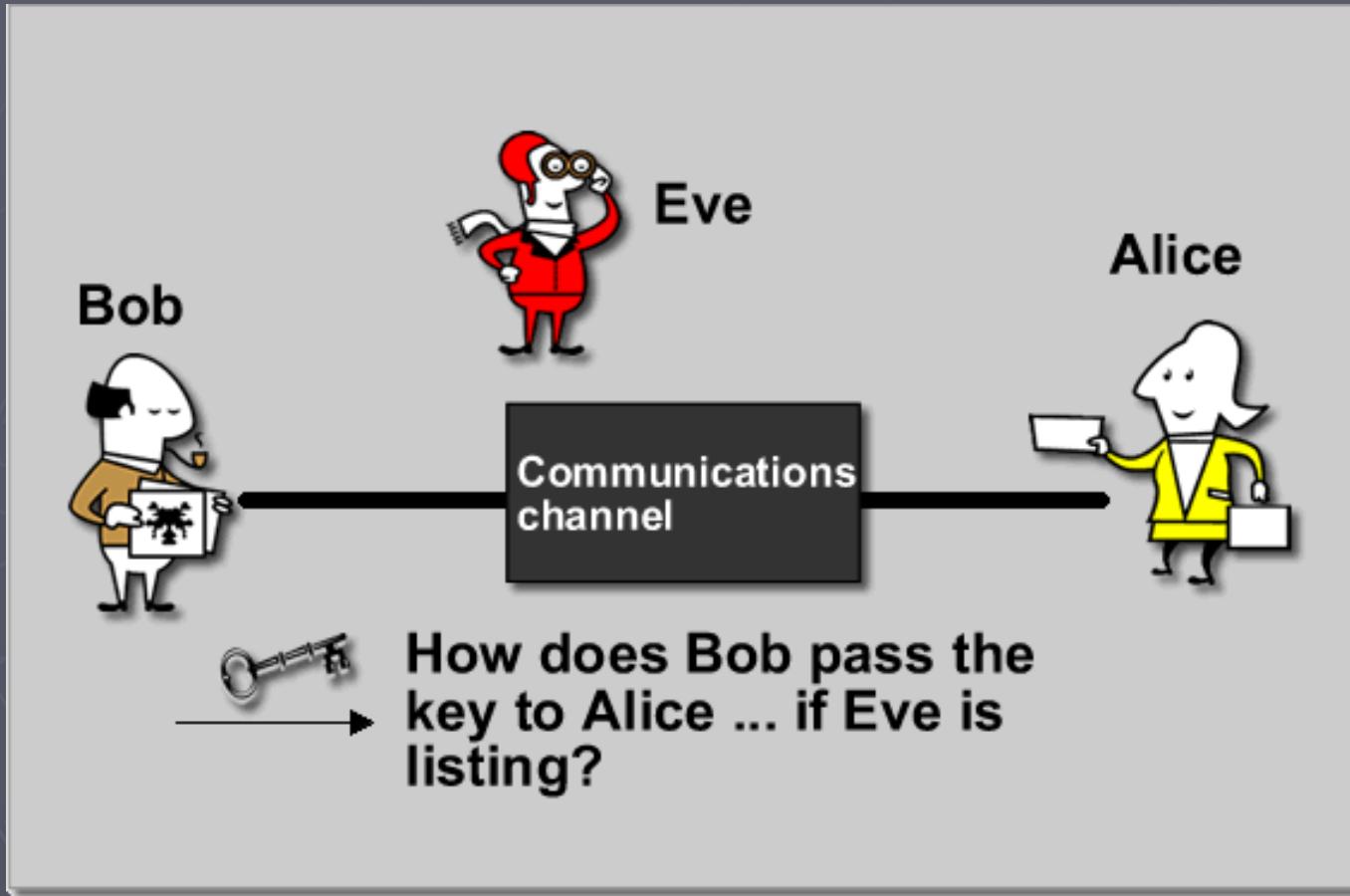


Diffie- Hellman Key Agreement

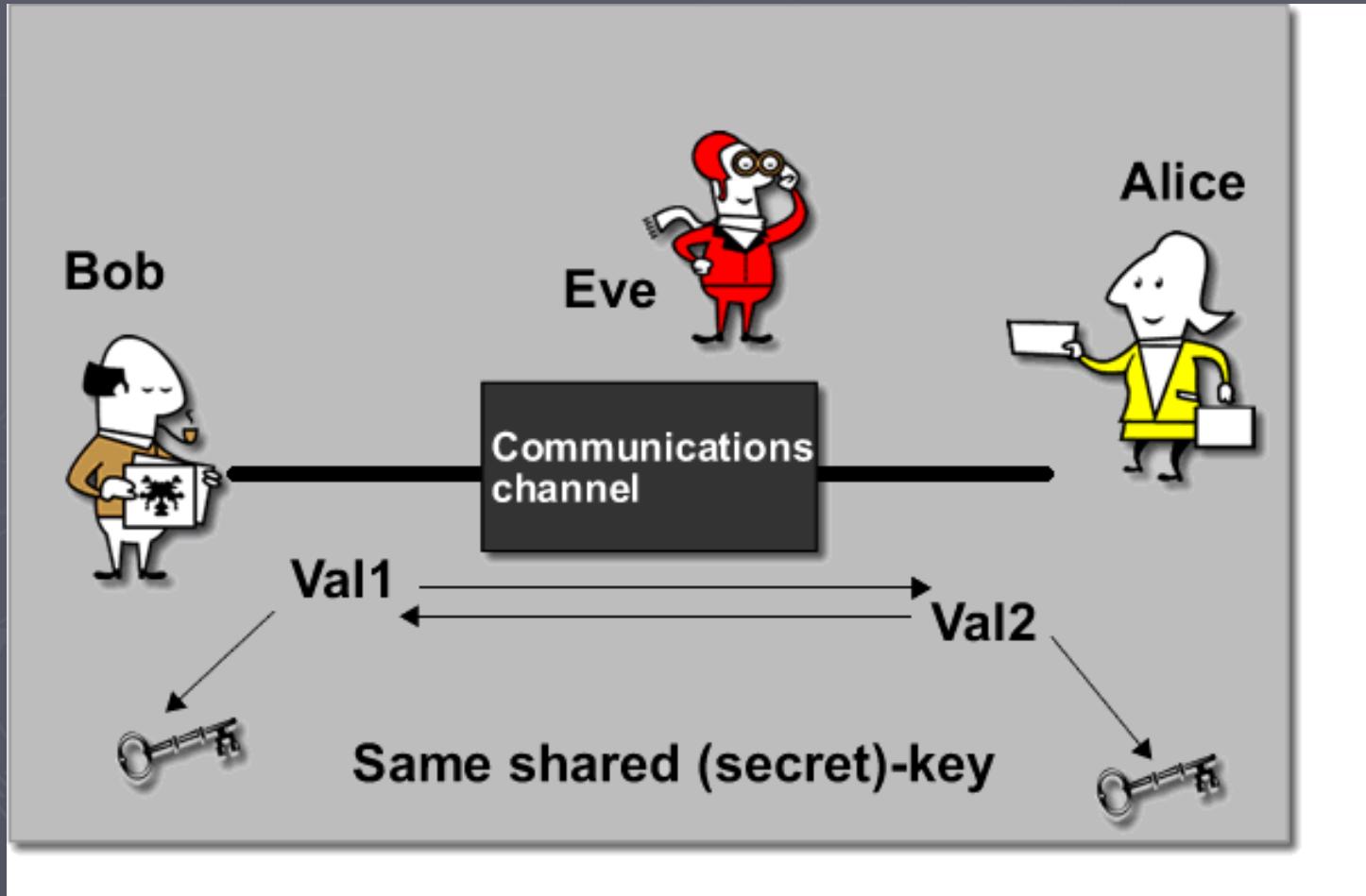
Diffie-Hellman

- ▶ Diffie-Hellman is a key exchange protocol developed by Diffie and Hellman in 1976.
- ▶ The purpose of Diffie-Hellman is to allow two entities to exchange a secret over a public (insecure) medium without having any prior secrets.

Key Establishment: The problem (cont.)



Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange

- ▶ Suppose we have two people wishing to communicate: Alice and Bob.
- ▶ They do not want Eve (eavesdropper) to know their message.

Algorithm

- ▶ Requires two large numbers, one prime p , and generator g is a primitive root of mod p , (p and g are both publicly available numbers).

Note: Anyone has access to these numbers.

- ▶ Users pick random private values x ($x < p$) and y ($y < p$)
- ▶ Compute public values
 - $R_1 = g^x \text{ mod } p$
 - $R_2 = g^y \text{ mod } p$
- ▶ Public values R_1 and R_2 are exchanged
- ▶ Compute shared, private key
 - $k_{\text{alice}} = (R_2)^x \text{ mod } p$
 - $k_{\text{bob}} = (R_1)^y \text{ mod } p$
- ▶ Algebraically it can be shown that $k_{\text{alice}} = k_{\text{bob}}$
 - Users now have a symmetric secret key to encrypt

Proof

- We know

$$R1 = g^x \bmod p$$

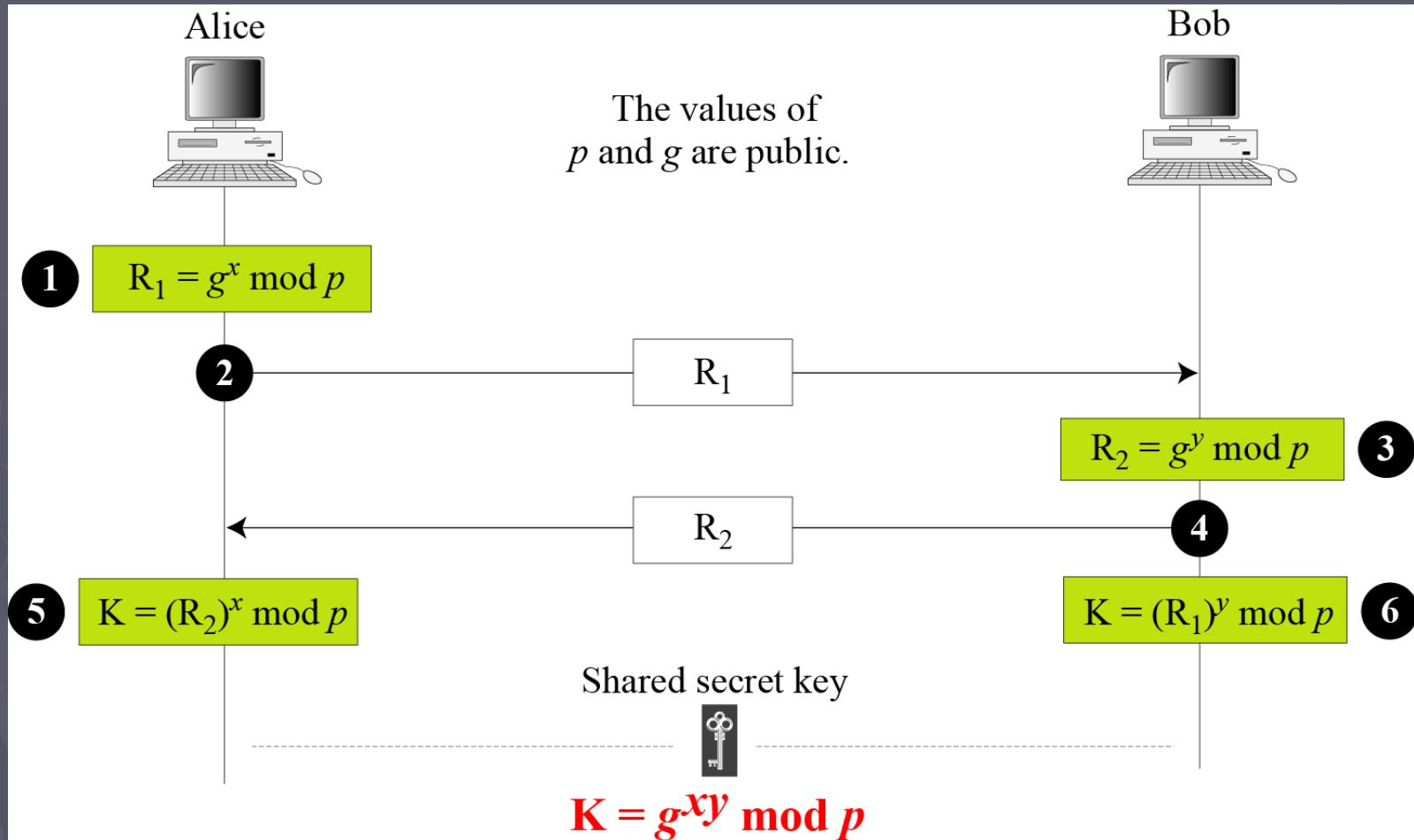
$$R2 = g^y \bmod p$$

- $k_{\text{alice}} = (R2)^x \bmod p$
= $(g^y \bmod p)^x \bmod p$
= $(g^y)^x \bmod p$
= $(g)^{yx} \bmod p$
= $(g^x)^y \bmod p$
= $(g^x \bmod p)^y \bmod p$
= $(R1)^y \bmod p$
= k_{bob}

Diffie-Hellman Key Exchange

- ▶ If Eve wants to compute k , *then she would* need either a or b .
- ▶ Otherwise, Eve would need to solve a Discrete Logarithm Problem.
 - There is no known algorithm to achieve this in a reasonable amount of time.

Diffie-Hellman Key Exchange



Example

- ▶ Alice and Bob get public numbers
 - $P = 23, G = 9$
- ▶ Alice and Bob pick private values $x=4$ & $y=3$ respectively
- ▶ Alice and Bob compute public values
 - $R_1 = 9^4 \text{ mod } 23 = 6561 \text{ mod } 23 = 6$
 - $R_2 = 9^3 \text{ mod } 23 = 729 \text{ mod } 23 = 16$
- ▶ Alice and Bob exchange public numbers
- ▶ Alice and Bob compute symmetric keys
 - $k_{\text{alice}} = (R_2)^x \text{ mod } p = 16^4 \text{ mod } 23 = 9$
 - $k_{\text{bob}} = (R_1)^y \text{ mod } p = 6^3 \text{ mod } 23 = 9$
- ▶ Alice and Bob now can talk securely!

Example

- ▶ Alice and Bob get public numbers
 - $P = 17, G = 2$
- ▶ Alice and Bob pick private values $x=3$ & $y=7$ respectively
- ▶ Alice and Bob compute public values
 - $R_1 = 2^3 \text{ mod } 17 = 8 \text{ mod } 17 = 8$
 - $R_2 = 2^7 \text{ mod } 17 = 128 \text{ mod } 17 = 9$
- ▶ Alice and Bob exchange public numbers
- ▶ Alice and Bob compute symmetric keys
 - $k_{\text{alice}} = (R_2)^x \text{ mod } p = 9^3 \text{ mod } 17 = 15$
 - $k_{\text{bob}} = (R_1)^y \text{ mod } p = 8^7 \text{ mod } 17 = 15$
- ▶ Alice and Bob now can talk securely!

Example in Two Steps

$$p = 17, g = 2, x = 3, y = 7$$

$$(2^3)^7 \bmod 17 = (2^7)3 \bmod 17$$

$$2^{21} \bmod 17 = 2^{21} \bmod 17$$

Alice			Bob			
Secret	Public	Calculates	Sends	Calculates	Public	Secret
a	p, g		$p, g \rightarrow$			b
a	p, g, A	$g^a \text{ mod } p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \text{ mod } p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \text{ mod } p = s$		$A^b \text{ mod } p = s$	p, g, A, B	b, s

1. Alice and Bob agree to use a prime number $p=23$ and base $g=5$.
2. Alice chooses a secret integer $a=6$, then sends Bob $A = g^a \text{ mod } p$
 - $A = 5^6 \text{ mod } 23$
 - $A = 15,625 \text{ mod } 23$
 - $A = 8$
3. Bob chooses a secret integer $b=15$, then sends Alice $B = g^b \text{ mod } p$
 - $B = 5^{15} \text{ mod } 23$
 - $B = 30,517,578,125 \text{ mod } 23$
 - $B = 19$
4. Alice computes $s = B^a \text{ mod } p$
 - $s = 19^6 \text{ mod } 23$
 - $s = 47,045,881 \text{ mod } 23$
 - $s = 2$
5. Bob computes $s = A^b \text{ mod } p$
 - $s = 8^{15} \text{ mod } 23$
 - $s = 35,184,372,088,832 \text{ mod } 23$
 - $s = 2$
6. Alice and Bob now share a secret: $s = 2$. This is because $6*15$ is the same as $15*6$. So somebody who had known both these private integers might also have calculated s as follows:
 - $s = 5^{6*15} \text{ mod } 23$
 - $s = 5^{15*6} \text{ mod } 23$
 - $s = 5^{90} \text{ mod } 23$
 - $s =$
 $807,793,566,946,316,088,741,610,050,849,573,099,185,363,389,551,639,556,884,765,625$
 $\text{mod } 23$
 - $s = 2$

Security of Diffie-Hellamn

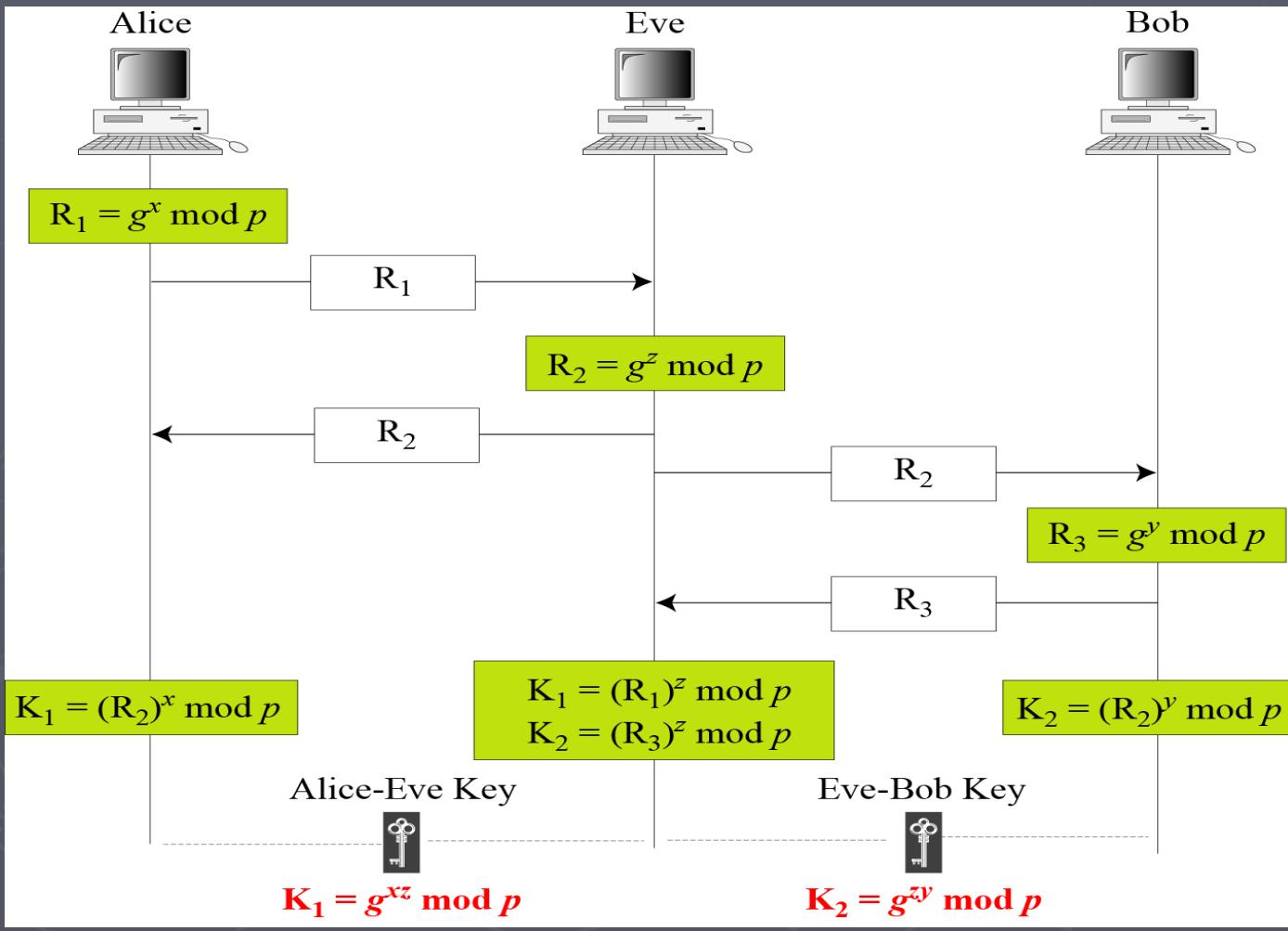
► This protocol vulnerable to two attacks:

- The Man-in-the-middle attack
- The Discrete logarithmic attack

Man-in-the-middle attack

- ▶ (p and g are publicly known)
- ▶ An adversary Eve intercepts Alice's public value and sends her own public value to Bob.
- ▶ When Bob transmits his public value, Eve substitutes it with her own and sends it to Alice.
- ▶ **Eve and Alice** thus agree on one shared key and **Eve and Bob** agree on another shared key.
- ▶ After this exchange, **Eve 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 is present because Diffie-Hellman key exchange does not authenticate the participants.

Man-in-the-middle attack (cont.)



Solution to Man-in-the-middle attack

- ▶ The basic idea is as follows.
 - Prior to execution of the protocol, the two parties Alice and Bob each obtain a public/private key pair and a certificate for the public key.
 - During the protocol, Alice calculates a signature on certain messages, covering the public value $g^a \text{ mod } p$. Bob proceeds in a similar way. Even though **Eve** is still able to intercept messages between Alice and Bob,
 - She cannot forge signatures without Alice's private key and Bob's private key. Hence, the enhanced protocol defeats the man-in-the-middle attack.

Discrete Logarithmic Attack

- ▶ The security of the key exchange is based on the difficulty of the discrete logarithm problem.
- ▶ Eve can intercept R₁ and R₂.
- ▶ If she can find x from $R_1 = g^x \text{ mod } p$ and y from $R_2 = g^y \text{ mod } p$,
- ▶ Then she calculate the symmetric key $K = g^{xy} \text{ mod } p$.
- ▶ The secret key is not secret anymore.

Discrete Logarithmic Attack *(cont.)*

- ▶ To make Diffie-Hellman safe from the discrete logarithm attack, the following are recommended.
 - The prime p must be very large. Then it is computationally infeasible to calculate the shared secret key $k = (g^{xy} \bmod p)$ given the two public values $(g^x \bmod p)$ and $(g^y \bmod p)$.
 - Bob and Alice must destroy x and y after they have calculated the symmetric key. The values of x and y must be used only once.

Summary

- ▶ Key agreement protocol- is a specific method of exchanging cryptographic keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography.
- ▶ The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
- ▶ This key can then be used to encrypt subsequent communications using a symmetric key cipher.
- ▶ Does not
 - Authenticate
- ▶ Defeats man-in-the-middle attack
- ▶ Defeats Discrete Logarithmic Attack

CO403 : CRYPTOGRAPHY AND NETWORK SECURITY (CS-II)

Dr. Udai Pratap Rao, CoED, SVNIT Surat

CO403 : CRYPTOGRAPHY AND NETWORK SECURITY (CS-II)**INTRODUCTION AND OVERVIEW**

(02 Hours)

ELEMENTARY NUMBER THEORY

(04 Hours)

Finite fields, Arithmetic and algebraic algorithms

PSEUDO RANDOM BIT GENERATORS

(02 Hours)

FORMAL DEFINITION OF SECURE ENCRYPTION

(04 Hours)

Perfect secrecy, Semantic security, IND-CPA, IND-CCA

STREAM CIPHERS

(04 Hours)

One time pad, Security proof of one time pad

BLOCK CIPHERS

(04 Hours)

Need for block ciphers, Luby-rackoff construction and its security proof, Modes of operation

HASH AND MAC FUNCTIONS

(04 Hours)

Definitions, Notions of security and unaffordability (EUF-CMA), Merkle-Damgard family of hash functions

HARD PROBLEMS

(04 Hours)

Discrete logarithm, Factorization

PUBLIC KEY CRYPTO SYSTEMS

(06 Hours)

Diffie Hellman, RSA encryption; Proofs of security under hardness assumptions, Digital Signature

NETWORK SECURITY

(03 Hours)

IDENTITY MANAGEMENT

(03 Hours)

ADVANCED TOPICS

(02 Hours)

(Total Contact Time: 42 Hours + 14 Hours = 56 Hours)

PRACTICALS

1) Implementation of Client side scripting

2) Implementation of Server side scripting

3) Implementation of mini project using above technology including the database connectivity

BOOKS RECOMMENDED

1). Dhiren Patel, Information Security: Theory and Practice, PHI, 2008/2010

2). William Stallings, "Cryptography and Network Security - Principles and Practice", 6/E, Pearson Education, 2013.

3). Douglas Stinson: "Cryptography: Theory and Practice, Third Edition", 3/E, Chapman and Hall/CRC, 2005

4). Menezes Bernard, Network Security and Cryptography, Cengage Learning India, 2010

5). Alfred. J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: "Handbook of Applied Cryptography", 1/E, CRC, 1996

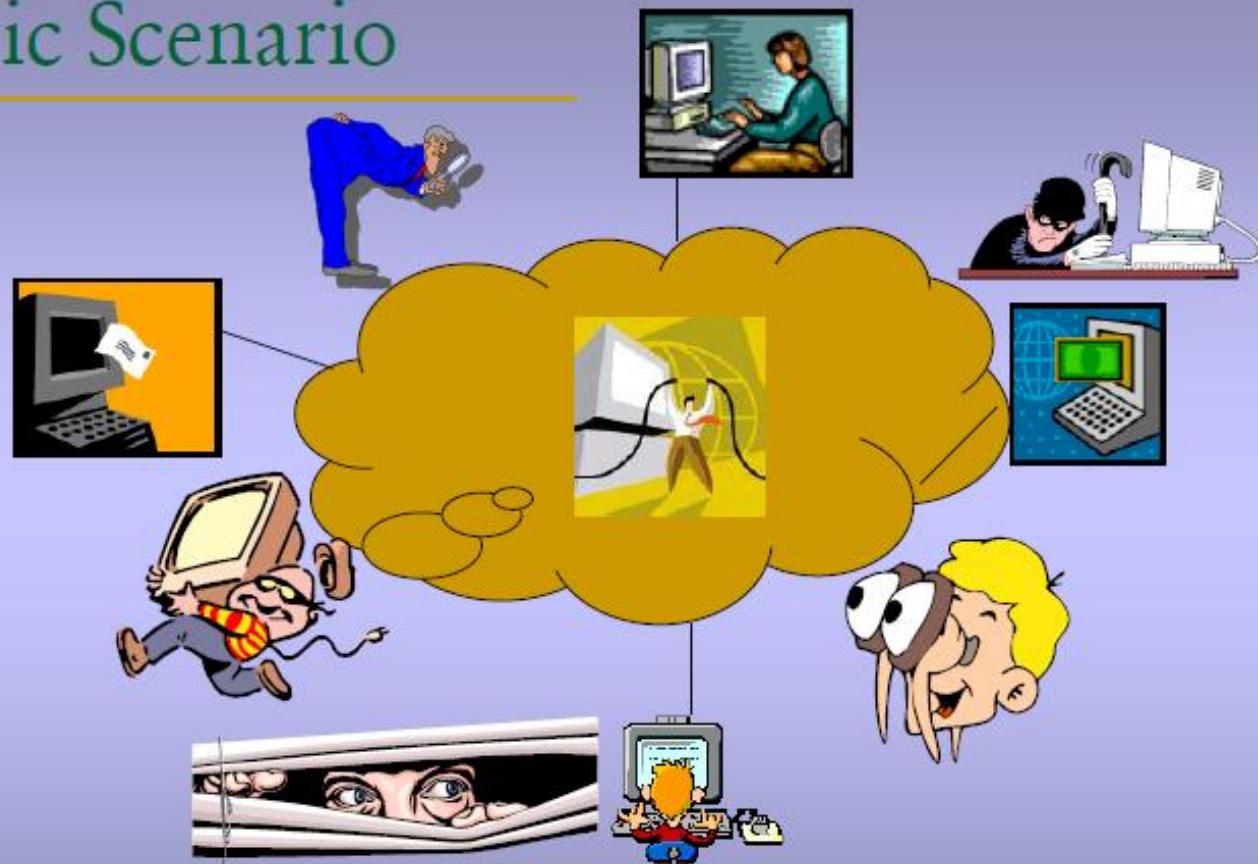
COURSE OUTCOMES

- After successful completion of this course,
 - Student will be able to Understand the concepts related to the basics of cryptography and computer security.
 - Deduce the mechanisms to be employed while trying to satisfy any of the security services
 - Apply the concept of security services and mechanisms from the application developers and network administrator's perspective.

Basic definitions

- Cryptography
 - study of encryption principles/methods
- Computer Security
 - generic name for the collection of tools designed to protect data
- Network Security
 - measures to protect data during their transmission
- Internet Security
 - measures to protect data during their transmission over a collection of interconnected networks

Basic Scenario



Basic Tasks to Secure the Information

- When an organization secures its information, it completes a few basic tasks:
 - It must analyze its assets and the threats these assets face from threat agents
 - It identifies its vulnerabilities and how they might be exploited
 - It regularly reviews the security policy to ensure it is adequately protecting its information

Basic Tasks to Secure the Information

- **Bottom-up approach:** major tasks of securing information are achieved from the lower levels(grassroots workers) of the organization upwards
 - This approach has one **key advantage**: the bottom-level employees have the technical expertise to understand how to secure information
 - It has a **weakness**: without approval from top levels of management, security schemas created by grassroots workers has small chance of success

Basic Tasks to Secure the Information

- **Top-down approach:** starts at the highest levels of the organization and works its way down
 - **Advantage:** the security plan initiated by top-level managers has the backing to make the plan work (funding and timing has the high level of support)

Defining Information Security

- Information security:
 - Tasks of guarding digital information, which is typically processed by a computer (such as a personal computer), stored on a magnetic or optical storage device (such as a hard drive or DVD), and transmitted over a network spacing.

Defining Information Security (continued)

- Ensures that protective measures are properly implemented
- is intended to protect information

Defining Information Security (continued)

- Analogy-
 - Alice places an object in a metal box, and then locks it with a combinational lock left there by Bob. Bob is the only person who can open the box since only he knows the combination.
 - number of digits in a combinational lock, **longer the key; stronger the encryption**, as extending the length of the key exponentially increases the number of possible key combinations.

Defining Information Security (continued)

- Information security is intended to protect information that has value to people and organizations
 - This value comes from the characteristics of the information:
 - **Confidentiality**
 - **Integrity**
 - **Availability**
- Information security is achieved through a combination of three entities

Defining Information Security (continued)

- **Confidentiality:** Prevention of unauthorized disclosure of information.
- **Integrity:** Prevention of unauthorized modification of information.
- **Availability:** Prevention of unauthorized withholding/custody of information or resources. Or keeping system available...

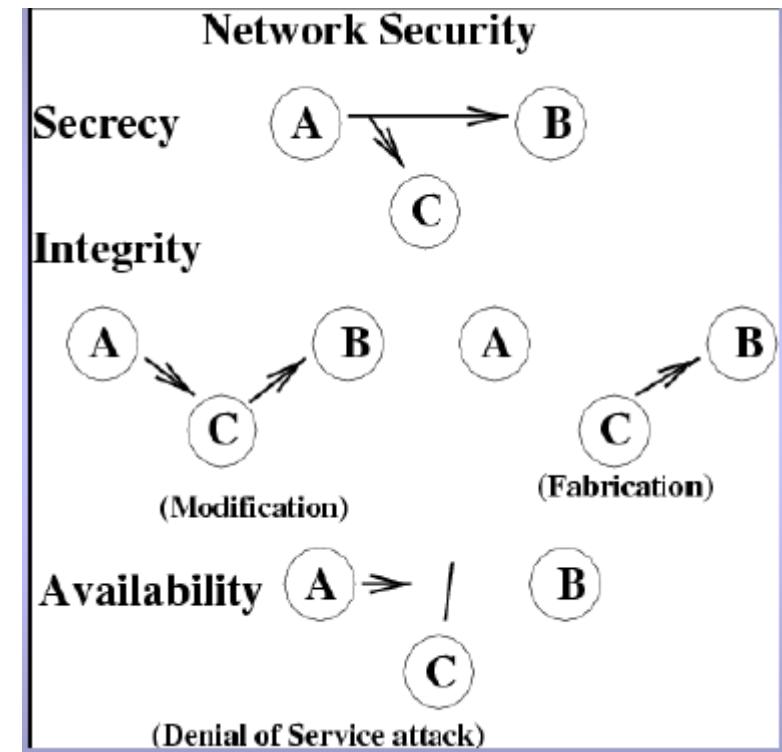
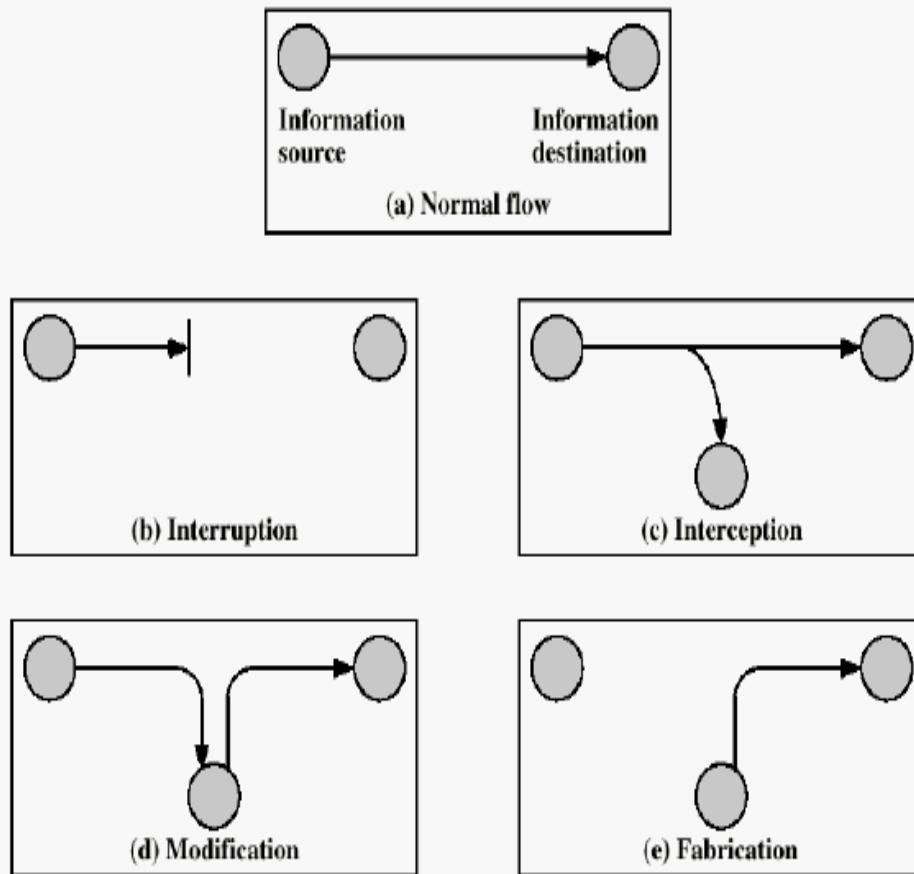
Defining Information Security (continued)

- **Nonrepudiation**: provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. In particular,
 - *Nonrepudiation of origin* proofs that the message was sent by the specified party.
 - *Nonrepudiation of destination* proofs that the message was received by the received party.

Defining Information Security (continued)

- ***Attacks against confidentiality***
 - Eavesdropping
 - traffic flow analysis
- ***Attacks against integrity***
 - Man-in-the-middle attack
- ***Attacks against availability***
 - Denial of Service attack

Security attacks

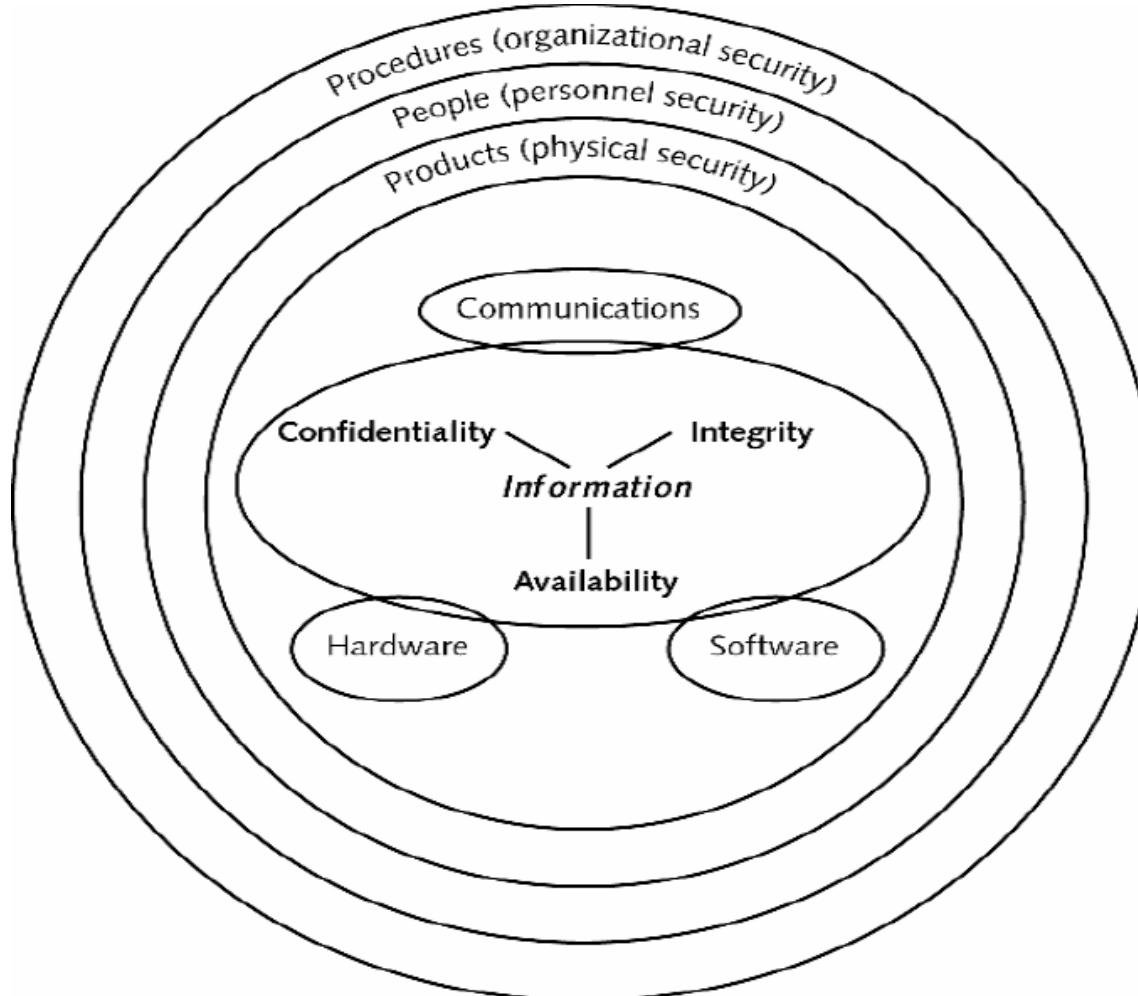


Defining Information Security (continued)

Security Service	Supporting Security Mechanisms
Confidentiality	encipherment
Traffic flow confidentiality	encipherment, traffic padding
Data integrity	encipherment, digital signature
Availability	authentication exchange
Nonrepudiation	digital signature

Relationship between security services and mechanisms

Defining Information Security (continued)



Information Security Components
CRYPTOGRAPHY AND NETWORK SECURITY
(B.Tech-IV)- UPR, CoED

Defining Information Security (continued)

Layer	Description
Products	The physical security around the data. May be as basic as door locks or as complicated as intrusion-detection systems and firewalls.
People	Those who implement and properly use security products to protect the data.
Procedures	Plans and policies established by an organization to ensure that people correctly use the products.

