payment Card Industry Data Security Standard (PCI DSS):** The PCI DSS is a set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.

Financial Privacy in India

- In India, privacy in finance is governed by several laws and regulations
 - **The Reserve Bank of India (RBI) guidelines:**
 - The RBI has issued guidelines for banks and other financial institutions on the management and protection of customer information. The guidelines require financial institutions to implement information security measures to protect customer information, limit access to customer information to only authorized personnel, and have policies in place for data retention and disposal.
 - **The Information Technology Act, 2000 (IT Act):**
 - The IT Act includes provisions for the protection of personal information and sensitive personal data or information (SPDI). Under the act, entities collecting and processing personal information must obtain the individual's consent and take reasonable security measures to protect the data

Financial Privacy in India

- **The Aadhaar Act, 2016:**
 - The Aadhaar Act provides for the collection and use of the Aadhaar number, a unique identification number issued to residents of India. The act sets out privacy protections for the collection and use of Aadhaar data, including requirements for obtaining consent and protecting the confidentiality of the data.
- **The Personal Data Protection Bill, 2019:**
 - The bill includes provisions for the protection of personal data and sensitive personal data, requirements for obtaining consent for data processing, and penalties for non-compliance.

Example Documentations

- https://www.meity.gov.in/writereadda/files/The%20Digital%20Personal%20Data%20Protection%20Bill%2C%202022_0.pdf

Privacy policy in medical domain

- Medical privacy, or health privacy, is the practice of **maintaining the security and confidentiality of patient records**. It involves both the conversational discretion of health care providers and the security of medical records.
- Medical confidentiality is a **concept of doctors maintaining all the information received during the course of the patient's treatment**.
- The patient's confidentiality regarding his/her treatment is of vital importance and should be protected.
- **It is the right of an individual that** his/her personal and medical information is kept private or confidential.
- Such delicate and confidential information about the individual should only be in between him and the doctor, physician, healthcare or health insurance company.

Privacy policy in medical domain

- The medical information of the patient given to a health care provider shall not be disclosed to others unless the patient gives his consent to disclose such information to others.
- The confidentiality of a patient should be maintained because the communication of personal information or records may create personal or professional problems while the patients depend on doctors to keep their medical information private.
- Privacy applies to a person. Privacy means keeping the medical records restricted from the public release because it protects the patient's identity.
- Confidentiality applies to the data. It refers to an individual's right to have personal and detectable medical information which remains private between the patient and the physician. It is an extension of privacy.

Privacy policy in medical domain

- The medical information of the patient given to a health care provider shall not be disclosed to others unless the patient gives his consent to disclose such information to others.
- The confidentiality of a patient should be maintained because the communication of personal information or records may create personal or professional problems while the patients depend on doctors to keep their medical information private.
- Privacy applies to a person. Privacy means **keeping the medical records restricted** from the public release because it protects the patient's identity.
- confidentiality applies to the data. It refers to an individual's right to have personal and detectable medical information which remains private between the patient and the physician. It is an extension of privacy.

Laws governing the Confidentiality and Privacy of a patient in India

According to the [Indian Medical Council \(Professional Conduct, Etiquette and Ethics\) Regulations, 2002](#), it has been stated under chapter 7- (7.14) that the registered medical practitioner shall not divulge any of the secrets of a patient that have been acquired in the exercise of his/her professional skill or while conducting the treatment.

Privacy policy in medical domain

- Federal law in the United States that establishes national standards for protecting the privacy, security, and confidentiality of individuals' health information.
- The law applies to covered entities, such as healthcare providers, health plans, and healthcare clearinghouses, as well as their business associates, who handle health information on their behalf.

HIPAA (Health Insurance Portability and Accountability Act)

<https://www.cdc.gov/phlp/publications/topic/hipaa.html>

- HIPAA includes several provisions related to the privacy and security of protected health information (PHI), including:
 - **Privacy rule:** This rule sets standards for **how covered entities must protect the privacy of individuals' PHI**, including how it can be used and disclosed, and requires that individuals be provided with certain rights regarding their PHI.
 - **Security rule:** **This rule establishes standards for the security of electronic PHI (ePHI)**, including administrative, physical, and technical safeguards that covered entities must implement to ensure the confidentiality, integrity, and availability of ePHI.
 - **Breach notification rule:** This rule requires covered entities to notify individuals in the event of a breach of their unsecured PHI, as well as the Department of Health and Human Services (HHS) and, in some cases, the media.
 - **Enforcement rule:** This rule establishes the procedures for investigations, penalties, and corrective action plans when a covered entity violates HIPAA's privacy and security rules.

HIPAA Privacy Rule - covered entities

- **Healthcare providers:** Every healthcare provider, regardless of size of practice, who electronically transmits health information in connection with certain transactions. These transactions include:
 - Claims
 - Benefit eligibility inquiries
- **Health plans:**
- Health plans include:
 - Health, dental, vision, and prescription drug insurers
 - Health maintenance organizations (HMOs)
 - Medicare, Medicaid, Medicare+Choice, and Medicare supplement insurers
 - Long-term care insurers (excluding nursing home fixed-indemnity policies)
 - Employer-sponsored group health plans

HIPAA Privacy Rule - covered entities

- **Healthcare clearinghouses:** Entities that process nonstandard information they receive from another entity into a standard (i.e., standard format or data content), or vice versa. In most instances, healthcare clearinghouses will receive individually identifiable health information only when they are providing these processing services to a health plan or healthcare provider as a business associate.
- **Business associates:** A person or organization (other than a member of a covered entity's workforce) using or disclosing individually identifiable health information to perform or provide functions, activities, or services for a covered entity. These functions, activities, or services include:
 - Claims processing
 - Data analysis
 - Utilization review
 - Billing

Computation systems for protecting delimited Data-Min Gen, Datafly, Mu-Argus, k-Similar.

- **Goal:**
- Release useful information in such a way that the identity of any individual or entity contained in data cannot be recognized while the data remains practically useful.
- What is delimited data?
 - ✓ Data separated by a delimiter such as a comma character(,) or a tab.
 - ✓ Generally used in hospital records, office records etc.

Race	Birth	Gender	ZIP	Problem
Black	09/20/65	m	02141	short of breath
Black	02/14/65	m	02141	chest pain
Black	10/23/65	f	02138	hypertension
Black	08/24/65	f	02138	hypertension
Black	11/07/64	f	02138	obesity
Black	12/01/64	f	02138	chest pain
White	10/23/64	m	02138	chest pain
White	03/15/65	f	02139	hypertension
White	08/13/64	m	02139	obesity
White	05/05/64	m	02139	short of breath
White	02/13/67	m	02138	chest pain
White	03/21/67	m	02138	chest pain

Computation systems for protecting delimited Data-Min Gen, Datafly, Mu-Argus, k-Similar.

- MinGen:
 - MinGen is a data protection tool that uses data generalization to create a set of k-anonymous data.
 - The technique involves grouping individuals into anonymized groups such that each group has at least **k** individuals. This approach allows for data analysis without compromising individual privacy.
- Datafly:
 - Datafly is a tool that uses data perturbation to provide privacy protection. The technique involves adding noise to the original data so that it becomes difficult to identify individual data points.
 - Datafly works by computing the correlation between columns of the original data and adding random noise to the columns with the highest correlation.
 - Maintains anonymity in released data by automatically substituting, generalizing and suppressing information as appropriate.
 - Decisions are made at the attribute and tuple level at the time of database access
 - The end result - a subset of the original database that provides minimal linking and matching of data because each tuple matches as many people as the data holder specifies.

Computation systems for protecting delimited Data-Min Gen, Datafly, Mu-Argus, k-Similar.

- Mu-Argus:
 - Mu-Argus is a tool designed for protecting confidential data in official statistics. It uses a combination of data masking, data swapping, and data perturbation techniques to protect sensitive information.
 - The tool can also perform data analysis on the protected data without compromising privacy.
- k-Similar:
 - k-Similar is a technique that generates a dataset that is similar to the original dataset but that does not contain any sensitive information.
 - The approach involves creating a similarity graph of the original dataset and then removing the edges that correspond to sensitive data.
 - k-Similar is an effective technique for providing privacy protection while maintaining data accuracy.

Computing privacy and risk measurements

- The goal of privacy and risk measurements is to evaluate the level of risk associated with the collection, storage, use, and sharing of personal data.
- Risk management is a *systematic application of management policies, procedures, and practices to the activities of communicating, consulting, establishing the context and identifying, analyzing, evaluating, treating, monitoring and reviewing risk.*

Common Methods

- **Privacy Impact Assessments (PIAs)**
 - A PIA is a systematic process for identifying and evaluating the potential privacy risks associated with a particular project, program, or system. It involves assessing the sensitivity of the data being collected, the purposes for which it will be used, the security measures in place to protect it, and the potential impact on individuals.
- **Threat Modeling**
 - Threat modeling is a structured approach to identifying and assessing potential threats to a system or application. It involves identifying potential attackers, their motivations and capabilities, and the potential impact of their actions.

Common Methods

- **Risk Assessment**

Risk assessment is a process for evaluating the likelihood and potential impact of different types of risks to an organization or system. It involves identifying potential threats, assessing the likelihood and impact of those threats, and determining the appropriate risk mitigation measures.