Information Security Layers

Defining Information Security (continued)

- A more comprehensive definition of information security is:
 - *That which **protects the integrity, confidentiality, and availability of information** on the devices that store, and transmit the information through products, people, and procedures.*

Information Security Terminology

- **Asset**
 - Something that has a value
- **Threat**
 - An event or object that may **defeat the security measures** in place and result in a loss
- **Threat agent**
 - A person or thing that has the power to carry out a threat

Information Security Terminology (continued)

- **Vulnerability**
 - Weakness that allows a threat agent to bypass security
- **Risk**
 - The likelihood that a threat agent will exploit a vulnerability
 - Realistically, risk cannot ever be entirely eliminated

Information Security Terminology (continued)

- **Security Mechanism**
 - a mechanism that is designed to detect, prevent, or recover from a security attack.
- **Security Service**
 - It makes use of security mechanisms to counter security attacks.

Information Security Terminology (continued)

Term	Example in Information Security
Asset	Employee database
Threat	Steal data
Threat Agent	Attacker
Vulnerability	Software defect
Exploit	Send virus to unprotected e-mail server
Risk	Educate users
Security Mechanism	IPS, IDS etc.....

Some More Terminologies

- plaintext - original message
- ciphertext - coded message
- cipher - algorithm for transforming plaintext to ciphertext
- key - info used in cipher and known only to sender/receiver
- encipher (encrypt)
 - converting plaintext to ciphertext
- decipher (decrypt)
 - recovering plaintext from ciphertext
- cryptography
 - study of encryption principles/methods
- cryptanalysis (codebreaking)
 - study of principles/ methods of deciphering ciphertext without knowing key
- cryptology
 - field of both cryptography and cryptanalysis

Attacker Profiles

- Six categories:
 - Hackers
 - Crackers
 - Script kiddies
 - Spies
 - Employees
 - Cyberterrorists

Hackers

- Person who uses **advanced computer skills** to attack computers, but **not** with a malicious intent
- Use their skills to expose security flaws.
- **Hacker Code of ethics:** Breaking into another person's computer is ethically acceptable as long as they don't commit theft, damage, or break of confidentiality.

Crackers

- Person who violates system security with malicious intent
- Have **advanced knowledge** of computers and networks and the skills to exploit them
- Destroy data, deny legitimate users of service, or otherwise cause serious problems on computers and networks
- "crackers are often mistakenly called hackers"

Script Kiddies

- Break into computers to create damage
- Are **unskilled** users
- Download **automated hacking software** from Web sites and use it to break into computers
- Tend to be young computer users with **almost unlimited amounts of free time** , which they can use to attack systems

Spies

- Person hired to break into a computer and steal information
- Do not randomly search for unsecured computers to attack
- Hired to attack a specific computer that contains sensitive information
- Motivation is almost always **financial.**

Employees

- One of the largest information security threats to business
- Employees break into their company's computer for these reasons:
 - To show the company a weakness in their security
 - To say, "I'm smarter than all of you"
 - For money.
 - A dissatisfied employee wanting to get back at the company

Cyberterrorists

- Experts fear that terrorists will attack the network and computer infrastructure to cause panic
- Cyberterrorists' motivation may be defined as ideology, or attacking for the sake of their principles
- One of the targets highest on the list of cyberterrorists is the Internet itself

Approaches to provide the security

- The different approaches for providing security can be categorized into the following six areas:
 - ***Attack Deterrence*** –
 - Attack deterrence refers to persuading an attacker not to launch an attack by increasing the perceived **risk of negative consequences** for the attacker.
 - Having a strong legal system may be helpful in attack deterrence.
 - However, it requires strong evidence against the attacker in case an attack was launched.

Approaches to provide the security

- ***Attack Prevention –***

- Attack prevention aims to prevent an attack by blocking it before an attack can reach the target.
- However, it is very difficult to prevent all attacks. This is because, to prevent an attack, the system requires **complete knowledge of all possible attacks** as well as the complete knowledge of **all the allowed normal activities**.
- An example of attack prevention system is a firewall .

Approaches to provide the security

– *Attack Deflection* –

- Attack deflection refers to tricking an attacker by making the attacker believe that the attack was successful,
- though in reality, the attacker was trapped by the system and deliberately made to reveal the attack.
- Research in this area focuses on attack deflection systems such as the honey pots .

– *Attack Avoidance* –

- Attack avoidance aims to make the resource unusable by an attacker even though the attacker is able to access that resource.
- An example of security mechanism for attack avoidance is the use of cryptography . Encrypting data renders the data useless to the attacker, thus, avoiding possible threat.

Approaches to provide the security

– ***Attack Detection*** –

- Attack detection refers to detecting an attack while the attack is still in progress or to detect an attack which has already occurred in the past.
- Detecting an attack is significant for two reasons; first the system must recover from the damage caused by the attack and second, it allows the system to take measures to prevent similar attacks in future.
- Research in this area focuses on building intrusion detection systems and it is equally important for database protection.

– ***Attack Reaction and Recovery*** –

- Once an attack is detected, the system must react to an attack and perform the recovery mechanisms as defined in the security policy.

Components of Intrusion Detection Systems

- An intrusion detection system typically consists of three sub systems or components:
 1. Data Preprocessor :
 - Data preprocessor is responsible for collecting and providing the audit data (in a specified form) that will be used by the next component (analyzer) to make a decision.
 - Data preprocessor is, thus, concerned with collecting the data from the desired source and converting it into a format that is comprehensible by the analyzer.

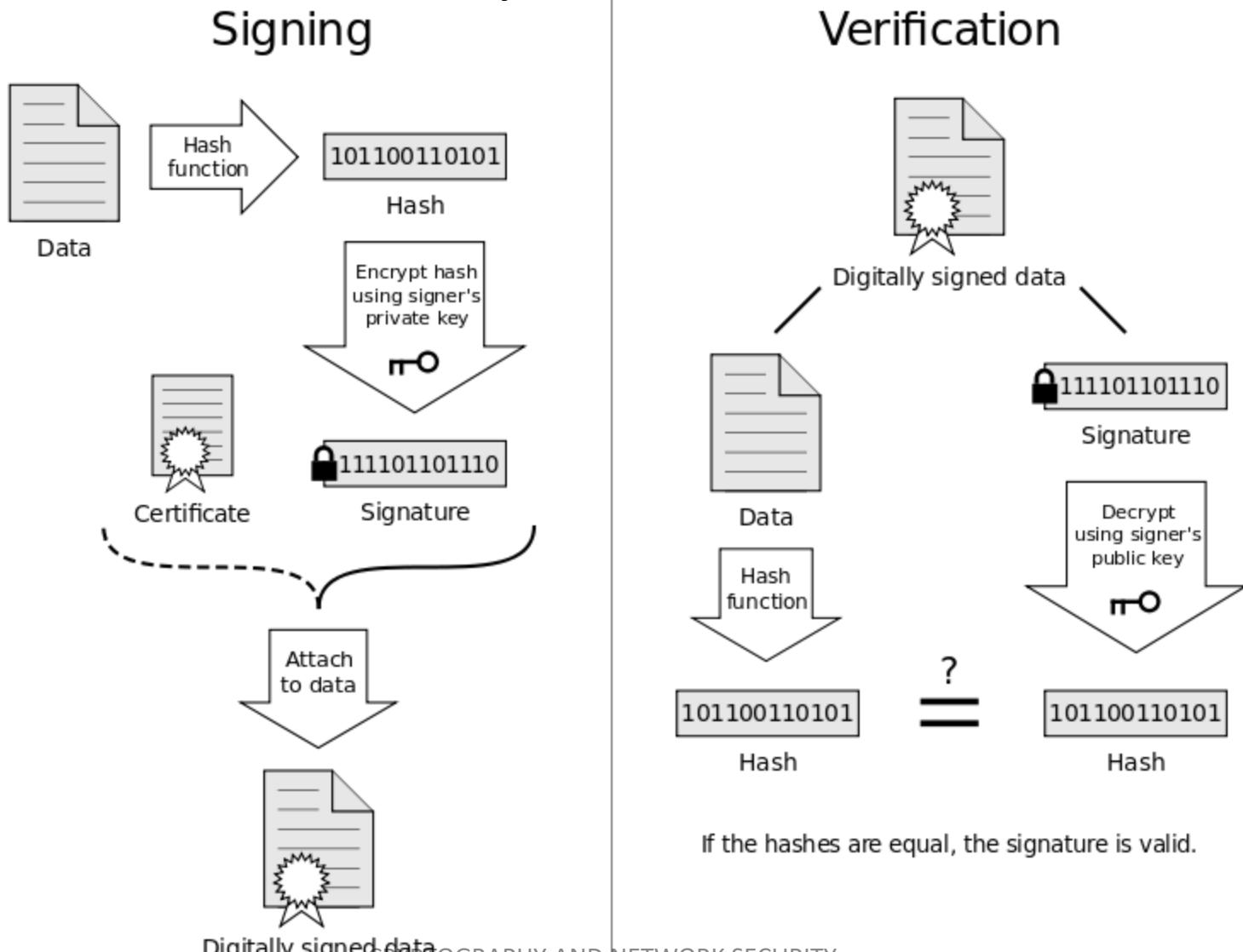
Data used for detecting intrusions range from user access patterns (for example, the sequence of commands issued at the terminal and the resources requested) to network packet level features (such as the source and destination IP addresses, type of packets and rate of occurrence of packets) to application and system level behaviour (such as the sequence of system calls generated by a process.) We refer to this data as the audit patterns.

Components of Intrusion Detection Systems

2. **Analyzer (Intrusion Detector)** — The analyzer or the intrusion detector is the core component which analyzes the audit patterns to detect attacks. This is a critical component and one of the most researched. Various ***pattern matching, machine learning, data mining and statistical techniques*** can be used as intrusion detectors. The capability of the analyzer to detect an attack often determines the strength of the overall system.
3. **Response Engine** – The response engine controls the reaction mechanism and determines how to respond when the analyzer detects an attack. The system may decide either to raise an alert without taking any action against the source or may decide to block the source for a predefined period of time. Such an action depends upon the predefined security policy of the network.

digital signature

an additional data unit that is added to the principal data unit to enable recipient to verify the source



Metrics for Performance Evaluation of Classifier

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes (Positive)	Class>No (Negative)
	Class=Yes (Positive)	a	b
	Class=No (Negative)	c	d

- The entries in the confusion matrix have the following meaning :
 - a is the number of **correct** predictions that an instance is **positive**,
 - b is the number of **incorrect** predictions that an instance **negative**,
 - c is the number of **incorrect** predictions that an instance is **positive**, and
 - d is the number of **correct** predictions that an instance is **negative**.

Metrics for Performance Evaluation of Classifier

- The *accuracy* (AC)- is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Consider a 2-class problem
 - Number of Class 0 examples = 9990
 - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is $9990/10000 = 99.9\%$
 - Accuracy is **misleading** because model does not detect any class 1 example

Contd...

- The *recall* or *true positive rate (TP)* is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

- The *false positive rate (FP)* is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = \frac{c}{c+d} = \frac{FP}{FP+TN}$$

Contd...

- The *true negative rate (TN)* is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$TN = \frac{d}{d+c} = \frac{TN}{TN+FP}$$

- The *false negative rate (FN)* is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{b}{b+a} = \frac{FN}{FN+TP}$$

Contd.....

- *The precision (P)* is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{a}{c + a} = \frac{TP}{FP + TP}$$

IDS Example

- Suppose we train a IDS model to predict whether an action is **Malicious** or **Not malicious**. After training the IDS model, we apply it to a test set of 500 new actions (also labeled) and the IDS model produces the contingency matrix below.

		True Class	
		Malicious	Not malicious
Predicted Class	Malicious	70	10
	Not Malicious	40	380

- Compute the precision of this IDS model with respect to the malicious class.
- Compute the recall of this IDS model with respect to the malicious class.

Cond...

- **High-precision and low recall with respect to Malicious:** whatever the model classifies as malicious is probably malicious. However, many actions that are truly malicious are misclassified as NOT malicious i.e <False Negative (False Acceptance)>
- **High recall and low precision with respect to malicious:** the model filters all the malicious actions, but also incorrectly classifies some genuine actions as malicious i.e. <False Positive (False Rejectance)>.

MATHEMATICS OF CRYPTOGRAPHY

PART I

MODULAR ARITHMETIC AND

CONGRUENCE

Objectives

- To review integer arithmetic, concentrating on divisibility and finding the GCD using Euclidean algorithm.
- To understand how the extended Euclidean algorithm can be used to solve linear Diophantine equations.
- To solve linear congruent equations.
- To find **additive and multiplicative inverses**.
- To **emphasize the importance of modular arithmetic and the modulo operators**, because they are extensively used in cryptography.

Book : Cryptography and Network security by Behrouz A. Forouzan

Integer Arithmetic

- In integer arithmetic, we use a set and a few operations.
- Reviewed here to create a background for modular arithmetic.

Set of Integers

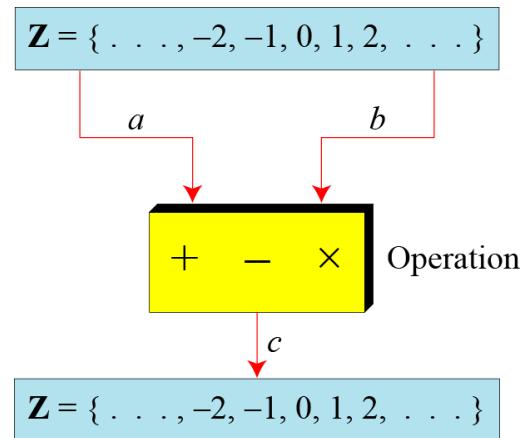
- The set of integers, denoted by \mathbb{Z} , contains all integral numbers (with no fraction) from negative infinity to positive infinity

$$\mathbb{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$$

The set of integers

Binary Operations

- In cryptography, we are interested in three binary operations applied to the set of integers. **division?**
- A binary operation takes two inputs and creates one output.



Three binary operations for the set of integers

Integer Division

- In integer arithmetic, if we divide a by n, we can get q and r.
- The relationship between these four integers can be shown as

$$a = q \times n + r$$

a= dividend

n= divisor

q= quotient

r= remainder

Integer Division(cont.)

- Assume that $a = 255$ and $n = 11$. We can find $q = 23$ and $r = 2$ using the division algorithm.

$$\begin{array}{r} 2 \ 3 \quad \longleftarrow \ q \\ \hline 2 \ 5 \ 5 \quad \longleftarrow \ a \\ 2 \ 2 \\ \hline 3 \ 5 \\ 3 \ 3 \\ \hline 2 \quad \longleftarrow \ r \\ \end{array}$$

The diagram shows the long division of 255 by 11. The quotient is 23 and the remainder is 2. The numbers 255 and 23 are highlighted in blue, while 11 and 2 are highlighted in red. Red arrows point from the labels *n*, *a*, *q*, and *r* to their respective values in the calculation.

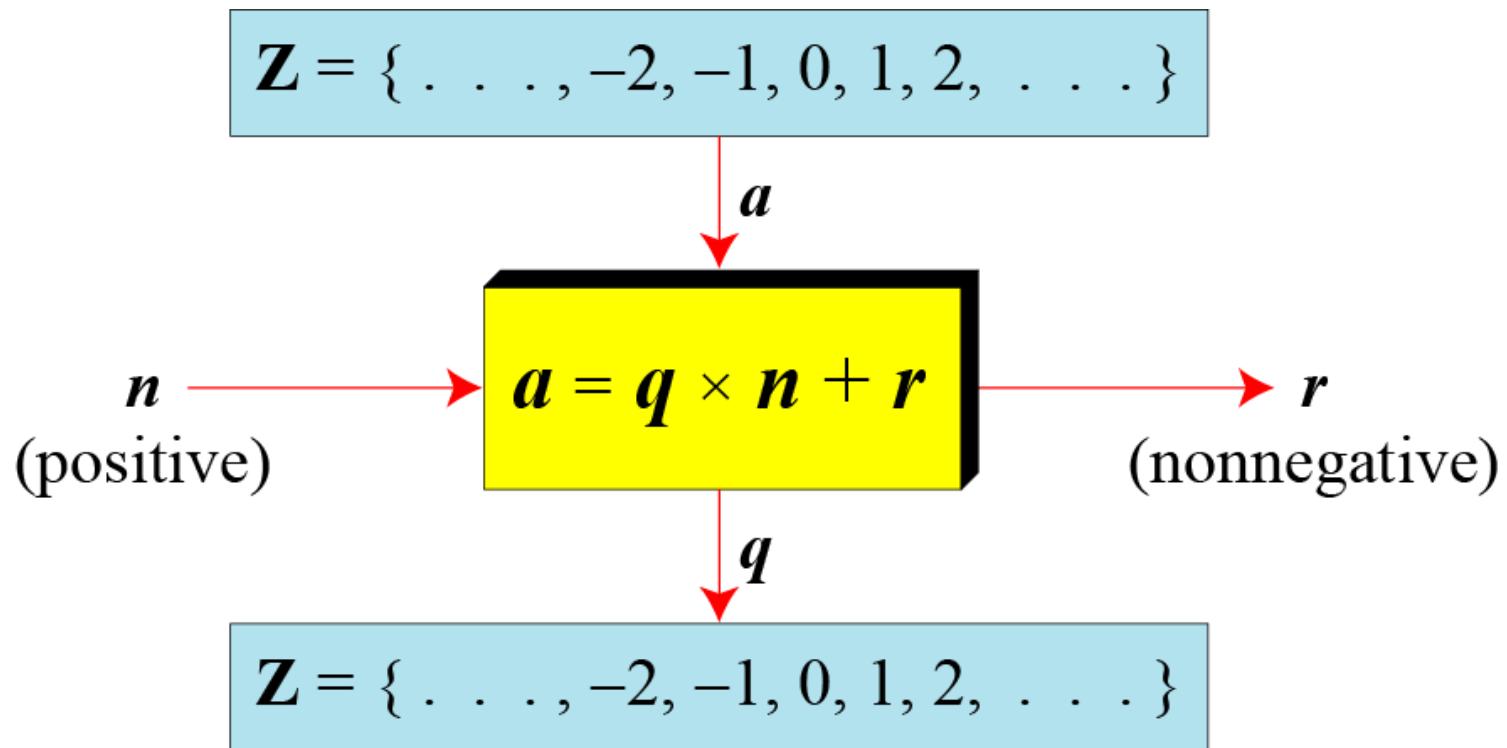
Finding the quotient and the remainder

- How to find the *quotient* and the *remainder* using language specific operators??? <in case of C language>

Integer Division(cont.)

In case of division relationship in **cryptography**, we impose two restrictions.

1. We require that divisor be a positive integer ($n > 0$)
2. We require that the remainder be a nonnegative integer ($r \geq 0$)



Division algorithm for integers

Integer Division(cont.)

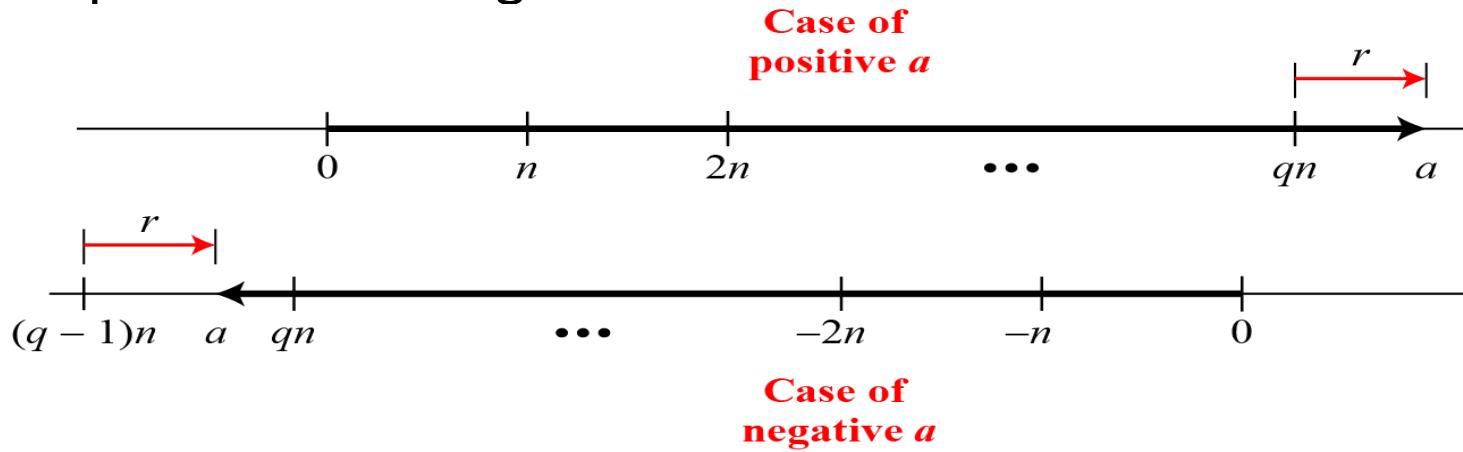
- When we use a computer or a calculator, r and q are negative when a is negative.
- How can we apply the restriction that r needs to be positive?
- The solution is simple, we decrement the value of q by 1 and we add the value of n to r to make it positive.

$$-255 = (-23 \times 11) + (-2) \quad \leftrightarrow \quad -255 = (-24 \times 11) + 9$$

- The above relation is still valid.

Integer Division(cont.)

- Graph of division algorithm



- If a is positive- move $q \times n$ units to right and move extra r units in the same direction.
- If a is negative- move $(q-1) \times n$ units to the left (q is negative in this case) and then r units in the opposite direction. <in both cases the value of r is positive>

Divisibility

- If a is not zero and we let $r = 0$ in the division relation, we get

$$a = q \times n$$

- If the remainder is zero, $n|a$ (n divides a)
- If the remainder is not zero, $n \nmid a$ (n does not divide a)

Divisibility(cont.)

- Properties

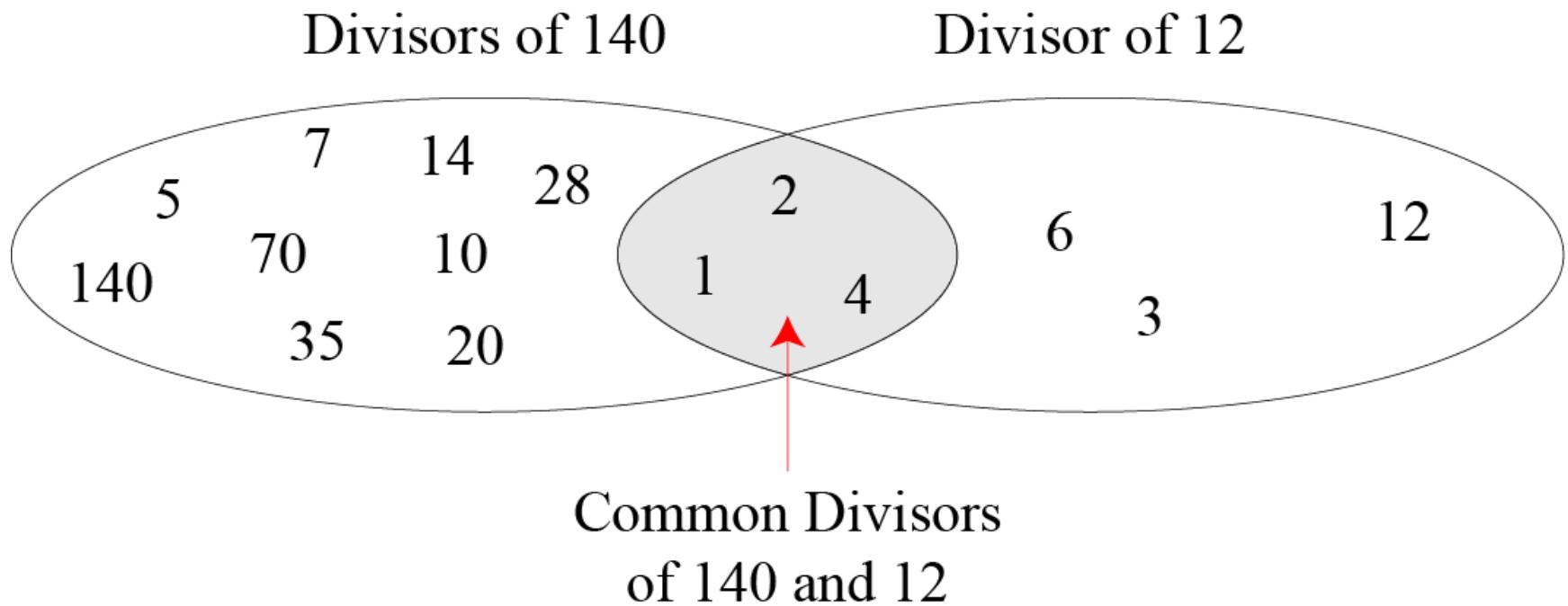
Property 1: if $a|1$, then $a = \pm 1$.

Property 2: if $a|b$ and $b|a$, then $a = \pm b$.

Property 3: if $a|b$ and $b|c$, then $a|c$.

Property 4: if $a|b$ and $a|c$, then
 $a|(m \times b + n \times c)$, where m
and n are arbitrary integers

Divisibility(cont.)



Divisibility(cont.)

Greatest Common Divisor

The greatest common divisor of two positive integers is the largest integer that can divide both integers.

Euclidean Algorithm

Fact 1: $\gcd(a, 0) = a$

Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b

Divisibility(cont.)

- For example, to calculate the $\text{gcd}(36,10)$, we use following steps:

$\text{gcd}(36, 10) = \text{gcd}(10, 6) \dots \dots \text{by fact 2}$

$\text{gcd}(10, 6) = \text{gcd}(6, 4) \dots \dots \text{by fact 2}$

$\text{gcd}(6, 4) = \text{gcd}(4, 2) \dots \dots \text{by fact 2}$

$\text{gcd}(4, 2) = \text{gcd}(2, 0) \dots \dots \text{by fact 2}$

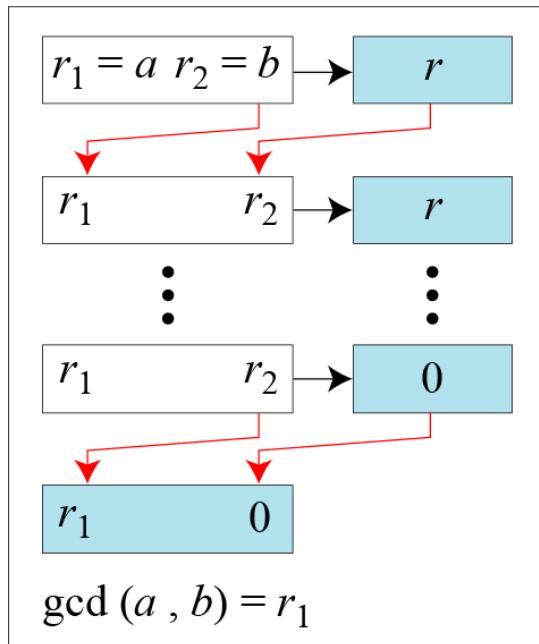
$\text{gcd}(2, 0) = 2 \dots \dots \text{by fact 1}$

Hence, Answer = 2

Divisibility(cont.)

Euclidean Algorithm:

- Two variables, $r1$ and $r2$, to hold the changing values during the process of reduction.



a. Process

```
 $r_1 \leftarrow a; \quad r_2 \leftarrow b;$  (Initialization)  
while ( $r_2 > 0$ )  
{  
     $q \leftarrow r_1 / r_2;$   
     $r \leftarrow r_1 - q \times r_2;$   
     $r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$   
}  
 $\gcd(a, b) \leftarrow r_1$ 
```

b. Algorithm

When $\gcd(a, b) = 1$, we say that a and b are relatively prime.

Divisibility(cont.)

Find the greatest common divisor of 2740 and 1760.

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Answer: $\gcd(2740, 1760) = 20$.

Divisibility(cont.)

Find the greatest common divisor of 25 and 60.

Divisibility(cont.)

Extended Euclidean Algorithm

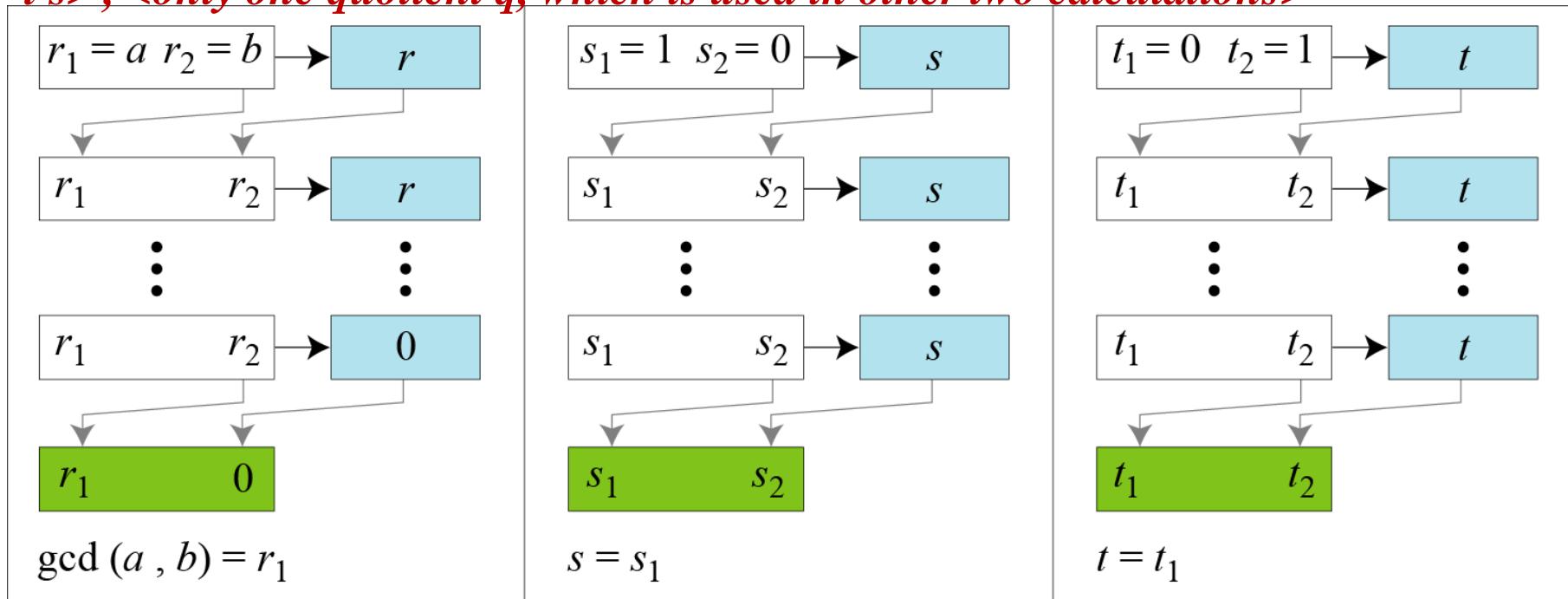
Given two integers a and b , we often need to find other two integers, s and t , such that

$$s \times a + t \times b = \gcd(a, b)$$

The extended Euclidean algorithm can calculate the $\gcd(a, b)$ and at the same time calculate the value of s and t .

Divisibility(cont.)

Extended Euclidean algorithm, part a- <use of three set of variables, r's, s's, and t's>, <only one quotient q, which is used in other two calculations>



a. Process

Divisibility(cont.)

Extended Euclidean algorithm, part b

```
r1 ← a;      r2 ← b;  
s1 ← 1;      s2 ← 0;  
t1 ← 0;      t2 ← 1;
```

(Initialization)

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2;$

```
  r ← r1 - q × r2;  
  r1 ← r2; r2 ← r;
```

(Updating r 's)

```
  s ← s1 - q × s2;  
  s1 ← s2; s2 ← s;
```

(Updating s 's)

```
  t ← t1 - q × t2;  
  t1 ← t2; t2 ← t;
```

(Updating t 's)

}

gcd (a , b) ← r₁; s ← s₁; t ← t₁

b. Algorithm

Divisibility(cont.)

Given $a = 161$ and $b = 28$, find gcd (a, b) and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

We get gcd (161, 28) = 7, $s = -1$ and $t = 6$.

Divisibility(cont.)

Given $a = 17$ and $b = 0$, find gcd (a, b) and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	17	0		1	0		0	1	

We get gcd (17, 0) = 17, $s = 1$, and $t = 0$

Divisibility(cont.)

Given $a = 0$ and $b = 45$, find $\gcd(a, b)$ and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
0	0	45	0	1	0	1	0	1	0
	45	0		0	1		1	0	

We get $\gcd(0, 45) = 45$, $s = 0$, and $t = 1$.

Divisibility(cont.)

Exercise:

Given $a = 84$ and $b = 320$, find $\gcd(a, b)$ and the values of s and t .

Divisibility(cont.)

Exercise:

Given $a = 84$ and $b = 320$, find gcd (a, b) and the values of s and t .

Solution:

$$\text{gcd}(84, 320) = 4, s = -19, t = 5$$

Linear Diophantine Equations

- A linear Diophantine equation of two variables is,
$$ax + by = c.$$
- We want to find integer values for x and y that satisfy the equation.
- Either no solution or an infinite number of solutions
- Let $d = \gcd(a,b)$; if $d \nmid c$, the equation has no solution.
- If $d \mid c$, the equation has infinite number of solutions : one of them is particular and the rest are general

Linear Diophantine Equations(cont.)

Particular Solution: $ax + by = c$

If $d \mid c$, a particular solution to the above equation can be found using
Following steps:

1. Reduce the equation to $a_1x + b_1y = c_1$ by dividing both sides of the equation by d . This is possible because d divides a , b , and c by the assumption.
2. Solve for s and t in the relation $a_1s + b_1t = 1$ using the extended Euclidean Algorithm.
3. The particular solution can be found:

Particular solution:

$$x_0 = (c/d)s \text{ and } y_0 = (c/d)t$$

Linear Diophantine Equations(cont.)

General Solution:

After finding the particular solution, the general solutions can be found:

General solutions:

$$x = x_0 + k(b/d) \text{ and } y = y_0 - k(a/d)$$

where k is an integer

Linear Diophantine Equations(cont.)

Example:

Find the particular and general solutions for the equation

$$21x + 14y = 35.$$

Particular solution:

$$x_0 = (c/d)s \text{ and } y_0 = (c/d)t$$

General solutions:

$$x = x_0 + k(b/d) \text{ and } y = y_0 - k(a/d)$$

where k is an integer

Linear Diophantine Equations(cont.)

Example:

Find the particular and general solutions for the equation

$$21x + 14y = 35.$$

Solution:

$d = \gcd(21, 14) = 7$, since $7 \mid 35$

We have $s=1$ and $t=-1$

Particular solution: $(x_0, y_0) = (5, -5)$

General solutions : $(5, -5), (7, -8), (9, -11) \dots$

Particular solution:

$$x_0 = (c/d)s \text{ and } y_0 = (c/d)t$$

General solutions:

$$x = x_0 + k(b/d) \text{ and } y = y_0 - k(a/d) \\ \text{where } k \text{ is an integer}$$

Linear Diophantine Equations(cont.)

Example:

Imagine we want to cash a Rs.100 cheque and get some Rs.20 notes and some Rs.5 notes.

Find out the possible choices if any exist for the given problem

Particular solution:

$$x_0 = (c/d)s \text{ and } y_0 = (c/d)t$$

General solutions:

$$x = x_0 + k(b/d) \text{ and } y = y_0 - k(a/d)$$

where k is an integer

Linear Diophantine Equations(cont.)

Solution:

$$20x + 5y = 100$$

Since $d = \gcd(20, 5)$ and 5 divides 100

Divide both sides by 5 to get $4x + y = 20$

Then solve the equation

$$4s + t = 1$$

Where,

$s=0$, and $t=1$ using the extended Euclidean algorithm

The particular solutions are $x_0 = 0 \times 20 = 0$ and $y_0 = 1 \times 20 = 20$

The general solutions with x and y nonnegative are $(0, 20), (1, 16), (2, 12), (3, 8), (4, 4), (5, 0)$

Particular solution:

$$x_0 = (c/d)s \text{ and } y_0 = (c/d)t$$

General solutions:

$$x = x_0 + k(b/d) \text{ and } y = y_0 - k(a/d)$$

where k is an integer

Modular Arithmetic

Preliminary

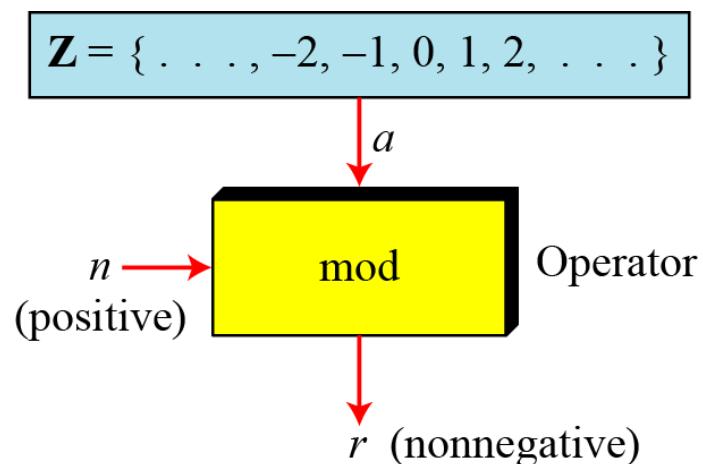
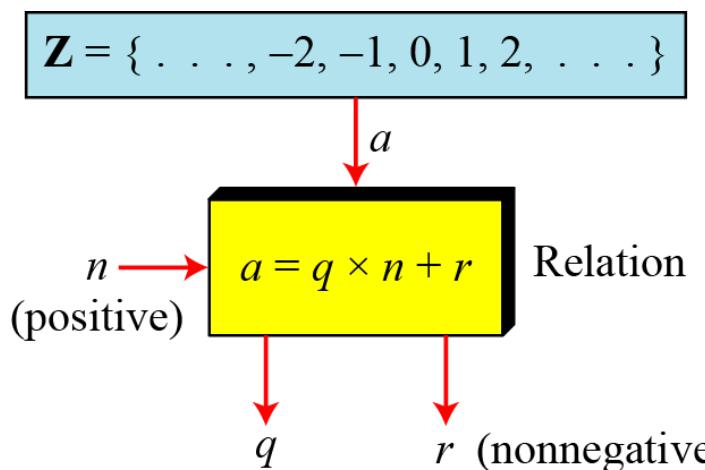
- The division relationship ($a = q \times n + r$) discussed in the previous section has two inputs (a and n) and two outputs (q and r).
- In modular arithmetic, we are interested in only one of the outputs, the remainder r .

Preliminary(cont.)

- We use modular arithmetic in our daily life; for example, we use a clock to measure time. Our clock system uses modulo 12 arithmetic.

Modulo Operator

- The modulo operator is shown as **mod**. The second **input (n)** is called the **modulus**. The **output r** is called the **residue**.



Division algorithm and modulo operator

Modulo Operator(cont.)

- Find the result of the following operations:
 - a. $27 \bmod 5$
 - b. $36 \bmod 12$
 - c. $-18 \bmod 14$
 - d. $-7 \bmod 10$

Modulo Operator(cont.)

- **Solution**
 - a. Dividing 27 by 5 results in $r = 2$
 - b. Dividing 36 by 12 results in $r = 0$
 - c. Dividing -18 by 14 results in $r = -4$. After adding the modulus $r = 10$
 - d. Dividing -7 by 10 results in $r = -7$. After adding the modulus to -7 , $r = 3$

Set of Residues : Z_n

- The results of the modulo operation with modulus n is always an integer between 0 and $n-1$. <i.e. result of $a \bmod n$ is always a nonnegative integer>
- The modulo operation creates a set, which in modular arithmetic is referred to as **the set of least residues modulo n , or Z_n** .

$$Z_n = \{ 0, 1, 2, 3, \dots, (n-1) \}$$

$$Z_2 = \{ 0, 1 \}$$

$$Z_6 = \{ 0, 1, 2, 3, 4, 5 \}$$

$$Z_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$$

Some Z_n sets

*Udai Pratap Rao: CRYPTOGRAPHY AND
NETWORK SECURITY @ B.Tech IV*

Congruence

- In *cryptography*, we often used the concept of **congruence** instead of equality.
- To show that two integers are congruent, we use the congruence operator (\equiv).
- We say that a is congruent to b modulo m , and we write $a \equiv b \pmod{m}$, if m divides $b-a$.
- Example:

$$2 \equiv 12 \pmod{10}$$

$$3 \equiv 8 \pmod{5}$$

$$13 \equiv 23 \pmod{10}$$

$$8 \equiv 13 \pmod{5}$$

Congruence(cont.)

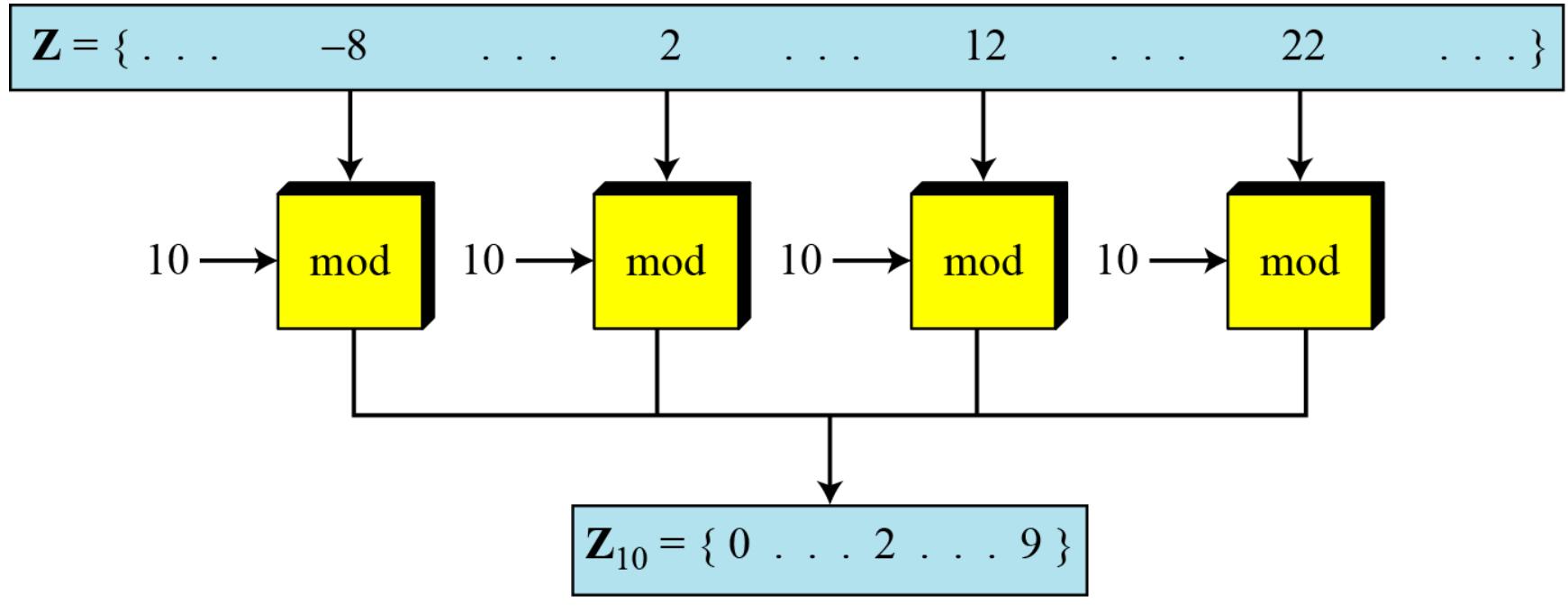
- Congruence operator Vs. equality operator
 - A equality operator maps a member of \mathbf{Z} to itself,
 - The congruence operator maps a member from \mathbf{Z} to member of \mathbf{Z}_n .
- The equality operator is one-to-one,
- The congruence operator is many-to-one
- The phrase $(\text{mod } n)$ is an indication of destination set \mathbf{Z}_n . <Example- $2 \equiv 12 \pmod{10}$, means that the destination set is \mathbf{Z}_{10} .>

Congruence(cont.)

Properties-

- $a \equiv b \pmod{m}$, implies that $b \equiv a \pmod{m}$ (*symmetry*)
- $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, implies that $a \equiv c \pmod{m}$ (*transitivity*)

Congruence(cont.)

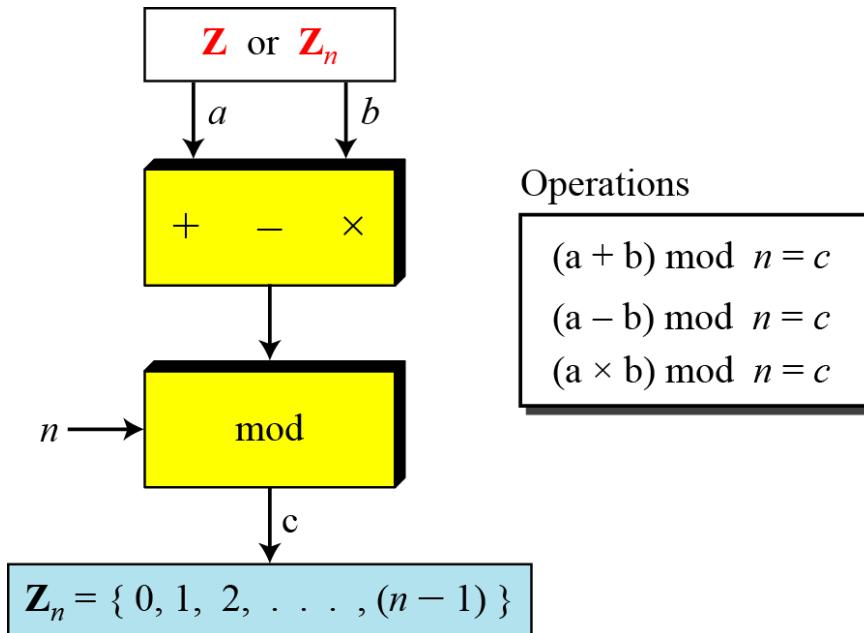


$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

Congruence Relationship

Operation in Z_n

- The three binary operations that we discussed for the set Z can also be defined for the set Z_n . The result may need to be mapped to Z_n using the mod operator.



Operation in Z_n (cont.)

- Perform the following operations (the inputs come from Z_n):
 - a. Add 7 to 14 in Z_{15} .
 - b. Subtract 11 from 7 in Z_{13} .
 - c. Multiply 11 by 7 in Z_{20} .

Operation in Z_n (cont.)

- **Solution**

$$(14 + 7) \bmod 15 \rightarrow (21) \bmod 15 = 6$$

$$(7 - 11) \bmod 13 \rightarrow (-4) \bmod 13 = 9$$

$$(7 \times 11) \bmod 20 \rightarrow (77) \bmod 20 = 17$$

Operation in Z_n (cont.)

- Perform the following operations (the inputs come from either Z or Z_n):
 - a. Add 17 to 27 in Z_{14} .
 - b. Subtract 43 from 12 in Z_{13} .
 - c. Multiply 123 by -10 in Z_{19} .

Operation in Z_n (cont.)

- Solution
- Add 17 to 27 in Z_{14} .
 - $(17+27)\text{mod } 14 = 2$
- Subtract 43 from 12 in Z_{13} .
 - $(12-43)\text{mod } 13 = 5$
- Multiply 123 by -10 in Z_{19} .
 - $(123 \times (-10)) \text{ mod } 19 = 5$

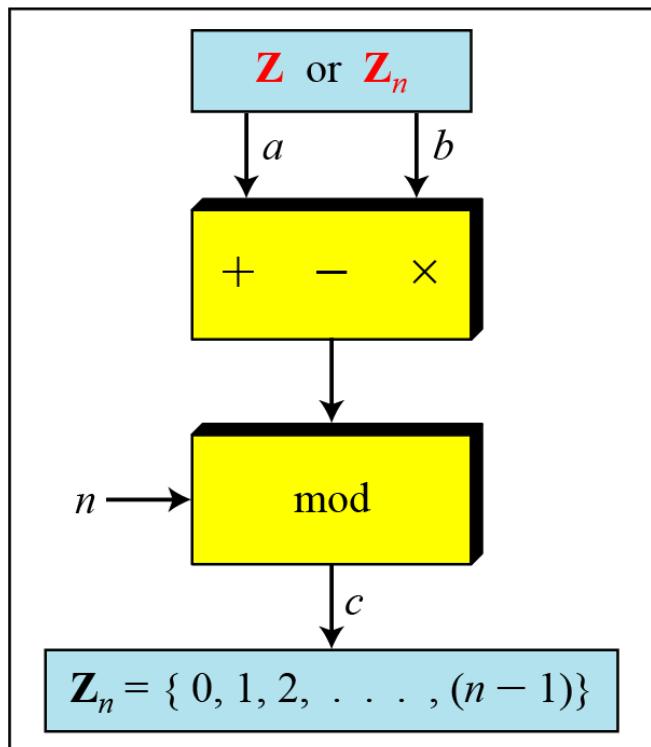
Operation in Z_n (cont.)

First Property: $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$

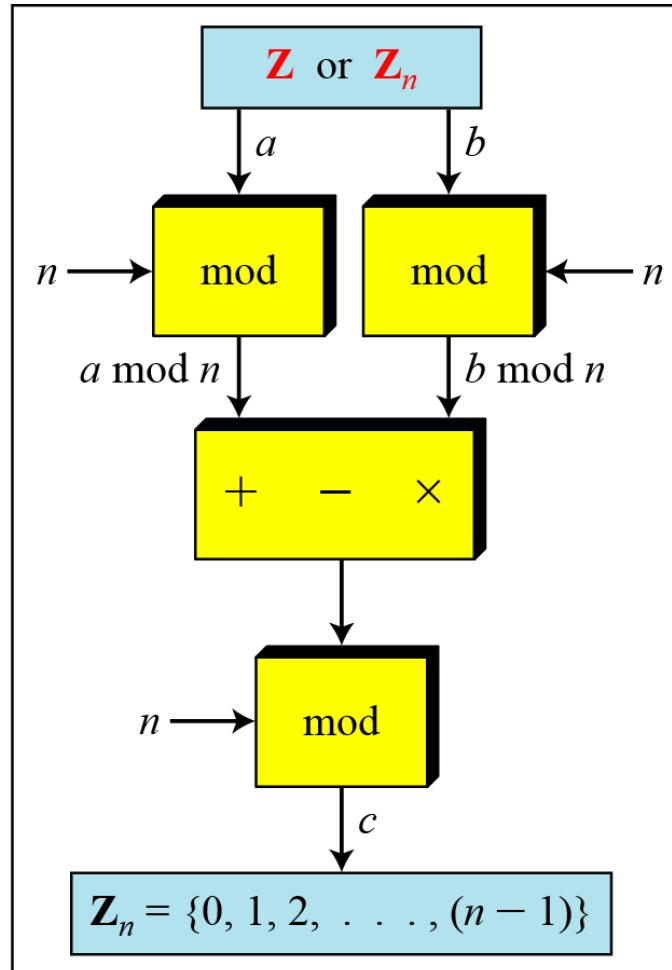
Second Property: $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$

Third Property: $(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

Operation in \mathbb{Z}_n (cont.)



a. Original process



b. Applying properties

Operation in Z_n (cont.)

- The following shows the application of the above properties:

1. $(1,723,345 + 2,124,945) \text{ mod } 11 = (8 + 9) \text{ mod } 11 = 6$
2. $(1,723,345 - 2,124,945) \text{ mod } 16 = (8 - 9) \text{ mod } 11 = 10$
3. $(1,723,345 \times 2,124,945) \text{ mod } 16 = (8 \times 9) \text{ mod } 11 = 6$

Inverses

- Working in modular arithmetic, we often need to find the inverse of a number relative to an **operation**.
- **additive inverse** (relative to an addition operation) or
- a **multiplicative inverse** (relative to a multiplication operation).

Inverses- relevancy with cryptography

- In cryptography we often work with inverses.
- If the sender uses an integer (as the encryption key), the receiver uses the inverse of that integer (as the decryption key).
- If the operation (encryption/decryption algorithm) is addition, \mathbf{Z}_n can be used as the set of possible keys because each integer in this set has an additive inverse.
- On the other hand, if the operation (encryption/decryption algorithm) is multiplication , \mathbf{Z}_n cannot be the set of possible keys because only some numbers of this set have a multiplicative inverse.

Additive Inverses

- In \mathbb{Z}_n , two numbers a and b are additive inverses of each other if

$$a + b \equiv 0 \pmod{n}$$

In modular arithmetic, each integer has an additive inverse. The sum of an integer and its additive inverse is congruent to 0 modulo n.

Additive Inverses

- Find additive inverse of 4 in Z_{10} .
- Solution
 - In Z_n , the additive inverse of a can be calculated as $b=n-a$.
 - The additive inverse 4 in Z_{10} is $10-4=6$.

Additive Inverses

- Find all additive inverse pairs in \mathbb{Z}_{10} .
- Solution
 - The six pairs of additive inverses are $(0, 0)$, $(1, 9)$, $(2, 8)$, $(3, 7)$, $(4, 6)$, and $(5, 5)$.

Multiplicative Inverses

- In Z_n , two numbers a and b are the multiplicative inverse of each other if,

$$a \times b \equiv 1 \pmod{n}$$

In modular arithmetic, an integer may or may not have a multiplicative inverse.

When it does, the product of the integer and its multiplicative inverse is congruent to 1 modulo n.

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 8 in \mathbb{Z}_{10} .
 - There is no multiplicative inverse because $\gcd(10, 8) = 2 \neq 1$.
 - In other words, we cannot find any number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.
- Find all multiplicative inverses in \mathbb{Z}_{10} .
 - There are only three pairs: (1, 1), (3, 7) and (9, 9). The numbers 0, 2, 4, 5, 6, and 8 do not have a multiplicative inverse.

Multiplicative Inverses(cont.)

- Find all multiplicative inverse pairs in \mathbb{Z}_{11} .

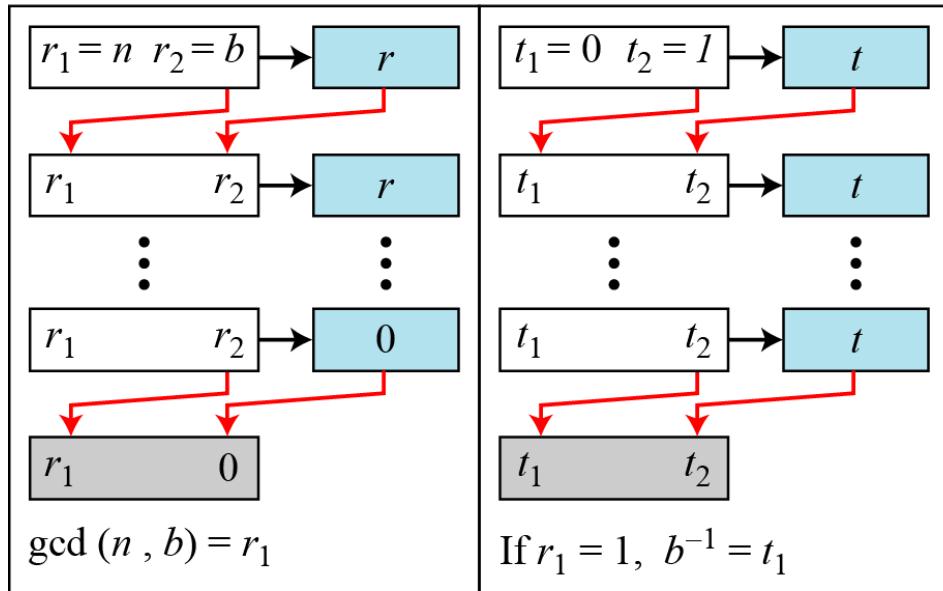
Multiplicative Inverses(cont.)

- Find all multiplicative inverse pairs in Z_{11} .
 - Solution
 - We have seven pairs: (1, 1), (2, 6), (3, 4), (5, 9), (7, 8), (9, 9), and (10, 10).
 - The reason is that in Z_{11} , $\gcd(11, a)$ is 1 (relatively prime) for all value of a except 0. it means all integers 1 to 10 have multiplicative inverse.

Multiplicative Inverses(cont.)

- The **extended Euclidean algorithm** finds the multiplicative inverses of b in Z_n when n and b are given and $\gcd(n, b) = 1$.
- The multiplicative inverse of b is the value of t_1 after being mapped to Z_n .

Multiplicative Inverses



a. Process

```

 $r_1 \leftarrow n; \quad r_2 \leftarrow b;$ 
 $t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$ 

while ( $r_2 > 0$ )
{
   $q \leftarrow r_1 / r_2;$ 
   $r \leftarrow r_1 - q \times r_2;$ 
   $r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$ 
   $t \leftarrow t_1 - q \times t_2;$ 
   $t_1 \leftarrow t_2; \quad t_2 \leftarrow t;$ 
}
if ( $r_1 = 1$ ) then  $b^{-1} \leftarrow t_1$ 

```

b. Algorithm

Using extended Euclidean algorithm to find multiplicative inverse

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 11 in \mathbb{Z}_{26} .

Solution

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
	1	0		-7	26	

The gcd (26, 11) is 1; the inverse of 11 is -7 or 19.

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 23 in \mathbb{Z}_{100} .

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 23 in \mathbb{Z}_{100} .

Solution

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	-4
2	23	8	7	1	-4	19
1	8	7	1	-4	9	-13
7	7	1	0	9	-13	100
	1	0		-13	100	

The gcd (100, 23) is 1; the inverse of 23 is -13 or 87.

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 12 in \mathbb{Z}_{26} .

Solution

q	r_1	r_2	r	t_1	t_2	t
2	26	12	2	0	1	-2
6	12	2	0	1	-2	13
	2	0		-2	13	

The gcd (26, 12) is 2; the inverse does not exist.

Addition and Multiplication Tables

- Addition and multiplication table for Z_{10}
- Additive Inverse: Inverse pair can be found when the result of addition is zero.
- Multiplicative Inverse: Inverse pair can be found when the result of multiplication is 1.

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Addition Table in Z_{10}

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	0	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

Multiplication Table in Z_{10}

Different Sets of Addition and Multiplication

- Some Z_n and Z_n^* sets

$$Z_6 = \{0, 1, 2, 3, 4, 5\}$$

$$Z_6^* = \{1, 5\}$$

$$Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$$

$$Z_7^* = \{1, 2, 3, 4, 5, 6\}$$

$$Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$Z_{10}^* = \{1, 3, 7, 9\}$$

We need to use Z_n when additive inverses are needed; we need to use Z_n^* when multiplicative inverses are needed.

Linear Congruence

Cryptography often involves solving an equation or a set of equations of one or more variables with coefficient in Z_n . This section shows how to solve equations when the power of each variable is 1 (linear equation).

Topics discussed in this section:

- 2.4.1 Single-Variable Linear Equations
- 2.4.2 Set of Linear Equations

Single-Variable Linear Equations

Equations of the form $ax \equiv b \pmod{n}$ might have no solution or a limited number of solutions.

Assume that the $\gcd(a, n) = d$.

If $d \nmid b$, there is no solution.

If $d|b$, there are d solutions.

Continued

If $d \mid b$, we use the following strategy to find the solutions.

1. Reduce the equation by dividing both sides of the equation (including the modulus) by d.
2. Multiply both sides of the reduced equation by the multiplicative inverse of a to find the particular solution x_0 .
3. The general solutions are $x_I = x_0 + k(n/d)$ for $k=0, 1, \dots, (d-1)$.

Continued

Example

Solve the equation $10x \equiv 2 \pmod{15}$.

Solution

First we find the gcd (10 and 15) = 5. Since 5 does not divide 2, we have no solution.

Example

Solve the equation $14x \equiv 12 \pmod{18}$.

Solution

$$14x \equiv 12 \pmod{18} \rightarrow 7x \equiv 6 \pmod{9} \rightarrow x \equiv 6(7^{-1}) \pmod{9}$$
$$x_0 = (6 \times 7^{-1}) \pmod{9} = (6 \times 4) \pmod{9} = 6$$
$$x_1 = x_0 + 1 \times (18/2) = 15$$

Continued

Example

Solve the equation $3x + 4 \equiv 6 \pmod{13}$.

Solution

First we change the equation to the form $ax \equiv b \pmod{n}$. We add -4 (the additive inverse of 4) to both sides, which give $3x \equiv 2 \pmod{13}$. Because $\gcd(3, 13) = 1$, the equation has only one solution, which is $x_0 = (2 \times 3^{-1}) \pmod{13} = 18 \pmod{13} = 5$. We can see that the answer satisfies the original equation: $3 \times 5 + 4 \equiv 6 \pmod{13}$.

Set of Linear Equations

We can also solve a set of linear equations with the same modulus if the matrix formed from the coefficients of the variables is invertible.

Figure 2.27 Set of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\equiv b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\equiv b_2 \\ \vdots &\quad \vdots \quad \vdots \quad \vdots \\ \vdots &\quad \vdots \quad \vdots \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &\equiv b_n \end{aligned}$$

a. Equations

$$\left[\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right] \equiv \left[\begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_n \end{array} \right] \quad \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right] \equiv \left[\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right]^{-1} \left[\begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_n \end{array} \right]$$

Continued

Example

Solve the set of following three equations:

$$3x + 5y + 7z \equiv 3 \pmod{16}$$

$$x + 4y + 13z \equiv 5 \pmod{16}$$

$$2x + 7y + 3z \equiv 4 \pmod{16}$$

Solution

The result is $x \equiv 15 \pmod{16}$, $y \equiv 4 \pmod{16}$, and $z \equiv 14 \pmod{16}$. We can check the answer by inserting these values into the equations.

MATHEMATICS OF CRYPTOGRAPHY

PART II

ALGEBRAIC STRUCTURES

Objectives

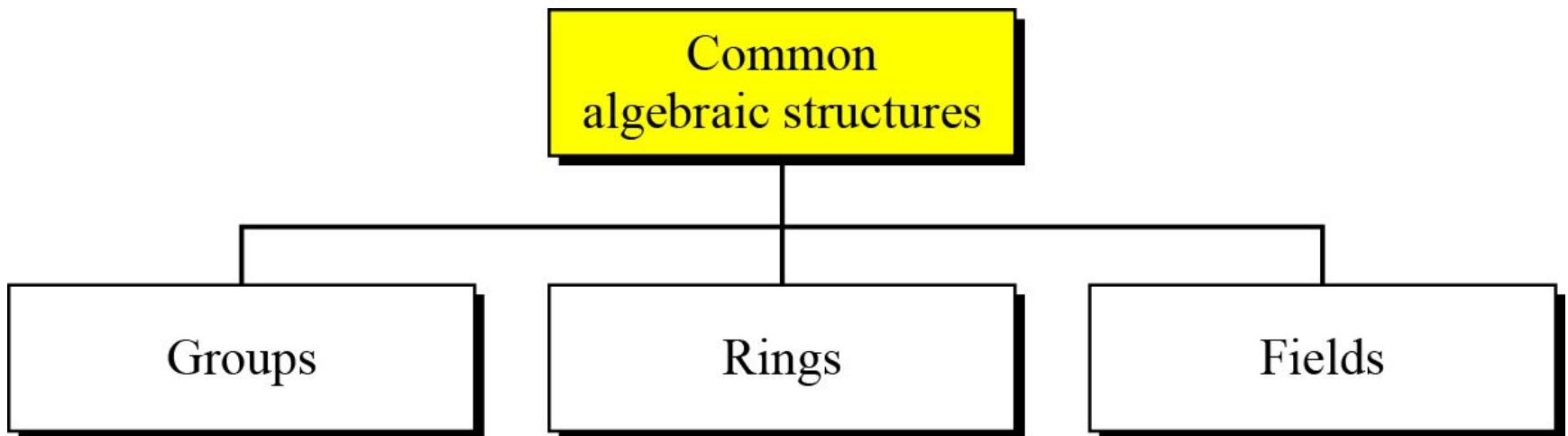
- To review the concept of algebraic structures
- To define and give some examples of groups
- To define and give some examples of rings
- To define and give some examples of fields
- To emphasize the finite fields of type GF (2^n) that make it possible to perform operations such as addition, subtraction, multiplication, and division on n-bit words in modern block ciphers

<modern symmetric-key ciphers are based on algebraic structure>

ALGEBRAIC STRUCTURES

- Cryptography requires sets of integers and specific operations that are defined for those sets.
- The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure.
- Three common algebraic structures: groups, rings, and fields.

ALGEBRAIC STRUCTURES(cont.)



Common algebraic structure

Groups

- A group (**G**) is a set of elements with a binary operation (**•**) that satisfies four properties.
 - Closure
 - Associativity
 - Existence of identity
 - Existence of inverse

Groups(cont.)

- Closure
 - If a and b are elements of G , then $c = a \bullet b$ is also an element of G .
- Associativity
 - If a , b and c are elements of G , then $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
- Existence of identity
 - For all a in G , there exist an element e , called the identity element, such that $e \bullet a = a \bullet e = a$
- Existence of inverse
 - For each a in G , there exists an element a' , called the inverse of a , such that $a \bullet a' = a' \bullet a = e$

Groups(cont.)

- A Commutative group (**Abelian group**), is a group in which the operator satisfies four properties plus an extra property that is **commutativity**.
 - For all a and b in G , we have $a \bullet b = b \bullet a$

Groups(cont.)

- Although a group involves a single operation, the properties imposed on the operation allow the use of a pair of operations as log as they are inverses of each other.
- If the defined operation is addition, the group supports both addition and subtraction, because subtraction is addition using the additive inverse.
- This is also true for multiplication and division.
- However, a group can support only addition/subtraction or multiplication/division operations, but not the both at the same time.

Groups(cont.)

- Example

The set of residue integers with the addition operator,

$$G = \langle \mathbb{Z}_n, + \rangle,$$

is a commutative group.

Check the properties....

“We can perform addition and subtraction on the elements of this set without moving out of the set.”

Assume- $a=2$, $b=3$, $c=4$, and identity element is 0

Example (cont..)

1. Closure is satisfied. The result of adding two integers in Z_n is another integer in Z_n .
2. Associativity is satisfied. The result of $4+(3+2)$ is same as $(4+3)+2$.
3. Commutativity is satisfied. We have $3+5=5+3$.
4. The identity element is 0. we have $3+0=0+3=3$.
5. Every element has an additive inverse. The inverse of a element is its complement. For example, the inverse of 3 is -3 ($n-3$ in Z_n) and the inverse of -3 is 3. the inverse allows us to perform subtraction on the set.

Groups(cont.)

- Example:
 - The set Z_n^* with the multiplication operator, $G = \langle Z_n^*, \times \rangle$, is also an abelian group.
 - We can perform multiplication and division on the elements of this set without moving out of the set.
- Example:
 - Let us define a set $G = \langle \{a, b, c, d\}, \bullet \rangle$ and the operation as shown in Table.

\bullet	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

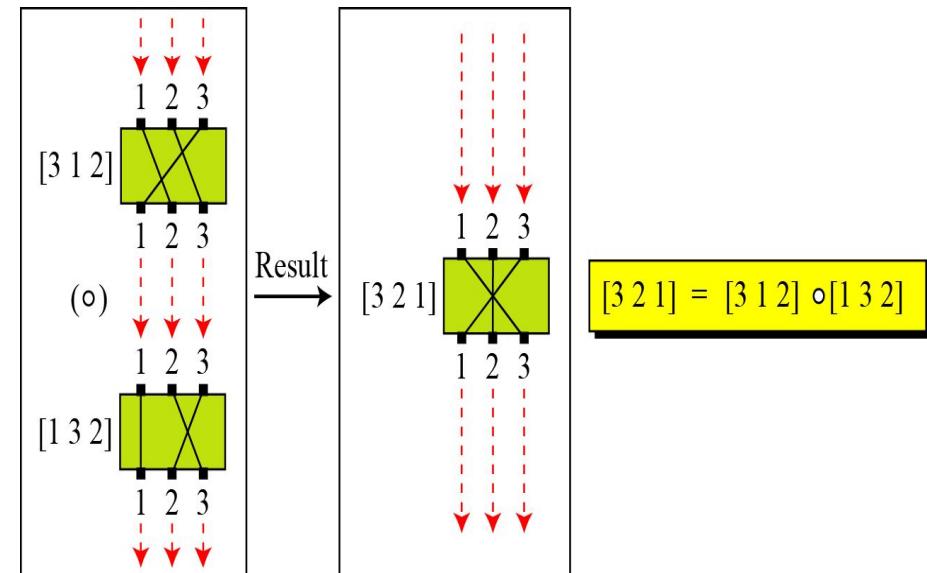
•	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

- The group is abelian????
- Closure ?
- Associativity?
- Commutativity?
- The group has an identity element, which is a.
- Each element has an inverse. The inverse pairs can be found by finding the identity in each row (shaded). The pairs are (a,a), (b,d), (c,c).

Groups(cont.)

- Example:
 - A very interesting group is the permutation group.
 - The set is the set of all permutations, and the operation is composition: applying one permutation after another.
 - Check for properties....
 - Is the group abelian???

Composition of two permutations



Groups(cont.)

- Example(cont.):

\circ	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 2 3]	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 3 2]	[1 3 2]	[1 2 3]	[2 3 1]	[2 1 3]	[3 2 1]	[3 1 2]
[2 1 3]	[2 1 3]	[3 1 2]	[1 2 3]	[3 2 1]	[1 3 2]	[2 3 1]
[2 3 1]	[2 3 1]	[3 2 1]	[1 3 2]	[3 1 2]	[1 2 3]	[2 1 3]
[3 1 2]	[3 1 2]	[2 1 3]	[3 2 1]	[1 2 3]	[2 3 1]	[1 3 2]
[3 2 1]	[3 2 1]	[2 3 1]	[3 1 2]	[1 3 2]	[2 1 3]	[1 2 3]

Operation table for permutation group

With three inputs and three outputs, there can be $3!$ or 6 different permutations.

Only four properties are satisfied.

1. Closure
2. Associativity.
3. Commutative?
4. The set has an identity element, which is [1 2 3].
5. Each element has inverse.

Groups(cont.)

- In the previous example, we showed that a set of permutations with the composition operation is a group.
- This implies that using two permutations one after another cannot **strengthen the security of a cipher**.
- Because we can always find a permutation that can do the same job because of the closure property.

Groups(cont.)

- Finite Group
 - If the set has a finite number of elements; otherwise, it is an infinite group.
- Order of a Group $|G|$
 - The number of elements in the group.
 - If the group is finite, its order is finite
- Subgroups
 - A subset H of a group G is a subgroup of G if H itself is a group with respect to the operation on G

Groups(cont.)

- Subgroups(cont.)
 - If $G = \langle S, \bullet \rangle$ is a group, $H = \langle T, \bullet \rangle$ is a group under the same operation, and T is a nonempty subset of S , then H is a subgroup of G
 - The above definition implies that:
 - If a and b are members of both groups, then $c = a \bullet b$ is also member of both groups
 - The group share the same identity element
 - If a is a member of both groups, the inverse of a is also a member of both groups
 - Each group is a subgroup of itself

Groups(cont.)

- Exercise:
 - Is the group $H = \langle \mathbb{Z}_{10}, + \rangle$ a subgroup of the group $G = \langle \mathbb{Z}_{12}, + \rangle$?

Groups(cont.)

- Exercise:
 - Is the group $H = \langle \mathbb{Z}_{10}, + \rangle$ a subgroup of the group $G = \langle \mathbb{Z}_{12}, + \rangle$?
- Solution:
 - The answer is no. Although H is a subset of G , the operations defined for these two groups are different. The operation in H is addition modulo 10; the operation in G is addition modulo 12.

Groups(cont.)

- Cyclic subgroups
 - If a subgroup of a group can be generated using the power of an element, the subgroup is called the **cyclic subgroup**.
 - The term *power* here means repeatedly applying the **group operation** to the element.

$$a^n \rightarrow a \bullet a \bullet \dots \bullet a \quad (n \text{ times})$$

Groups(cont.)

- Four cyclic subgroups can be made from the group $G = \langle \mathbb{Z}_6, + \rangle$.
- They are $H_1 = \langle \{0\}, + \rangle$, $H_2 = \langle \{0, 2, 4\}, + \rangle$, $H_3 = \langle \{0, 3\}, + \rangle$, and $H_4 = G$.

$$0^0 \bmod 6 = 0$$

$$1^0 \bmod 6 = 0$$

$$1^1 \bmod 6 = 1$$

$$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$$

$$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$$

$$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$$

$$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$$

$$2^0 \bmod 6 = 0$$

$$2^1 \bmod 6 = 2$$

$$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4$$

$$3^0 \bmod 6 = 0$$

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

$$4^0 \bmod 6 = 0$$

$$4^1 \bmod 6 = 4$$

$$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2$$

$$5^0 \bmod 6 = 0$$

$$5^1 \bmod 6 = 5$$

$$5^2 \bmod 6 = 4$$

$$5^3 \bmod 6 = 3$$

$$5^4 \bmod 6 = 2$$

$$5^5 \bmod 6 = 1$$

Groups(cont.)

- Exercise:
 - Find out the cyclic subgroups for group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$.

Groups(cont.)

- Three cyclic subgroups can be made from the group $G = \langle Z_{10}^*, \times \rangle$. G has only four elements: 1, 3, 7, and 9. The cyclic subgroups are $H_1 = \langle \{1\}, \times \rangle$, $H_2 = \langle \{1, 9\}, \times \rangle$, and $H_3 = G$.

$$1^0 \bmod 10 = 1$$

$$\begin{aligned}3^0 \bmod 10 &= 1 \\3^1 \bmod 10 &= 3 \\3^2 \bmod 10 &= 9 \\3^3 \bmod 10 &= 7\end{aligned}$$

$$\begin{aligned}7^0 \bmod 10 &= 1 \\7^1 \bmod 10 &= 7 \\7^2 \bmod 10 &= 9 \\7^3 \bmod 10 &= 3\end{aligned}$$

$$\begin{aligned}9^0 \bmod 10 &= 1 \\9^1 \bmod 10 &= 9\end{aligned}$$

Groups(cont.)

- Cyclic group
 - The group **G** has a cyclic subgroup $H_4 = G$. This means that the group **G** is a cyclic group.
 - In this case, the elements that generates the cyclic subgroup can also generate the group itself.
 - This element is referred to as a *generator*.

Groups(cont.)

- Cyclic group(cont.)
- Example:
 - Three cyclic subgroups can be made from the group $G = \langle Z_{10}^*, \times \rangle$.
 - The cyclic subgroups are $H_1 = \langle \{1\}, \times \rangle$, $H_2 = \langle \{1, 9\}, \times \rangle$, and $H_3 = G$.
 - The group $G = \langle Z_{10}^*, \times \rangle$ is a cyclic group with two **generators**, $g = 3$ and $g = 7$.
 - The group $G = \langle Z_6, + \rangle$ is a cyclic group with two **generators**, $g = 1$ and $g = 5$.

Groups(cont.)

- Lagrange's Theorem
 - Assume that G is a group, and H is a subgroup of G . If the order of G and H are $|G|$ and $|H|$, respectively, then, based on this theorem, $|H|$ divides $|G|$.
 - Application of Lagrange's Theorem: Given a group G of order $|G|$, the orders of the potential subgroups can be easily determined if the divisors of $|G|$ can be found.
- Order of an Element
 - The order of an element is the order of the cyclic group it generates.

Groups(cont.)

- Example:
 - In the group $G = \langle \mathbb{Z}_6, + \rangle$, the orders of the elements are:
 $\text{ord}(0) = 1, \text{ord}(1) = 6, \text{ord}(2) = 3, \text{ord}(3) = 2, \text{ord}(4) = 3,$
 $\text{ord}(5) = 6.$
 - In the group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$, the orders of the elements are:
 $\text{ord}(1) = 1, \text{ord}(3) = 4, \text{ord}(7) = 4, \text{ord}(9) = 2.$

Examples

Let us consider the following statements

- (i) $(\mathbb{Z}_{10}, +)$ is a cyclic group.
- (ii) $(\mathbb{Z}, +)$ is not a cyclic group.

Select the correct option from below.

- A. (i) and (ii) both are true.
- B. Only (ii) is true.
- C. Only (i) is true.
- D. (i) and (ii) both are false.

Examples

Order of 3 in the group $(\mathbb{Z}_5, +)$ is ____.

- A. 2
- B. 5
- C. 1
- D. 3

Examples

Let $G = \{a \in \mathbb{Z}_{10} \mid \gcd(a, 10) = 1\}$. Let us consider the following statements

- (i) G is a group under multiplication modulo 10.
- (ii) The number of elements in set G is 5.

Select the correct option from below.

- A. (i) and (ii) both are true.
- B. Only (ii) is true.
- C. Only (i) is true.
- D. (i) and (ii) both are false.

Ring

- A ring, $R = \langle \dots, \bullet, \square \rangle$, is an algebraic structure with two operations.
- First operation must satisfy all five properties
- Second operation must satisfy only the first two
- In addition, second operation must be distributed over first
 - i.e. for all a, b , and c elements of R , we have,
 - $(b \bullet c) = (a \bullet b) \bullet (a \bullet c)$ and
 - $(a \bullet b) \square c = (a \square c) \bullet (a \square c)$

Ring(cont.)

- Commutative Ring- is a ring in which the commutative property is also satisfied for the second operation.

Distribution of \square over \bullet

1. Closure	\bullet
2. Associativity	
3. Commutativity	
4. Existence of identity	
5. Existence of inverse	

1. Closure	\square
2. Associativity	
3. Commutativity	

Note:
The third property is
only satisfied for a
commutative ring.



Ring(cont.)

- The set Z with two operations, addition and multiplication, is a commutative ring.
- We show it by $R = \langle Z, +, \times \rangle$.
- Addition satisfies all of the five properties; multiplication satisfies only three properties.

Field

- A field, denoted by $F = \langle \{ \dots \}, \bullet, \square \rangle$ is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse.

Distribution of \square over \bullet

- 1. Closure \bullet
- 2. Associativity
- 3. Commutativity
- 4. Existence of identity
- 5. Existence of inverse

- 1. Closure \square
- 2. Associativity
- 3. Commutativity
- 4. Existence of identity
- 5. Existence of inverse

Note:
The identity element of the first operation has no inverse with respect to the second operation.

$\{a, b, c, \dots\}$
Set

\bullet \square
Operations

Field

Field(cont.)