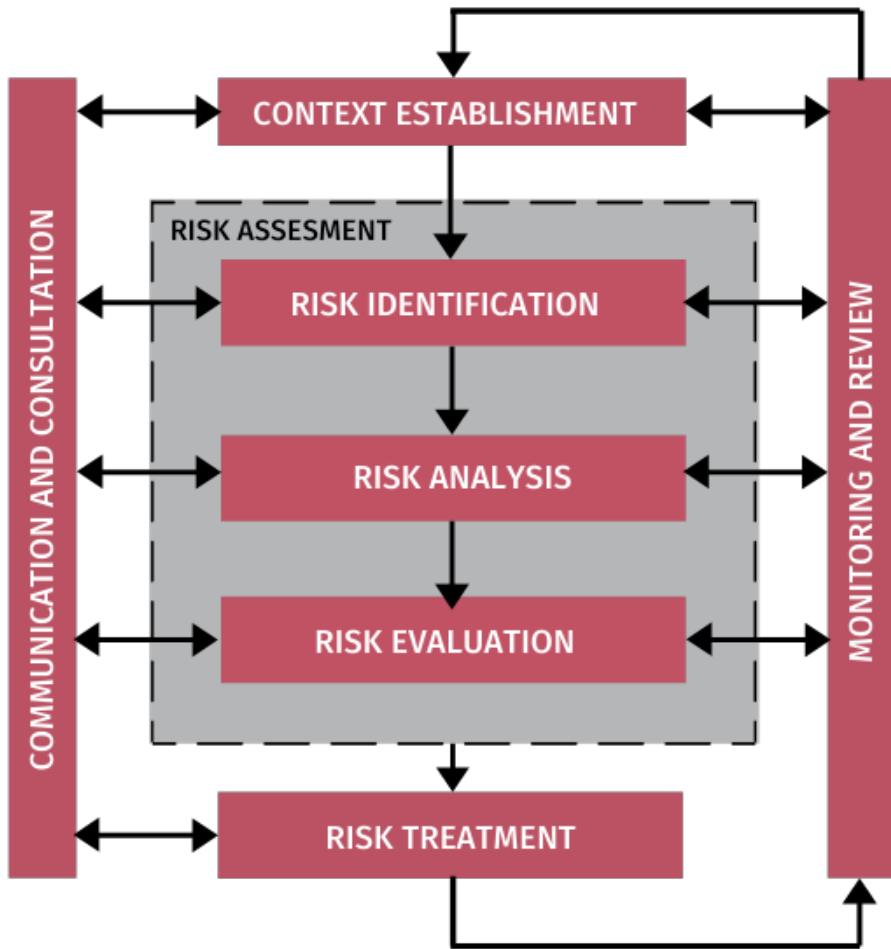
- **Privacy Metrics**

Privacy metrics are **quantitative measures of the level of privacy protection provided** by a system or process. Examples of privacy metrics include the proportion of data subjects who are identifiable, the proportion of data that is sensitive, and the level of encryption or anonymization used.

- **Risk Metrics**

Risk metrics are quantitative measures of the level of risk associated with a particular system or process. Examples of risk metrics include the likelihood of a data breach, the potential impact of a data breach, and the cost of mitigation measures.

7 Steps in Privacy Risk Management



Risk identification, analysis, and evaluation are collectively referred to as **risk assessment**, a sub-process of the overall **risk management process**.

7 Steps in Privacy Risk Management

Context establishment

- outlining specific areas where risk management will be applied.
- From a data privacy perspective, the scope would be **records of processing activity**, as this aligns with *the nature, scope, context, and purposes of processing*.
- It might also consider an organization's drivers for protecting personal data, such as protecting individuals' privacy, meeting legal and regulatory requirements, practicing corporate responsibility, and enhancing consumer trust.
- During the **context establishment phase**, you will need to develop the following criteria:
 - **Risk evaluation criteria** – used to evaluate the criticality of the assets involved
 - **Risk impact criteria** – used to describe the degree of damage caused by an incident
 - **Risk acceptance criteria** – used to decide whether a risk is already at an acceptable level

- **Establishing Context in Data Privacy**
- In **data privacy risk management**, the impacted asset would be personal data, and its classification level would be higher or lower depending on whether personal data is a **special category** of data.

- **2. Risk identification**
- Risk identification in information security aims to determine **what could cause a potential loss** to an organization's assets and gain insight into how, where, and why the loss might happen.
- To do this, several sub-steps need to be performed:
 - Identification of assets
 - Identification of threats
 - Identification of existing controls
 - Identification of vulnerabilities
 - Identification of consequences

- **3. Risk analysis**
- The purpose of risk analysis is to assign levels to risks and can be calculated as shown below:

$$\begin{array}{lcl} \text{data privacy} & = & \text{likelihood} \times \text{severity} \\ \text{risk level} & & \end{array}$$

- Not a mathematical formula. It merely emphasizes that the risk level is a function of these two qualities.

- **4. Risk evaluation**
- Levels of all risks need to be compared against **risk evaluation criteria** and **risk acceptance criteria**, which have been developed during the context establishment phase.

Risk value	Acceptance criteria action
Low	Can be accepted without documented justification.
Moderate	Can be accepted provided that continual monitoring is in place. Treatment plans need to be investigated and implemented where required.
High	Can be accepted by senior management with adequate documented justification and where possible mitigating treatment plans are implemented immediately.

Risk value	Evaluation criteria action
Low	Reduce risk considering the cost of prevention compared to a reduction in risk.
Moderate	Action must be taken. Where the impact is major, urgent action must be taken.
High	Urgent action must be taken.

Demographics and Uniqueness

- The statistical characteristics of human populations (such as age or income) are used to identify market.
- Demographic analysis
 - Demographic analysis is the collection and **study of data regarding the general characteristics of specific populations.**
 - It is frequently **used as a business marketing tool** to determine the best way to reach customers and assess their behavior.
 - Segmenting a population by using demographics allows companies to determine the size of a potential market.

Impact on privacy

- Demographics, such as age, gender, race, and socio-economic status, can impact privacy in a number of ways.
- For example, certain demographic groups may be more vulnerable to privacy risks, such as older adults who may be less familiar with technology and more susceptible to scams and fraud.
- Additionally, certain types of personal data, such as health or financial information, may be more sensitive for certain demographic groups.

Uniqueness

- Uniqueness refers to the degree to which an individual's personal data can be used to identify them.
- For example, certain types of personal data, such as a Social Security number or a biometric identifier, may be more unique and therefore more identifiable than other types of data, such as a name or address.
- This uniqueness can increase the risk of identity theft and other privacy violations.

Privacy attacks on Demographics and uniqueness

- **Data breaches:**
 - Data breaches can expose personal information, including demographic and unique identifiers, to unauthorized parties. This can lead to identity theft and other privacy violations.
- **Data aggregation:**
 - Data aggregation involves combining multiple sources of personal information to create a more detailed profile of an individual. This can increase the risk of privacy violations, as it may reveal sensitive information about a person's demographics or uniqueness that they may not have intended to share.
- **Discrimination:**
 - Demographic and unique identifiers can be used to discriminate against individuals in various contexts, such as employment or housing. For example, if an employer uses demographic data to make hiring decisions, it may result in discrimination against certain groups.

Privacy attacks on Demographics and uniqueness

- **De-anonymization:**

- De-anonymization involves identifying individuals from supposedly anonymous data sets. This can be done by combining data from different sources, such as social media profiles and public records, to re-identify individuals.

- **Social engineering:**

- Social engineering involves manipulating individuals into disclosing sensitive information, such as demographic or unique identifiers, through tactics such as phishing or pretexting. This can lead to identity theft and other privacy violations.

Protection Models

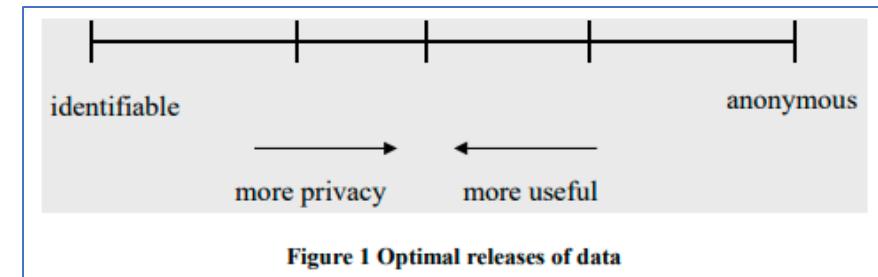
- Society is experiencing exponential growth in the number and variety of data collections as computer technology, network connectivity and disk storage space becomes increasingly affordable.
- Data holders, operating autonomously and with limited knowledge, are left with the difficulty of releasing information that does not compromise privacy, confidentiality or national interests.
- In many cases the **survival of the database itself depends on the data holder's ability to produce anonymous data** because not releasing such information at all may diminish the need for the data, while on the other hand, failing to provide proper protection within a release may create circumstances that harm the public or others.

Quality versus anonymity

- There is a natural tension between the quality of data and the techniques that provide anonymity protection
 - Consider a continuum that characterizes possible data releases.
 - At one end of the continuum are person-specific data that are fully identified.
 - At the other end are anonymous data that are derived from the original person-specific data, but in which no person can be identified.
 - Between these two endpoints is a finite partial ordering of data releases, where each release is derived from the original data but for which privacy protection is less than fully anonymous.

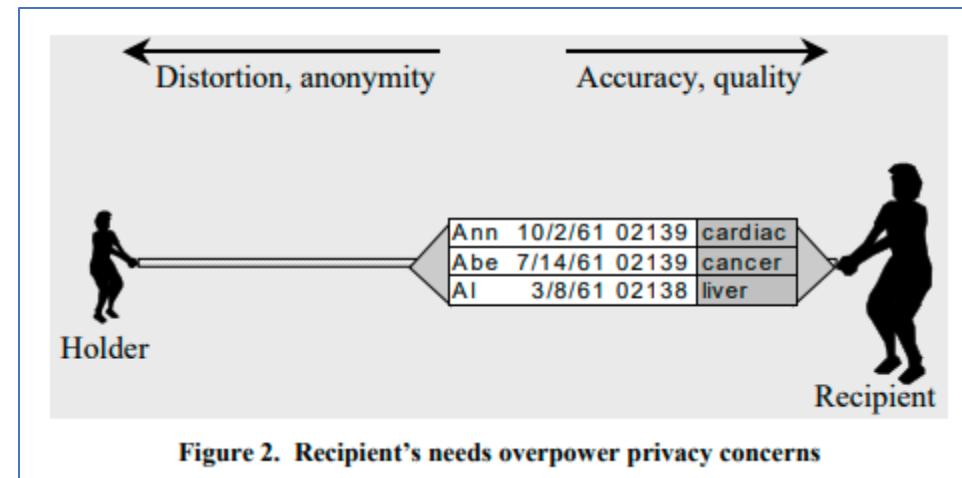
Quality versus anonymity

- The first realization is that any attempt to provide some anonymity protection, no matter how minimal, involves modifying the data and thereby distorting its contents.
- Movement along the continuum from the fully identified data towards the anonymous data adds more privacy protection, but renders the resulting data less useful.



Data holders and recipients

- The data holder and the need to protect the confidentiality or privacy of the information overshadows the recipient and the recipient's use for the data and so the data is completely suppressed and not released at all.
- Data collected and associated with national security concerns provides an example.



Levels of access restrictions by data holders to person-specific data

- Insiders only (Pr) “private”.
 - Data collections that are available to authorized “insiders only” are considered to be privately held information because the only people who gain access are almost exclusively those who directly collected the information.
- Limited Access (SPr) “semi-private”.
 - Data collections denoted as having “limited access” are those where access extends beyond those who originally collected the information, but only an identifiable small number of people are eligible for access in comparison to a substantially larger number of people who are not eligible for access. This access policy typically includes an extensive application and review process.
- Deniable Access (SPu) “semi-public”.
 - Data collections having “deniable access” are those where an application and review process may exist but only an identifiable small number of people are denied access in comparison to a substantially larger number of people who are eligible for access.
- No restrictions (Pu) “public”.
 - Data collections having “no restrictions” are those where an application process may or may not exist, but the data collections are generally made available to all who request them.