- Finite Fields
 - Only finite fields are extensively used in cryptography.
 - It is a field with finite number of elements, are very important structures in cryptography.
 - Galois showed that for a field to be finite, the number of elements should be p^n , where p is a prime and n is a positive integer.

A Galois field, $GF(p^n)$, is a finite field with p^n elements.

Field(cont.)

- GF(p) Fields
 - When $n = 1$, we have GF(p) field.
 - This field can be the set \mathbb{Z}_p , $\{0, 1, \dots, p - 1\}$, with two arithmetic operations (addition and multiplication).
 - In this set each element has an additive inverse and that nonzero elements have a multiplicative inverse (no multiplicative inverse for 0).

Field(cont.)

- A very common field in this category is GF(2) with the set {0, 1} and two operations, addition and multiplication.

GF(2)

{0, 1}	+ ×
--------	--------

+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	0
1	0	1

Multiplication

a	0	1	a	0	1
-a	1	0	a ⁻¹	—	1

Inverses

GF(2) field

Field(cont.)

- We can define GF(5) on the set Z_5 (5 is a prime) with addition and multiplication operators.

GF(5)

$\{0, 1, 2, 3, 4\}$ + ×

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Addition

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Multiplication

Additive inverse

a	0	1	2	3	4
-a	0	4	3	2	1
a	0	1	2	3	4
a^{-1}	-1	3	2	4	

Multiplicative inverse

GF(5) field

- Summary:

<i>Algebraic Structure</i>	<i>Supported Typical Operations</i>	<i>Supported Typical Sets of Integers</i>
Group	(+ −) or (× ÷)	\mathbf{Z}_n or \mathbf{Z}_n^*
Ring	(+ −) and (×)	\mathbf{Z}
Field	(+ −) and (× ÷)	\mathbf{Z}_p

GF(2^n) FIELDS

- In cryptography, we often need to use four operations(addition,subtraction,multiplication, and division).
- In other words, we need to use fields.
- However, when we work with computers, the positive integers are stored in the computers as n-bit words in which n is usually 8,16,32 and so on.
- Range of integers is 0 to $2^n - 1$
- Hence modulus is 2^n
- What if we want to use field???? <there are two choices>

GF(2^n) FIELDS (cont.)

- Solution 1
 - Use GF(p), with the set Z_p , where p is the largest prime number less than 2^n
 - But the problem ???
 - It is inefficient because we can not use the integers from p to $2^n - 1$.
 - Example- if n=4, the largest prime less than 2^4 is 13. this means we cannot use integers 13,14, and 15.
 - If n=8 ????????
- Solution 2
 - Use GF(2^n)
 - Use a set of 2^n elements
 - The elements in this set are n-bit words
 - E.g. for n=3, the set is {000,001,010,011,100,101,110,111}

GF(2^n) FIELDS (cont.)

- Let us define a GF(2^2) field in which the set has four 2-bit words: {00, 01, 10, 11}.
- We can redefine addition and multiplication for this field in such a way that all properties of these operations are satisfied. Addition and multiplication are defined in terms of polynomials.

	Addition			
	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Identity: 00

An example of GF(2^2) field

	Multiplication			
	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Identity: 01

Polynomials

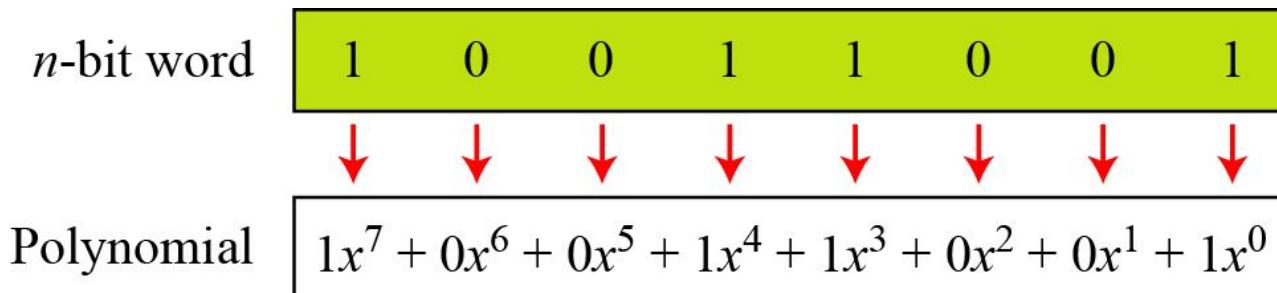
- We can directly define the rules for addition and multiplication operations on n -bit words that satisfy the properties in $\text{GF}(2^n)$.
- A polynomial of degree $n - 1$ is an expression of the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x^1 + a_0x^0$$

- where x^i is called the i th term and a_i is called coefficient of the i th term.

Polynomials (cont.)

- We can represent the 8-bit word (10011001) using a polynomial.



First simplification
$$1x^7 + 1x^4 + 1x^3 + 1x^0$$

Second simplification
$$x^7 + x^4 + x^3 + 1$$

Polynomials (cont.)

- Find the 8-bit word related to the polynomial $x^5 + x^2 + x$, we first supply the omitted terms.
- Since $n = 8$, it means the polynomial is of degree 7. The expanded polynomial is,

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$$

- This is related to the 8-bit word **00100110**.

Polynomials (cont.)

- Operations on polynomials
 - Actually involves two operations
 - Operation on coefficients and operation on polynomials
 - Hence, need to define two fields
 - What for coefficient??
 - What for polynomials???

Polynomials (cont.)

- Operations on polynomials
 - Actually involves two operations
 - Operation on coefficients and operation on polynomials
 - Hence, need to define two fields
 - What for coefficient??
 - What for polynomials???
 - Coefficients are made of 0 or 1; we can use GF(2) and GF(2^n) for polynomials....

Polynomials (cont.)

- Modulus
 - Why Modulus?
 - Addition of two polynomials never creates a polynomial out of the set.
 - However, multiplication of two polynomials may create a polynomial with degree more than $n-1$.
 - This means we need to divide the result by a modulus and keep only the remainder, as we do in modular arithmetic.
 - For the sets of polynomials in $\text{GF}(2^n)$, a group of polynomials of degree n is defined as the modulus.

Polynomials (cont.)

- **irreducible polynomials.**
 - No polynomial in the set can divide this polynomial
 - Can not be factored into a polynomial with degree of less than n

Degree	Irreducible Polynomials
1	$(x + 1), (x)$
2	$(x^2 + x + 1)$
3	$(x^3 + x^2 + 1), (x^3 + x + 1)$
4	$(x^4 + x^3 + x^2 + x + 1), (x^4 + x^3 + 1), (x^4 + x + 1)$
5	$(x^5 + x^2 + 1), (x^5 + x^3 + x^2 + x + 1), (x^5 + x^4 + x^3 + x + 1),$ $(x^5 + x^4 + x^3 + x^2 + 1), (x^5 + x^4 + x^2 + x + 1)$

Polynomials (cont.)

- Polynomial addition

Addition and subtraction operations on polynomials are the same operation.

Polynomials (cont.)

- Example
- Let us do $(x^5 + x^2 + x) \oplus (x^3 + x^2 + 1)$ in $GF(2^8)$. We use the symbol \oplus to show that we mean polynomial addition. The following shows the procedure:
- <keep the uncommon term and delete the common term>

$$\begin{array}{r} 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0 \quad \oplus \\ 0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0 \\ \hline 0x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0 \end{array} \rightarrow x^5 + x^3 + x + 1$$

Polynomials (cont.)

- Short cut method
 - Addition in GF(2) means the exclusive-or (XOR) operation.
 - So we can exclusive-or the two words, bits by bits, to get the result.
 - In the previous example, $x^5 + x^2 + x$ is 00100110 and $x^3 + x^2 + 1$ is 00001101.
 - The result is 00101011 or in polynomial notation $x^5 + x^3 + x + 1$.

Polynomials (cont.)

- Multiplication
 - The coefficient multiplication is done in GF(2).
 - The multiplying x^i by x^j results in x^{i+j} .
 - The multiplication may create terms with degree more than $n - 1$, which means the result needs to be reduced using a modulus polynomial.

Polynomials (cont.)

- Example
 - Find the result of $(x^5 + x^2 + x) \otimes (x^7 + x^4 + x^3 + x^2 + x)$ in $GF(2^8)$ with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$. <

$$P_1 \otimes P_2 = x^5(x^7 + x^4 + x^3 + x^2 + x) + x^2(x^7 + x^4 + x^3 + x^2 + x) + x(x^7 + x^4 + x^3 + x^2 + x)$$

$$P_1 \otimes P_2 = x^{12} + x^9 + x^8 + x^7 + x^6 + x^9 + x^6 + x^5 + x^4 + x^3 + x^8 + x^5 + x^4 + x^3 + x^2$$

$$P_1 \otimes P_2 = (x^{12} + x^7 + x^2) \bmod (x^8 + x^4 + x^3 + x + 1) = x^5 + x^3 + x^2 + x + 1$$

- To find the final result, divide the polynomial of degree 12 by the polynomial of degree 8 (the modulus) and keep only the remainder.

Polynomials (cont.)

- Polynomial division with coefficients in GF(2)

$$\begin{array}{r} x^4 + 1 \\ \hline x^8 + x^4 + x^3 + x + 1 \quad | \quad x^{12} + x^7 + x^2 \\ \hline x^{12} + x^8 + x^7 + x^5 + x^4 \\ \hline x^8 + x^5 + x^4 + x^2 \\ x^8 + x^4 + x^3 + x + 1 \\ \hline \end{array}$$

Remainder $x^5 + x^3 + x^2 + x + 1$

Polynomials (cont.)

- Example:
 - In GF (2⁴), find the inverse of (x² + 1) modulo (x⁴ + x + 1).
 - Finding multiplicative inverse- use extended Euclidean Algorithm
- Solution
 - The answer is (x³ + x + 1) <How to proof the answer>

q	r_1	r_2	r	t_1	t_2	t
(x ² + 1)	(x ⁴ + x + 1)	(x ² + 1)	(x)	(0)	(1)	(x ² + 1)
(x)	(x ² + 1)	(x)	(1)	(1)	(x ² + 1)	(x ³ + x + 1)
(x)	(x)	(1)	(0)	(x ² + 1)	(x ³ + x + 1)	(0)
	(1)	(0)		(x ³ + x + 1)	(0)	

Polynomials (cont.)

- Example:
 - In $\text{GF}(2^8)$, find the inverse of (x^5) modulo $(x^8 + x^4 + x^3 + x + 1)$.

• Solution

q	r_I	r_2	r	t_I	t_2	t
(x^3)	$(x^8 + x^4 + x^3 + x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	(0)	(1)	(x^3)
$(x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$
(x)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$
$(x^3 + x^2 + 1)$	$(x^3 + x^2 + 1)$	(1)	(0)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$	(0)
	(1)	(0)		$(x^5 + x^4 + x^3 + x)$	(0)	

Polynomials (cont.)

- A better algorithm: Obtain the result by repeatedly multiplying a reduced polynomial by x .
- Example:
 - Find the result of multiplying $P_1 = (x^5 + x^2 + x)$ by $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$ in $\text{GF}(2^8)$ with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$

Polynomials (cont.)

- Solution:

- We first find the partial result of multiplying x^0, x^1, x^2, x^3, x^4 , and x^5 by P_2 . Note that although only three terms are needed, the product of $x^m \otimes P_2$ for m from 0 to 5 because each calculation depends on the previous result.

<i>Powers</i>	<i>Operation</i>	<i>New Result</i>	<i>Reduction</i>
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	No
$x^1 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2 + x)$	$x^5 + x^2 + x + 1$	Yes
$x^2 \otimes P_2$	$x \otimes (x^5 + x^2 + x + 1)$	$x^6 + x^3 + x^2 + x$	No
$x^3 \otimes P_2$	$x \otimes (x^6 + x^3 + x^2 + x)$	$x^7 + x^4 + x^3 + x^2$	No
$x^4 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2)$	$x^5 + x + 1$	Yes
$x^5 \otimes P_2$	$x \otimes (x^5 + x + 1)$	$x^6 + x^2 + x$	No
$\mathbf{P_1} \times \mathbf{P_2} = (x^6 + x^2 + x) + (x^6 + x^3 + x^2 + x) + (x^5 + x^2 + x + 1) = x^5 + x^3 + x^2 + x + 1$			

MATHEMATICS OF CRYPTOGRAPHY

PART III

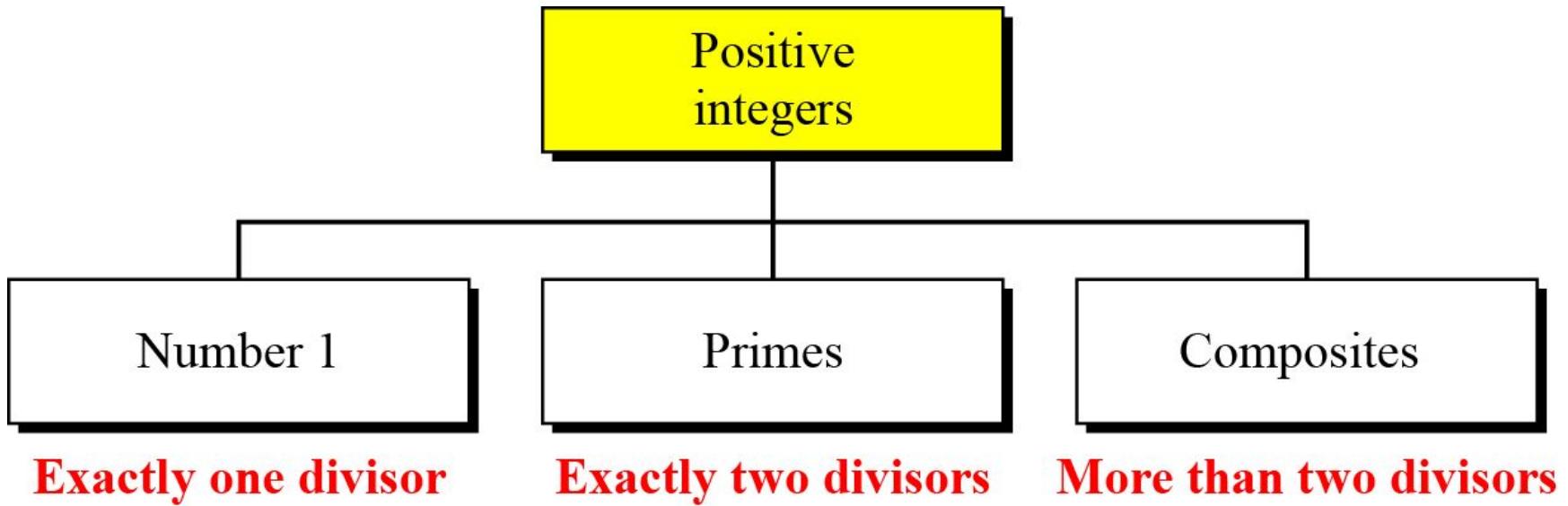
Primes and Related Congruence Equations

Objectives

- To introduce prime numbers and their applications in cryptography
- To discuss some primality test algorithms and their efficiencies.
- To discuss factorization algorithms and their applications in cryptography
- To discuss the Chinese remainder theorem and its application
- To introduce modular exponentiation and algorithm

Primes

Three groups of positive integers



*A prime is divisible only by itself and 1.
The smallest prime????*

Primes(cont.)

- Number of Primes

$$[n / (\ln n)] < \pi(n) < [n / (\ln n - 1.08366)]$$

- E.g. Find the number of primes less than 1,000,000.
 - The approximation gives the range 72,383 to 78,543. The actual number of primes is 78,498

Checking for Primeness

- Given a number n , how can we determine if n is a prime?
 - The answer is that we need to see if the number is divisible by primes less than \sqrt{n}
- Is 97 a prime?
 - The floor of $\sqrt{97} = 9$. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.

Checking for Primeness(cont.)

- Is 301 a prime?
 - The floor of $\sqrt{301} = 17$. We need to check 2, 3, 5, 7, 11, and 13. The numbers 2, 3, and 5 do not divide 301, but 7 does. Therefore 301 is not a prime.

Euler's Phi-Function

- *Euler's phi-function*, $\varphi(n)$, which is sometimes called the *Euler's totient function* plays a very important role in cryptography.
- The function finds the number of integers that are both smaller than n and relatively prime to n
- The followings helps to find the value of $\varphi(n)$.
 1. $\varphi(1) = 0$.
 2. $\varphi(p) = p - 1$ if p is a prime.
 3. $\varphi(m \times n) = \varphi(m) \times \varphi(n)$ if m and n are relatively prime.
 4. $\varphi(p^e) = p^e - p^{e-1}$ if p is a prime.

Euler's Phi-Function(cont.)

- We can combine all four rules to find the value of $\varphi(n)$. For example, if n can be factored as

$$n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$$

- Then we combine the third and the fourth rule to find

$$\phi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_2^{e_2-1}) \times \dots \times (p_k^{e_k} - p_k^{e_k-1})$$

- The value of $\varphi(n)$ for large composites can be found only if the number n can be factored into primes.

The difficulty of finding $\varphi(n)$ depends on the difficulty of finding the factorization of n .

“ n can be factored into primes”

Euler's Phi-Function(cont)

$$1. \phi(1) = 0.$$

$$2. \phi(p) = p - 1 \text{ if } p \text{ is a prime.}$$

$$3. \phi(m \times n) = \phi(m) \times \phi(n) \text{ if } m \text{ and } n \text{ are relatively prime.}$$

$$4. \phi(p^e) = p^e - p^{e-1} \text{ if } p \text{ is a prime.}$$

- Example 1

- What is the value of $\phi(13)$?

- Solution

- Because 13 is a prime, $\phi(13) = (13 - 1) = 12$.

- Example 2

- What is the value of $\phi(10)$?

- Solution

- We can use the third rule: $\phi(10) = \phi(2) \times \phi(5) = 1 \times 4 = 4$, because 2 and 5 are primes.

Euler's Phi-Function(cont)

1. $\phi(1) = 0.$
2. $\phi(p) = p - 1$ if p is a prime.
3. $\phi(m \times n) = \phi(m) \times \phi(n)$ if m and n are relatively prime
4. $\phi(p^e) = p^e - p^{e-1}$ if p is a prime.

- **Example 3**

- What is the value of $\phi(240)?$

- **Solution**

- We can write $240 = 2^4 \times 3^1 \times 5^1$. Then

$$\phi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$$

- **Example 4**

- Can we say that $\phi(49) = \phi(7) \times \phi(7) = 6 \times 6 = 36????$

- **Solution**

- No. The third rule applies when m and n are relatively prime. Here $49 = 7^2$. We need to use the fourth rule: $\phi(49) = 7^2 - 7^1 = 42$.

Euler's Phi-Function(cont.)

- Example 5
 - What is the number of elements in Z_{14}^* ?
- Solution
 - The answer is $\varphi(14) = \varphi(7) \times \varphi(2) = 6 \times 1 = 6$. The members are 1, 3, 5, 9, 11, and 13.

Interesting point: If $n > 2$, the value of $\varphi(n)$ is even.

Examples

If $n = 2020$, then $\Phi(n)$ is

- A. 200
- B. 400
- C. 600
- D. 800

prime factorization of 2,020 is $2^2 \times 5 \times 101$

Accepted Answers:

D.

Fermat's Little Theorem

- It plays important role in cryptography. It has two versions.
- First Version
 - If p is a prime and a is an integer such that p does not divide a , then

$$a^{p-1} \equiv 1 \pmod{p}$$

- Second Version
 - Removes the condition on a
 - It says that if p is prime and a is an integer,

$$a^p \equiv a \pmod{p} \quad \text{<exponent and modulus are same>}$$

Fermat's Little Theorem(cont.)

- Application- <exponentiation> it is helpful for quickly finding a solution to some exponentiation.
- Example 1
 - Find the result of $6^{10} \bmod 11$.
 $a^{p-1} \equiv 1 \bmod p$
 $a^p \equiv a \bmod p$
- Solution
 - We have $6^{10} \bmod 11 = 1$. This is the first version of Fermat's little theorem where $p = 11$.
- Example 2
 - Find the result of $3^{12} \bmod 11$.
- Solution
 - Here the exponent (12) and the modulus (11) are not the same. With substitution this can be solved using Fermat's little theorem.

$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11) (3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

Fermat's Little Theorem(cont.)

- Application- Multiplicative Inverses <if modulus is prime>
- If p is a prime and a is an integer such that p does not divide a .

$$a^{-1} \bmod p = a^{p-2} \bmod p$$

- The answers to multiplicative inverses modulo a prime can be found **without** using the extended Euclidean algorithm:
 - a. $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$
 - b. $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$
 - c. $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$
 - d. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

Euler's Theorem

- It is the generalization of Fermat's little theorem. The **modulus in Euler's theorem** is an integer not prime.

- First Version

- If a and n are coprime,

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

- Second Version

- Removes the condition that a and n should be coprime. If $n=p\times q$, $a < n$, and k is integer, then

$$a^{k \times \varphi(n) + 1} \equiv a \pmod{n}$$

The second version of Euler's theorem is used in the RSA cryptosystem

Euler's Theorem(cont.)

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

- Application- Exponentiation
- Example 1
 - Find the result of $6^{24} \pmod{35}$.

$$a^{k \times \phi(n) + 1} \equiv a \pmod{n}$$

- Solution
 - We have $6^{24} \pmod{35} = 6^{\phi(35)} \pmod{35} = 1$.
- Example 2
 - Find the result of $20^{62} \pmod{77}$.
- Solution

If we let $k = 1$ on the second version, we have

$$\begin{aligned}20^{62} \pmod{77} &= (20 \pmod{77}) (20^{\phi(77)+1} \pmod{77}) \pmod{77} \\&= (20)(20) \pmod{77} = 15.\end{aligned}$$

Euler's Theorem(cont.)

- Application-Multiplicative Inverses
 - Euler's theorem can be used to find multiplicative inverses modulo a composite. If n and a are coprime, then

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

Euler's Theorem(cont.)

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

- Example

- The answers to multiplicative inverses modulo a composite can be found **without** using the extended Euclidean algorithm.

- a. $8^{-1} \bmod 77 = 8^{\phi(77)-1} \bmod 77 = 8^{59} \bmod 77 = 29 \bmod 77$
- b. $7^{-1} \bmod 15 = 7^{\phi(15)-1} \bmod 15 = 7^7 \bmod 15 = 13 \bmod 15$
- c. $60^{-1} \bmod 187 = 60^{\phi(187)-1} \bmod 187 = 60^{159} \bmod 187 = 53 \bmod 187$
- d. $71^{-1} \bmod 100 = 71^{\phi(100)-1} \bmod 100 = 71^{39} \bmod 100 = 31 \bmod 100$

Generating Primes

- Mersenne Primes- $M_p = 2^p - 1$
- If p in the above formula is a prime, then M_p was thought to be prime.

$$M_2 = 2^2 - 1 = 3$$

$$M_3 = 2^3 - 1 = 7$$

$$M_5 = 2^5 - 1 = 31$$

$$M_7 = 2^7 - 1 = 127$$

$$M_{11} = 2^{11} - 1 = 2047$$

Not a prime ($2047 = 23 \times 89$)

$$M_{13} = 2^{13} - 1 = 8191$$

$$M_{17} = 2^{17} - 1 = 131071$$

A number in the form $M_p = 2^p - 1$ is called a Mersenne number and may or may not be a prime.

Generating Primes(cont.)

- Fermat Primes

$$F_n = 2^{2^n} + 1$$

$F_0 = 3$ $F_1 = 5$ $F_2 = 17$ $F_3 = 257$ $F_4 = 65537$
 $F_5 = 4294967297 = 641 \times 6700417$ *Not a prime*

Primality Testing

- Finding an algorithm to correctly and efficiently test a very large integer and output *a prime or a composite* has always been a challenge in number theory.
- Two types
 - Deterministic Algorithms <gives correct answer>
 - Probabilistic Algorithms <gives an answer that is correct most of the time, but not all of time>

Deterministic Algorithms

- Divisibility Algorithm

Algorithm 9.1 *Pseudocode for the divisibility test*

```
Divisibility_Test (n)                                // n is the number to test for primality
{
    r ← 2
    while (r <  $\sqrt{n}$ )
    {
        if ( $r \mid n$ ) return "a composite"
        r ← r + 1
    }
    return "a prime"
}
```

Probabilistic Algorithms

- Fermat Test

If n is a prime, then $a^{n-1} \equiv 1 \pmod{n}$.

If n is a prime, $a^{n-1} \equiv 1 \pmod{n}$

If n is a composite, it is possible that $a^{n-1} \equiv 1 \pmod{n}$

- Example
 - Does the number 561 pass the Fermat test?

Probabilistic Algorithms(cont.)

- Example
 - Does the number 561 pass the Fermat test?
- Solution
 - Use base 2

$$2^{561-1} = 1 \bmod 561$$

- The number passes the Fermat test, but it is not a prime, because $561 = 33 \times 17$.

FACTORIZATION

Fundamental Theorem of Arithmetic

$$n = p_1^{e1} \times p_2^{e2} \times \dots \times p_k^{ek}$$

- Greatest Common Divisor

$$a = p_1^{a1} \times p_2^{a2} \times \dots \times p_k^{ak}$$

$$b = p_1^{b1} \times p_2^{b2} \times \dots \times p_k^{bk}$$

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \dots \times p_k^{\min(a_k, b_k)}$$

- Least Common Multiplier- smallest integer that is multiple of both a&b

$$a = p_1^{a1} \times p_2^{a2} \times \dots \times p_k^{ak}$$

$$b = p_1^{b1} \times p_2^{b2} \times \dots \times p_k^{bk}$$

$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} \times p_2^{\max(a_2, b_2)} \times \dots \times p_k^{\max(a_k, b_k)}$$

- Example- GCD & LCM OF 16 and 64

$$\text{lcm}(a, b) \times \gcd(a, b) = a \times b$$

CHINESE REMAINDER THEOREM

- Used to solve a set of congruent equations with **one variable** but **different moduli**, which are relatively prime

$$\begin{aligned}x &\equiv a_1 \pmod{m_1} \\x &\equiv a_2 \pmod{m_2}\end{aligned}$$

...

$$x \equiv a_k \pmod{m_k}$$

- The above equations have a unique solution if the moduli are relatively prime

Continued...

- Example
 - The following is an example of a set of equations with different moduli:
$$\begin{aligned}x &\equiv 2 \pmod{3} \\x &\equiv 3 \pmod{5} \\x &\equiv 2 \pmod{7}\end{aligned}$$
 - The solution to this set of equations is given in the next section; for the moment, note that the answer to this set of equations is $x = 23$. This value satisfies all equations: $23 \equiv 2 \pmod{3}$, $23 \equiv 3 \pmod{5}$, and $23 \equiv 2 \pmod{7}$.

Continued...

- Solution To Chinese Remainder Theorem
 - Find $M = m_1 \times m_2 \times \dots \times m_k$. This is the common modulus.
 - Find $M_1 = M/m_1, M_2 = M/m_2, \dots, M_k = M/m_k$.
 - Find the multiplicative inverse of M_1, M_2, \dots, M_k using the corresponding moduli (m_1, m_2, \dots, m_k). Call the inverses $M_1^{-1}, M_2^{-1}, \dots, M_k^{-1}$.
 - The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \dots + a_k \times M_k \times M_k^{-1}) \bmod M$$

Continued...

- Example
 - Find the solution to the simultaneous equations:

$$\begin{aligned}x &\equiv 2 \pmod{3} \\x &\equiv 3 \pmod{5} \\x &\equiv 2 \pmod{7}\end{aligned}$$

- Solution: We follow the four steps.
 1. $M = 3 \times 5 \times 7 = 105$
 2. $M_1 = 105 / 3 = 35, M_2 = 105 / 5 = 21, M_3 = 105 / 7 = 15$
 3. The inverses are $M_1^{-1} = 2, M_2^{-1} = 1, M_3^{-1} = 1$
 4. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \pmod{105} = 23 \pmod{105}$

Continued...

- Example
 - Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.
- Solution ????

Continued...

- Example
 - Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.
- Solution
 - This is a CRT problem. We can form three equations and solve them to find the value of x .
$$x \equiv 3 \pmod{7}$$
$$x \equiv 3 \pmod{13}$$
$$x \equiv 0 \pmod{12}$$
 - If we follow the four steps, we find $x = 276$. We can check that $276 \equiv 3 \pmod{7}$, $276 \equiv 3 \pmod{13}$ and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

Continued...

- Assume we need to calculate $z = x + y$ where $x = 123$ and $y = 334$. These numbers can be represented as follows:

$$\begin{array}{ll} x \equiv 24 \pmod{99} & y \equiv 37 \pmod{99} \\ x \equiv 25 \pmod{98} & y \equiv 40 \pmod{98} \\ x \equiv 26 \pmod{97} & y \equiv 43 \pmod{97} \end{array}$$

- Adding each congruence in x with the corresponding congruence in y gives

$$\begin{array}{ll} x + y \equiv 61 \pmod{99} & \rightarrow z \equiv 61 \pmod{99} \\ x + y \equiv 65 \pmod{98} & \rightarrow z \equiv 65 \pmod{98} \\ x + y \equiv 69 \pmod{97} & \rightarrow z \equiv 69 \pmod{97} \end{array}$$

- Now three equations can be solved using the Chinese remainder theorem to find z . One of the acceptable answers is $z = 457$.

Continued...

Secret Sharing scheme in cryptography **aims to distribute and later recover secret S among n parties**. Secret S is distributed in form of **shares** which are generated from secret. Without cooperation of k no. of parties, the secret cannot be reconstructed from shares directly. Consider the following example:

Say our secret is S. The shares for n=4 no. of parties are generated taking modulus 11,13,17 and 19. They are respectively 1,12,2 and 3 and given by following equations:

$$\begin{aligned}S &\equiv 1 \pmod{11}, \\S &\equiv 12 \pmod{13}, \\S &\equiv 2 \pmod{17}, \\S &\equiv 3 \pmod{19}.\end{aligned}$$

Now, from four possible sets of k=3 shares (as **k shares are necessary to reconstruct the secret**), consider one possible set {1, 12, 2} and recover the secret S from it.

Continued...

Solution: The problem can be solved by Chinese remainder theorem.

For the set {1,12,2}, the equations available are,

$$S \equiv 1 \pmod{11},$$

$$S \equiv 12 \pmod{13},$$

$$S \equiv 2 \pmod{17},$$

Now solving this equation using CRT, $M=11 * 13 * 17 = 2431$,

$$M_1 = 2431/11 = 221,$$

$$M_2 = 2431/13 = 187,$$

$$M_3 = 2431/17 = 143$$

M_1^{-1} , M_2^{-1} and M_3^{-1} can be calculated using Extended Euclidean Algorithm.

$$M_1^{-1} = 1$$

$$M_2^{-1} = 8$$

$$M_3^{-1} = 5$$

Now, secret $S = ((1 * 221 * 1) + (12 * 187 * 8) + (2 * 143 * 5)) \pmod{2431}$

$$S = 155 \pmod{2431}$$

EXPONENTIATION AND LOGARITHM

EXPONENTIATION AND LOGARITHM

- Exponentiation and logarithm are inverses of each other.
- a is called the base of the exponentiation or logarithm

Exponentiation: $y = a^x$

\rightarrow

Logarithm: $x = \log_a y$

EXPONENTIATION

- In cryptography, a common modular operation is **exponentiation**. That is we often need to calculate.

$$y = a^x \bmod n$$

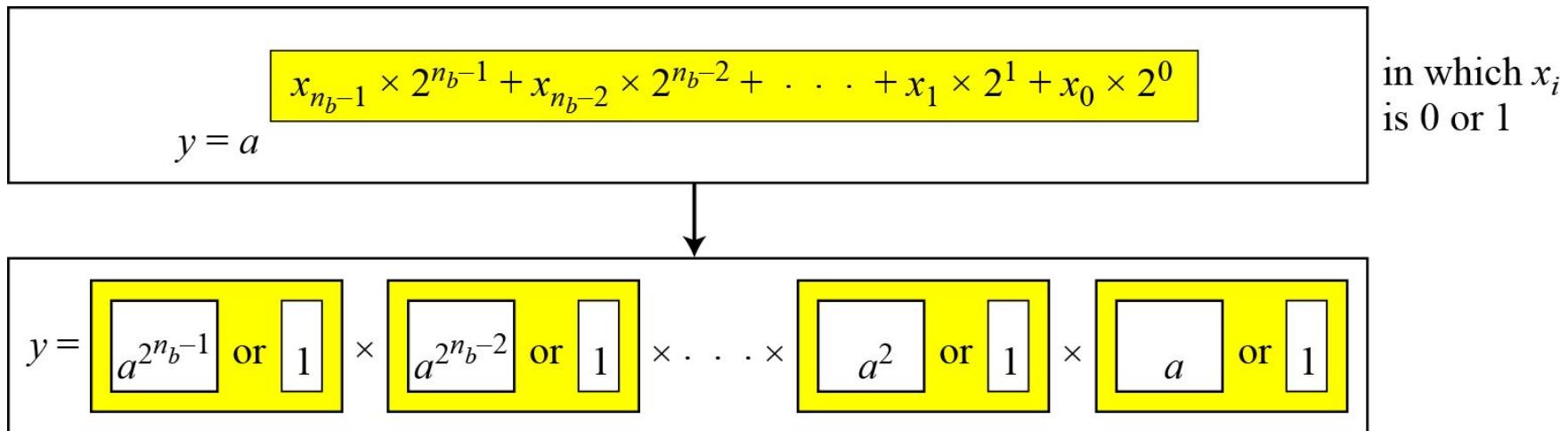
- The RSA cryptosystem, which uses exponentiation for both encryption and decryption with very large exponents.
- Unfortunately, most computer languages have no operator that can efficiently compute exponentiation, particularly when the exponent is very large.
- To make this type of calculation, we need more efficient algorithms.

EXPONENTIATION

- Fast Exponentiation
 - The idea behind the *square-and-multiply method*
- In traditional algorithms only *multiplication* is used to simulate exponentiation, but the fast exponentiation uses both *squaring* and *multiplication*.
- *square-and-multiply method* - treat the exponent as a binary number of n_b bits (x_0 to x_{nb-1})

Exponentiation

- Fast Exponentiation
 - The idea behind the square-and-multiply method



Example:

$$y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a$$

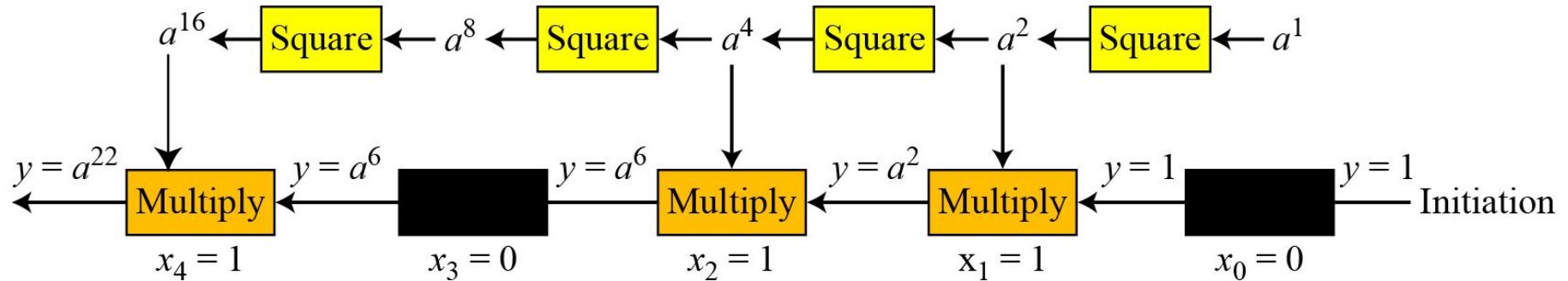
Continued...

Algorithm 9.7 *Pseudocode for square-and-multiply algorithm*

```
Square_and_Multiply (a, x, n)
{
    y ← 1
    for (i ← 0 to  $n_b - 1$ ) //  $n_b$  is the number of bits in x
    {
        if ( $x_i = 1$ ) y ←  $a \times y \bmod n$  // multiply only if the bit is 1
        a ←  $a^2 \bmod n$  // squaring is not needed in the last iteration
    }
    return y
}
```

Continued...

- The process for calculating $y = a^x$
- In this case, $x = 22 = (10110)_2$ in binary.



Continued...

Table 9.3 Calculation of $17^{22} \bmod 21$

i	x_i	<i>Multiplication (Initialization: $y = 1$)</i>	<i>Squaring (Initialization: $a = 17$)</i>
0	0		$a = 17^2 \bmod 21 = 16$
1	1	$y = 1 \times 16 \bmod 21 = 16$	$a = 16^2 \bmod 21 = 4$
2	1	$y = 16 \times 4 \bmod 21 = 1$	$a = 4^2 \bmod 21 = 16$
3	0		$a = 16^2 \bmod 21 = 4$
4	1	$y = 1 \times 4 \bmod 21 = 4$	

Logarithm

- In cryptography we need to discuss modular logarithm.
- If we use exponentiation to encrypt or decrypt, the adversary can use logarithm to attack.
- We need to know how hard it is to reverse the exponentiation.

Logarithm(cont.)

- Order of the Group.
- Example:
 - What is the order of group $G = \langle \mathbb{Z}_{21}^*, \times \rangle$?
 - $|G| = \phi(21) = \phi(3) \times \phi(7) = 2 \times 6 = 12$. There are 12 elements in this group: 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, and 20. All are relatively prime with 21.

Logarithm(cont.)

- Order of an element: The order of an element is the order of the cyclic group it generates.
- Example:
 - Find the order of all elements in $G = \langle \mathbb{Z}_{10}^*, \times \rangle$.
 - This group has only $\varphi(10) = 4$ elements: 1, 3, 7, 9.
 - **Lagrange Theorem**- The order of an element divides the order of the group. The only integers that divide 4 are 1, 2, and 4, which means in each case we need to check only these powers to find the order of the element.
 - a. $1^1 \equiv 1 \pmod{10} \rightarrow \text{ord}(1) = 1$.
 - b. $3^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(3) = 4$.
 - c. $7^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(7) = 4$.
 - d. $9^2 \equiv 1 \pmod{10} \rightarrow \text{ord}(9) = 2$.

Logarithm(cont.)

- Primitive roots
 - In the group $G = \langle \mathbb{Z}_n^*, \times \rangle$, when the order of an element is the same as $\varphi(n)$, that element is called the primitive root of the group.
 - Example
 - There are no primitive roots in $G = \langle \mathbb{Z}_8^*, \times \rangle$ because no element has the order equal to $\varphi(8) = 4$.

Logarithm(cont.)

- Example
 - the result of $a^i \equiv x \pmod{7}$ for the group $G = \langle \mathbb{Z}_7^*, \times \rangle$. In this group, $\phi(7) = 6$.

Table 9.5 Example 9.50

Primitive root →

Primitive root →

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
$a = 1$	x: 1					
$a = 2$	x: 2	x: 4	x: 1	x: 2	x: 4	x: 1
$a = 3$	x: 3	x: 2	x: 6	x: 4	x: 5	x: 1
$a = 4$	x: 4	x: 2	x: 1	x: 4	x: 2	x: 1
$a = 5$	x: 5	x: 4	x: 6	x: 2	x: 3	x: 1
$a = 6$	x: 6	x: 1	x: 6	x: 1	x: 6	x: 1

Logarithm(cont.)