Protection Models : Definitions (informal)

- **anonymous data:**
 - The term anonymous data implies that the data cannot be manipulated or linked to identify an individual.
- **Anonymous data system:**

An anonymous data system is one that releases entity-specific data such that particular properties, such as

 - ✓ identity, of the entities that are the subject of the data cannot be inferred from the released data.
- **Disclosure control function**
 - ✓ Disclosure control function (Statistical disclosure control), **seeks to protect data in such a way that they can be publicly released and mined without giving away private information** that can be linked to specific individuals or entities.
- **de-identified data**
 - De-identified data result when all explicit identifiers such as name, address, and phone number are removed, generalized, or replaced with a made up alternative.

Definitions

Definition. attributes

Let $B(A_1, \dots, A_n)$ be a *table* with a finite number of tuples. The finite set of *attributes* of B are $\{A_1, \dots, A_n\}$.

Definition. entity

Let $p_i = \{ (A_i, v_i) : A_i \text{ is an attribute and } v_i \text{ is its associated value} \}$. I say p_i is an *entity*.
 $U = \{p_i : p_i \text{ is an entity}\}$ is a finite set I term a *population of entities*.

Definition. collection function

Given a population of entities U and a table T , I say f_c is a collection function on U . That is, $f_c: U \rightarrow T$ is a *collection function* and T is an *entity-specific table*. I say that T is a *person-specific table* if the entities are people.

Definition. disclosure control function

Given a table T and a finite set of tables B , I say f is a disclosure control function on $\{T\}$. That is, $f: \{T\} \rightarrow B$ is a *disclosure control function*.

Definition. disclosure control function

Given a table T and a finite set of tables B , I say f is a disclosure control function on $\{T\}$. That is, $f: \{T\} \rightarrow B$ is a *disclosure control function*.

Definition. re-identification relation

Given a population of entities U , an entity-specific table T and $f_c: U \rightarrow T$,

I say f_g is a *re-identification relation* if and only if:

$$\exists p_i \in U \text{ such that } p_i \in f_g(f_c(p_i)) \text{ and } |f_g(f_c(p_i))| = k, \text{ where } 1 \leq k < |U|.$$

I also say that f_g is a re-identification of p_i and I say that f_g uniquely identifies p_i if $k=1$.

Definition. pseudo-entities

Given a population of entities U , an entity-specific table T , $f_c: U \rightarrow T$ and a re-identification relation $f_g: T \rightarrow U'$ where $U \subseteq U'$. I say $(U' - U)$ is the finite set of *pseudo-entities*.

Definition. quasi-identifier

Given a population of entities U , an entity-specific table T , $f_c: U \rightarrow T$ and $f_g: T \rightarrow U'$, where $U \subseteq U'$. A quasi-identifier of T , written Q_T , is a set of attributes $\{A_i, \dots, A_j\} \subseteq \{A_1, \dots, A_n\}$ where:

$$\exists p_i \in U \text{ such that } f_g(f_c(p_i)[Q_T]) = p_i.$$

Definition. basic anonymous data system

A *basic anonymous data system*, **ADS₀**, is a nine-tuple $(S, P, PT, QI, U, R, E, G, f)$, where the following conditions are satisfied:

1. S is the finite set of entities with attributes to be protected.
2. P is the finite set of possible entities. $S \subseteq P$.
3. PT is the finite multi-set of privately held information about each member of S . There exists a collection function, $f_c : S \rightarrow PT$, where $PT = \{k \bullet t_s : t_s = f_c(s) \text{ and } |f_c^{-1}(f_c(s))| = k, \forall s \in S\}$.
4. QI is the quasi-identifier of PT denoting attributes to be protected.
5. U is a finite set of possible entities and pseudo-entities. $P \subseteq U$.
6. R is the set of possible releases. Each release $RT \in R$ is a finite multi-set.
7. E is the collection of possible external information. $\forall T_{i=1, \dots, m}$ where T_i is a collection of external information about a subset of the members of P , then $E = T_1 \times \dots \times T_n$.

8. \mathbf{G} is the set of possible relations from $\mathbf{R} \rightarrow \mathbf{U}$.

$$G = \left\{ (g_1, g_2) : g_1 \circ g_2 \text{ where } R \xrightarrow{g_1} E \xrightarrow{g_2} U \right\}$$

Given a QI for \mathbf{PT} , written $QI_{PT} = A_1, \dots, A_m$, a release $RT \in \mathbf{R}$ where $RT = f(\mathbf{PT}[QI])$, and a set of explicit identifiers named EI_{g2} where $g_2(g_1(RT)[EI_{g2}]) \subseteq U$, then

$$\begin{aligned} g_1(RT) &= \{k \bullet t_u[A_1, \dots, A_m] : t_u[QI_{PT}] \in RT, t_u[EI_{g2}] \in E \text{ and } |t_u[QI_{PT}, EI_{g2}]| = k, \\ &\forall t_u \in E, QI_{PT} \subseteq A_1, \dots, A_m \text{ and } EI_{g2} \subseteq A_1, \dots, A_m\}. \end{aligned}$$

g_1 and g_2 are relations and g_2 is a direct communication method.

9. f is a disclosure control function such that $f: \{\mathbf{PT}\} \rightarrow \mathbf{R}$ and given a release $RT \in \mathbf{R}$ where $RT = f(\mathbf{PT}[QI])$, one of the following conditions must be satisfied:

- a. if $\exists g \in G, \exists t \in RT$, where $f(f_c(s)) = t$ and $g(f(f_c(s))) = s$ then $\exists u \in U$, such that $u \neq s$ and $g(f(f_c(s))) = u$.
- b. if $\exists (g_1, g_2) \in G$ where $GT = g_1(f(t_s[QI]))$, $\exists t_s[QI] \in RT$ and $t_s[QI, EI_{g2}] \in GT$ where $f_c(s) = t_s$ and $g_2(g_1(f(t_s[QI]))[EI_{g2}]) = s$, then $\exists t_u[QI, EI_{g2}] \in GT$ such that $t_s \neq t_u$ and $g_2(t_u[QI, EI_{g2}]) = s$.
- c. Given $\mathbf{PT}(A_1, \dots, A_n)$ and $RT(A_w, \dots, A_y)$, let $A_p, \dots, A_q = (\{A_1, \dots, A_n\} - QI) \cap \{A_w, \dots, A_y\}$. If $\exists g \in G, \exists t_{s1}[A_p, \dots, A_q] \in RT$, where $f_c(s) = t_{s1}$ and $g(f(t_{s1}[QI])) = s$ and $t_{s1}[A_p, \dots, A_q] \neq \emptyset$ and if $\exists t_{s2}[A_p, \dots, A_q] \in \mathbf{PT}$ such that $f_c(s) = t_{s2}$ and $f(t_{s2}) = t_{s1}$ and $t_{s2}[A_p, \dots, A_q] = t_{s1}[A_p, \dots, A_q]$, then condition (a) or condition (b) above must be satisfied on t_{s1} .

Values for S, P, PT, QI, U and E

$S = \{(name, Ann), (name, Abe), (name, Al)\}$

$P = \{(name, Dan), (name, Don), (name, Dave), (name, Ann), (name, Abe), (name, Al)\}$

$PT(name, birth\ date, ZIP, diagnosis) :$

Name	Birth date	ZIP	Diagnosis
Ann	10/2/61	02139	Cardiac
Abe	7/14/61	02139	Cancer
Al	3/8/61	02138	Liver

$QI = \{name, birth\ date, ZIP\}$

$U = \{(name, Jcd), (name, Jwq), (name, Jxy), (name, Dan), (name, Don), (name, Dave),$

$(name, Ann), (name, Abe), (name, Al)\}$

$E(name, birth\ date, ZIP) :$

Name	Birth date	ZIP
Ann	10/2/61	02139
Abe	7/14/61	02139
Al	3/8/61	02138

g_2 = a direct communication channel that operates on the *name* attribute.

G as the set of all possible relations from R to U consistent with property 8 in the definition of an **ADS₀**

Protection Models

There are three protection models associated with data privacy

- Null Map
- K-Map
- Wrong Map

Null Map

- In **null-map protection** each tuple in the released information may or may not map to an actual entity in the population P , but none of the tuples can be mapped to an entity in the set of subjects S .
- Examples of disclosure limitation techniques that can achieve null-map protection include strong encryption of the QI, extensive swapping of the values in QI and systematic use of additive noise.

Definition. null-map protection

Let \mathbf{A} be an ADS_0 , $f(\text{PT}) = \text{RT}$ and $R \in \text{RT}$. If $\forall t \in RT$, there does not exist $g \in G$ where $g(t) \in S$, then \mathbf{A} adheres to *null map protection*.

K-Map

- k-map protection maintains the invariant that each tuple in the released information refers indistinctly to at least k members of U. Notice that k does not rely on ISI > k or on IRTI >k.

Definition (k-map protection).

Let \mathbf{A} be an ADS_0 , $f(\text{PT}) = \text{RT}$ and $R \in \text{RT}$. If $\forall t \in \text{RT}, \exists g \in G$, where $f(f_c(s)) = t$ and $g(f(f_c(s))) = s$ and $\{u_1, u_2, u_{k-1}\} \in U$ such that for $i=1, \dots, k-1$, $u_i \neq s$, and $g(f(f_c(s))) = u_i$, then \mathbf{A} adheres to *k-map protection*.

Wrong map

- Wrong map protection requires each tuple in the released information to be identified to only one entity in subjects but that entity is not the entity to which the original information was collected. The ADS_0 requirement ensures the values with attributes outside QI contained in the release are not the same as those originally collected. Notice if there exists only one entity in the subjects S , then wrong-map protection cannot be done and with only two entities in S , the release is compromised.

Definition (wrong-map protection).

Let \mathbf{A} be an ADS_0 , $f(\text{PT}) = \text{RT}$ and $R \in \text{RT}$. If $|\text{RT}| > 2$ and $\forall t \in \text{RT}, \exists g \in G$ where $f(f_c(s)) = t$, and $g(f(f_c(s))) = s$ and there does not exist $g' \in G$ where $g' \neq g$ such that $g'(t) \in S$, then \mathbf{A} adheres to *wrong map protection*.

Survey of Techniques-Protection models (Null-map)

- The null-map technique involves removing all data elements that might lead to the identification of individuals.
- This technique ensures that no personally identifiable information (PII) is present in the dataset, thus preventing any potential disclosure of sensitive information.
- However, this technique also results in the loss of data accuracy and can make it difficult to perform certain types of data analysis.

Survey of Techniques-Protection models (K-map)

- The k-map technique involves grouping similar records together in order to reduce the risk of individual-level identification.
- This technique is often used when dealing with categorical data, such as demographic information or survey responses.
- K-map is an effective technique for preserving data accuracy while still maintaining privacy, but it is vulnerable to re-identification attacks.

Survey of Techniques-Protection models (Wrong map)

- The wrong map technique involves intentionally altering some data elements in a dataset to make it more difficult to identify individuals.
- This technique involves modifying sensitive data in a way that is still statistically useful for analysis but that no longer reveals sensitive information.
- This technique can be effective at preventing disclosure of sensitive information, but it can also result in a loss of data accuracy.

Data Security and Privacy

DSE 3258

L11 —Acquisition and Duplication

What is Computer Forensics?

- Computer Forensics is the science of obtaining, preserving, and documenting evidence from digital electronic storage devices, such as computers, digital cameras, mobile phones, and various memory storage devices.
- Digital evidence **is any information or data of value to an investigation** that is stored on, received by, or transmitted by an electronic device.
- **Text messages, emails, pictures and videos, and internet searches** are some of the most common types of digital evidence.
- **Computer forensics** is a set of **methodological methods** and **techniques** for identifying, gathering, preserving, extracting, interpreting, documenting, and presenting evidence from computing equipment in legal or administrative proceedings
- Computer Forensics is primarily concerned with the proper acquisition, preservation and analysis of digital evidence.

Acquisition and Duplication

- Data acquisition is the **process of gathering evidence or information**.
- This can be done by using established methods to acquire data from a suspected storage media outlet to gain access to information about the crime or other incident, and potentially using that data as evidence to convict a suspect.
- It is the use of established methods to extract Electronically Stored Information (ESI) from suspect computer or storage media to gain insight into a crime or an incident.
- It is a critical step in digital forensics, as an improper acquisition may alter data in the evidence media, and render it inadmissible in a court of law.
- Investigators must be able to verify the accuracy of acquired data, and the complete process should be auditable and acceptable in the court.

Acquisition and Duplication

- **Types of Data Acquisition**
 - Live Acquisition
 - Involves collecting data from a system that is powered ON
 - Dead Acquisition (Static Acquisition)
 - Involves collecting data from a system that is powered OFF

<https://johntai.net/posts/chfi-notes/module-04/>

Live Acquisition

- Live data acquisition involves collecting volatile data from a live system.
- Volatile information assists in determining the logical timeline of the security incident, and the possible users responsible.
- Live acquisition can then be followed by static/dead acquisition, where an investigator shuts down the suspect machine, removes the hard disk, and then acquires its forensic image.
- Types of data captured during live acquisition
 - System Data
 - Current configuration, Running state, Date and time, Command history
 - Current system uptime, Running processes, etc
 - Network Data
 - Routing tables, , Network configuration, Network connections, etc.
 - Live acquisition can help investigators obtain : Data from unencrypted containers or disks that are open on the system, automatically get encrypted when the system shuts down
 - Private browsing history and data from remote storage services such as Dropbox (cloud service) by examining the Random-Access Memory (RAM)

Dead Acquisition

- Dead acquisition is defined as the acquisition of data from a suspect machine that is powered off
- Dead acquisition usually involves acquiring data from storage devices such as hard drives, DVD-ROMs, USB drives, flashcards, and smartphones
- Examples of static data: emails, word documents, web activity, spreadsheets, slack space, unallocated drive space, and various deleted files

Rules of Thumb for Data Acquisition

- **Do not work on original digital evidence.** Create a bit-stream/logical image of a suspicious drive/file to work on.
- Produce two or more copies of the original media
 - The first is the **working copy** to be used for analysis
 - The other copies act as the **library/control copies** that are stored for **disclosure** purposes or in the event that the working copy gets corrupt
- Use **clean media** to store the copies
- Upon creating copies of original media, verify the **integrity of copies** with the original

Types of Data Acquisition

➤ Logical Acquisition and Sparse Acquisition

- In a situation with time constraints and when the investigator is aware of what files need to be acquired, logical acquisition is an ideal method.
- Logical acquisition allows an investigator to capture only selected **files or files types** of interest to the case
- Examples of logical acquisition include:
 - Email investigation that requires collection of **Outlook .pst or .ost files**
- Sparse acquisition is similar to logical acquisition, which in addition **collects fragments of unallocated data, allowing investigators to acquire deleted files**. Use this method when inspection of the entire drive is not required.

Types of Data Acquisition

➤ Bit-Stream Image

- Bit-stream imaging creates a **bit-by-bit** copy of a suspect drive, which is a cloned copy of the entire drive including all its sectors and clusters.
- This image contains not just a copy of all the files and folders, but also the ambient data, which allows forensic investigators to **retrieve deleted files or folders**.

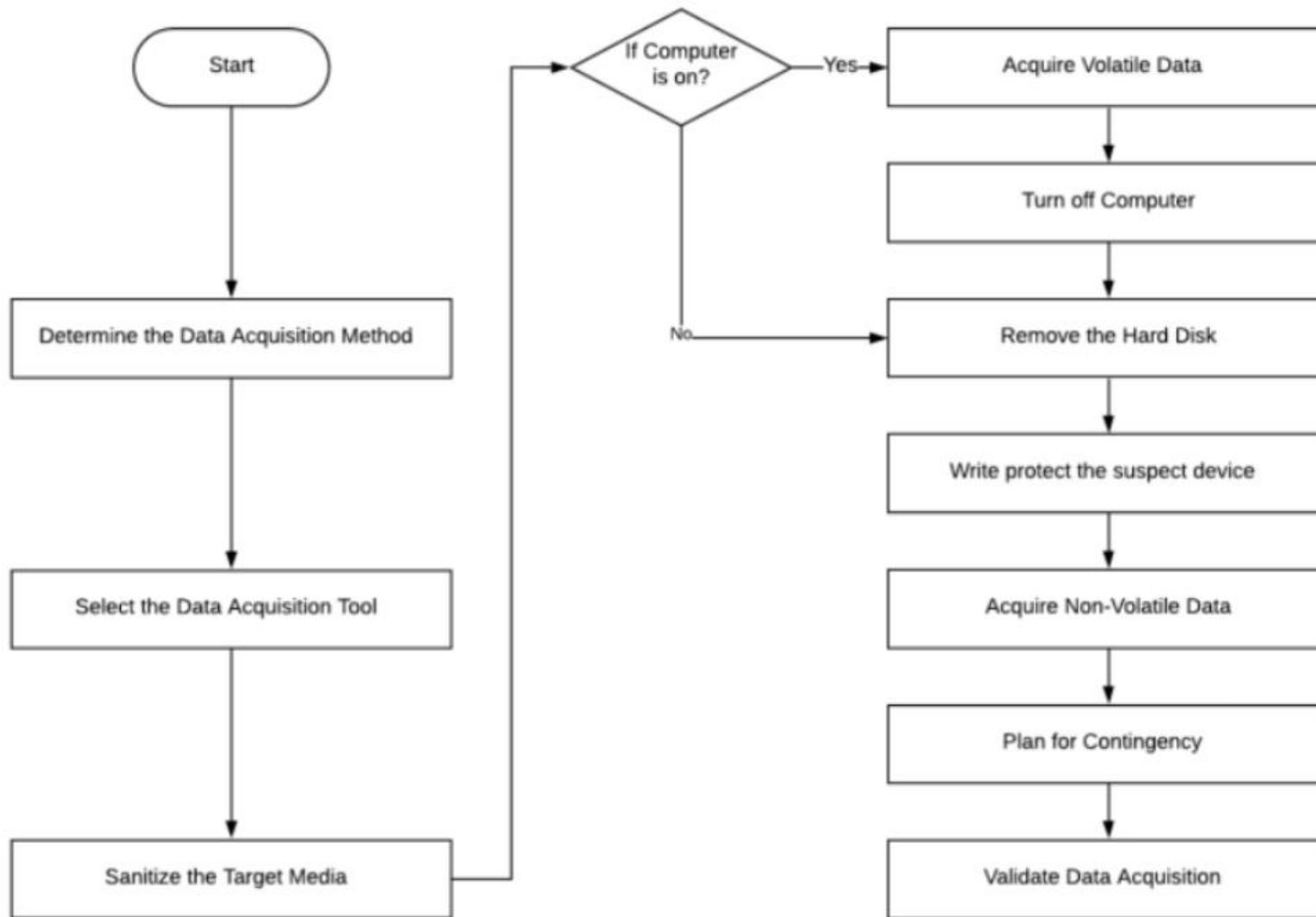
➤ Bit-stream disk-to-image file

- It is the most common method used by **forensic investigators**
- With this method, one or many copies of the suspect **drive** can be generated
- The created image file is a bit-by-bit replica of the suspect drive
- Tools used: ProDiscover, EnCase, FTK, etc.

Types of Data Acquisition

- **Bit-stream disk-to-disk**
 - Disk-to-image copying is not possible in situations where
 - The suspect drive is very old and incompatible with the imaging software
 - Investigator needs to recover credentials used for websites and user accounts
 - To overcome this situation, investigators can create a **disk-to-disk bit-stream copy** of the target media
 - Tools used for this imaging process such as **EnCase**, **tableau Forensic Imager**, etc. enable investigators to modify the internal components of the target disk so that data obtained aligns well with the suspect drive.

Data Acquisition Methodology



Data Acquisition Methodology

Step 1: Determine the Best Data Acquisition Method

- An investigator needs to identify the best data acquisition method suitable for the investigation, depending on the investigator's situation. These situations include
 - ✓ Size of the suspect's drive
 - ✓ Time required to acquire the image
 - ✓ Whether the investigator can retain the suspect's drive
- Investigators need to acquire only the data that is intended to be acquired

Data Acquisition Methodology

Step 2: Select the Data Acquisition Tool

- Investigators need to choose the right tool for data acquisition based on the type of acquisition technique they choose. When it comes to imaging tools, they need to choose the tools that satisfy certain requirements.
- **Mandatory Requirements**
 - ✓ The tool should not change the original content
 - ✓ The tool should log I/O errors in an accessible and readable form, including the type of the error and location of the error
 - ✓ The tool must have the ability to pass scientific and peer review. Results must be repeatable and verifiable by a third party if necessary.
 - ✓ The tool should alert the user if the source is larger than the destination
 - ✓ The tool should create a bit-stream copy of the original content when there are no errors in accessing the source media
 - ✓ Tool documentation should be correct, i.e., the user should get expected results by implementing it in accordance with the tool's documented procedures

Data Acquisition Methodology

Step 3: Sanitize the Target Media

- ✓ Investigators must properly sanitize the target media in order to delete any prior data residing on it, before it is used for collecting forensic data
- ✓ Post investigation, they must dispose of this media by following the same standards, so as to mitigate the risk of unauthorized disclosure of information and ensure its confidentiality

Step 4: Acquire Volatile Data

- ✓ Volatile data acquisition involves collecting data that is lost when the computer is shut down or restarted
- ✓ This data usually corresponds to running processes, logged-on users, registries, open files, etc.
- ✓ While most of this data is recovered by examining the live system, the approximately same amount of data can be obtained by examining the image acquired from the memory of the system.