The group $G = \langle \mathbb{Z}_n^, \times \rangle$ has primitive roots only if n is 2, 4, p^t , or $2p^t$. $\langle p$ is an odd prime (not 2) and t is an integer*

For which value of n , does the group $G = \langle \mathbb{Z}_n^, \times \rangle$ have primitive roots: 17, 20, 38, and 50?*

Solution

- a. $G = \langle \mathbb{Z}_{17}^*, \times \rangle$ has primitive roots, 17 is a prime.
- b. $G = \langle \mathbb{Z}_{20}^*, \times \rangle$ has no primitive roots.
- c. $G = \langle \mathbb{Z}_{38}^*, \times \rangle$ has primitive roots, $38 = 2 \times 19$ prime.
- d. $G = \langle \mathbb{Z}_{50}^*, \times \rangle$ has primitive roots, $50 = 2 \times 5^2$ and 5 is a prime.

Logarithm(cont.)

If the group $G = \langle \mathbb{Z}_n^, \times \rangle$ has any primitive root, the number of primitive roots is $\varphi(\varphi(n))$.*

Cyclic Group If g is a primitive root in the group, we can generate the set \mathbb{Z}_n^* as $\mathbb{Z}_n^* = \{g^1, g^2, g^3, \dots, g^{\varphi(n)}\}$

The group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$ has two primitive roots because $\varphi(10) = 4$ and $\varphi(\varphi(10)) = 2$. It can be found that the primitive roots are 3 and 7. The following shows how we can create the whole set \mathbb{Z}_{10}^* using each primitive root.

$$\begin{array}{lllll} g = 3 \rightarrow & g^1 \bmod 10 = 3 & g^2 \bmod 10 = 9 & g^3 \bmod 10 = 7 & g^4 \bmod 10 = 1 \\ g = 7 \rightarrow & g^1 \bmod 10 = 7 & g^2 \bmod 10 = 9 & g^3 \bmod 10 = 3 & g^4 \bmod 10 = 1 \end{array}$$

The group $G = \langle \mathbb{Z}_n^*, \times \rangle$ is a cyclic group if it has primitive roots. The group $G = \langle \mathbb{Z}_p^*, \times \rangle$ is always cyclic.

Logarithm(cont.)

The idea of Discrete Logarithm

Properties of $G = \langle \mathbb{Z}_p^, \times \rangle$:*

- 1.** *Its elements include all integers from 1 to $p - 1$.*
- 2.** *It always has primitive roots.*
- 3.** *It is cyclic. The elements can be created using g^x where x is an integer from 1 to $\varphi(n) = p - 1$.*
- 4.** *The primitive roots can be thought as the base of logarithm.*

Logarithm(cont.)

Solution to Modular Logarithm Using Discrete Logs

Tabulation of Discrete Logarithms

Table 9.6 Discrete logarithm for $\mathbf{G} = \langle \mathbf{Z}_7^*, \times \rangle$

y	1	2	3	4	5	6
$x = L_3 y$	6	2	1	4	5	3
$x = L_5 y$	6	4	5	2	1	3

Logarithm(cont.)

Find x in each of the following cases:

a. $4 = 3^x \pmod{7}$.

b. $6 = 5^x \pmod{7}$.

Solution

We can easily use the tabulation of the discrete logarithm:

a. $4 = 3^x \pmod{7} \rightarrow x = L_3 4 \pmod{7} = 4 \pmod{7}$

b. $6 = 5^x \pmod{7} \rightarrow x = L_5 6 \pmod{7} = 3 \pmod{7}$

One-Time Pad:

One of the goals of cryptography is perfect secrecy. A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain. This idea is used in a cipher called one-time pad, invented by **Vernam**.

- The key has the same length as the plaintext and is chosen completely random.
- A one-time pad is the perfect cipher, but it is almost impossible to implement commercially. If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?
- However there are some occasions when a one-time pad can be used. **For example:** if the president of a country needs to send a completely secret message to the president of another country, she can send a trusted envoy with the random key before sending the message.
- The encryption and decryption algorithms each use a single exclusive-or operation. Based on the properties of the exclusive-or operation, the encryption and decryption algorithms are inverses of each other. It is important to note that in this cipher the exclusive-or operation is used one bit at a time. In other words, the operation is over 1-bit word and the field is GF(2).

Security analysis:

- There is no way that an adversary can guess the key or the plaintext and ciphertext statistics.
- There is no relationship between the plaintext and ciphertext, either. In other words, the ciphertext is a true random stream of bits even if the plaintext contains some pattern.
- Eve cannot break the cipher unless she tries all possible random key streams, which would be 2^n if the size of the plaintext is n bits.

Implementation Issue:

How can the sender and the receiver share a one-time pad key each time they want to communicate? They need to somehow agree on the random key. So this perfect and ideal cipher is very difficult to achieve.

PKC algorithm

RSA

RSA & Diffie-Hellman

- RSA - Ron Rivest, Adi Shamir and Len Adleman at MIT, in 1977.
 - RSA is a block cipher
 - The most widely implemented
- Diffie-Hellman
 - Exchange a secret key securely



RSA

- best known & widely used public-key scheme
 - can be used to provide both secrecy & digital signatures
 - security due to cost of factoring large numbers
-

Key Lengths

SKE length	PKC length
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
112 bits	1792 bits
128 bits	2304 bits

certification authorities use 4096 bits

RSA Key Setup

- each user generates a public/private key pair by
 - select two large distinct primes at random - p, q .
 - compute their system modulus $n=p \cdot q$
 - note $\phi(n) = (p-1)(q-1)$
 - select at random the encryption key e
 - where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
 - solve the following equation to find decryption key d
 - $e \cdot d \equiv 1 \pmod{\phi(n)}$ and $0 \leq d \leq n$. How ?
 - publish the public encryption key: $PU = \{e, n\}$
 - keep secret private decryption key: $PR = \{d, n\}$
 - It is critically important that the factors p & q of the modulus n are kept secret
-

A hacker problem

- If public key = (31,3599), then what is the private key?
 - From the problem $e = 31$, $n = 3599$
 - From this find p and q easily
 - Finding p and q , find $\phi(n)$
 - Having found out $\phi(n)$, apply Extended Euclidean to find d
-

RSA Use

- 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
-

RSA Example - Key Setup

- Select primes

$$p=17 \quad \& \quad q=11$$

- Compute n

$$n = pq = 17 \times 11 = 187$$

- Compute phi value

$$\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$$

- Select encryption parameter

$$e: \gcd(e, 160) = 1; \text{ choose } e=7$$

- Determine decryption parameter

$$d: de \equiv 1 \pmod{160} \text{ and } d < 160$$

Value is $d=23$?

- Publish public key $PU=\{7, 187\}$
- Keep secret private key $PR=\{23, 187\}$



RSA Example - En/Decryption

- The sample RSA private/public operations are:

- Given message $M = 88$ (note that $88 < 187$)

- Encryption is

$$\begin{aligned}C &= 88^7 \bmod 187 \\&= 88^{(3+3+1)} \bmod 187 \\&= (88^3 \bmod 187) (88^3 \bmod 187) (88 \bmod 187) \bmod 187 \\&= (44 * 44 * 88) \bmod 187 \\&= 11\end{aligned}$$

- Decryption is

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

RSA Example - Key Setup

- Select primes

$$p=11 \quad \& \quad q=3$$

- Compute n

$$n = pq = 11 \times 3 = 33$$

- Compute phi value

$$\phi(n) = (p-1)(q-1) = 10 \times 2 = 20$$

- Select encryption parameter

$$\gcd(e: \gcd(e, 20) = 1; \text{ choose } e = 3)$$

- Determine decryption parameter

$$d: de \equiv 1 \pmod{20} \text{ and } d < 20$$

Value is $d=7$ since $7 \times 3 = 21 = 20 \times 1 + 1$

- Publish public key

$$PU = \{3, 20\}$$

- Keep secret private key

$$PR = \{7, 20\}$$



RSA Example - En/Decryption

- The sample RSA private/public operations are:

- Given message $M = 7$ (note that $7 < 33$)

- Encryption is

$$C = 7^3 \bmod 33$$

$$= 343 \bmod 33$$

$$= 13$$

- Decryption is

$$M = 13^7 \bmod 33$$

$$= 13^{(3+3+1)} \bmod 33$$

$$= (13^3 \bmod 33) (13^3 \bmod 33) (13 \bmod 33) \bmod 33$$

$$= (2197 \bmod 33) (2197 \bmod 33) (13) \bmod 33$$

$$= 19 \cdot 19 \cdot 13 \bmod 33 = 4693 \bmod 33$$

$$= 7$$



Another Example

- Consider the text grouping in the groups of three i.e.
 - **ATTACKXATXSEVEN = ATT ACK XAT XSE VEN**
 - Represent the blocks in base 26 using A=0, B=1, C=2.....
 - $ATT = 0 * 26^2 + 19 * 26^1 + 19 = 513$
 - $ACK = 0 * 26^2 + 2 * 26^1 + 10 = 62$
 - $XAT = 23 * 26^2 + 0 * 26^1 + 19 = 15567$
 - $XSE = 23 * 26^2 + 18 * 26^1 + 4 = 16020$
 - $VEN = 21 * 26^2 + 4 * 26^1 + 13 = 14313$
 - Next issue is designing the cryptosystem – selecting the parameters.
 - What should be the value of n ?
 - The value of n should be greater than 17575. How & why ?
 - Let $p = 137$ and $q = 131$; so that $n = pq = 17947$
-

Another Example – Key Setup

- Compute phi value

$$\phi(n) = (p-1)(q-1) = 136 \times 130 = 17680$$

- Select encryption parameter

$$\gcd(e: \gcd(e, 17680) = 1; \text{ choose } e = 3)$$

- Determine decryption parameter

$$d: de \equiv 1 \pmod{17680} \text{ and } d < 17680$$

Value is $d=11787$

- Publish public key

$$PU = \{ 3, 17947 \}$$

- Keep secret private key

$$PR = \{ 11787, 17947 \}$$



Another Example – En/Decryption

- The sample RSA private/public operations are:
 - Given message $M = ATT = 513$
 - Encryption is
$$\begin{aligned} C &= 513^3 \bmod 17947 \\ &= 8363 \end{aligned}$$
 - Decryption is
$$\begin{aligned} M &= 8363^{11787} \bmod 17947 \\ &= 513 \end{aligned}$$
 - Overall the plaintext is represented as the set of integers m
$$\{513, 62, 15567, 16020, 14313\}$$
 - Overall the ciphertext is represented as the set of integers c
$$\{8363, 5017, 11884, 9546, 13366\}$$
-

Speed of RSA

- In H/W, the speed of RSA is 1000 times slower than DES
 - In S/W, the speed of RSA is 100 times slower than DES
 - It is assumed, the difference will remain....
 - even with the advancement in technology
 - How to speed up the RSA operations ?
-

RSA Key Generation

- The users of RSA must
 - determine two primes at random - p , q
 - select either e or d and compute the other
- primes p, q must not be easily derived from modulus $n=p \cdot q$
 - means must be sufficiently large
 - typically guess and use probabilistic test



RSA Key Generation (contd)

- exponents e , d are inverses, so use inverse algorithm to compute the other
 - So, the basic operation involved, in either case, is
 - modular exponentiation
 - hence need to optimize the same
 - Use square-and-multiply method
-

Computational complexity: Encryption

- encryption uses exponentiation to power e
- hence if e small, this will be faster
 - the most common choices are 3, 17 and 65537
 - X.509 recommends 65537
 - PEM recommends 3 while
 - PKCS#1 recommends either 3 or 65537
- but if e too small (eg e=3) security attack is possible
- if e fixed one must ensure $\gcd(e, \phi(n)) = 1$
 - i.e. reject any p or q not relatively prime to e



Computational complexity: Decryption

- decryption uses exponentiation to power d
 - this is likely to be large and insecure if not
 - can use the Chinese Remainder Theorem (CRT)
 - to compute mod p & q separately and then combine to get the desired answer
 - approx 4 times faster than doing directly
 - only owner of private key who knows values of p & q can use this technique
-

RSA Security

- RSA has been extensively analyzed for vulnerabilities by many researchers.
 - After thirty years, one finds interesting attacks but,
 - none of them is critical
 - they mostly illustrate the dangers of improper usage of RSA
 - hence, securely implementing RSA is a nontrivial task.
 - At an outset, the security depends on two mathematical problems
 - the problem of factoring large numbers
 - the RSA problem
 - both the above problems are considered to be hard
 - How do we define the RSA problem?
 - How do we define the RSA function ?
 - What is breaking the RSA security?
 - The most promising approach to solve the RSA problem appears to be
 - being able to factor n into p and q not knowing them a priori
 - How to use the factors then, to break the security?
-

Week 1: Lecture Notes

Topics:

Introduction to Cryptography.

Classical Cryptosystem.

Cryptanalysis on substitution cipher.

Playfair Cipher.

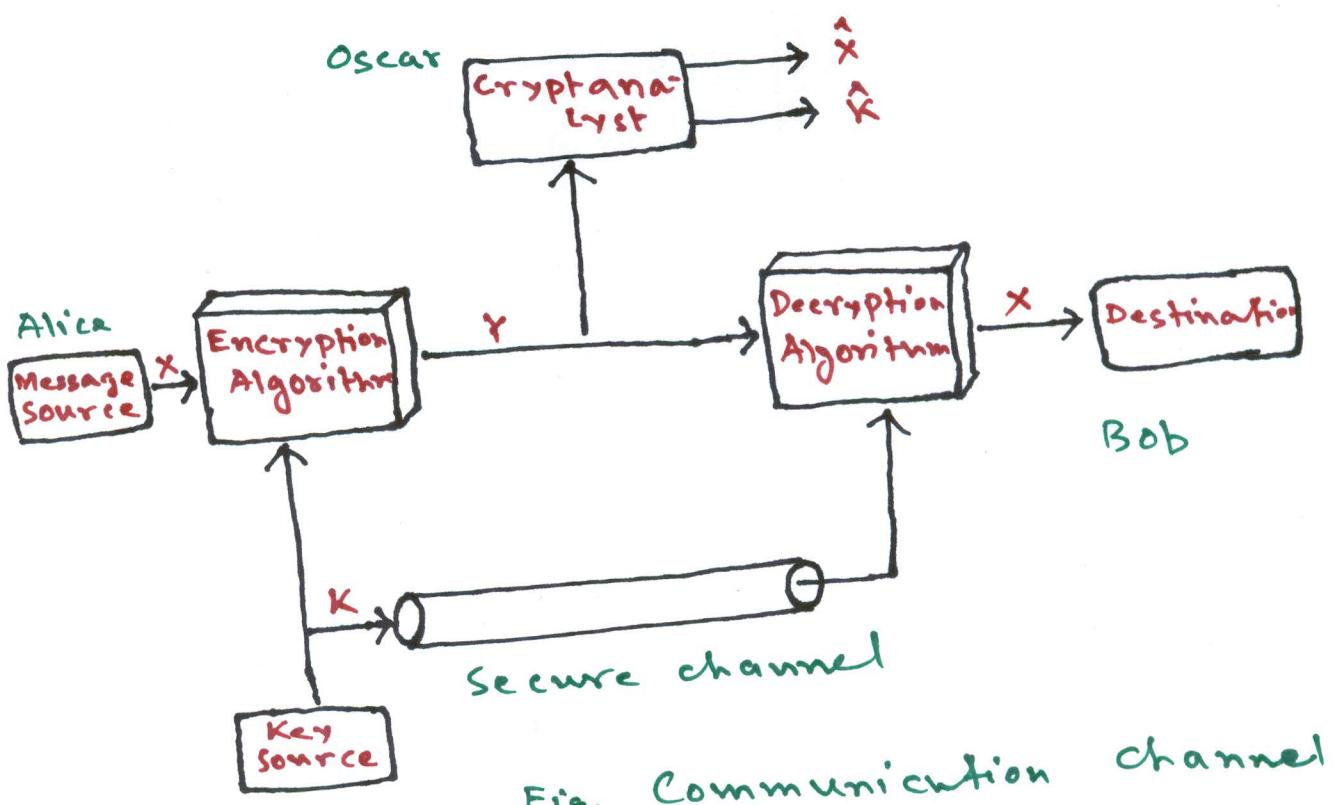
Block Cipher.

Introduction to Cryptography :

- Cryptography is the science or art of secret writing.
- The fundamental objective of cryptography is to enable two people (Alice and Bob) to communicate over an insecure channel in such a way that an opponent (Oscar) can not understand what is being said.
- **Plaintext:** the information that

Alice wants to send to Bob.

- Alice encrypts the plaintext, using a predetermined key, and send the resulting ciphertext to Bob over the public channel.
- Upon receiving the ciphertext
 - Oscar can not determine what the plaintext was.
 - But Bob knows the encryption key, can decrypt the ciphertext and get the plaintext.



- **Cryptology**: two competing areas:
 - Cryptography — Art of converting information to a form that will be unintelligible to an unintended recipient, carried out by Cryptographer.
 - Cryptanalysis — Art of breaking cryptographic systems, carried out by Cryptanalyst.
- Two main types of cryptography in use today:
 - Symmetric or secret key cryptography.
 - Asymmetric or public key cryptography.

Conventional Encryption :

- Also termed single key or symmetric encryption .

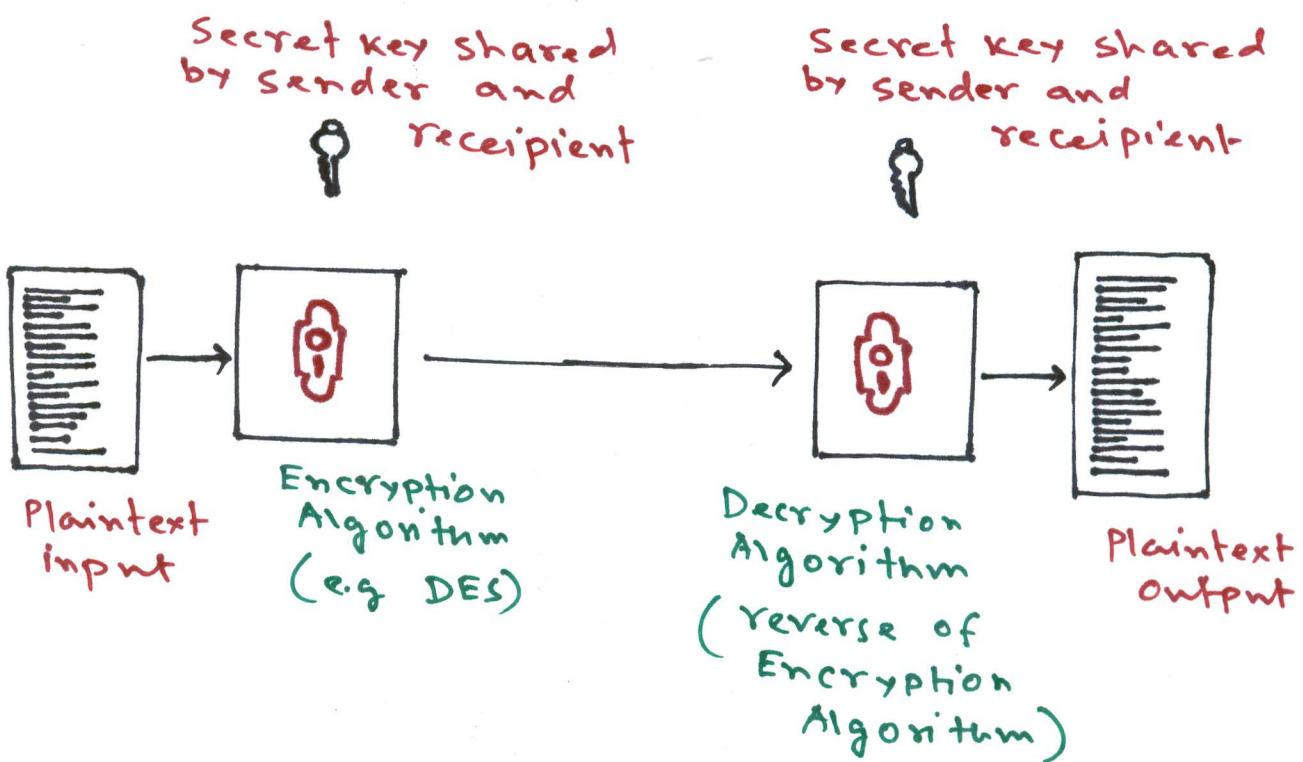


Fig: Simplified model of conventional encryption .

Cryptosystem

- Cryptosystem is a five tuple (P, C, X, E, D)

- Plaintext Space (P): set of all possible plaintext.
 - Ciphertext Space (C): set of all possible ciphertext.
 - Key Space (K): set of all possible keys.
 - E : set of all encryption rules
 - D : set of all possible decryption rules.
- For each $k \in K$, there is an encryption rule $e_k \in E$ and corresponding decryption rule $d_k \in D$ such that
- $$d_k(e_k(x)) = x \text{ for every plaintext } x \in P.$$
- A practical cryptosystem should satisfy
- Each encryption function e_k

and each decryption function d_k should be efficiently computable.

- An opponent, upon seeing the ciphertext string y , should be unable to determine the key k that was used or the plaintext string x .
- The process of attempting to compute the key k , given a string of ciphertext y , is called cryptanalysis
 - If the opponent can determine k , then he can decrypt y just as Bob would, using d_k .
 - Determining k should be as difficult as determining the plaintext string x , given the ciphertext string y

Classical Cryptosystem

Shift cipher

- $Z_{26} = \{0, 1, 2, \dots, 24, 25\}$

- $P = C = K = Z_{26}$

- For $k \in K$,

$$e_k(x) = (x+k) \bmod 26 \text{ for } x \in P$$

$$d_k(y) = (y-k) \bmod 26 \text{ for } y \in C$$

- Caesar Cipher is a particular case
 $(k=3)$

Example

- Plaintext is ordinary English text.
- Correspondence between alphabetic characters and integer : $A=0, B=1, \dots, Y=24, Z=25$

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Q	R	S	T	U	V	W	X	Y	Z
16	17	18	19	20	21	22	23	24	25

Encryption

- Key K = 11
 - Plaintext is "We will meet at midnight"
 - corresponding sequence of integers:
22, 4 22 8 11 11 12 4 4 19 0 19 12 8 3 13 8 6 7
19
 - We add 11 (key) to each value (rounding modulo 26):

7 15 7 19 22 22 23 15 15 4 11 4 23 19 14 24
19 17 18 9

- Convert the sequence of integers to alphabetic characters.
Ciphertext is:
“ HPHTWWXPPPELEXTOYTRSE ”

Decryption

- ciphertext : "HPHTWWXPPELEXTOYTRSE"
- convert the ciphertext to sequence of integers :

7 15 7 19 22 22 23 15 15 4 11 9 23 19 19
24 19 17 18 4

- Subtract 11 from each value (reducing modulo 26) :

22 4 22 8 11 11 12 4 4 19 0 19 12 8 3
13 8 6 7 19

- convert the sequence of integers to alphabetic characters :

Plaintext is "we will meet at midnight"

Caesar Cipher

- Caesar cipher is the earliest known (and the simplest). It involves

replacing each letter of the alphabet with the letter standing three places further down. This is then wrapped around on itself when the end is reached. For example

Key :

$K=3$

Plaintext : meetmeaftertheparty

Ciphertext : PHHWPHDIWHVWKHSUWB

Shift cipher is not secure

- Brute-force cryptanalysis easily performed on the shift cipher by trying all 25 possible keys.
- Given a ciphertext string, Oscar successively try the decryption process with $K=0, 1, 2$ etc. until get a meaningful text.

- Ciphertext: JBCRC~~L~~^GRWCRVNBJENBW
RWN
 - $k=0 \rightarrow$ jberelqrwervnbjenbwrrwn
 - $k=1 \rightarrow$ iabqbkpqr**b**qumaidmarqvm
 - $k=2 \rightarrow$ hzapajopuaptlzhclzupul
 - $k=3 \rightarrow$ gyzozi~~not~~^zoskygbkytotk
 - $k=4 \rightarrow$ fxyhymnsynrjxfajxsnsj
 - $k=5 \rightarrow$ ewxm~~xg~~^fmrxmqiweziwrnri
 - $k=6 \rightarrow$ dvwlwfklqwlp hvdyhvqlqh
 - $k=7 \rightarrow$ cuvkvejkprkoju~~c~~^xjupkpg
 - $k=8 \rightarrow$ btujudijoujnftbwftojof.
 - $k=9 \rightarrow$ astitchintimesavesnine
- The key $k=9$.

Substitution Cipher

- $P = C =$ set of 26-letter English alphabet

$$P = \{a, b, c, \dots, y, z\}$$

$$C = \{A, B, C, \dots, Y, Z\}.$$

- $X = \text{set of all possible permutations of 26 alphabet characters.}$
- For each permutation $\phi \in X$
 $e_\phi(x) = \phi(x)$ for $x \in P$
 $d_\phi(y) = \phi^{-1}(y)$ for $y \in C$,
 where ϕ^{-1} is the inverse permutation of ϕ .
- Encryption function is the permutation ϕ :

a	b	c	d	e	f	g	h	i	j	k	l	m	n
X	N	Y	A	H	P	O	G	Z	Q	W	B	T	S

o	p	q	r	s	t	u	v	w	x	y	z		
F	L	R	C	V	M	U	E	K	J	D	I		

- Decryption function is the inverse permutation ϕ^{-1}

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
d	l	r	y	v	o	h	e	z	x	w	p	t	b	g

P	Q	R	S	T	U	V	W	X	Y	Z				
f	j	q	u	m	u	s	k	a	c	i				

- Key : $K = \emptyset$

- ciphertext :

MGIZVYZLGHCMHJMYXSNHAYCDL
MHA

- Find the plain text ???

Vigenere Cipher

- Polyalphabetic cipher : use different monoalphabetic substitutions while moving through the plaintext.
- Let m be a positive integer
- $\mathcal{P} = \mathcal{C} = \mathcal{X} = (\mathbb{Z}_{26})^m$
- For $K = (K_1, K_2, \dots, K_m) \in \mathcal{K}^m$

$$e_K(x_1, \dots, x_m) = (x_1 + K_1, \dots, x_m + K_m)$$

$$d_K(y_1, \dots, y_m) = (y_1 - K_1, \dots, y_m - K_m)$$
- All above operations are performed in \mathbb{Z}_{26} .

- Example

- Correspondence between alphabetic characters and integers:

$$A=0, B=1, \dots, Y=24, Z=25$$

- $m = 6$
- Keyword is "CIPHER", this corresponds to numerical equivalent $k = (2, 8, 15, 7, 4, 17)$
- Plaintext: "this cryptosystem is not secure"
- Encryption: add modulo 26

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15
18	19	4	12	8	18	13	14	19	18	9	2
2	8	15	7	4	17	2	8	15	7	4	17
20	1	19	19	12	9	15	22	8	25	8	19
20	17	9									
2	8	15									
22	25	19									

- ciphertext: "VPX~~Z~~GIAXIVWPUBTTM~~J~~PWIZ ITWZT"

- Transposition techniques: so far all the ciphers we have looked at involved only substitution. A very different kind of mapping is achieved using transposition.
- In its simplest form, the rail fence technique involves writing down the plaintext as a sequence of columns and the ciphertext is read off as a sequence of rows. For example, if we use a rail fence of depth 2 with the plaintext 'meet me after the party is over' we get:

m	e	m	a	t	r	h	p	r	y	s	v	r
e	t	e	f	e	t	r	a	t	i	o	e	

- ciphertext is 'mematrhprysvrete feteatiae' which is simply the first row concatenated with the second.

Transposition/Permutation Cipher

- Let m be a positive integer
- $P = C = (\mathbb{Z}_{26})^m$
- $X =$ set of all possible permutations of $\{1, 2, \dots, m\}$
- For each permutation $\pi \in X$
 $e_\pi(x_1, \dots, x_m) = (x_{\pi(1)}, \dots, x_{\pi(m)})$
 $d_\pi(y_1, \dots, y_m) = (y_{\pi^{-1}(1)}, \dots, y_{\pi^{-1}(m)})$
- π' being the inverse permutation of π

Example :

- $m = 6$
- Key is the following permutation π :

x	1	2	3	4	5	6
$\pi(x)$	3	5	1	6	4	2

- inverse permutation π^{-1}

x	1	2	3	4	5	6
$\pi^{-1}(x)$	3	6	1	5	2	4

- Plaintext: "defendthehilltopatsunset"
- partition the plaintext into group of six letters:
 defend | thehil | ltopat | sunset
- rearrange according to π :
 fn ddee | eitlhh | oaltp | nestsu
- Ciphertext: " FNDDEEEITLHHOALTPT
 NESTSU "
- Decryption can be done using π^{-1}

Frequency Analysis

- Suppose we have a long ciphertext, the challenge is to decipher it.
- Let us know the text is in English and has been encrypted using a monoalphabetic substitution cipher.

- searching all possible keys is impractical as the key space size is 26!
- In English, e is the most common letter, followed by t, then a, and so on as shown in the figure

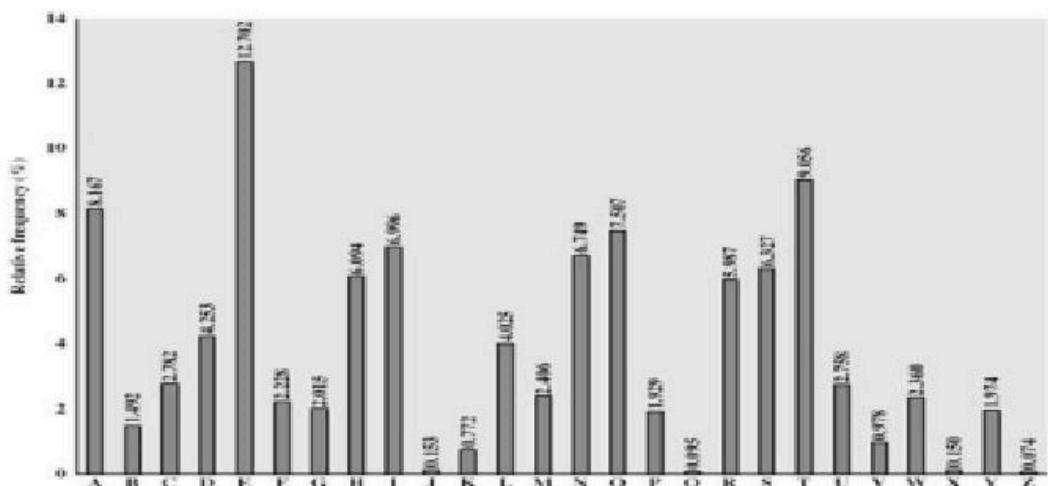


Figure 3: Relative Frequency of letters in the English Language.

- examine the ciphertext in question, and work out the frequency of each letter.
- if most common letter in the ciphertext is, for example, J then it would

seems likely that this is a substitution for e

- if the second most common letter in the ciphertext is P, then this is probably a substitution for t, and so on.
- however, regularities of the language may be exploited, e.g., relative frequency.
- frequency analysis requires logical thinking, intuition, flexibility and guesswork.

Playfair Cipher

- Use the key word CHARLES (Charles Wheatstone invented the cipher)
- Draw up a 5×5 matrix with the keyword first removing any repeating letters as follows:

c	h	a	r	t
e	s	b	d	f
g	i/j	k	m	n
o	p	q	t	u
v	w	x	y	z

- Plaintext: "meet me at the bridge"
 - split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex
- — Repeating plaintext letters that are in the same pair are separated with a filler letter such as 'x'
'ballon' would be treated as
ba lx lo on
- Two plaintext letters in the same row are each replaced by the letter to the right, with the first element of the row

circularly following the last.

eb is replaced by sd

ng is replaced by gi (or gi as preferred)

- 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.

dt would be replaced by my

ty would be replaced by yr

- 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.

me becomes gd

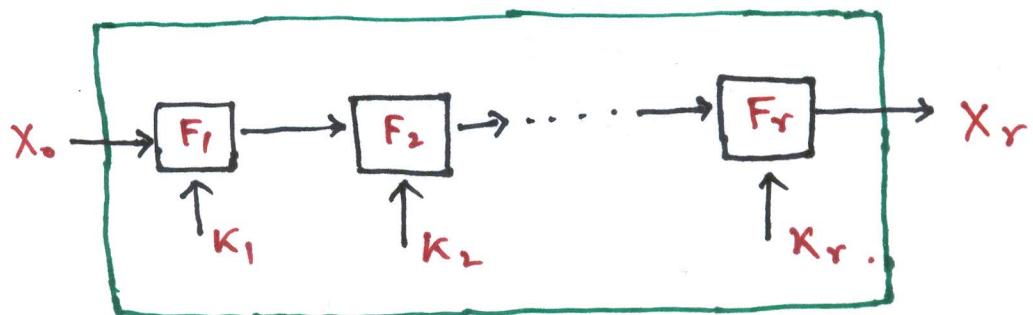
- Ciphertext therefore is:

"gd do gd rq pr sd hm em bv"

Block cipher

- Divided message (plaintext) into fixed sized blocks M_1, \dots, M_i, \dots
- Encrypt each message block separately to obtain $C_i = E_K(M_i)$ where
 - K is the secret symmetric key
 - $E_K()$ is the encryption function
- The cipher is $C_1, C_2, \dots, C_i, \dots$
- Decrypt each cipher block separately to obtain $M_i = D_K(C_i)$ where $D_K()$ is the decryption function.

r-round block cipher



- K_1, K_2, \dots, K_r is the list of round keys derived from the secret key using a publicly known key scheduling algorithm

- $x_i = F_i(x_{i-1}, k_i)$

Encryption

- Let X be plaintext.
- Let F be the round function for all the round.
- The encryption operation is carried out as follows:

$$\begin{aligned}
 x_0 &\leftarrow X \\
 x_1 &\leftarrow F(x_0, k_1) \\
 x_2 &\leftarrow F(x_1, k_2) \\
 &\vdots \\
 x_r &\leftarrow F(x_{r-1}, k_r)
 \end{aligned}$$

- Ciphertext $Y = x_r$.

Decryption

- F is injective function if its second argument is fixed
- There exists a function F' such that $F'(F(x, Y), Y) = x$. Then the decryption operation is carried out as follows:

$$\begin{aligned}
 x_r &\leftarrow Y \\
 x_{r-1} &\leftarrow F'(x_r, k_r) \\
 &\vdots \\
 x_1 &\leftarrow F'(x_2, k_2) \\
 x_0 &\leftarrow F'(x_1, k_1) \\
 x &\leftarrow x_0.
 \end{aligned}$$

Substitution-Permutation Network

- plaintext: $l m$ -bit binary string,
 $X = (x_1, x_2, \dots, x_m)$
- We can regard X as the concatenation
of m l bit substring:
 $X = X_{(1)} || X_{(2)} || \dots || X_{(m)}$ and for
 $1 \leq i \leq m$, we have that
 $X_{(i)} = (X_{(i-1)l+1}, \dots, X_{il})$
- S-box is a permutation $\pi_S : \{0,1\}^l \rightarrow \{0,1\}^l$
- $\pi_P : \{1, 2, 3, \dots, lm\} \rightarrow \{1, 2, 3, \dots, lm\}$
- k_1, k_2, \dots, k_{r+1} is the list of round
keys derived from the secret key
 k .

- The encryption algorithm is as follows:

Algorithm SPN

input : $x, \pi_s, \pi_p, (k_1, \dots, k_{r+1})$

output : Y

1. for $\omega^0 \leftarrow x$
2. for $i \leftarrow 1$ to $r-1$ do
3. $u^i \leftarrow \omega^{i-1} \oplus k_i$
4. for $j \leftarrow 1$ to m
5. do $v_{(ij)}^i \leftarrow \pi_s(u_{(ij)}^i)$
6. $\omega^i \leftarrow (v_{\pi_p(1)}, \dots, v_{\pi_p(m)}^i)$
7. end do
8. $u^r \leftarrow \omega^{r-1} \oplus k_r$.
9. for $j \leftarrow 1$ to m
10. do $v_{(jj)}^r \leftarrow \pi_s(u_{(jj)}^r)$
11. $Y \leftarrow v^r \oplus k_{r+1}$

- Let $l=m=r=4$ and π_s be defined as follows, where the input (i.e z) and the output (i.e $\pi_s(z)$) are written in hexadecimal notation,
 $(0=(0,0,0,0), 1=(0,0,0,1), \dots, 9=(1,0,0,1), A=(1,0,1,0), \dots, F=(1,1,1,1))$

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_s(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

- Let π_p be defined as follows:

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_p(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

- Key Scheduling Algorithm : $K = (K_1, \dots, K_{32})$.

For $1 \leq i \leq 5$, define K_i to consist of 16 consecutive bits of K , beginning with K_{4i-3}

- Example : Suppose the key is

$$K = 0011 \ 1010 \ 1001 \ 0100 \ 1101 \ 0110 \\ 0011 \ 1111$$

- The round keys are as follows:

$$K_1 = 0011 \ 1010 \ 1001 \ 0100$$

$$K_2 = 1010 \ 1001 \ 0100 \ 1101$$

$$K_3 = 1001 \ 0100 \ 1101 \ 0110$$

$$K_4 = 0100 \ 1101 \ 0110 \ 0011$$

$$K_5 = 1101 \ 0110 \ 0011 \ 1111$$

- Suppose that the plaintext is

$$X = 0010 \quad 0110 \quad 1011 \quad 0111$$

- Then the encryption of X proceeds as follows:

$$\omega^0 = 0010 \quad 0110 \quad 1011 \quad 0111$$

$$K_1 = 0011 \quad 1010 \quad 1001 \quad 0100$$

$$u' = 0001 \quad 1100 \quad 0010 \quad 0011$$

$$v^1 = 0100 \quad 0101 \quad 1101 \quad 0001$$

$$\omega^1 = 0010 \quad 1110 \quad 0000 \quad 0111$$

$$K_2 = 1010 \quad 1001 \quad 0100 \quad 1101$$

$$u^2 = 1000 \quad 0111 \quad 0100 \quad 1010$$

$$v^2 = 0011 \quad 1000 \quad 0010 \quad 0110$$

$$\omega^2 = 0100 \quad 0001 \quad 1011 \quad 1000$$

$$K_3 = 1001 \quad 0100 \quad 1101 \quad 0110$$

$$u^3 = 1101 \quad 0101 \quad 0110 \quad 1110$$

$$v^3 = 1001 \quad 1111 \quad 1011 \quad 0000$$

$$\omega^3 = 1110 \quad 0100 \quad 0110 \quad 1110$$

$$K_4 = 0100 \quad 1101 \quad 0110 \quad 0011$$

$$u^4 = 1010 \quad 1001 \quad 0000 \quad 1101$$

$$v^4 = 0110 \quad 1010 \quad 1110 \quad 1001$$

$$K_5 = 1101 \quad 0110 \quad 0011 \quad 1111$$

$$Y = 1011 \quad 1100 \quad 1101 \quad 0110$$

- Y is the ciphertext.

Introduction to Modern Symmetric-key Ciphers

Objectives

- ❑ To distinguish between traditional and modern symmetric-key ciphers.
- ❑ To introduce modern block ciphers and discuss their characteristics.
- ❑ To explain why modern block ciphers need to be designed as substitution ciphers.
- ❑ To introduce components of block ciphers such as P-boxes and S-boxes.