Data Acquisition Methodology

Step 5: Enable Write Protection on the Evidence Media

- ✓ It is necessary to write protect the suspect drive using write blockers to preserve and protect the evidence contained in it.
- ✓ A write blocker is a hardware device or software application that allows data acquisition from the storage media without altering its contents.
- ✓ It blocks write commands, thus allowing read-only access to the storage media

Step 6: Acquire Non-volatile Data

- ✓ Non-volatile data can be acquired in both live acquisition and dead acquisition. It mainly involves acquiring data from a hard disk.
- ✓ There is no significant difference in the amount of data acquired from a hard disk between the live and dead acquisition methods.

Data Acquisition Methodology

Step 7: Plan for Contingency

- Investigators must prepare for contingencies such as when the hardware or software does not work, or a failure occurs during acquisition
 - Hard Disk Data Acquisition
 - ✓ Investigators must create at least two images of the digital evidence collected, in order to preserve it. If one copy of the digital evidence recovered becomes corrupt, investigators can then use the other copy.
 - If you have access to two or more imaging tools, you must create two images of the evidence using at least two of them. In case, you have access to only one tool, make two or more images of the drive using the same tool.
 - Be prepared to deal with encrypted drives that need the user to provide the decryption key for decrypting. Microsoft includes a full disk encryption feature (BitLocker) with select editions of Windows Vista and later.

Data Acquisition Methodology

Step 8: Validate Data Acquisition

- ✓ Validating data acquisition involves calculating the hash value of the target media and comparing it with its forensic counterpart to ensure that the data is completely acquired
- ✓ Hash value calculation generates a unique numeric value for files which is used for preserving data integrity and preventing data alteration
- ✓ If two files contain the same hash value, they are taken to be completely identical even if those are named differently

Sterilizing evidence media

- Any previous data must be removed from the copy media with a software tool that is proven to remove all data from the drive.
- If media is not sterilized properly, the forensic output would be contaminated and in case of disposal of media, sensitive data may be leaked.
- By using forensically sterile media, the Computer Forensics specialist ensures that the media itself will not contaminate the evidence.
- This sterilization process should be documented and visually verified by the forensic examiner, leaving no doubt that whatever is found on the working copy during a forensic examination is/was also present on the original media.

Acquiring Forensics Images

- A forensic image (forensic copy) is a bit-by-bit, sector-by-sector direct copy of a physical storage device, including all files, folders and unallocated, free and slack space.
- Forensic images include all the files visible to the operating system (OS), as well as deleted files and pieces of files left in the slack and free space.
- This includes both the logical file structure (files and folders) and all the associated metadata for that logical structure.
- Electronic evidence can be gathered from a variety of sources, including computers, mobile devices, remote storage devices, internet of things (IoT) devices, and virtually any other computerized system.
- This image is created using various third-party tools which can easily capture the image of a hard drive bit by bit without changing even a shred of data.
- Forensic software copies data by creating a bitstream which is an exact duplicate.
- The best thing about creating a forensic image is that it also copies the deleted data, including files that are left behind in swap and free spaces.

Acquiring Live Volatile Data

- Volatile memory or random access memory stores information such as running process, incognito browsing sessions, clipboard data , information stored in plain text files and much more.
- **Volatile data refers to the information stored in a system's temporary storage areas, like the RAM or physical memory, and in active processes or services.** This data is characterized by its transient nature; it exists only as long as the system is powered on and can be lost or altered upon shutdown or restart.
- Volatile memory contains the following system artifacts which gets lost when the device is restarted or shut down. The following artifacts can be expected out of the volatile memory acquisition process:
 1. System Process
 2. Running services
 3. Clipboard Information
 4. Browsing Sessions (Incognito Sessions)
 5. Passwords
 6. Accessed Files and Multimedia
 7. Chats/Running Application stored data
- During an incident response, an investigator has to analyze the suspected machines and the profitability of capturing live RAM. Device has to be unlocked authentically and RAM dumping modules has to be loaded according to the host operating system.

Acquiring Live Volatile Data

- Highly Volatile
 - **These types of evidence are unlikely to be recoverable after the system is powered down.** Examples include physical memory, running processes, running services, screen shots, active network sessions, operational drivers, system information, and mounted encrypted volumes.
 - **Physical Memory:** Provides the opportunity to examine and carve potential passwords, recent messages, partial documents, malicious processes, web history, financial data, phone numbers, contact information, etc.
 - **Running Processes:** Provides the investigator or auditor with a record of the processes that were running on the target computer at the time of the acquisition. This information can provide clues about what the suspect or victim was doing most recently.
 - **Running Services:** Furnishes insight into the system services that were running or stopped. For example - was the antivirus active, was the firewall running, was there a VPN in operation?
 - **Screenshots:** Gives information about the most recent user activity, images, videos, messages, documents, and open web pages.
 - **Active Network Sessions:** Affords insight into the connections to inside or outside services. These could be NAS devices, cloud infrastructures, accomplices, or compromised services.
 - **Operational Drivers:** Provides detailed information about which peripherals have been connected to the system. For example, cameras, GPS devices, USB devices, etc. that could be valuable to the investigation or audit.
 - **System Information:** Provides IP and MAC addresses, general system information to link this computer/device to the acquired evidence.
 - **Mounted Encrypted Volumes:** Access to information that may be vital for the investigation, yet only available while file systems are mounted and unlocked.

Acquiring Live Volatile Data

- Moderately Volatile
 - **These types of evidence might be recoverable, but the process can be complex, slow, or less accurate.** Examples include user events (login, shutdown), security events, registry events, recent images, recent multimedia, recent documents, actively inserted devices, recently inserted devices, web history, and email history.
 - **User Events (Login, Shutdown):** Provides information about when the systems were used, when users logged in and logged out. Gives investigators evidence that could be used in questioning users.
 - **Security Events:** Provides auditors and compliance officers with information about possible security violations, unsuccessful login attempts, and changes to important security settings that could affect operations.
 - **Registry Entries:** Delivers a wealth of information about Windows systems, security settings, application settings, and even user activities.
 - **Recent Images, Recent Multimedia, and Recent Documents:** Offers a glimpse at the most recent images, multimedia, and documents viewed and modified by users.
 - **Actively Inserted Devices and Recently Inserted Devices:** Provides quick access to information about inserted USB and other memory devices.
 - **Web History:** Gives investigators insight into the most recent browsing habits of users.
 - **Email History:** Gives investigators access to email history and address books in use by users.

Acquiring Live Volatile Data

- Possibly Volatile or Time-Sensitive
 - These types of evidence are most likely recoverable using postmortem procedures, but the recovery may be delayed. Examples include files and documents, drive images, directory structure, and installed applications.
 - **Files and Documents:** Certain files by type or content may provide immediate evidence to investigators or auditors. These files may have vital data related to the investigation or contain company proprietary data.
 - **Drive Images:** In some cases, the direct image of a logical volume may be essential either to preserve evidence or acquire evidence that may be lost during shutdown or only be available in a live environment.
 - **Directory Structure:** Taking a snapshot of a directory structure may provide information about user activities and tendencies.
 - **Installed Applications:** Can provide a glimpse into the tendencies and sophistication of the user.

Acquiring Live Volatile Data

- **The steps for acquisition are as follows:**
 1. Determine the state of the machine
 2. Identify the operating system
 3. Check for authentic device access
 4. Insert acquisition media
 5. Perform Volatile Memory Dump
 6. Collect SWAP, PAGEFILE.sys and system protected files
 7. Hash and verify the acquired files
 8. Create Investigator copies
- **Memory Acquisition Procedure**
 - It is host operating system dependent.

Acquiring volatile memory from windows OS

Pre- Acquisition Process

To acquire volatile memory of windows OS based system, investigator needs to ensure the following measures:

1. Host machine should not be connected to any external network

- Disconnect LAN/WIFI/Bluetooth connections by putting the device in airplane mode
- Ensure the device is unlocked without installing any password bypassing module

2. Remove any external connected devices

3. Host machine should be connected to a stable power source

- Windows memory management stores volatile memory in multiple ways, an investigator needs to acquire the following volatile information for proper volatile memory acquisition:

- ✓ **Pagefile.sys**: is a paging file which windows uses to store virtual memory contents.
- ✓ **Hiberfil.sys**: is created when windows hibernation is enabled. It stores volatile memory contents when the system needs to enter or has entered hibernation.
- ✓ **Swapfile.sys**: is used to store the idle and non active process data .

- **Acquisition Process**

- To perform acquisition dump the memory contents to file along with other on-disk virtual memory storage files.
- Example tools: FTK Imager lite, dumpit ,Lime
- Ensure the following:
 - ✓ Acquisition module should be executed from an external device
 - ✓ The memory dump should be stored in an external drive
 - ✓ Always hash the acquired data for integrity
 - ✓ Ensure external drive has adequate free space calculating pagefile, hiberfil, swapfile and memory dump.

Metadata extraction

What is meta data?

- Often referred to as **data that describes other data**, metadata is structured reference data that helps to sort and identify attributes of the information it describes
- Meta is a prefix that -- in most information technology usages -- means "**an underlying definition or description.**" Metadata summarizes basic information about data, which can make it easier to find, use and reuse particular instances of data.

Descriptive Meta data

- Descriptive metadata is basic information: who, what, when and where.
- Think of it as a description of a file or a piece of art with the plaque next to it; it's there to help individuals know what they are looking at and the description changes depending on the contents of the object or information piece.
- Types of descriptive metadata include:
 - Time and date of creation
 - Program or processes used for the creation of the data
 - Purpose of the data
 - Creator or author of the data
 - Location on a device where the data was created
 - Technical standards used
 - File size
 - Data quality
 - Source of the data
 - Modifications or programs used to modify the file

Metadata extraction

There are different types of metadata that can be found in a digital file, these include:

- **File Name:** This is the name given to the file when it was originally created and can be a useful way of identifying its source. This can also help identify a specific version of malware if there are multiple iterations or variants available in an attacker's toolkit.
- **File Size:** This can be used to identify the size of a file and whether it has been modified or tampered with. If a file is modified by an attacker, then there will be some changes in its size compared to the original. This can also be useful for identifying new versions of malware if they are released by their authors.
- **Date Modified:** The date a file was last modified can be useful in helping to identify if it has been modified by an attacker. If a file is not modified, then there will be no change in its date of modification.
- **Location on Disk:** If a file is stored in a compressed form, it can be identified by its location on disk. For example, if the file is stored in the same folder as other files that were not modified by an attacker and it has been modified, then it would be likely that this file was modified by someone else.
- **File Hash:** If a file has been modified by an attacker, it will have a different hash value than the original. A hash value is a unique number that can be used to identify particular files. This can be helpful in identifying if a file has been modified by an attacker.

Why Meta data is useful in cyber forensic?

- Metadata examination is extremely useful in the field of cyber forensics, especially if the metadata contains information that is not easily obtainable. When a file is moved from one directory to another, the modification time and access time may change, however the creation time will remain the same (if the OS supports it). The hash value of a file can be used to determine if a file has changed since its inception. If there are no changes made to a file from the time it was created until now then these values should be identical.
- For example, say that you have a spreadsheet containing data from your company's sales for the third quarter of 2022. The metadata on this file will tell you when it was created as well as any changes made since then.
- Metadata can be very useful in cyber forensics because it provides investigators with more information than simply retrieving the contents of files or computer hardware devices. Metadata can help investigators determine if files have been altered since they were first created or if they were written at all – something that may not be easily found through other means such as timestamping or hashing algorithms

Metadata extraction

- More sophisticated metadata might also contain data summaries or keywords, content ratings, location coordinates, or system file identification labels.
- Verifying metadata is crucial in digital forensic investigations as it helps establish the authenticity, integrity, and reliability of the evidence.
- Metadata extraction is a tool for understanding the content, context, and trends within large volumes of data—data that can expose insights about potentially malicious threats.

File-system analysis

- File systems analysis is a fundamental aspect of digital forensics, involving the understanding of how data is stored, organized, and retrieved on storage media such as hard drives, solid-state drives, and removable storage devices. Here's an overview of file systems analysis and the key concepts involved:
- File System Basics:
 - Definition: A file system is a method used by operating systems to organize and store data on storage devices, providing a structured way to store, retrieve, and manage files.
 - Components: A file system consists of various components, including the boot sector, file allocation table (FAT), master file table (MFT), inode table, directory structure, and file metadata.

File-system analysis

- Storage Media Structures:

- **Boot sector:** The boot sector is the first sector of a storage device and contains the boot loader program responsible for booting the operating system.
- **Partition table:** The partition table stores information about the partitions on a storage device, including their size, location, and file system type.
- **File allocation table (FAT):** FAT is a file system used by older versions of Windows to track the allocation of disk space to files and directories. It consists of entries that map file clusters to file names and attributes.
- **Master file table (MFT):** The MFT is a key component of the NTFS file system used by modern versions of Windows. It contains metadata about files and directories, including file attributes, timestamps, and data run extents.
- **Inode table:** Inode-based file systems like ext2/ext3/ext4 used in Linux store file metadata and pointers to data blocks in a data structure called an inode table.

File-system analysis

- File System Analysis Techniques:
 - **Data structure parsing:** Analysts parse and interpret the data structures of file systems to extract information about files, directories, and metadata.
 - **File signature analysis:** Analysts identify file types and formats by analyzing file signatures or magic numbers, which are unique identifiers found in file headers.
 - **Unallocated space analysis:** Analysts examine unallocated space on storage media to recover deleted or fragmented files and identify remnants of past user activity.
 - **Timestamp analysis:** Analysts analyze timestamps associated with files and directories (e.g., creation time, modification time, access time) to reconstruct timelines of user activity.

File-system analysis

- **Directory Structure Analysis:**

- Directory hierarchy: Analysts analyze the hierarchical structure of directories to understand how files are organized and stored on a storage device.
- Directory entry parsing: Analysts parse directory entries to extract information about file names, attributes, timestamps, and file paths.

- **Metadata Analysis:**

- File metadata: Analysts examine file metadata such as file attributes (e.g., read-only, hidden, system), timestamps (e.g., creation time, modification time), and file size.
- Directory metadata: Analysts analyze directory metadata such as timestamps, permissions, and file system quotas to understand directory properties and access controls

File-system analysis

- Forensic Significance:
 - Reconstruction of digital events: File systems analysis enables analysts to reconstruct digital events, such as file creation, modification, and deletion, to establish timelines of user activity and potential evidence tampering.
 - Identification of relevant artifacts: File systems analysis helps identify relevant artifacts and evidence for forensic investigations, including incriminating files, suspicious directories, and hidden data.

Data Security and Privacy

DSE 3258

Security metrics: Design, Data sources, Analysis of security metrics data, Measuring security cost and value, Different context for security process management

L12 –Security Metrics

TB 7 - Lance Hayden, *IT Security Metrics: A Practical Framework for Measuring Security & Protecting Data*, Tata McGraw Hill, 2016

What are Security metrics?

- As defined by the **National Institute of Standards and Technology (NIST)**, metrics are **tools that are designed to facilitate decision-making and improve performance and accountability** through collection, analysis, and reporting of relevant performance-related data.
- Security metrics can be naturally interpreted as a standard (or system) used for quantitatively measuring an organization's security posture.
- Security metrics are quantifiable measurements used to understand the status of systems and services through the collection, analysis and reporting of relevant data.
- They are based on security objectives that help inform decisions on how to improve the security of all components involved in delivering services and processing data.
- Without good metrics, analysts cannot answer many security related questions.
 - Some examples of such questions include “Is our network more secure today than it was before?”
- *IT Security Metrics* provides a comprehensive approach to measuring risks, threats, operational activities, and the effectiveness of data protection in the organization.

Why are Security metrics needed?

Security metrics are needed to:

- Provide a **quantitative and objective basis** for security operations,
- **Support decision making**, e.g. is investment in more security controls needed?
- Support **software quality** since software security is part of software quality,
- Support the **reliable maintenance of security operations**, e.g. how often do users need to change their passwords?
- Support the **incremental improvement** of software's resistance to attacks

Most security metrics follow the SMART structure, which stands for:

- **Specific**

The data must be targeted to the exact area being measured.

- **Measurable**

To be used as a security metric, the data needs to be accurate as well as complete.

- **Actionable**

Data should be easy to understand so action can be taken as soon as possible.

- **Relevant**

All metrics being measured should be important to the data being protected.

- **Timely**

The data should be available when you need it during an analysis.

Good vs bad metric

Collecting valuable data is important, however, if the generation and selection of metrics is done without care, all the data collected will produce useless and meaningless SM.

list of criteria for good metrics

- 1) Consistently measured, without subjective criteria;
- 2) Cheap to gather, preferably in an automated way;
- 3) Expressed as a cardinal number or percentage, not in a qualitative label like “high”, “medium” and “low”;
- 4) Expressed using at least one unit of measure, such as “defects”, “hours”, or “dollars”;
- 5) Contextually specific, and relevant enough to decision-makers that they can act.

- Designing Security Metrics:
 - Designing effective security metrics requires an understanding of the business objectives, security goals, and the types of data that are available.
 - Metrics should be designed to measure the effectiveness of security controls and policies and to provide insights into the security posture of the organization.
- Data Sources:
 - The data sources for security metrics can come from a variety of sources, including security tools, logs, and other data sources.
 - The data should be reliable, accurate, and relevant to the security metrics being measured.
- Analysis of Security Metrics Data:
 - Analysis of security metrics data is essential to identify trends, patterns, and anomalies. It also helps to identify areas where security controls need to be improved.
 - Data analysis should be done using appropriate tools and techniques to ensure the accuracy and reliability of the results.

- Measuring Security Cost and Value:
 - Measuring the cost and value of security controls and policies is an important aspect of security metrics. This involves measuring the cost of implementing security controls and policies, as well as the value that they provide to the organization.
- Different Context for Security Process Management:
 - Security metrics can be used in different contexts, such as risk management, compliance management, incident management, and vulnerability management. In each of these contexts, different metrics may be relevant, and different data sources may be used.

Designing Security Metrics

Choosing Good Metrics:

- Nothing Either Good or Bad, but Thinking Makes It So
 - As you develop your security metrics, you should be less concerned with what makes a metric intrinsically good or bad and much more concerned with how you develop measurement projects that provide value and organizational benefits to your security program.
 - This means taking the time to develop metrics that are based on your unique requirements and not relying on “out-of-the-box” metrics that you apply without thinking about what the measurement is supposed to achieve.

Purpose

- Designing security metrics serves several purposes, including:
 - **Improving Security Posture:** Security metrics can be used to identify areas of weakness in an organization's security posture and to track progress in implementing security controls and addressing vulnerabilities.
 - **Demonstrating Compliance:** Security metrics can be used to demonstrate compliance with regulatory requirements and industry standards, such as ISO 27001 or NIST Cybersecurity Framework.
 - **Supporting Risk Management:** Security metrics can be used to support risk management efforts by identifying and monitoring key risk indicators and measuring the effectiveness of risk mitigation strategies.
 - **Providing Management Insight:** Security metrics can be used to provide management with insight into the effectiveness of security controls and to identify areas where additional investments in security may be necessary.
 - **Improving Incident Response:** Security metrics can be used to measure the effectiveness of incident response processes and to identify areas where improvements can be made.