Objectives (Continued)

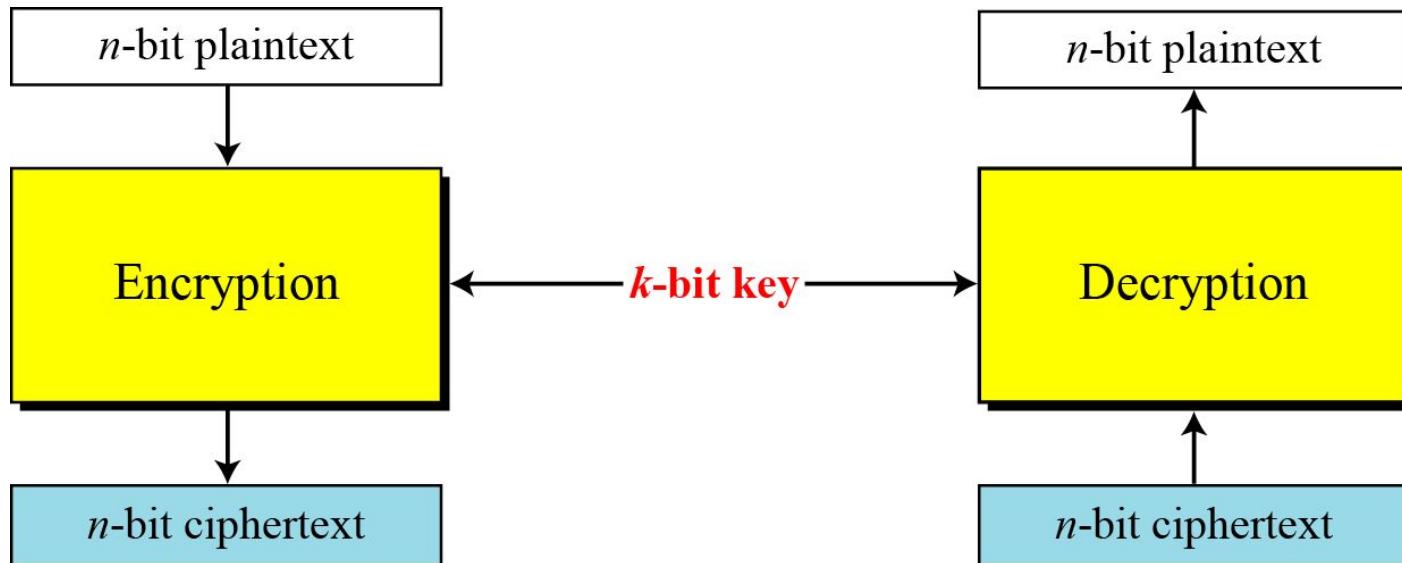
- To discuss product ciphers and distinguish between two classes of product ciphers: Feistel (Luby-rackoff) and non-Feistel ciphers.**
- To discuss two kinds of attacks particularly designed for modern block ciphers: differential and linear cryptanalysis.**

5.1 MODERN BLOCK CIPHERS

- Traditional symmetric-key ciphers- *character-oriented ciphers*
- Modern symmetric-key ciphers- *bit-oriented ciphers because the information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.*
- *It is convenient to convert these type of data into stream of bits, to encrypt the stream, and then send the encrypted stream.*
- *In addition, when text is treated at bit level, each character is replaced by 8 (or 16)bits, which means that the number of symbols becomes 8 (or 16) times larger. Mixing a large number of symbols increases security.*

5.1 *Continued*

Figure 5.1 *A modern block cipher*



- A *symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key.*
- *The decryption algorithm must be the **inverse** of encryption algorithm, and both operation must use the **same secret key**.*

5.1 *Continued*

- If the message has fewer than n bits, padding must be added to make it an n -bit block;
- If the message has more than n bits, it should be divided into n -bit block and the appropriate padding must be added to the last block if necessary.
- The common values for n are 64, 128, 256, or 512 bits.

5.1 *Continued*

Example 5.1

How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Solution

Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If $|M|$ and $|Pad|$ are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \rightarrow |Pad| = -800 \bmod 64 \rightarrow 32 \bmod 64$$

Note: only last block contains padding.

The cipher uses the encryption algorithm **thirteen** times to create thirteen ciphertext block.

5.1.1 Substitution or Transposition

- *A modern block cipher can be designed to act as a substitution cipher or a transposition cipher. Here symbols to be substituted or transposed are bits instead of characters.*
- *If the cipher is designed as a substitution cipher, a 1-bit or 0-bit in the plaintext can be replaced by either a 0 or 1. this means that the plaintext and the ciphertext can have a different number of 1's.*
- *A 64-bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.*
- *If the cipher is designed as a transposition cipher, the bits are only reordered (transposed); there is same number of 1's in plaintext and in the ciphertext.*

5.1.1 *Continued*

Note

**To be resistant to exhaustive-search attack,
a modern block cipher needs to be
designed as a substitution cipher.**

5.1.1 *Continued*

Example 5.2

Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the **intercepted ciphertext** in each of the following cases?

- a. The cipher is designed as a substitution cipher.
- b. The cipher is designed as a transposition cipher.

Solution

- a. In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
- b. In the second case, Eve knows that there are exactly 10 1's in the plaintext, because transposition does not change the number of 1's (or 0's) in the ciphertext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.

5.1.2 Block Ciphers as Permutation Groups

Full-Size Key Transposition Block Ciphers

In a full-size key transposition cipher We need to have $n!$ possible keys, so the key should have $\lceil \log_2 n! \rceil$ bits.

Example 5.3

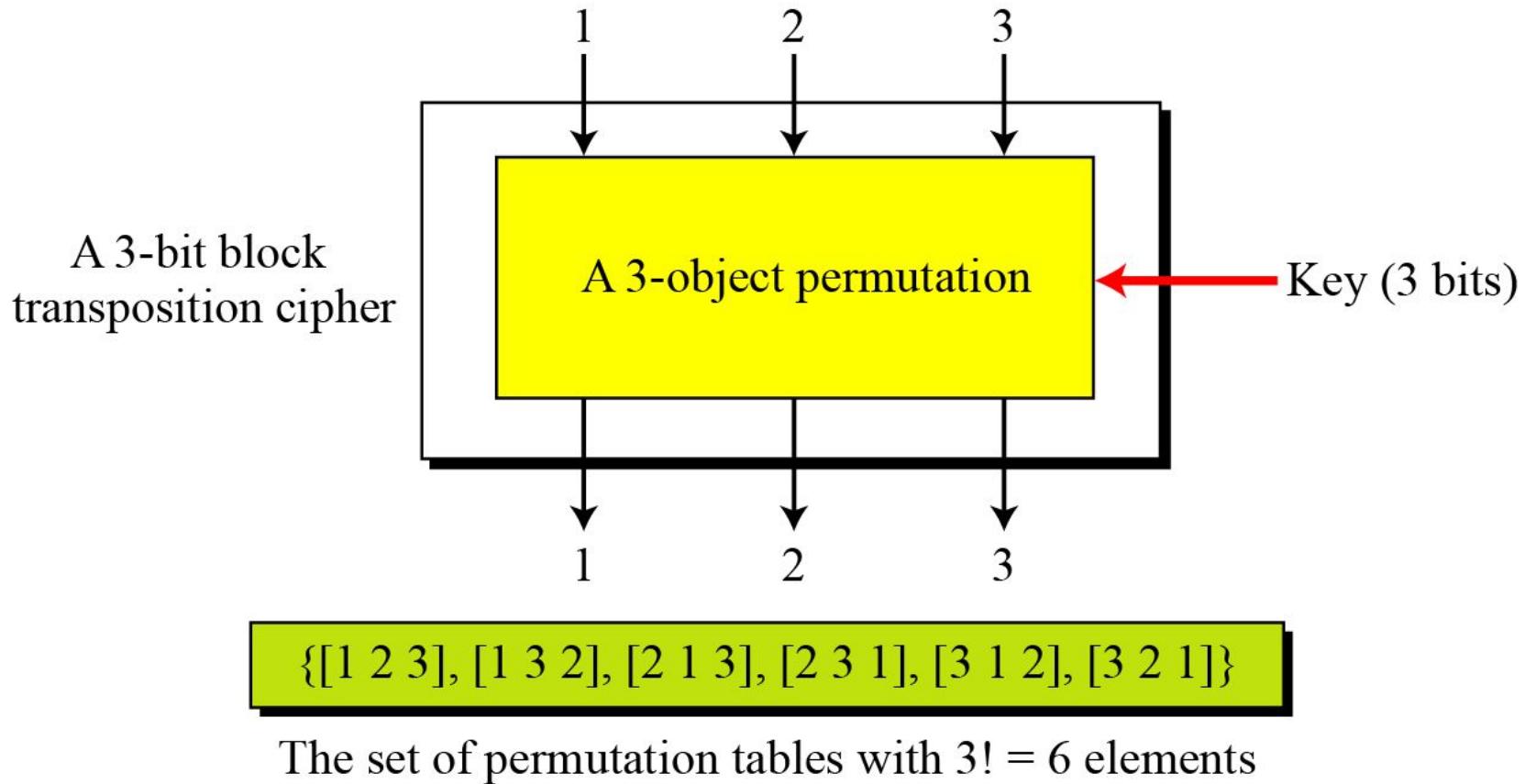
Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.

Solution

The set of permutation tables has $3! = 6$ elements, as shown in Figure 5.2.

5.1.2 *Continued*

Figure 5.2 *A transposition block cipher modeled as a permutation*



5.1.2 *Continued*

Full-Size Key Substitution Block Ciphers

A full-size key substitution cipher does not transpose bits; it substitutes bits. We can model the substitution cipher as a permutation if we can decode the input and encode the output.

Example 5.4

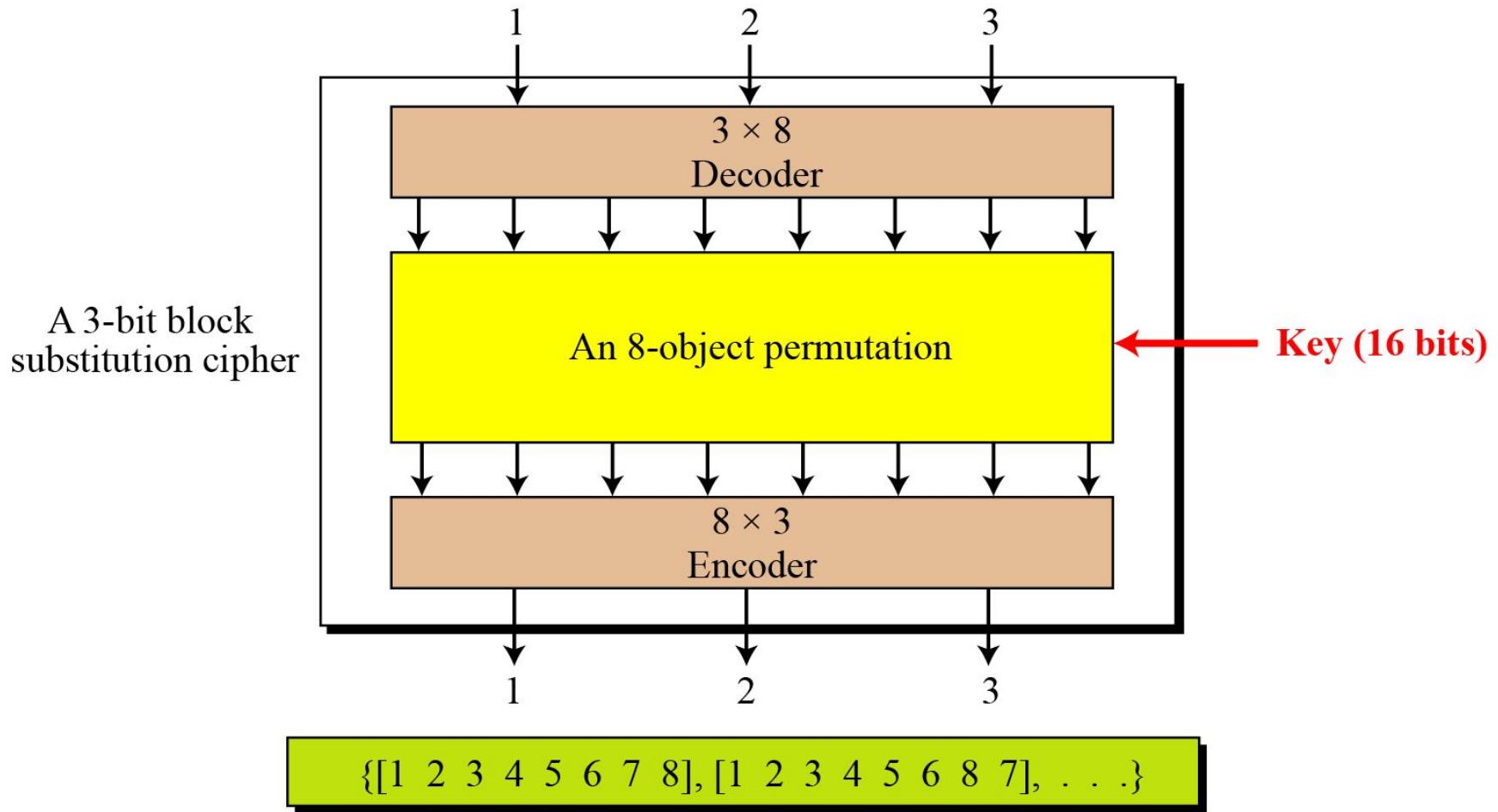
Show the model and the set of permutation tables for a 3-bit block substitution cipher.

Solution

Figure 5.3 shows the model and the set of permutation tables. The key is also much longer, $[\log_2 40,320] = 16$ bits.

5.1.2 *Continued*

Figure 5.3 *A substitution block cipher model as a permutation*



The set of permutation tables with $8! = 40,320$ elements

5.1.3 Components of a Modern Block Cipher

Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

P-Boxes

A P-box (permutation box) parallels the traditional transposition cipher for characters. It transposes bits.

S-Box

An S-box (substitution box): An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.

Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper- using idea of a product cipher
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
 - *substitution* (S-box) - confusion
 - *permutation* (P-box) - diffusion
- provide *confusion & diffusion* of message & key
- *Bit shuffling creates the diffusion effect, while substitution is used for confusion*

5.1.3 Continued

Confusion

The idea of confusion is to hide the relationship between the ciphertext and the key (means the key does not relate in a simple way to the ciphertext, each character of ciphertext should depend on several part of the key)

Note

Confusion hides the relationship between the ciphertext and the key.

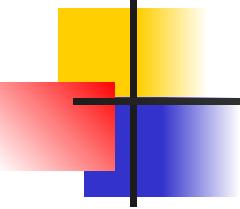
5.1.3 Continued

Diffusion

The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.

Note

Diffusion hides the relationship between the ciphertext and the plaintext.



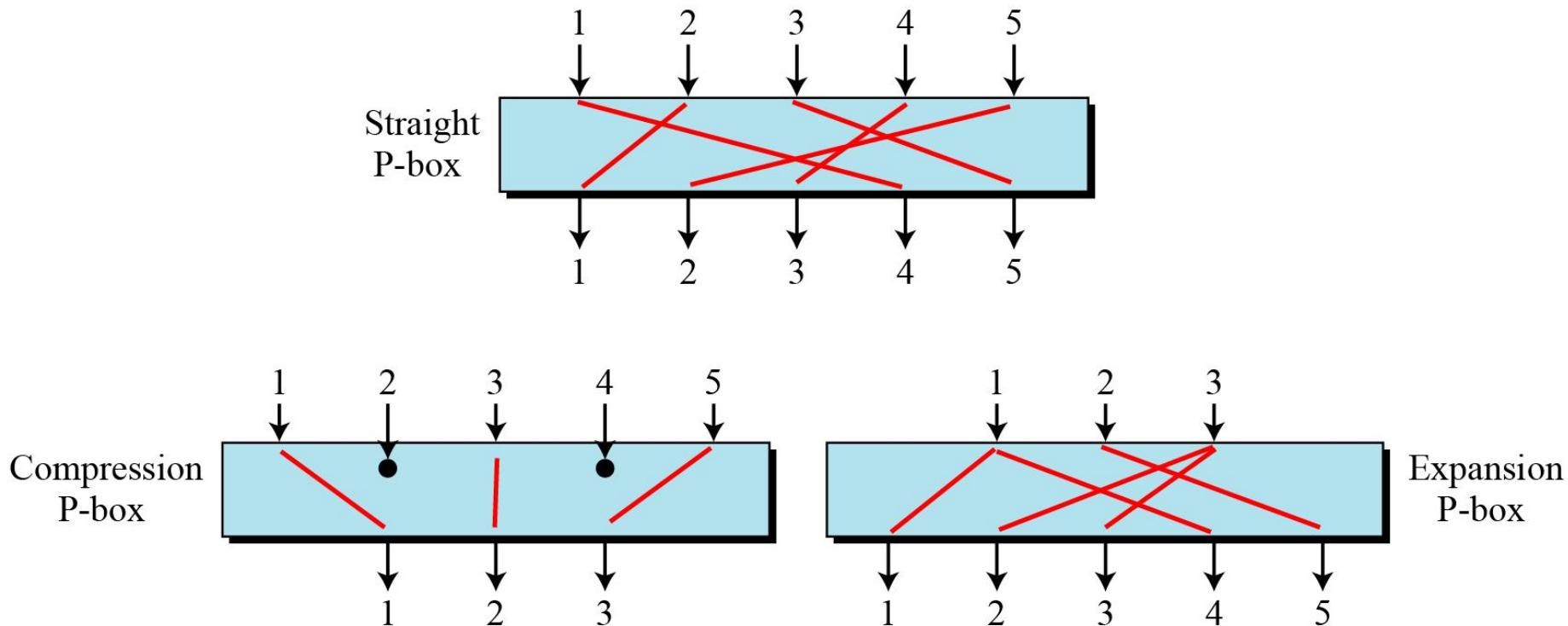
5.1.3 Continued

Example:

- Caesar ciphers have poor confusion.
- Polyalphabetic substitutions provide good confusion especially if the key length exceeds message length.
- Vernam cipher or one-time pad relies entirely on confusion, not on diffusion.
- Transposition ciphers provide good diffusion.

5.1.3 *Continued*

Figure 5.4 *Three types of P-boxes*

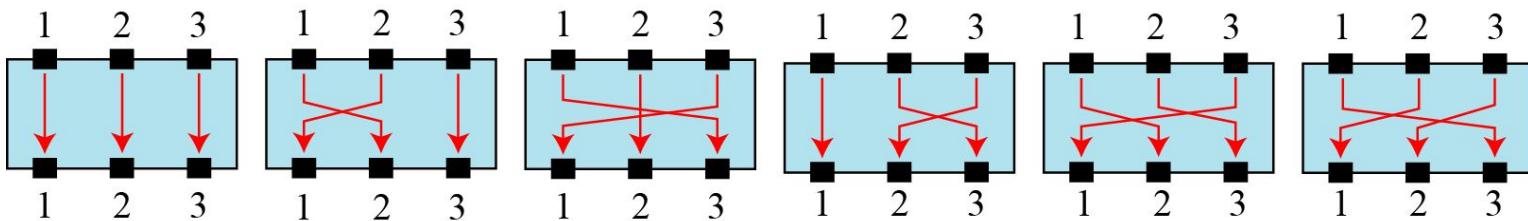


5.1.3 *Continued*

Example 5.5

Figure 5.5 shows all 6 possible mappings of a 3×3 P-box.

Figure 5.5 *The possible mappings of a 3×3 P-box*



5.1.3 *Continued*

Straight P-Boxes

Table 5.1 *Example of a permutation table for a straight P-box*

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

Table 5.1 has 64 entries, corresponding to the 64 inputs. The position (index) of the entry corresponds to the output. Because the first entry contains the number 58, we know that the **first output comes from the 58th input**. Because the last entry is 7, we know that the 64th output comes from the 7th input, and so on.

5.1.2 *Continued*

Example 5.6

Design an 8×8 permutation table for a straight P-box that moves the **two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8)** in the output words. Relative positions of other bits should not be changed.

Solution

We need a straight P-box with the table [4 1 2 3 6 7 8 5].

5.1.3 *Continued*

Compression P-Boxes

A compression P-box is a P-box with n inputs and m outputs where $m < n$.

Table 5.2 *Example of a 32×24 permutation table*

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

5.1.3 *Continued*

Compression P-Box

Table 5.2 *Example of a 32×24 permutation table*

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

5.1.3 *Continued*

Expansion P-Boxes

An expansion P-box is a P-box with n inputs and m outputs where $m > n$.

Table 5.3 *Example of a 12×16 permutation table*

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

5.1.3 *Continued*

P-Boxes: Invertibility

Note

A straight P-box is invertible, but compression and expansion P-boxes are not.

5.1.3 *Continued*

Example 5.7

Figure 5.6 shows how to invert a permutation table represented as a one-dimensional table.

Figure 5.6 *Inverting a permutation table*

1. Original table

6	3	4	5	2	1
---	---	---	---	---	---

2. Add indices

6	3	4	5	2	1
1	2	3	4	5	6

3. Swap contents
and indices

1	2	3	4	5	6
6	3	4	5	2	1

4. Sort based
on indices

6	5	2	3	4	1
1	2	3	4	5	6

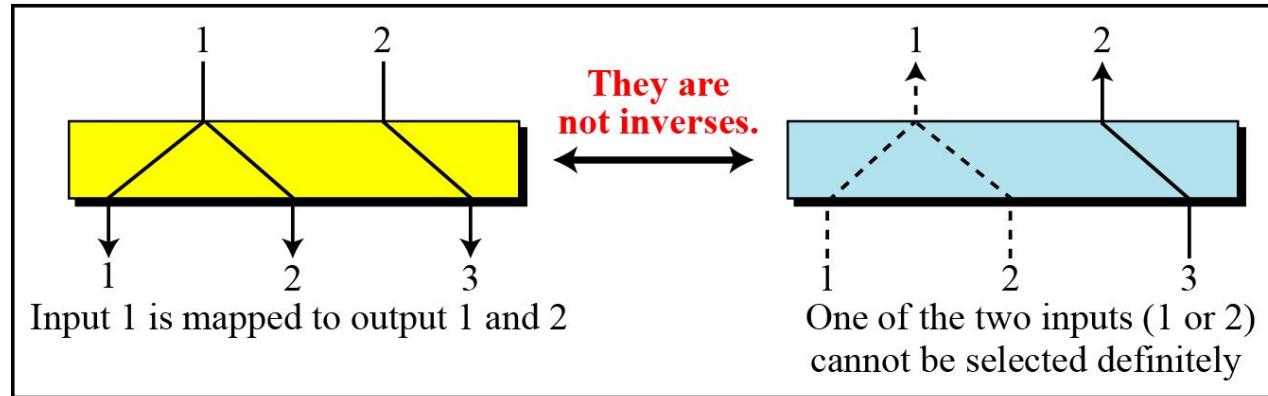
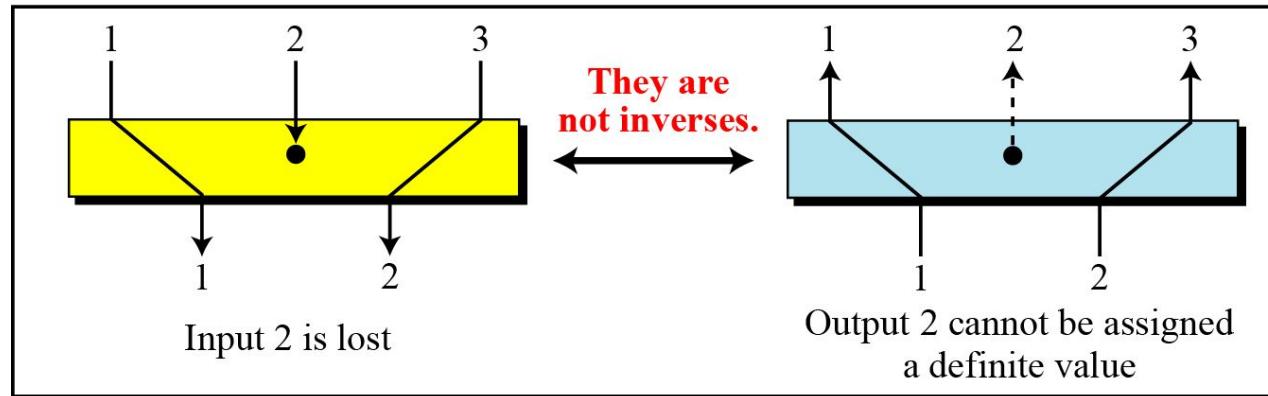
6	5	2	3	4	1
---	---	---	---	---	---

5. Inverted table

5.1.3 Continued

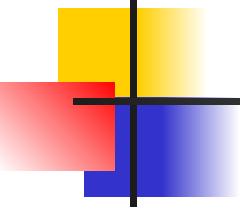
Figure 5.7 Compression and expansion P-boxes are non-invertible

Compression P-box



Expansion P-box

- In compression P-box, an input can be dropped during encryption; the decryption algorithm does not have a clue how to replace the dropped bit (a choice between a 0-bit or a 1-bit).
- In expansion P-box, an input may be mapped to more than one output during encryption; the decryption scheme algorithm does not have a clue which of the several inputs are mapped to an output.



5.1.3 Continued

- *Figure 5.7, shows that a compression P-box is not the inverse of an expansion P-box or vice versa.*
- *This means that if we use a compression P-box in the encryption cipher, we cannot use an expansion P-box in the decryption cipher; or vice versa.*

5.1.3 *Continued*

S-Box

An S-box (substitution box)

Note

An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.

- *S-box can have different number of inputs (n-bit word) and outputs (m-bit word).*
- *S-box can be keyed or keyless, modern block ciphers normally use keyless S-boxes, where mapping from the inputs to the outputs is predefined.*

5.1.3 *Continued*

Example 5.8

In an S-box with three inputs and two outputs, we have

$$y_1 = x_1 \oplus x_2 \oplus x_3 \quad y_2 = x_1$$

The S-box is linear because $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$ and $a_{2,2} = a_{2,3} = 0$. The relationship can be represented by matrices, as shown below:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

5.1.3 *Continued*

Example 5.10

The following table defines the input/output relationship for an **S-box of size 3×2** . The leftmost bit of the input defines the **row**; the two rightmost bits of the input define the **column**. The two output bits are values on the **cross section of the selected row and column**.

Leftmost bit

Rightmost bits

Output bits

		00	01	10	11
0	00	10	01	11	
1	10	00	11	01	

Based on the table, an input of 010 gives the output 01. An input of 101 gives the output of 00.

5.1.3 *Continued*

S-Boxes: Invertibility

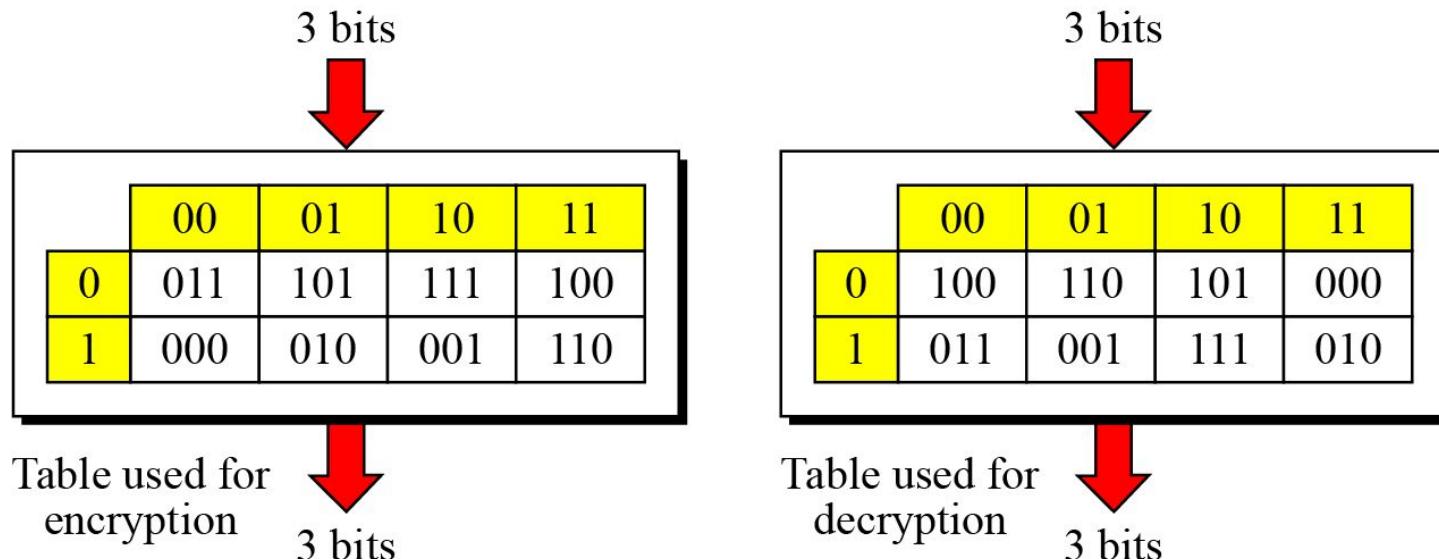
An S-box may or may not be invertible. In an invertible S-box, the number of input bits should be the same as the number of output bits.

5.1.3 *Continued*

Example 5.11

Figure 5.8 shows an **example of an invertible S-box**. For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.

Figure 5.8 *S-box tables for Example 5.11*

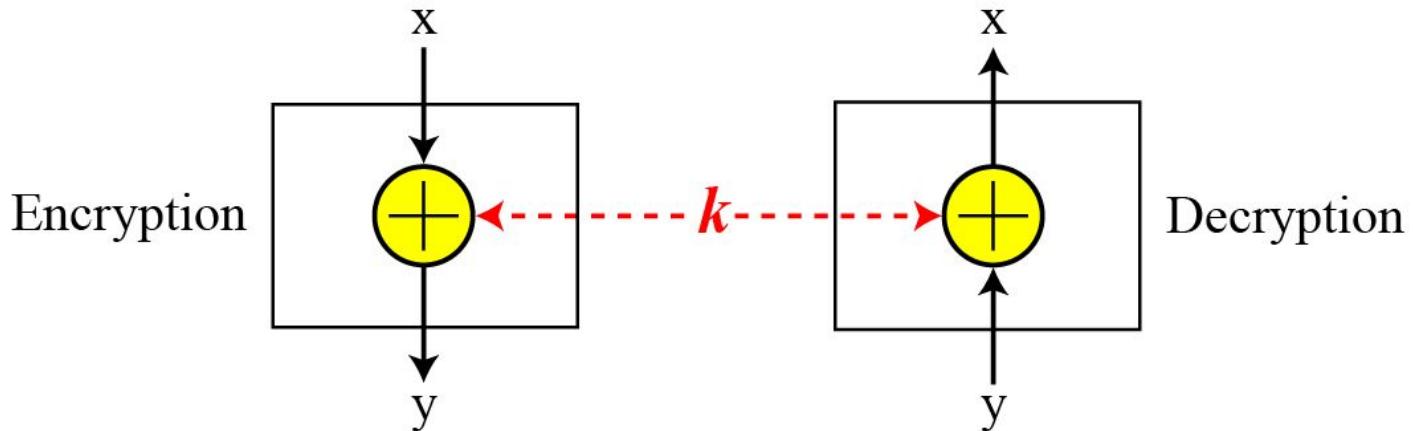


5.1.3 Continued

Exclusive-Or

An important component in most block ciphers is the exclusive-or operation.

Figure 5.9 Invertibility of the exclusive-or operation



5.1.3 Continued

Exclusive-Or (Continued)

An important component in most block ciphers is the exclusive-or operation. The addition and subtraction operations in the $GF(2^n)$ field are performed by a single operation called the exclusive-or (XOR).

*The five properties of the exclusive-or operation in the $GF(2^n)$ field makes this operation a very interesting component for use in a block cipher: **closure**, **associativity**, **commutativity**, **existence of identity**, and **existence of inverse**.*

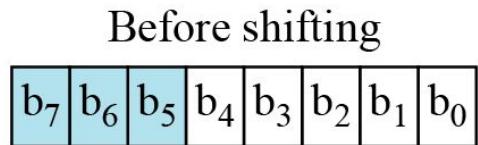
5.1.3 Continued

Circular Shift

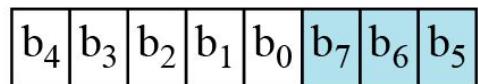
Another component found in some modern block ciphers is the circular shift operation. ($n=8$ and $k=3$)

The circular shift operation **mixes the bits in a word and helps hide the patterns in the original word.**

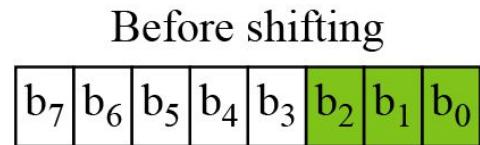
Figure 5.10 Circular shifting an 8-bit word to the left or right



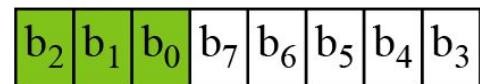
Shift left (3 bits)



After shifting



Shift right (3 bits)



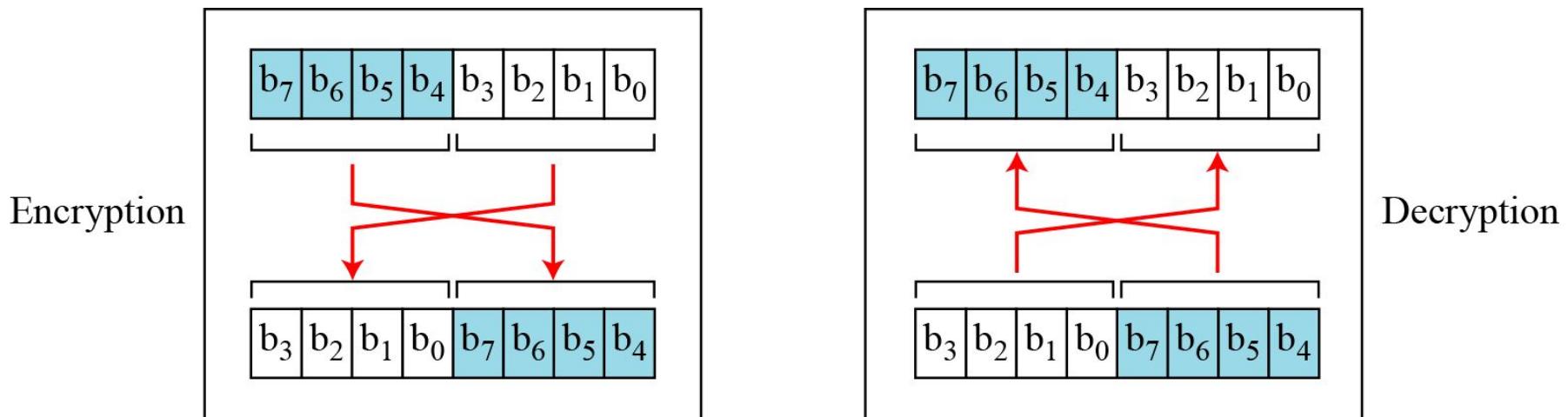
After shifting

5.1.3 Continued

Swap

The swap operation is a special case of the circular shift operation where $k = n/2$. <valid when n is an even number>, here left-shifting $n/2$ bits is same as right-shifting $n/2$, this component is self-invertible.

Figure 5.11 Swap operation on an 8-bit word

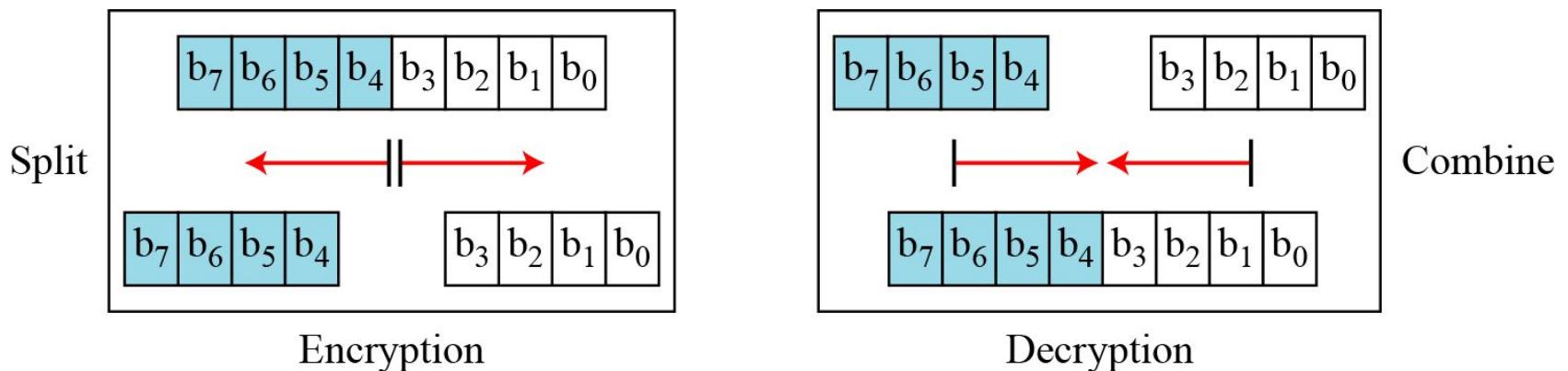


5.1.3 Continued

Split and Combine

Two other operations found in some block ciphers are split and combine.

Figure 5.12 *Split and combine operations on an 8-bit word*



5.1.4 Product Ciphers

Shannon introduced the concept of a product cipher. A product cipher is a complex cipher combining substitution, permutation, and other components discussed in previous sections.

5.1.4 Continued

Diffusion

The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.

Note

Diffusion hides the relationship between the ciphertext and the plaintext.

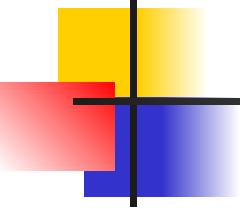
5.1.4 Continued

Confusion

The idea of confusion is to hide the relationship between the ciphertext and the key (means the key does not relate in a simple way to the ciphertext, each character of ciphertext should depend on several part of the key)

Note

Confusion hides the relationship between the ciphertext and the key.