Methods for Deriving Security Metrics from Security Goals

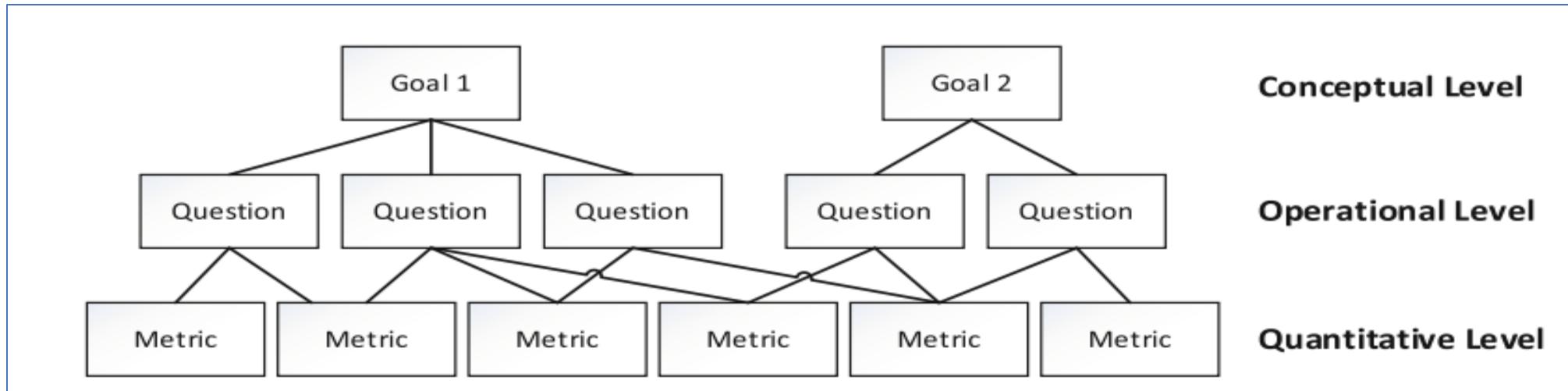
- Three approaches that support metrics derivation from goals:

Method	Proposed Year	Description
GQM (Goal-Question-Metric)	1994	Provides an outline of process that defines goals, refining them into questions and then specifying measurements and finally data to be collected.
BSC (Balanced Scorecard Framework)	2000	Goal-oriented methodology for defining measurement plans.
GAM (Goal-Argument-Metric)	2008	Framework that look into several dimensions for describing, implementing and managing strategy at different levels of an organization by linking objectives, initiatives and measures to an organization's strategy

Goal-Question-Metric (GQM) Approach

- GQM is a simple, three-step process for developing security metrics.
- The first step in the process involves defining specific goals that the organization hopes to achieve. These goals are not measurement goals, but objectives that measurement is supposed to help achieve.
- The goals are then translated into even more specific questions that must be answered before assessing whether the organization has achieved or is achieving the goals.
- Finally, these questions are answered by identifying and developing appropriate metrics and collecting empirical data associated with the measurements.
- The method ensures that the resulting metrics data remains explicitly aligned with the higher level goals and objectives of the measurement sponsors.

GQM Model is a hierarchical structure



- **Conceptual level (Goal)**

- A goal is defined for an object, for a range of reasons, with respect to different models of value, from different perspectives and relative to a specific domain.

- **Operational level (Question)**

- A set of questions is utilized to define models of the object of study and after that emphasizes on that object to describe the evaluation or accomplishment of a particular goal.

- **Quantitative level (Metric)**

- A set of measurements, taking into account the models, associated with every question in order to answer it quantifiably.

Setting Goals

- Goals give GQM measurements their power, so setting appropriate goals becomes the most important part of the metrics process.
- The GQM method includes a basic template concept for articulating the goals of a security measurement or improvement project quickly and succinctly.

Goal Component	Description	Example
Outcome	The purpose of the project, what will be achieved	Improvement, assessment, understanding
Elements	The boundaries and objects (systems, processes, characteristics) involved in or impacted by the goal	Vulnerabilities, network components, regulatory compliance, system users
Perspective	The point of view taken to understand the goal	External attackers, compliance auditors

Asking Questions

- Goal statements are conceptual in nature. They do not define how the attributes and targets of the goal will be operationally addressed.
- Individual goals are translated into a series of questions that enable the components of the goal to be achieved or evaluated for success.
- These questions articulate the goal and the measurement project in terms of what objects or activities must be observed and what data must be collected to address the individual components of the goal statement.

Assigning Metrics

- After questions have been developed to define the goal operationally, the goal can begin to be characterized at a data level, and metrics can be assigned that will provide answers.
- Designing metrics becomes much more intuitive, because only certain measurements will produce the data necessary to answer the very specific questions that the goal has produced.

Example: Security-Related Downtime

Understanding how long your systems are up and available to users is a common IT metric. Understanding how security impacts availability is also important, particularly when you need to compare security to other IT challenges.

Goal Statement *The goal of this project is to understand security impacts on system availability by comparing security-related downtime to general availability from the perspective of the security team.*

Question How often is the system down due to failure?

Metrics Time between failures

Failure duration

Mean system availability

Question How often is the system down due to maintenance?

Metrics Time between maintenance

Maintenance duration

Mean system availability

Metrics How often is downtime the result of a security event?

Question Number of security events in time period

Duration of event remediation

Goal Statement	<i>The goal of this project is to evaluate the company's compliance with the HIPAA security regulations by comparing company knowledge and activities to the HIPAA compliance guidance for IT systems provided in NIST SP 800-66 from the perspective of regulatory auditors.</i>
Question Metrics	Does the company have a security management process? Number of assets and information systems that create, receive, transmit, or maintain electronic personal health information (EHPI) Number (percentage) of assets and information systems that have not been assessed for EHPI
Question Metrics	What are the risks to EHPI under the company's custodianship? Number of risk assessments performed by the company in previous 12 months Mean time between risk assessments
Question Metrics	How does the company manage risks to EHPI? Number of approved controls in the company's security controls baseline Ratio of addressable or supplementary to required security controls and implementation specifications

Table 2-5. GQM Project for HIPAA Compliance Using NIST SP 800-66

Understanding Data

Data Types

- Quantitative Data
 - Expressed with numbers and analysed statistically
- Qualitative Data

Data Types

Quantitative Data

Quantitative data is expressed with numbers and analyzed statistically.

Nominal, Ordinal, interval, Ratio

Qualitative Data

Qualitative data **describes qualities or characteristics**. It is collected using questionnaires, interviews, or observation, and frequently appears in narrative form.

If you recorded your interview, the video, audio, and transcripts would also be qualitative data. Analysis of qualitative data is very different than quantitative analysis, as the data is messier, more complex, and requires more interpretation.

Data Sources for Security Metrics

- System Data
 - System and event logs
 - System configurations
 - Source code
 - Test results such as vulnerability assessments or patch testing
- Process Data
 - Activity reporting (budgets, time tracking, training records, meeting minutes)
 - Process tracking (trouble tickets, support call records, compliance monitoring)
 - Workflow breakdowns
 - Business process diagrams (a visual representation of a process that company carries out to achieve a goal.)

Data Sources for Security Metrics (Cont..)

- Documentary Data
 - Security policies and procedures
 - Other policies (which might have an impact on security operations)
 - Audit and review reports
 - Project plans and stakeholder documents
 - Corporate records (financial statements, customer lists, contracts, e-mail)
 - Corporate documents (annual reports, shareholders briefings, SEC filings)
 - Industry reports (analyst research, government reports, market research)
- People Data
 - Surveys and questionnaires (internal and external)
 - Interviews and focus groups
 - Case studies
 - Direct observations

Analysing Security Metrics Data

Reasons for Analysis

- Security metrics data analysis is designed to answer a known, specific question about an aspect of the security program

Date	Time	Action	IN/OUT	Source IP	Destination IP	Service
Oct 28	09:34:20	Accept	OUT	xxx.xxx.110.25	xxx.xxx.200.33	HTTP
Oct 28	09:34:50	Deny	IN	xxx.xxx.66.78	xxx.xxx.110.119	ICMP
Oct 28	09:35:01	Accept	OUT	xxx.xxx.110.25	xxx.xxx.200.33	HTTP
Oct 28	09:35:15	Drop	OUT	xxx.xxx.66.92	xxx.xxx.125.10	FTP
...						

Consider a situation in which a firewall administrator must report monthly on the number of accepted and rejected connections through the corporate perimeter.

Analysing Security Metrics Data

Types:

- Applied Analysis
 - When your security metrics data analysis is designed to answer a known, specific question about an aspect of the security program, is applied analysis.
 - Examples include analyses such as those mentioned in the preceding section, in which statistics on events or security operations are needed for reporting or compliance purposes.
- Exploratory Analysis
 - When you analyze data for the purposes of answering new questions, or even for developing those new questions on the basis of existing information or knowledge, you begin to move from applied analysis to exploratory analysis

Preparing for Data Analysis

- Source of the Data
 - System logs
 - Security event and incident management (SEIM) systems
 - Scanners and analysis tools
 - Audit reports
 - User surveys
 - Company databases (operational and historical)
 - Policies and other records and documents
- In some cases, you may be pulling data from one source that has been collected or aggregated from another source.

Scale of the Data

- **Nominal** Names or labels only, with no quantitative meaning involved even if numbers are used;

Category Value	Operating System
1	Windows XP
2	Windows Vista
3	HP-UX
4	Solaris
5	Linux
6	Mac OS X

Table 3-1. Nominal Categories for OS Type

- For analytical purposes, can use nominal data to build frequency distributions and perform cross-tabulation if you have more than one set of nominal data.
- It is not appropriate to use statistical techniques such as the *mean* (commonly called the average, although the two are different), or the *median* (the middle value) on nominal data

Scale of the Data

- **Ordinal** Indicates ranking order, but with no insight into the differences between rankings; first, second, and third place race results.
- That is Ordinal data does not provide any information regarding the amount of difference between the rankings, such as how much faster the winner of the race was compared to the runner up.
- Ordinal data uses numbers to describe a more complex relationship between the targets of observation that is found in nominal data
- Analysis techniques for ordinal data are much like those of nominal data, involving counts of which observations fall into which ranks and the distribution of the data.
- Although people often do it, it is still inappropriate to apply means or averages to ordinal data, because the ordinal scale does not give any insight into the differences between ordinal rankings.

Scale of the Data

- **Interval**
- Where ordinal data describes a ranking relationship, but with no real measure of the distance between individual rankings, interval data involves increases in rank in which the distance between the ranks is measured in some sort of standard unit. Thus the amount of difference between ranks means something.
- Measures of temperature on the Celsius and Fahrenheit scales are good examples of interval data, because the difference between 10 degrees and 20 degrees is the same as the distance between 0 degrees and 10 degrees on each scale .
- It is possible to do more analytically with interval data than with nominal or ordinal data because we are now playing with real numbers.
- We can add, subtract, and multiply measurements.
- We cannot divide or develop ratios between data, however, since the zero point on an interval scale is arbitrary and it is possible to use negative numbers (as with temperature), although this is not always part of the scale (as with academic grades).
- But most common statistical techniques become available with interval data, including the mean, the median, the mode, and the standard deviation.
- Interval data allows us to analyze dispersion, or how “spread out” our data is, and this in turn opens up some interesting probabilistic analysis techniques and the possibility of inferential statistics (those that generalize and predict) rather than more simple descriptive).

Scale of the Data

- **Ratio** Ratio data is pretty much the same as interval data, with the addition of an absolute zero point where nothing exists to measure.
- On a ratio scale, not only is the difference between 0 and 1 the same as the difference between 1 and 2 (as with interval data), but the difference between 0 and 1 is also half the difference between 0 and 2.
- Measurements such as weight and length are measured on ratio scales.
- Analytically, ratio and interval data are very similar, because the data is truly quantitative and allows for a variety of statistical techniques to be performed.