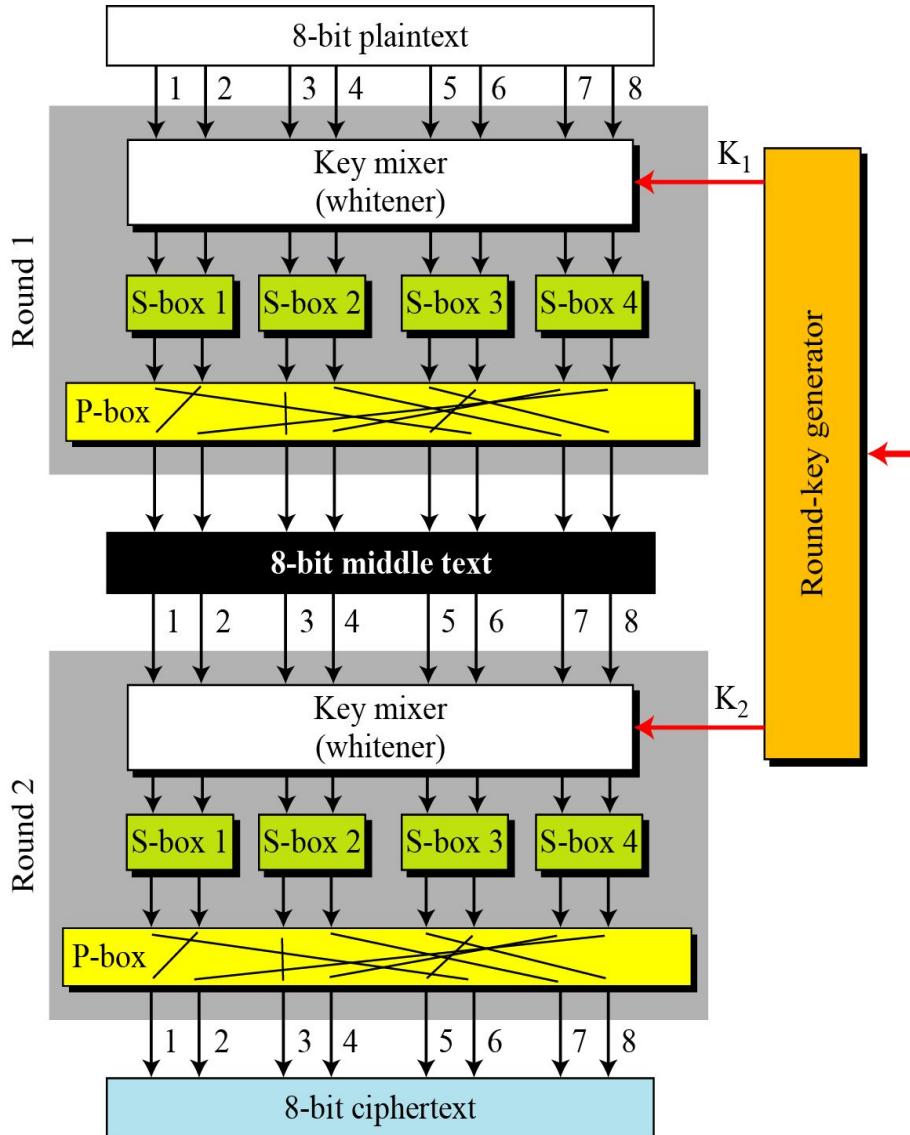
5.1.4 Continued

Rounds

Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

5.1.4 Continued

Figure 5.13 A product cipher made of two rounds

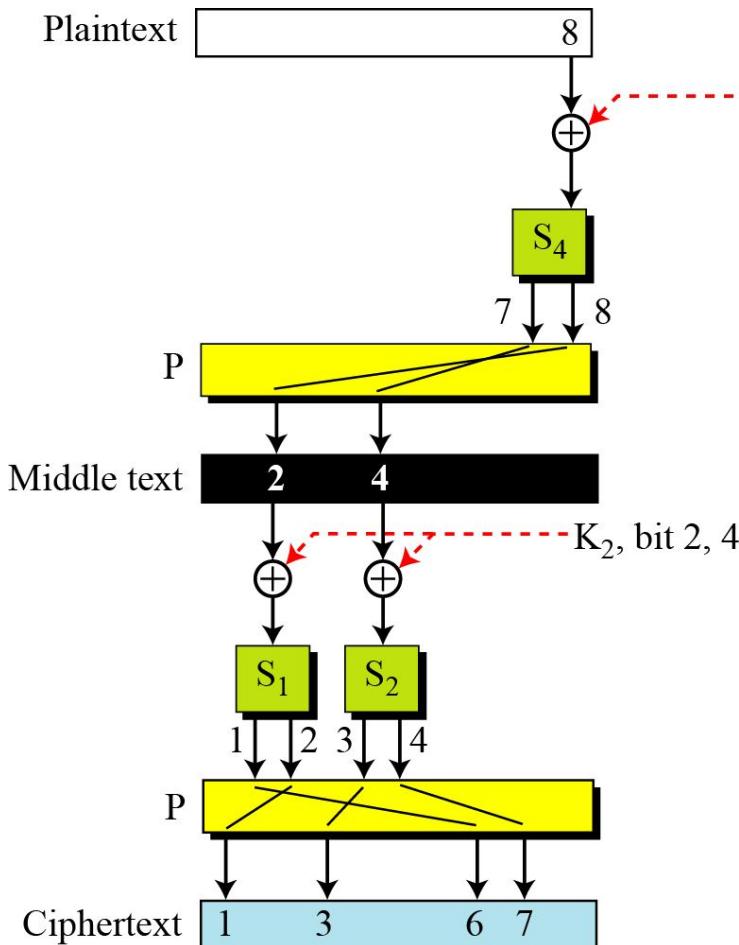


Three transformations happen at each round:

1. The 8-bit text is mixed with the key to whiten the text (hide the bits using the key). This is normally done by exclusive-oring the 8-bit word with the 8-bit key.
2. The output of the whitener are organized into four 2-bit groups and are fed into four S-boxes. The values of bits are changed based on the structure of the S-boxes in this transformation.
3. The output of S-boxes are passed through a P-box to permute the bits so that in the next round each box receives different inputs.

5.1.4 Continued

Figure 5.14 Diffusion and confusion in a block cipher



Diffusion: changing a single bit in the plaintext affects many bits in the ciphertext.

Confusion: the four bits of ciphertext, bits 1, 3, 6, and 7, are affected by three bits in the key (bit 8 in K_1 and bits 2 and 4 in K_2).

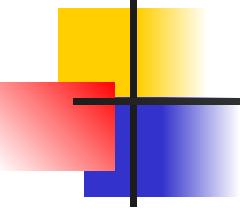
It shows that each bit in each round key effects several bits in the ciphertext. The relationship between ciphertext bits and key bits is unclear.

5.1.5 Two Classes of Product Ciphers

Modern block ciphers are all product ciphers, but they are divided into two classes.

1. Feistel ciphers

2. Non-Feistel ciphers



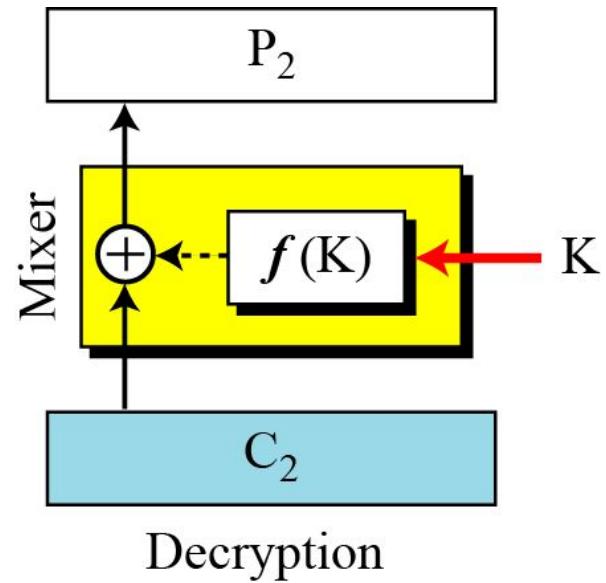
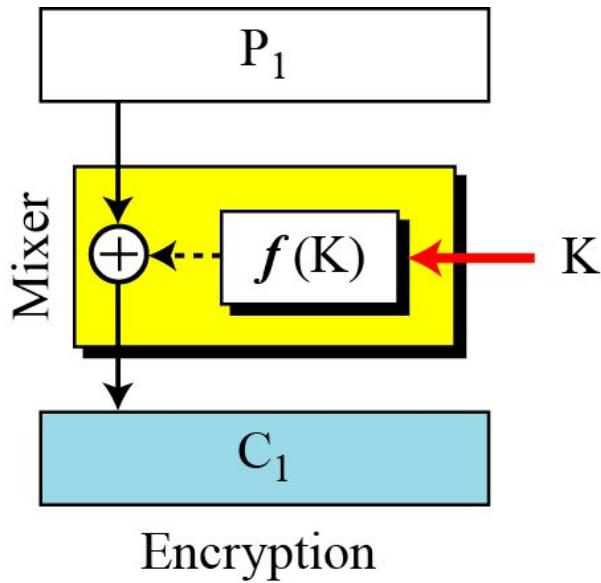
5.1.5 Continued

Feistel Ciphers

*Feistel designed a very intelligent and interesting cipher that has been used for decades. A Feistel cipher can have three types of components: **self-invertible**, **invertible**, and **noninvertible**.*

5.1.5 Continued

Figure 5.15 *The first thought in Feistel cipher design*



5.1.3 *Continued*

Example 5.12

This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the **function takes the first and third bits of the key**, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

Solution

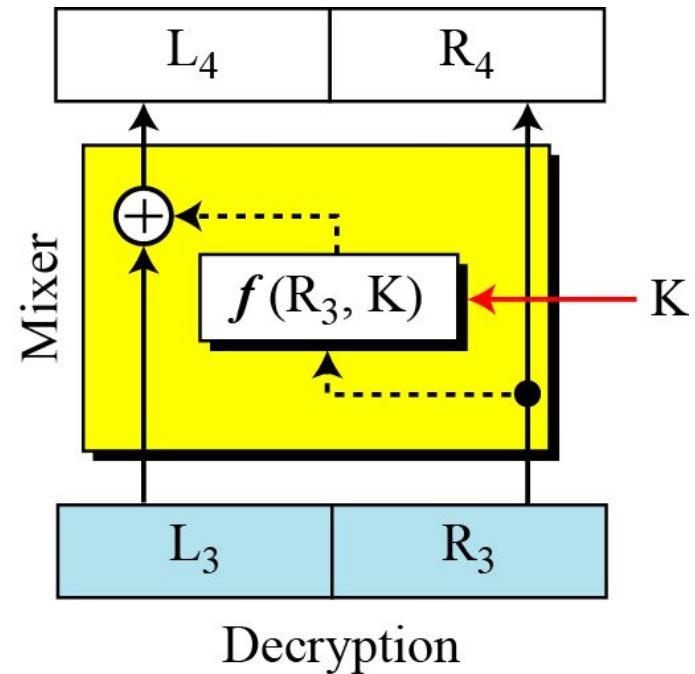
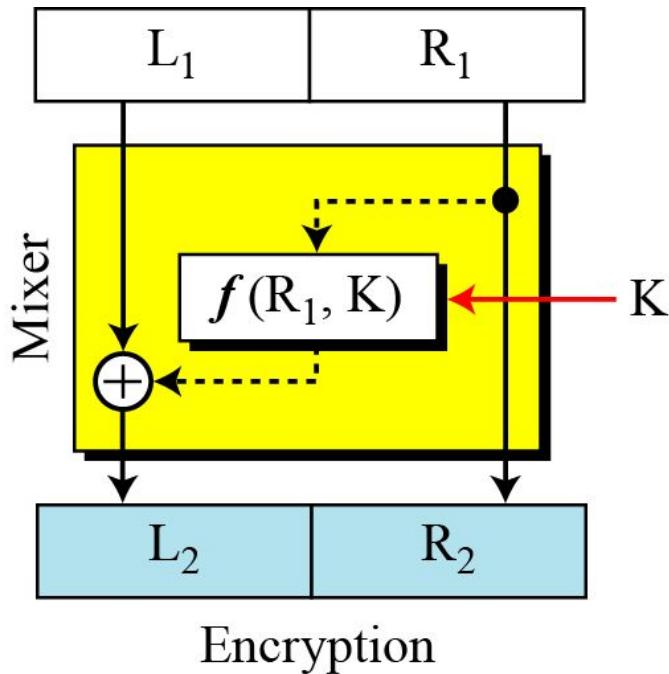
The function extracts the first and third bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

$$\text{Encryption: } C = P \oplus f(K) = 0111 \oplus 1001 = 1110$$

$$\text{Decryption: } P = C \oplus f(K) = 1110 \oplus 1001 = 0111$$

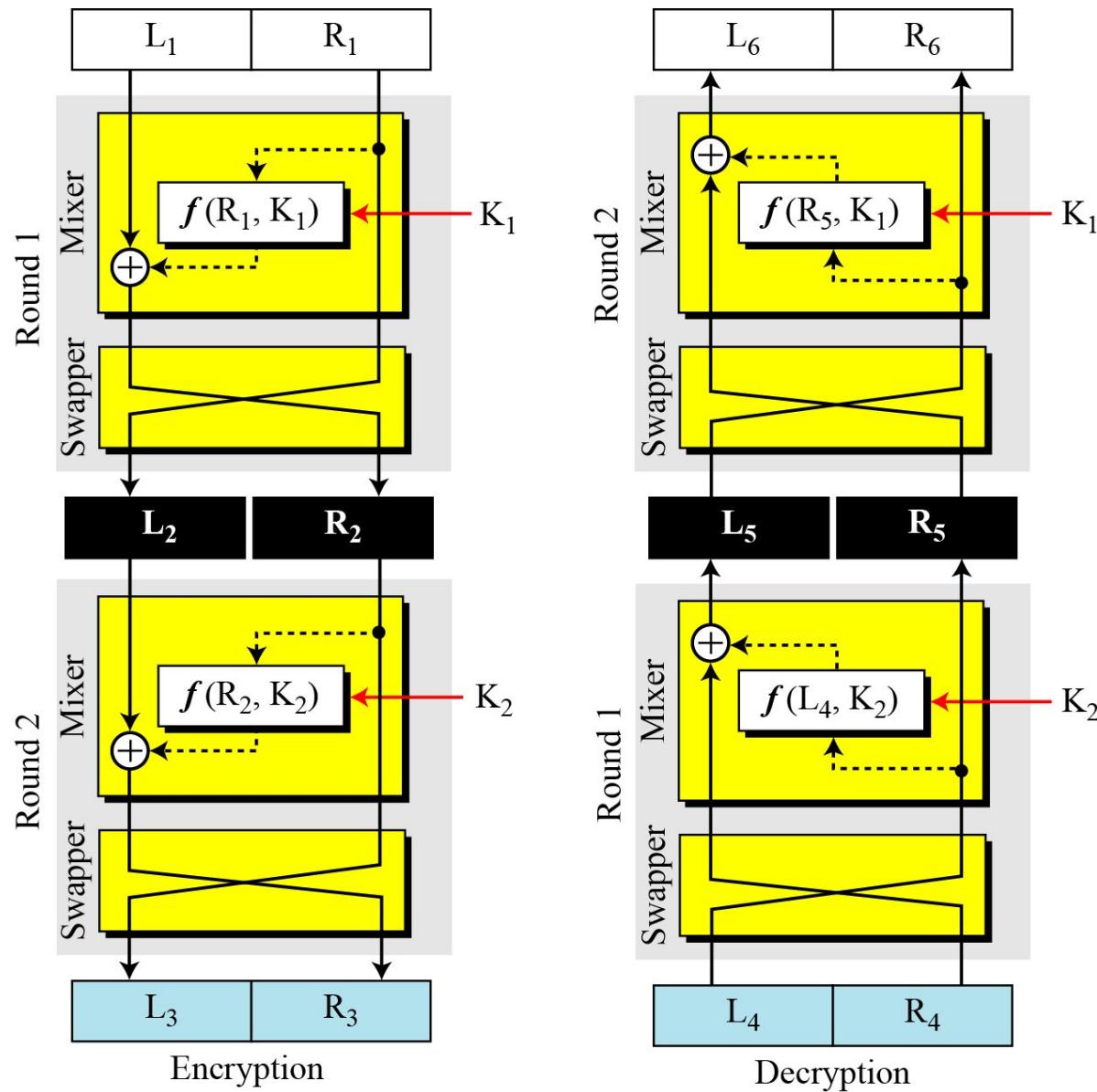
5.1.5 Continued

Figure 5.16 Improvement of the previous Feistel design



5.1.5 Continued

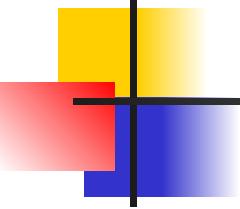
Figure 5.17 Final design of a Feistel cipher with two rounds



5.1.5 Continued

Non-Feistel Ciphers

A non-Feistel cipher uses only invertible components. A component in the encryption cipher has the corresponding component in the decryption cipher.



5.1.6 Attacks on Block Ciphers

Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks discussed in Chapter 3.

5.1.5 Continued

Differential Cryptanalysis

Eli Biham and Adi Shamir introduced the idea of differential cryptanalysis. This is a chosen-plaintext attack.

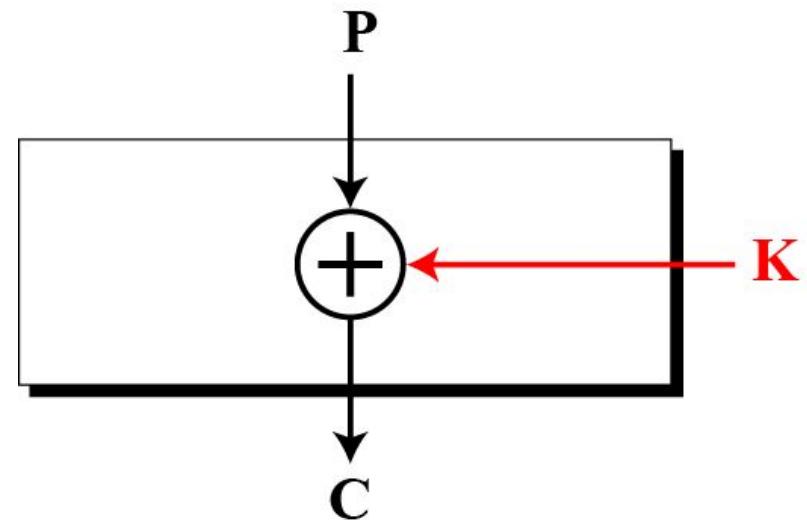
5.1.6 *Continued*

Example 5.13

Assume that the cipher is made only of one exclusive-or operation, as shown in Figure 5.18. Without knowing the value of the key, Eve can easily find the relationship between plaintext differences and ciphertext differences if by plaintext difference we mean $P_1 \oplus P_2$ and by ciphertext difference, we mean $C_1 \oplus C_2$. The following proves that $C_1 \oplus C_2 = P_1 \oplus P_2$:

$$C_1 = P_1 \oplus K \quad C_2 = P_2 \oplus K \quad \rightarrow \quad C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

Figure 5.18 *Diagram for Example 5.13*

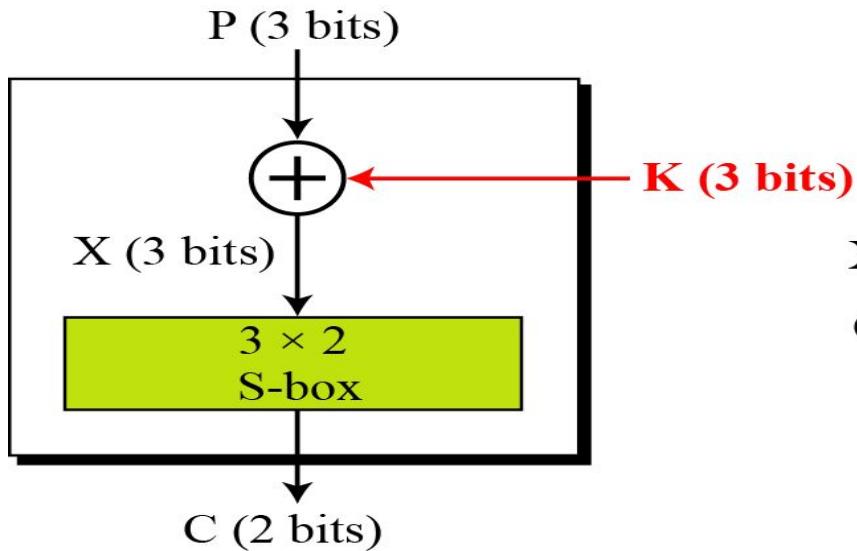


5.1.6 Continued

Example 5.14

We add one S-box to Example 5.13, as shown in Figure 5.19.

Figure 5.19 *Diagram for Example 5.14*



X	000	001	010	011	100	101	110	111
C	11	00	10	10	01	00	11	00

S-box table

The existence of the S-box prevents Eve from finding a definite relationship between the plain text differences and the ciphertext differences. Eve can create a probabilistic relationship.

5.1.6 *Continued*

Example 5.14 Continued

Eve now can create a probabilistic relationship as shown in Table 5.4.

The below table shows that for each plaintext difference, how many ciphertext differences the cipher may create.

Table 5.4 *Differential input/output*

		$C_1 \oplus C_2$			
		00	01	10	11
P ₁ ⊕ P ₂		000	8		
001		2	2		4
010		2	2	4	
011			4	2	2
100		2	2	4	
101			4	2	2
110		4		2	2
111				2	6

5.1.6 *Continued*

Example 5.15

The heuristic result of Example 5.14 can create probabilistic information for Eve as shown in Table 5.5.

Table 5.5 *Differential distribution table*

		$C_1 \oplus C_2$			
		00	01	10	11
P ₁ ⊕ P ₂		000	1	0	0
001	010	0.25	0.25	0	0.50
011	100	0.25	0.25	0.50	0
101	110	0	0.50	0.25	0.25
111	111	0.50	0	0.25	0.75

5.1.6 *Continued*

Example 5.16

Looking at Table 5.5, Eve knows that if $P_1 \oplus P_2 = 001$, then $C_1 \oplus C_2 = 11$ with the probability of 0.50 (50 percent). She tries $C_1 = 00$ and gets $P_1 = 010$ (chosen-ciphertext attack). She also tries $C_2 = 11$ and gets $P_2 = 011$ (another chosen-ciphertext attack). Now she tries to work backward, based on the first pair, P_1 and C_1 ,

$$C_1 = 00 \rightarrow X_1 = 001 \text{ or } X_1 = 111$$

$$\text{If } X_1 = 001 \rightarrow K = X_1 \oplus P_1 = 011$$

$$\text{If } X_1 = 111 \rightarrow K = X_1 \oplus P_1 = 101$$

$$C_2 = 11 \rightarrow X_2 = 000 \text{ or } X_2 = 110$$

$$\text{If } X_2 = 000 \rightarrow K = X_2 \oplus P_2 = 011$$

$$\text{If } X_2 = 110 \rightarrow K = X_2 \oplus P_2 = 101$$

The two tests confirm that $K = 011$ or $K = 101$.

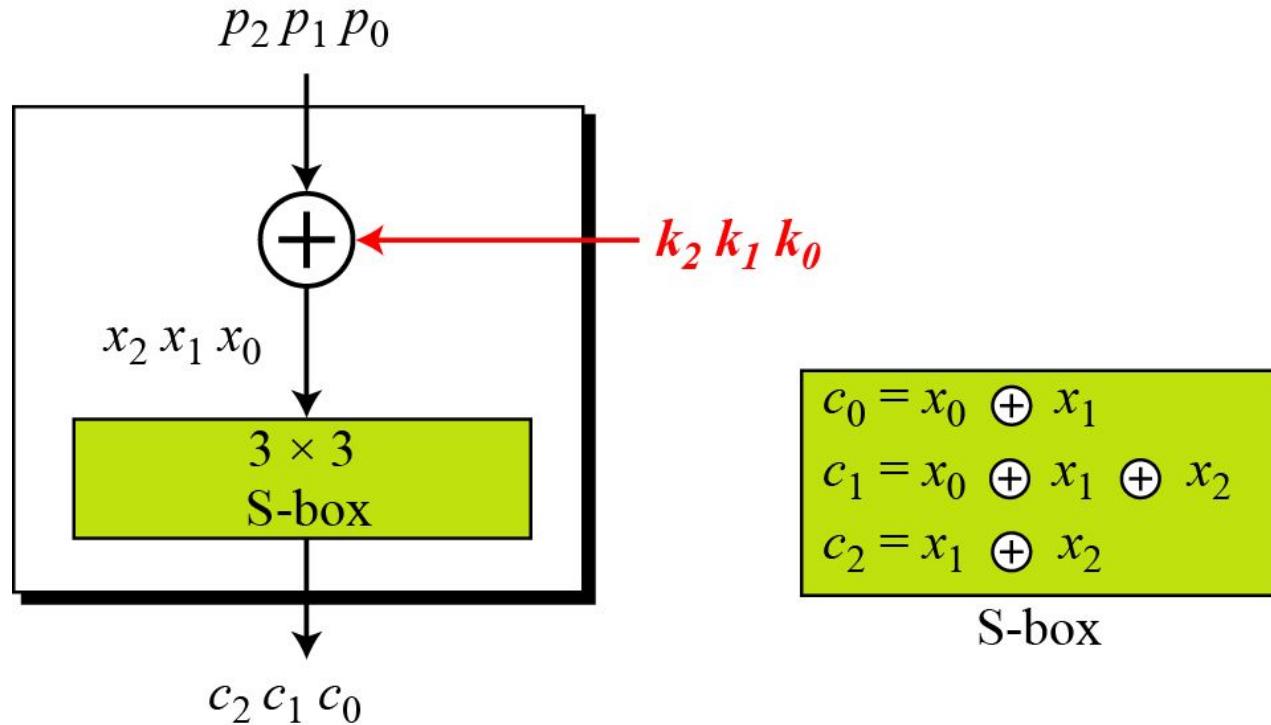
5.1.6 Continued

Linear Cryptanalysis

Linear cryptanalysis was presented by Mitsuru Matsui in 1993. The analysis uses known plaintext attacks.

5.1.6 Continued

Figure 5.20 A simple cipher with a linear S-box



5.1.6 Continued

$$c_0 = p_0 \oplus k_0 \oplus p_1 \oplus k_1$$

$$c_1 = p_0 \oplus k_0 \oplus p_1 \oplus k_1 \oplus p_2 \oplus k_2$$

$$c_2 = p_1 \oplus k_1 \oplus p_2 \oplus k_2$$

Solving for three unknowns, we get.

$$k_1 = (p_1) \oplus (c_0 \oplus c_1 \oplus c_2)$$

$$k_2 = (p_2) \oplus (c_0 \oplus c_1)$$

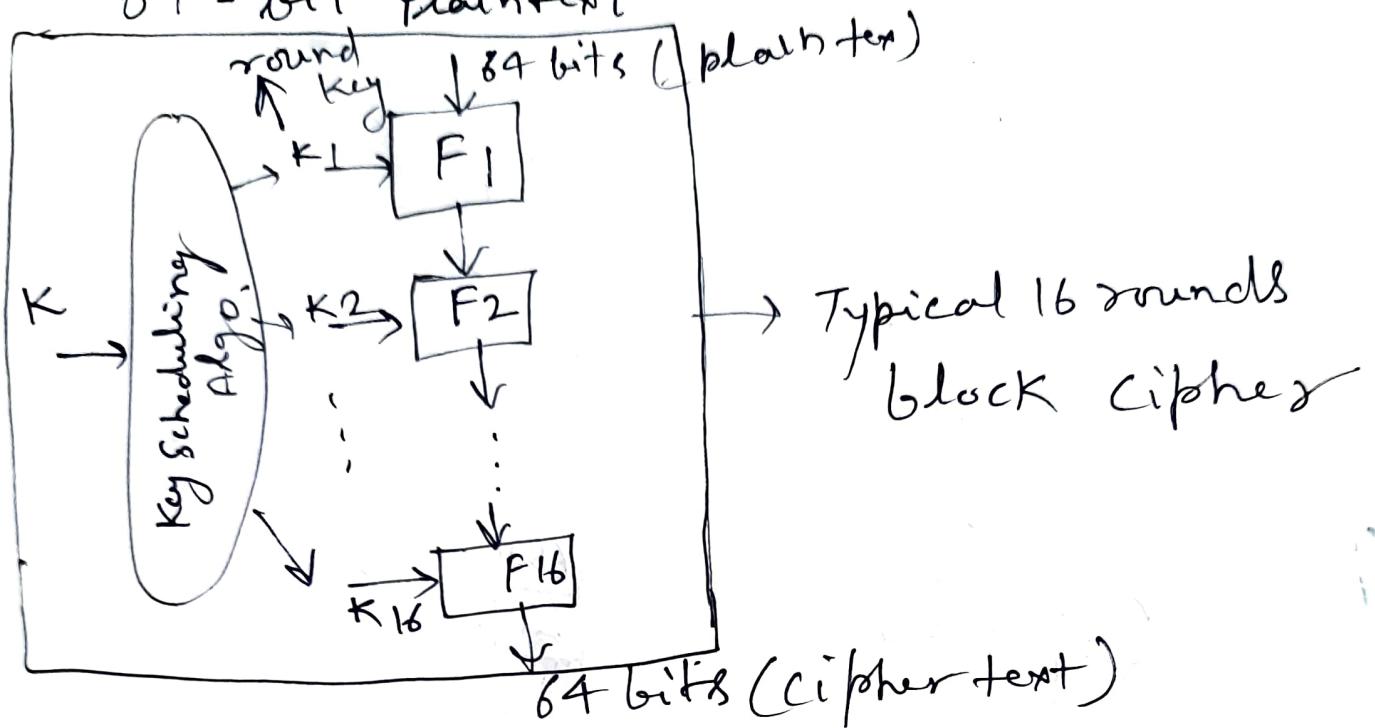
$$k_0 = (p_0) \oplus (c_1 \oplus c_2)$$

This means that three known-plaintext attacks can find the values of k_0 , k_1 , and k_2 .

DES

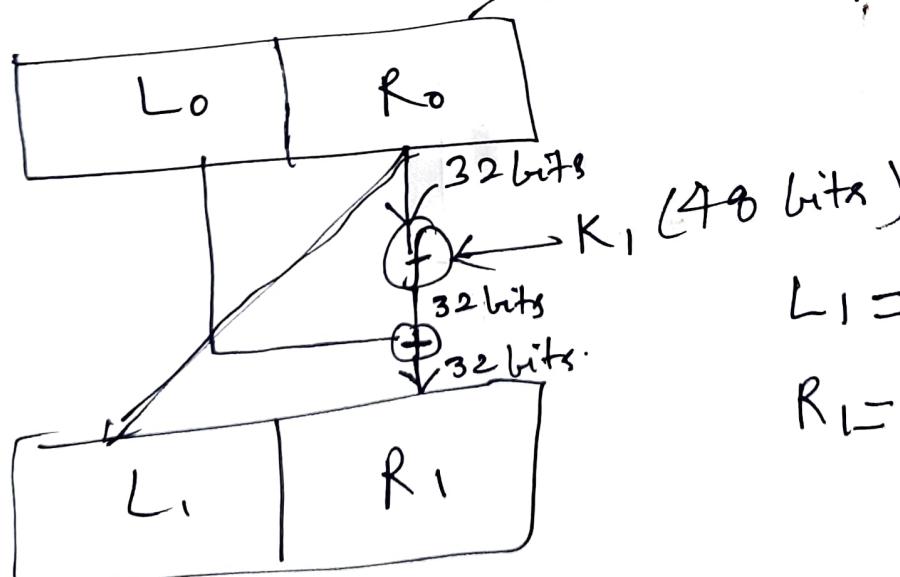
No. of Rounds $\Rightarrow 16$

64-bit plaintext

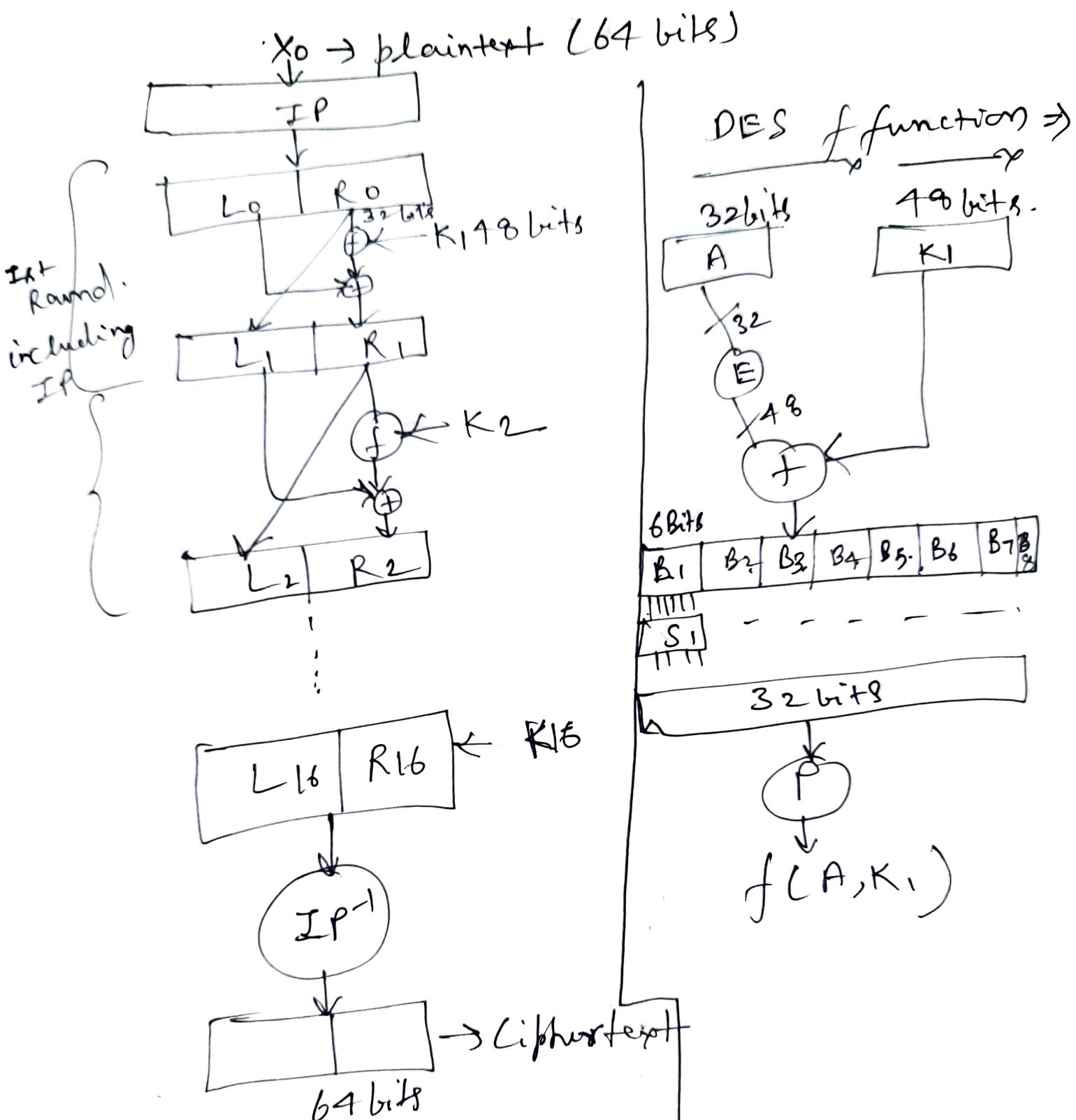


DES Round Function : it's called Feistel cipher

Feistel cipher \rightarrow 64 bits input

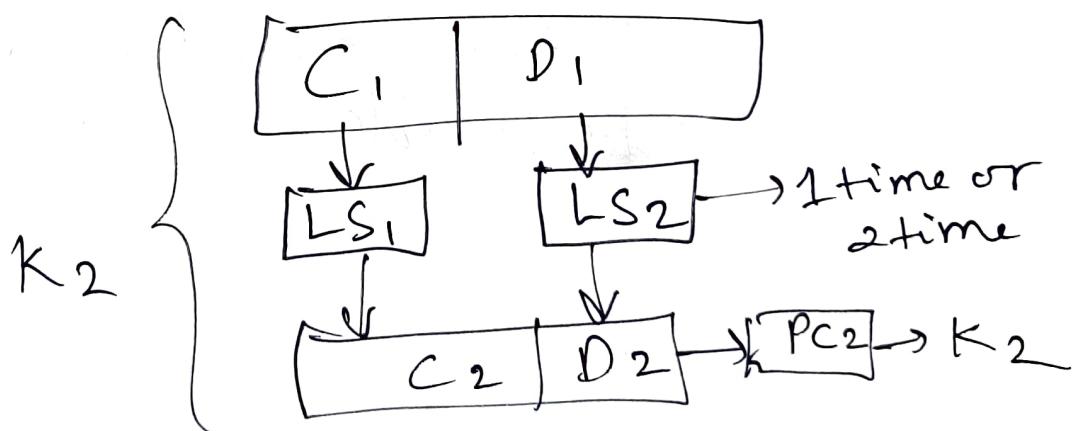
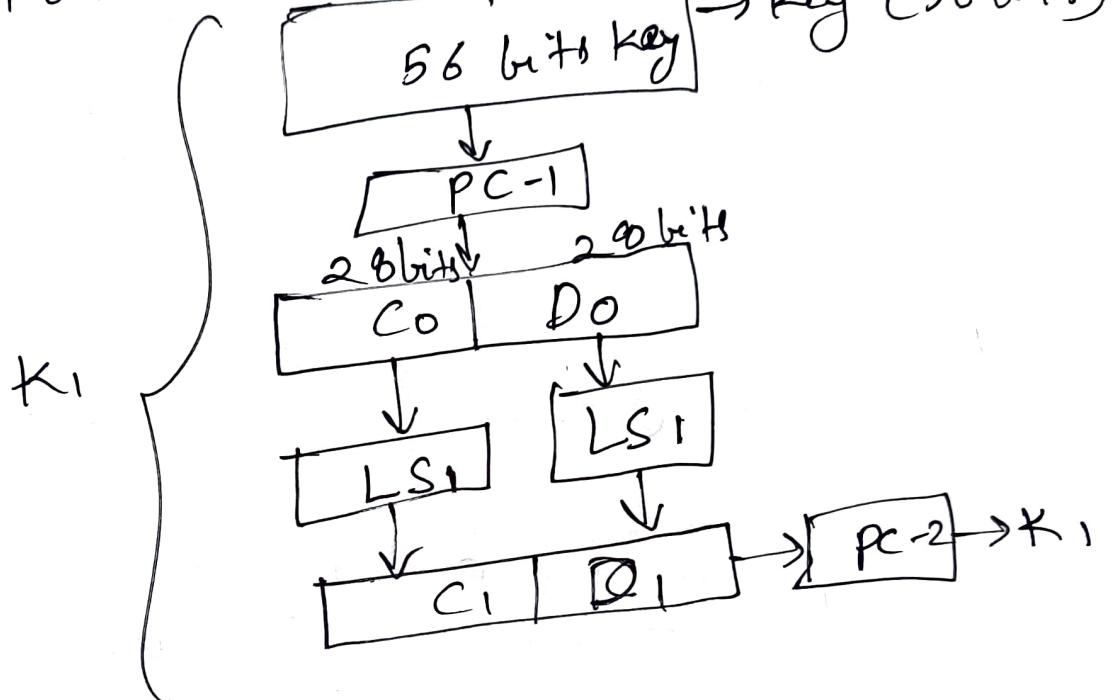
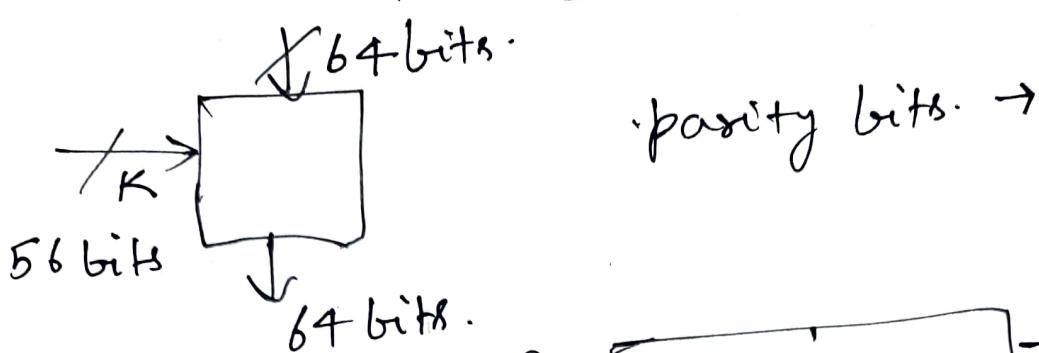


each round of DES is basically Feistel function.