Qualitative Data

- **Data from observations:** Empirical data is based on direct observation. Qualitative data can be highly empirical. Qualitative data of this kind may include written research notes, photographs and drawings, video or audio recordings, and transcriptions of such data.
- **Data from responses:** Response data comes from interviews and interactions with people as individuals and as groups. This type of qualitative data is in the form of records of these interactions, with one person asking questions that are answered by others. The data is still empirical, based on direct observation of the interviewees' responses, but response data tends to be more structured and specific than ethnographic observations,
- **Data from Records and Artifacts:** The third type of qualitative data comprises information produced by our activities. Written documents and texts are common examples of qualitative data, from books and periodicals, to policy documents and corporate reports, to HTML pages and source code. This type of data reflects what you are measuring or observing.

Cleaning or Normalizing

- Does the Data Require Cleaning or Normalizing?
 - For Consistency and Accuracy
 - Missing Data and Outliers
 - Transforming Data

Analysis Tools and Techniques

- Techniques
 - Descriptive Statistics
 - Distribution
 - Central Tendency
 - Mode
 - Median
 - Mean
 - Dispersion
 - Range
 - Variance
 - Standard Deviation
- Tools
 - Spreadsheets
 - Statistical Software

Data Security and Privacy

DSE 3258

L13 —Digital Signature

Digital Signature- Introduction

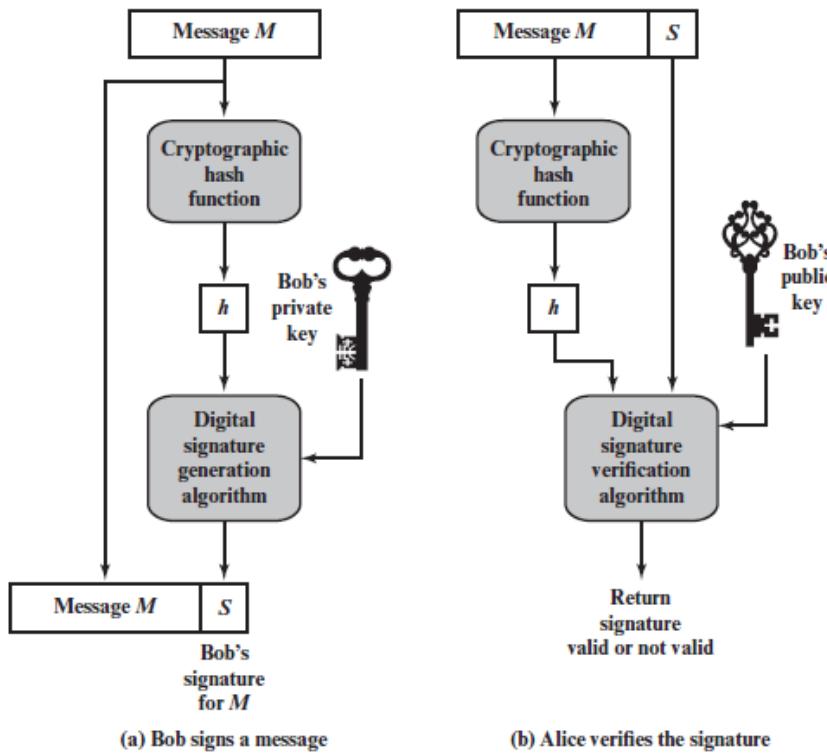


Figure 13.1 Simplified Depiction of Essential Elements of Digital Signature Process

Properties

- Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two parties are possible.
- For example, suppose that John sends an authenticated message to Mary. Consider the following disputes that could arise.
 - 1. Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.
 - 2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

- In situations where there is not complete trust between sender and receiver, something more than authentication is needed.
- The digital signature must have the following properties:
 - It must verify the author and the date and time of the signature.
 - It must authenticate the contents at the time of the signature.
 - It must be verifiable by third parties, to resolve disputes.
- Thus, the digital signature function includes the authentication function

DSA – attacks and forgeries

- Types of attacks.
- **Key-only attack:** C only knows A's public key.
- **Known message attack:** C is given access to a set of messages and their signatures.
- **Generic chosen message attack:** C chooses a list of messages before attempting to break A's signature scheme, independent of A's public key. C then obtains from A valid signatures for the chosen messages. The attack is generic, because it does not depend on A's public key; the same attack is used against everyone.
- **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.
- **Adaptive chosen message attack:** C is allowed to use A as an “oracle.” This means that C may request from A signatures of messages that depend on previously obtained message-signature pairs.

Note:

A – user

C - attacker

DSA – attacks and forgeries

- **Successful attacks:**

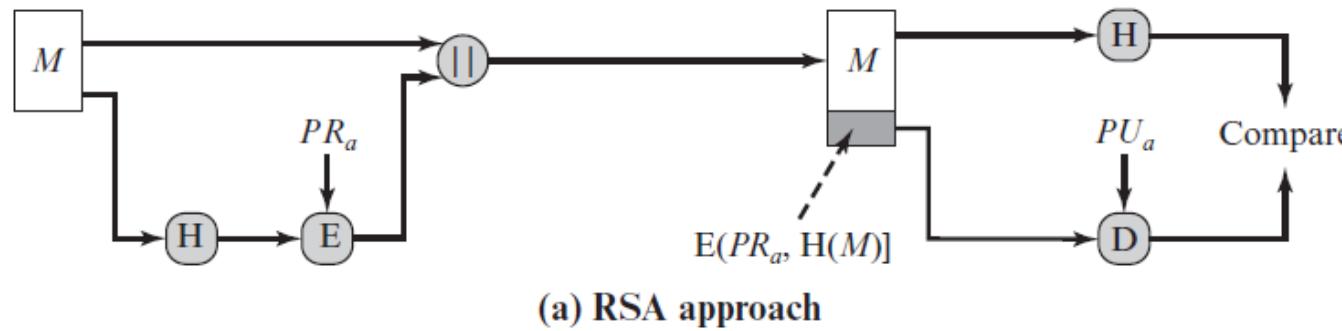
Note:
A – user
C - attacker

- **Total break:** C determines A's private key.
- **Universal forgery:** C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.
- **Selective forgery:** C forges a signature for a particular message chosen by C.
- **Existential forgery:** C forges a signature for at least one message. C has no control over the message. Consequently, this forgery may only be a minor nuisance to A.

Digital Signature Requirements

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information only known to the sender to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

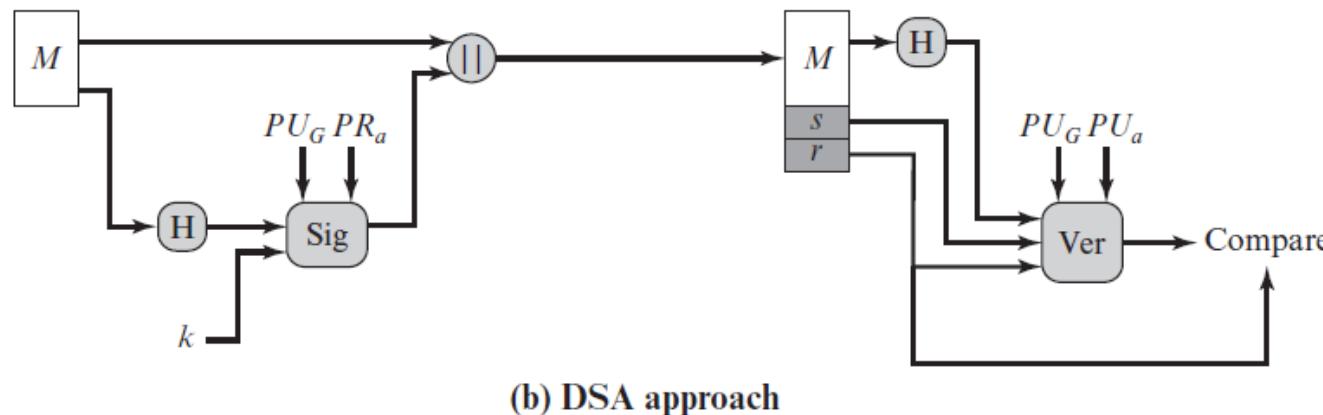
Digital Signature using RSA



- In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length.
- This hash code is then encrypted using the sender's private key to form the signature.
- Both the message and the signature are then transmitted.
- The recipient takes the message and produces a hash code.
- The recipient also decrypts the signature using the sender's public key.
- If the calculated hash code matches the decrypted signature, the signature is accepted as valid.
- Because only the sender knows the private key, only the sender could have produced a valid signature.

NIST Digital Signature Algorithm

- The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the *Digital Signature Algorithm (DSA)*.
- For hashing - Secure Hash Algorithm (SHA)
- DSA is used only for authentication not for confidentiality i.e. only signatures can be generated not to encrypt the message.



NIST DSA (contd..)

At sender end

- The hash code is provided as input to a signature function along with a random number k generated for this particular signature.
- The signature function also depends on the sender's private key (PR_a) and a set of parameters known to a group of communicating principals.
- We can consider this set to constitute a global public key (PU_G).
- The result is a signature consisting of two components, labeled s and r .

NIST DSA (contd..)

At receiver end

- The hash code of the incoming message is generated. The hash code and the signature are inputs to a verification function.
- The verification function also depends on the global public key as well as the sender's public key (PUa), which is paired with the sender's private key.
- The output of the verification function is a value that is equal to the signature component r if the signature is valid.
- The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

Global Public-Key Components

- p prime number where $2^{L-1} < p < 2^L$
for $512 \leq L \leq 1024$ and L a multiple of 64;
i.e., bit length L between 512 and 1024 bits
in increments of 64 bits
- q prime divisor of $(p - 1)$, where $2^{N-1} < q < 2^N$
i.e., bit length of N bits
- g = $h(p - 1)/q$ is an exponent mod p ,
where h is any integer with $1 < h < (p - 1)$
such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

- x random or pseudorandom integer with $0 < x < q$

User's Public Key

$$y = g^x \bmod p$$

User's Per-Message Secret Number

- k random or pseudorandom integer with $0 < k < q$

Signing

$$\begin{aligned}r &= (g^k \bmod p) \bmod q \\s &= [k^{-1} (H(M) + xr)] \bmod q \\ \text{Signature} &= (r, s)\end{aligned}$$

Verifying

$$\begin{aligned}w &= (s')^{-1} \bmod q \\u_1 &= [H(M')w] \bmod q \\u_2 &= (r')w \bmod q \\v &= [(g^{u_1} y^{u_2}) \bmod p] \bmod q \\ \text{TEST: } v &= r'\end{aligned}$$

M = message to be signed

$H(M)$ = hash of M using SHA-1

M', r', s' = received versions of M, r, s

Figure 13.3 The Digital Signature Algorithm (DSA)