(3)

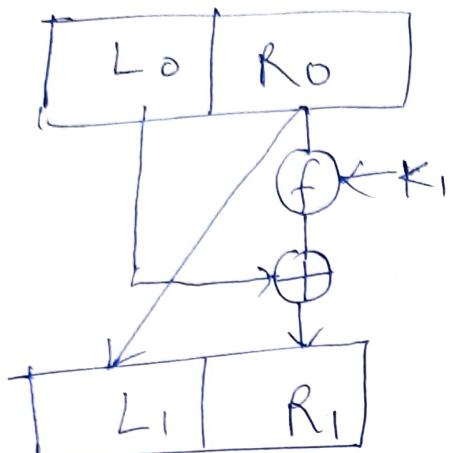
Key Scheduling for DES

 K_{16}

⊕

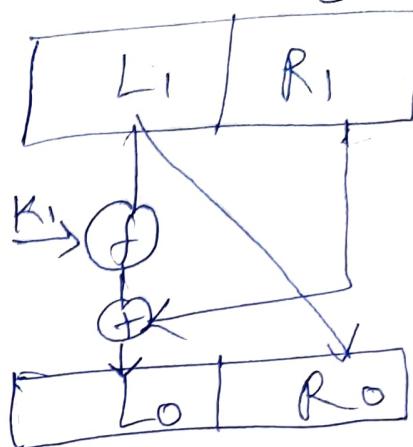
Convert Feistel cipher?

For Decryption

For Encryption (One Round)

$$L_0 = R_0$$

$$R_1 = f(R_0, K_1) \oplus L_0$$

For Decryption (One Round)

$$R_0 = L_1$$

$$L_0 = R_1 \oplus f(R_0, K_1)$$

Beauty: we are not inventing f , we are applying same f in Decryption,
 Because sometimes inverse may not be possible, we have S box in f .

How secure DES is?

Cryptanalysis on DESIssues: (i) ~~Security~~ It is not well documented.

For Example S-box, they have given just a table:

0	1	2	3	4	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

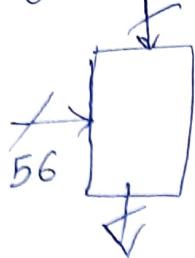
We have to do the table lookup.

But how values are coming there is no proper operation for that.

0	1	2	3	4	-	-	-	-	-	-	-	-	-	-	-
0	1	2	3	4	-	-	-	-	-	-	-	-	-	-	-

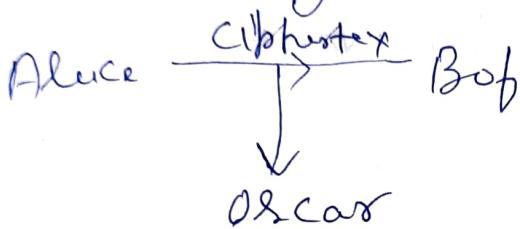
 4×16

(i) key size:



Attack Model

(i) Ciphertext only attack:



(ii) Known plaintext attack:

attacker has some plaintext & corresponding ciphertext $\langle p_i, c_i \rangle \quad i=1, \dots, n$

goal:- to get the k

- to guess new plaintext p^* from c^*
temporary

(iii) Chosen Plaintext Attack: (^{giving access to encryption system})

Attacker will choose some plaintext & get some ciphertext

goal:- get the key

- or guess the p^* from c^*

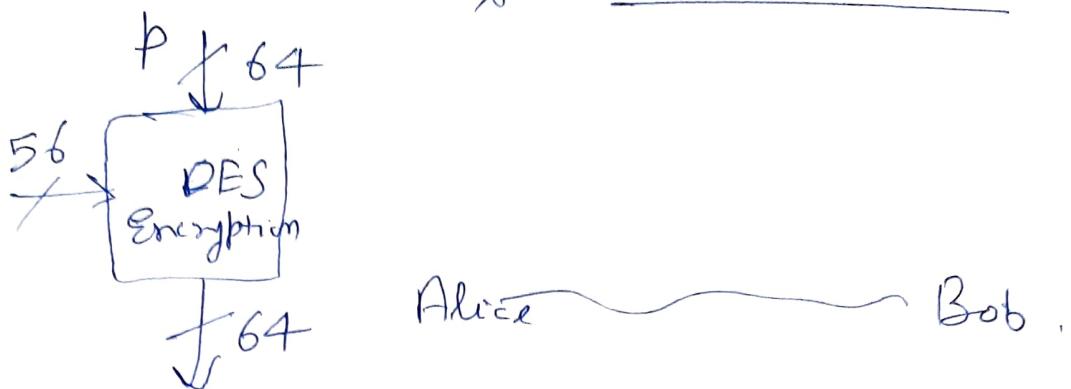
(iv) Chosen ciphertext Attack: (^{temporarily giving access to decryption system})

goal:- get the key

to guess new plaintext p^* from c^*

⑥

Exhaustive Search Attack on DES



$$|K| = 56$$

other is basically a known plaintext attack;

If Attacker knows P & its corresponding C .
 $\langle P, C \rangle$

- Attacker wants to guess the key

$$K = 56$$

$$\text{Size of key space} = 2^{56}$$

What attacker will do?

Attacker will try for all possible key.

To decrypt the C .

~~try~~ match k_i with P ---

Time complexity = ?

Time = $2^{56} \times$ (Time required for
DES decryption or
↓ encryption)

$$= 2^{56} \text{ sec.} \quad 1 \text{ sec.}$$

If two comb. involved then time will be reduced by
 $\frac{2^{56}}{2} = 2^{55}$

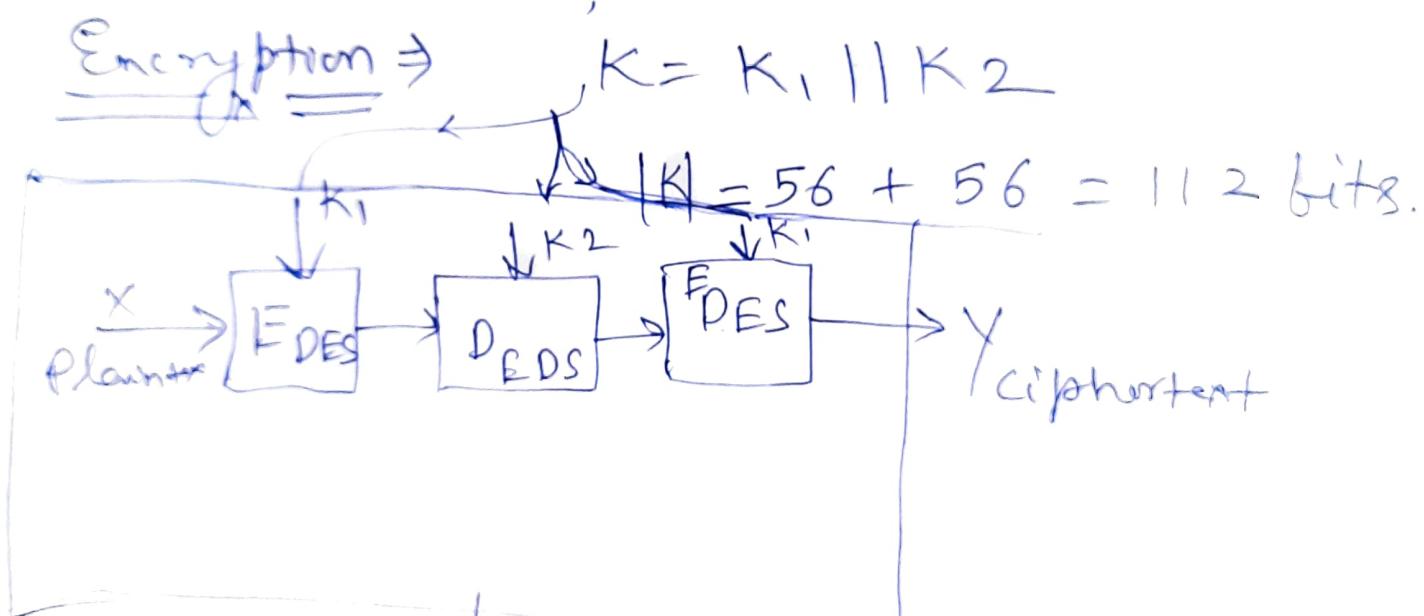
7

Exhaustive Search attack is known as generic attack, because attacker is not interested to know the internal structure of DES, only to know key size.

 X X X

Triple DES

Replacement of DES?

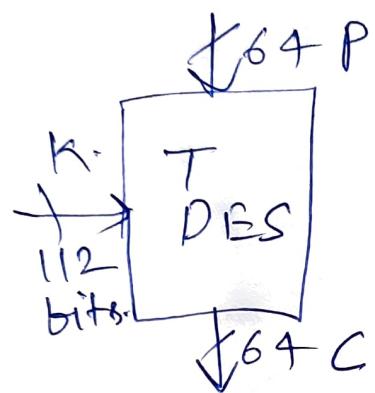
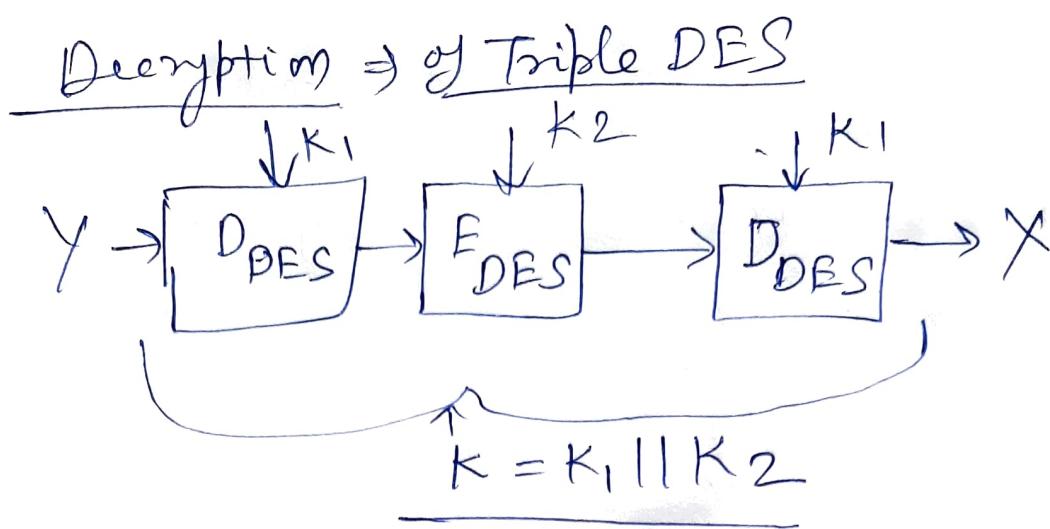


\rightarrow Encryption of triple DES

No. of Rounds = 48 rounds

Major Drawback
 \hookrightarrow Huge time required.

8



NETWORK SECURITY

Network security is any activity designed to protect the usability and integrity of your network and data. It includes both hardware and software technologies.

Effective network security manages access to the network. It targets a variety of threats and stops them from entering or spreading on your network.

NETWORK SECURITY

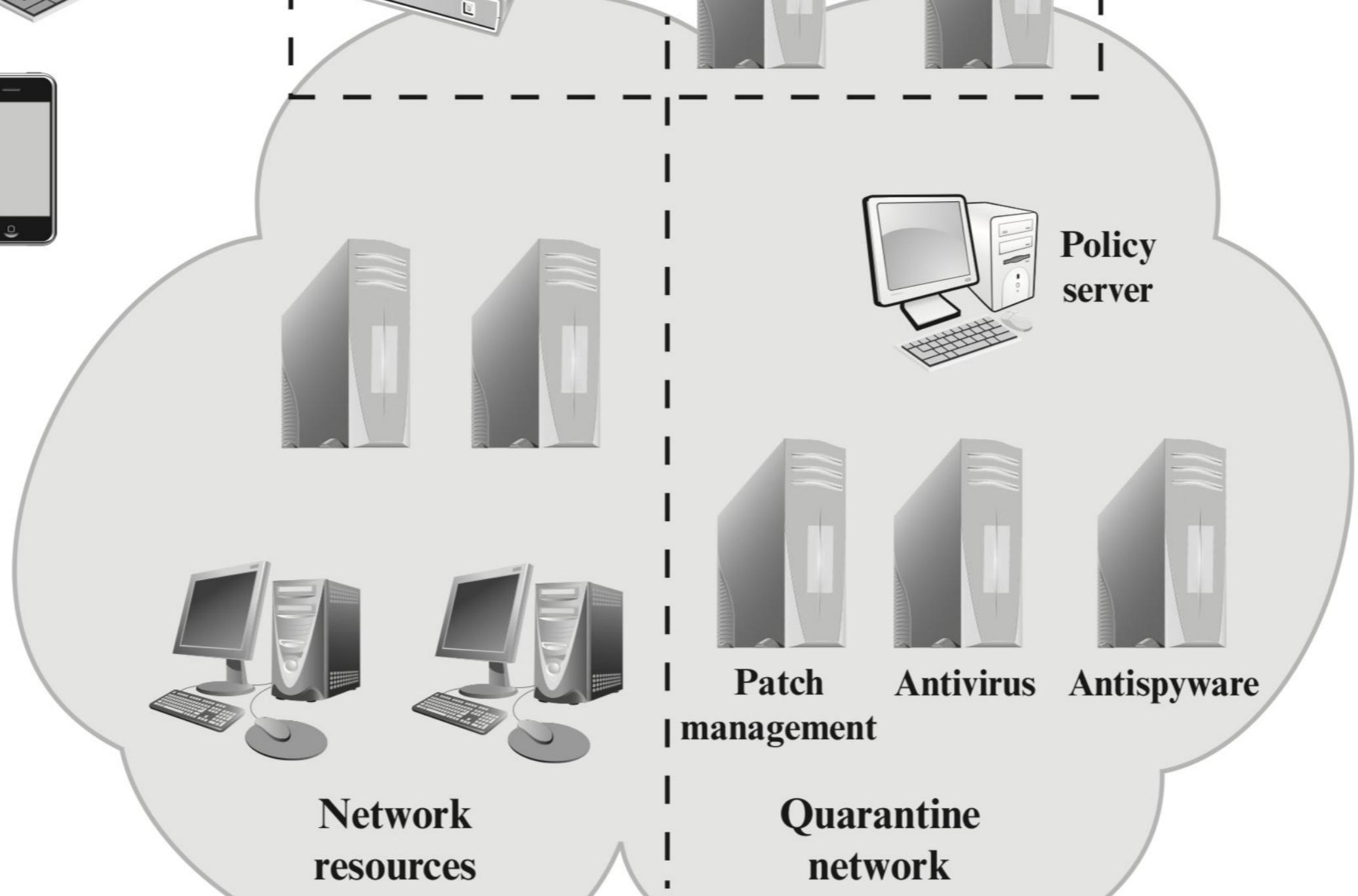
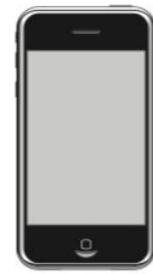
NETWORK ACCESS CONTROL (NAC)

NAC authenticates users logging into the network and determines what data they can access and actions they can perform

Elements of a Network Access Control System:

- **Access requestor (AR):** The AR is the node that is attempting to access the network and may be any device that is managed by the NAC system, including workstations, servers, printers, cameras, and other IP-enabled devices. ARs are also referred to as **suplicants**, or simply, clients.
- **Policy server:** Based on the AR's posture and an enterprise's defined policy, the policy server determines what access should be granted. The policy server often relies on backend systems, including antivirus, patch management, or a user directory, to help determine the host's condition.

- **Network access server (NAS):** The NAS functions as an access control point for users in remote locations connecting to an enterprise's internal network. Also called a **media gateway**, a **remote access server (RAS)**, or a **policy server**, an NAS may include its own authentication services or rely on a separate authentication service from the policy server.



A variety of different ARs seek access to an enterprise network by applying to some type of NAS.

The first step is generally to authenticate the AR.

Authentication typically involves some sort of secure protocol and the use of cryptographic keys.

Authentication may be performed by the NAS, or the NAS may mediate the authentication process.

In the latter case, authentication takes place between the supplicant and an authentication server that is part of the policy server or that is accessed by the policy server.

The authentication process serves a number of purposes.

It verifies a **supplicant's claimed identity**, which enables the policy server to determine what access privileges, if any, the AR may have.

The authentication exchange may result in the **establishment of session keys** to enable future secure communication between the supplicant and resources on the enterprise network.

Typically, the policy server or a supporting server will perform **checks on the AR** to determine if it should be **permitted interactive remote access connectivity**.

These checks sometimes called health, suitability, screening, or assessment checks require software on the user's system to verify compliance with certain requirements from the organization's secure configuration baseline.

For example, the user's antimalware software must be up-to-date, the operating system must be fully patched, and the remote computer must be owned and controlled by the organization.

These **checks should be performed before granting the AR access** to the enterprise network.

Based on the results of these checks, the organization can determine whether the remote computer should be permitted to use interactive remote access.

If the user has acceptable authorization credentials but the remote computer does not pass the health check, the user and remote computer should be denied network access or have limited access to network.

Once an AR has been authenticated and cleared for a certain level of access to the enterprise network, the NAS can enable the AR to interact with resources in the enterprise network.

The NAS may mediate every exchange to enforce a security policy for this AR, or may use other methods to limit the privileges of the AR.

Network Access Enforcement Methods

ENFORCEMENT METHODS ARE THE ACTIONS THAT ARE APPLIED TO AR TO REGULATE ACCESS TO THE ENTERPRISE NETWORK.

- **IEEE 802.1X** :This is a link layer protocol that enforces authorization before a port is assigned an IP address. IEEE 802.1X makes use of the Extensible Authentication Protocol for the authentication process.
- **Virtual local area networks (VLANs)** the enterprise network consisting of an interconnected set of LANs, is segmented logically into a number of virtual LANs. The NAC system decides to which of the network's VLANs it will direct an AR, based on whether the device needs security remediation, Internet access only, or some level of network access to enterprise resources. VLANs can be created dynamically and VLAN membership, of both enterprise servers and ARs, may overlap. That is, an enterprise server or an AR may belong to more than one VLAN.

- Firewall: A firewall provides a form of NAC by allowing or denying network traffic between an enterprise host and an external user.
- DHCP management: The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol that enables dynamic allocation of IP addresses to hosts. A DHCP server intercepts DHCP requests and assigns IP addresses instead. Thus, NAC enforcement occurs at the IP layer based on subnet and IP assignment. A DCHP server is easy to install and configure, but is subject to various forms of IP spoofing, providing limited security.

WEB SECURITY

businesses are enthusiastic about setting up facilities on the Web for electronic commerce.

But the reality is that the Internet and the Web are extremely vulnerable to compromises of various sorts.

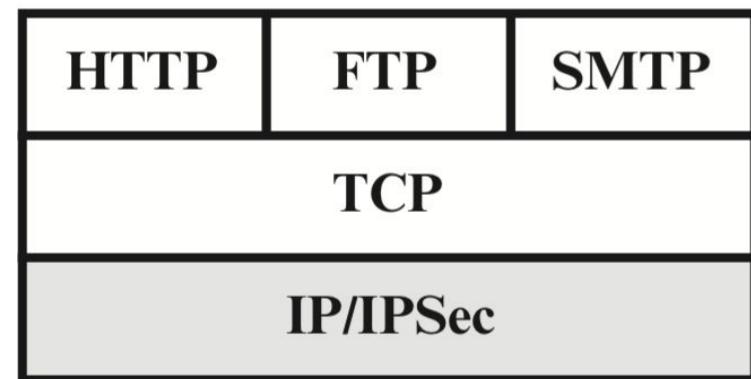
As businesses wake up to this reality, the demand for secure Web services grows

The World Wide Web is fundamentally a client/server application running over the Internet

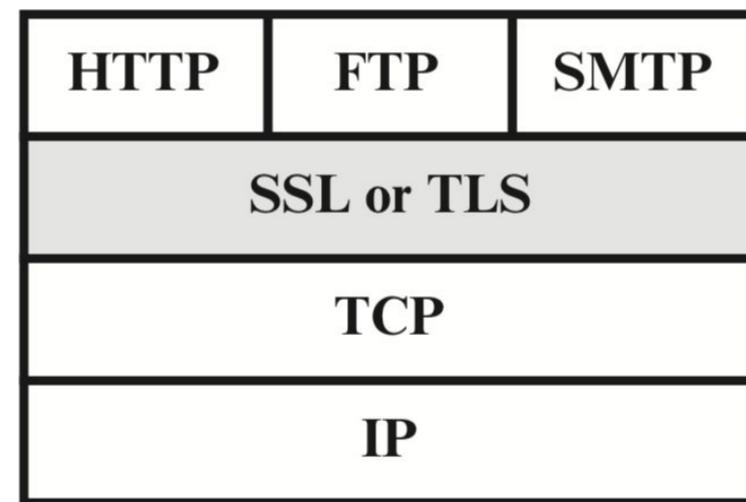
A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data

Table 17.1 A Comparison of Threats on the Web

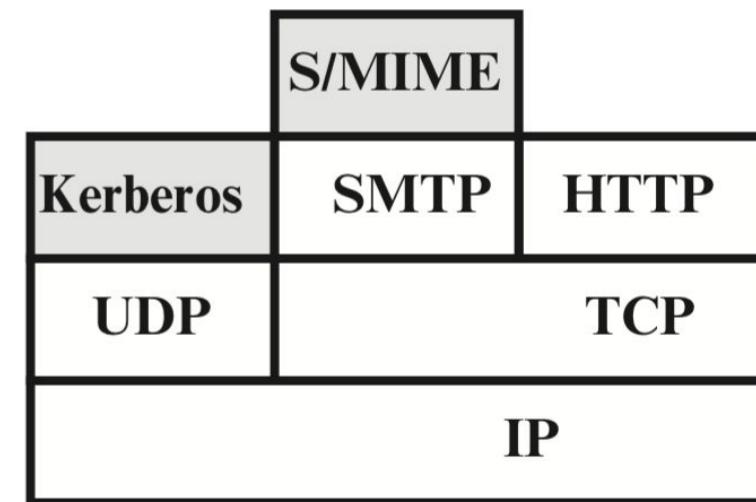
	Threats	Consequences	Countermeasures
Integrity	<ul style="list-style-type: none">• Modification of user data• Trojan horse browser• Modification of memory• Modification of message traffic in transit	<ul style="list-style-type: none">• Loss of information• Compromise of machine• Vulnerability to all other threats	Cryptographic checksums
Confidentiality	<ul style="list-style-type: none">• Eavesdropping on the net• Theft of info from server• Theft of data from client• Info about network configuration• Info about which client talks to server	<ul style="list-style-type: none">• Loss of information• Loss of privacy	Encryption, Web proxies
Denial of Service	<ul style="list-style-type: none">• Killing of user threads• Flooding machine with bogus requests• Filling up disk or memory• Isolating machine by DNS attacks	<ul style="list-style-type: none">• Disruptive• Annoying• Prevent user from getting work done	Difficult to prevent
Authentication	<ul style="list-style-type: none">• Impersonation of legitimate users• Data forgery	<ul style="list-style-type: none">• Misrepresentation of user• Belief that false information is valid	Cryptographic techniques



(a) Network level



(b) Transport level



(c) Application level

Figure 17.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

TRANSPORT LAYER SECURITY

At this level, there are two implementation choices.

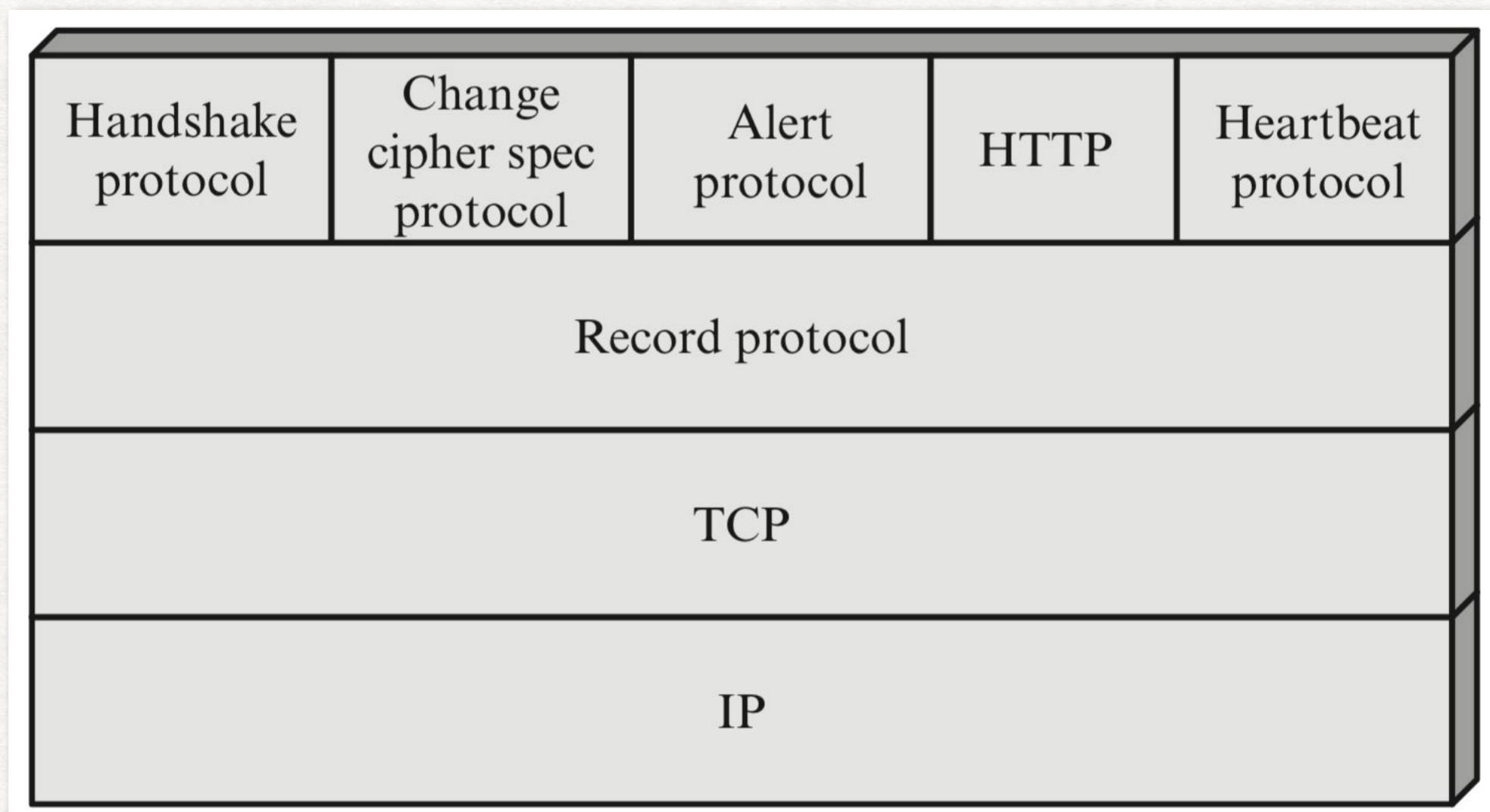
For full generality, TLS could be provided as part of the underlying protocol suite and therefore be transparent to applications.

Alternatively, TLS can be embedded in specific packages.

For example, most browsers come equipped with TLS, and most Web servers have implemented the protocol.

TLS ARCHITECTURE

TLS is designed to make use of TCP to provide a reliable end-to-end secure service. TLS is not a single protocol but rather two layers of protocols



The TLS Record Protocol provides basic security services to various higher-layer protocols.

In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of TLS.

Three higher-layer protocols are defined as part of TLS: the Handshake Protocol; the Change Cipher Spec Protocol; and the Alert Protocol.

TWO IMPORTANT TLS CONCEPTS ARE THE TLS SESSION AND THE TLS CONNECTION

- **CONNECTION:** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For TLS, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.
- **SESSION:** A TLS session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

TLS RECORD PROTOCOL

- The TLS Record Protocol provides two services for TLS connections:
 - Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of TLS payloads.
 - Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

- The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled before being delivered to higher-level users.

HMAC is defined as

$$\text{HMAC}_K(M) = \text{H}[(K^+ \oplus \text{opad}) \parallel \text{H}[(K^+ \oplus \text{ipad}) \parallel M]]$$

where

H = embedded hash function (for TLS, either MD5 or SHA-1)

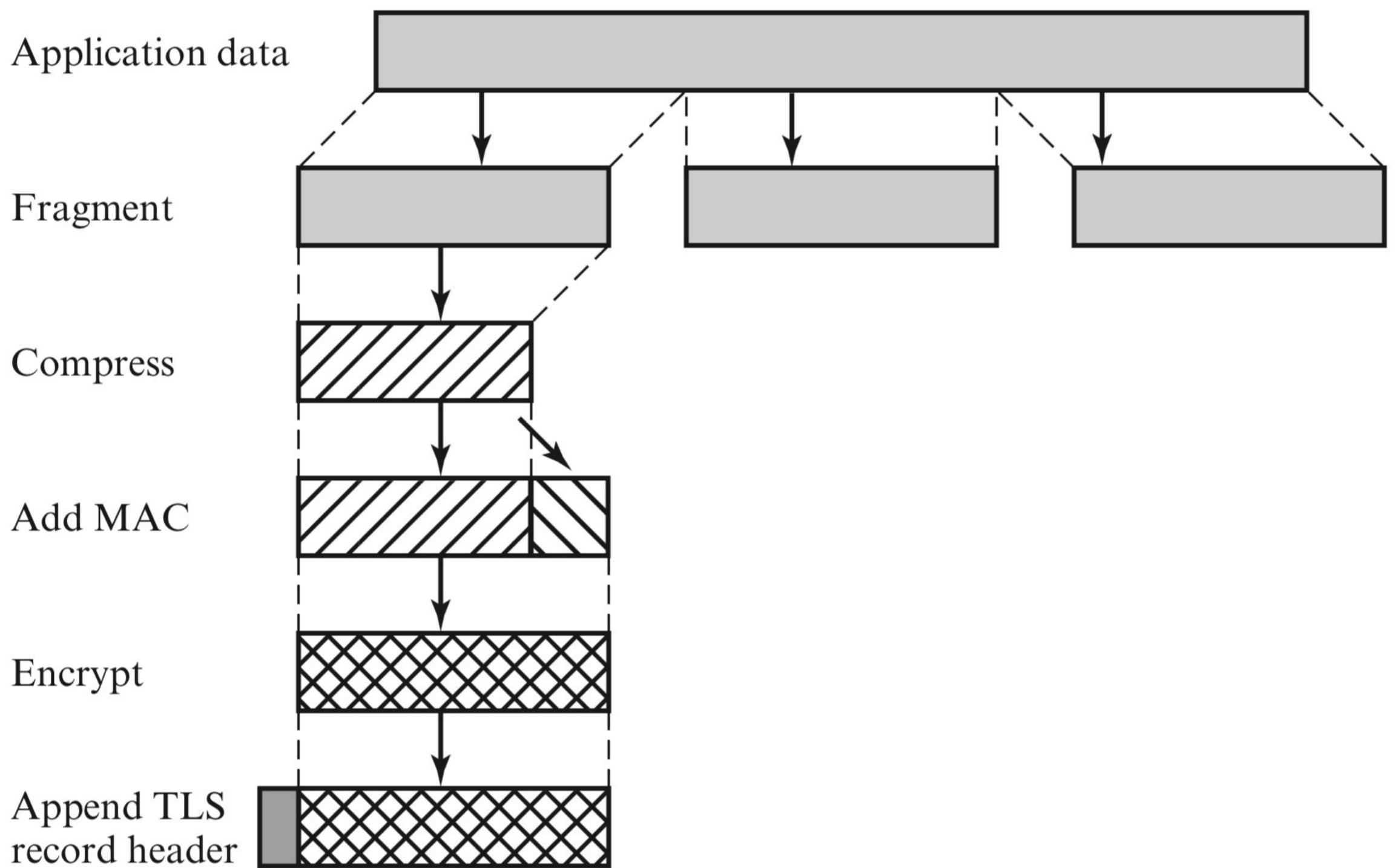
M = message input to HMAC

K^+ = secret key padded with zeros on the left so that the result is equal to the block length of the hash code (for MD5 and SHA-1, block length = 512 bits)

ipad = 00110110 (36 in hexadecimal) repeated 64 times (512 bits)

opad = 01011100 (5C in hexadecimal) repeated 64 times (512 bits)

TLS Record Protocol Operation



For stream encryption, the compressed message plus the MAC are encrypted. Note that the MAC is computed before encryption takes place and that the MAC is then encrypted along with the plaintext or compressed plaintext.

For block encryption, padding may be added after the MAC prior to encryption. The padding is in the form of a number of padding bytes followed by a one- byte indication of the length of the padding.

Change Cipher Spec Protocol

This protocol consists of a single message which consists of a single byte with the value 1.

The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

ALERT PROTOCOL

The Alert Protocol is used to convey TLS-related alerts to the peer entity. As with other applications that use TLS, alert messages are compressed and encrypted, as specified by the current state.

Each message in this protocol consists of two bytes. The first byte takes the value warning (1) or fatal (2) to convey the severity of the message.

If the level is fatal, TLS immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert.

The following alerts are always fatal:

- **unexpected_message:** An inappropriate message was received.
- **bad_record_mac:** An incorrect MAC was received.
- **decompression_failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).
- **handshake_failure:** Sender was unable to negotiate an acceptable set of security parameters given the options available.
- **illegal_parameter:** A field in a handshake message was out of range or inconsistent with other fields.

1 byte



(a) Change Cipher Spec Protocol

1 byte



3 bytes

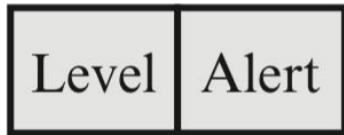


≥ 0 bytes



(c) Handshake Protocol

1 byte 1 byte



(b) Alert Protocol

≥ 1 byte

Opaque content

(d) Other Upper-Layer Protocol (e.g., HTTP)

HANDSHAKE PROTOCOL

The most complex part of TLS is the Handshake Protocol.

This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in a TLS record.

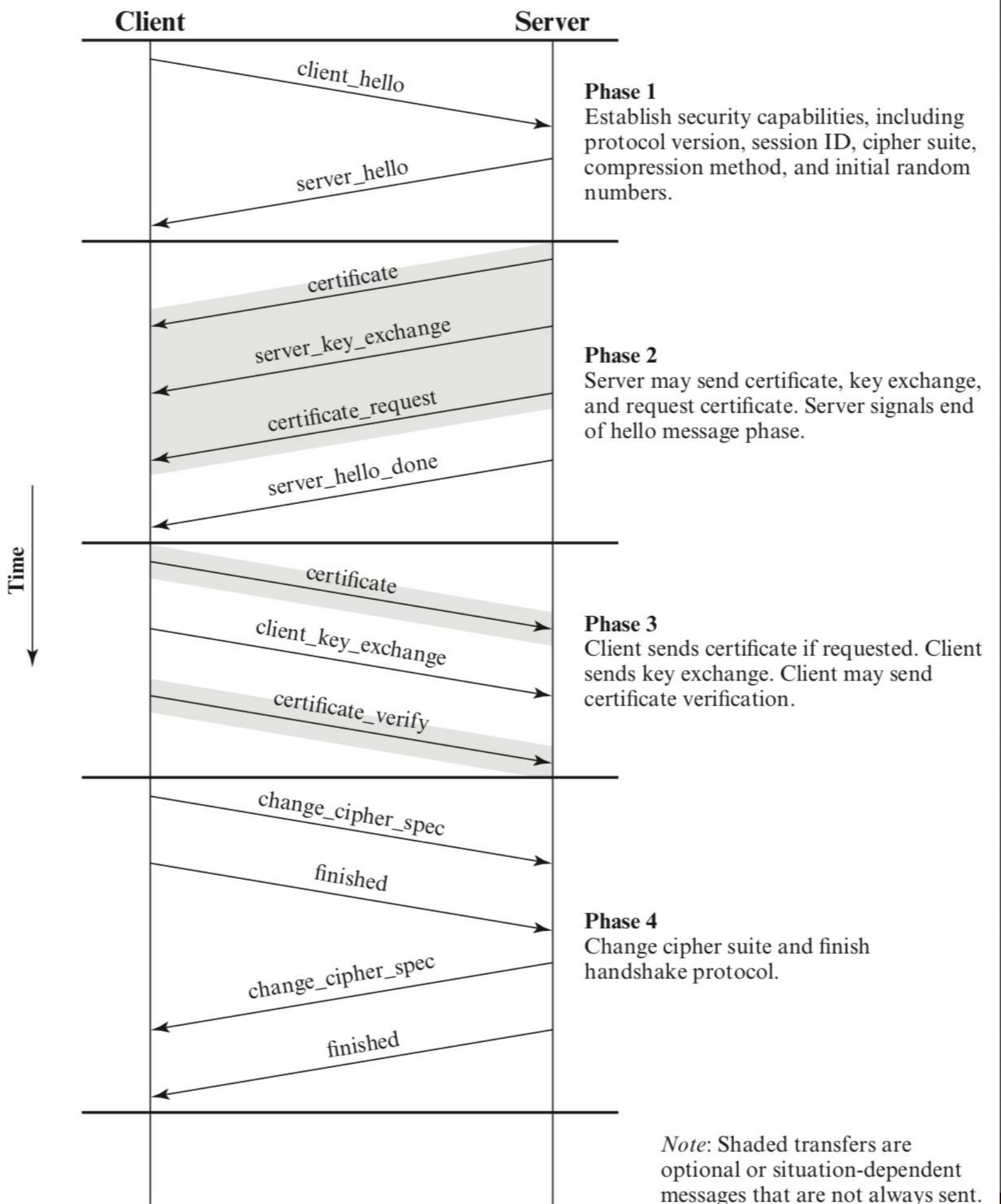
The Handshake Protocol is used before any application data is transmitted.

The Handshake Protocol consists of a series of messages exchanged by client and server.

- PHASE 1. ESTABLISH SECURITY CAPABILITIES
- Phase 1 initiates a logical connection and establishes the security capabilities that will be associated with it. The exchange is initiated by the client, which sends a `client_hello` message with the following parameters:
 - **Version:** The highest TLS version understood by the client.
 - **Random:** A client-generated random structure consisting of a 32-bit timestamp and 28 bytes generated by a secure random number generator. These values serve as nonces and are used during key exchange to prevent replay attacks.
 - **Session ID:** A variable-length session identifier. A nonzero value indicates that the client wishes to update the parameters of an existing connection or to create a new connection on this session. A zero value indicates that the client wishes to establish a new connection on a new session.

Table 17.2 TLS Handshake Protocol Message Types

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value



- CipherSuite: This is a list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference. Each element of the list (each cipher suite) defines both a key exchange algorithm
- Compression Method: This is a list of the compression methods the client supports.

HEARTBEAT PROTOCOL

In the context of computer networks, a heartbeat is a periodic signal generated by hardware or software to indicate normal operation or to synchronize other parts of a system. A heartbeat protocol is typically used to monitor the availability of a protocol entity. In the specific case of TLS, a Heartbeat protocol was defined in 2012.

The Heartbeat protocol runs on top of the TLS Record Protocol and consists of two message types: `heartbeat_request` and `heartbeat_response`. The use of the Heartbeat protocol is established during Phase 1 of the Handshake protocol.

Each peer indicates whether it supports heartbeats. If heartbeats are supported, the peer indicates whether it is willing to receive `heartbeat_request` messages and respond with `heartbeat_response` messages or only willing to send `heartbeat_request` messages.

A `heartbeat_request` message can be sent at any time. Whenever a `request` message is received, it should be answered promptly with a corresponding `heartbeat_response` message.

HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. The HTTPS capability is built into all modern Web browsers. Its use depends on the Web server supporting HTTPS communication. For example, some search engines do not support HTTPS.

The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://. A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which invokes SSL.

When HTTPS is used, the following elements of the communication are encrypted:

- URL of the requested document
- Contents of the document
- Contents of browser forms (filled in by browser user)
- Cookies sent from browser to server and from server to browser
- Contents of HTTP header

HTTPS is documented in RFC 2818, *HTTP Over TLS*. There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

Connection Initiation

The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake. When the TLS handshake has finished, the client may then initiate the first HTTP request.

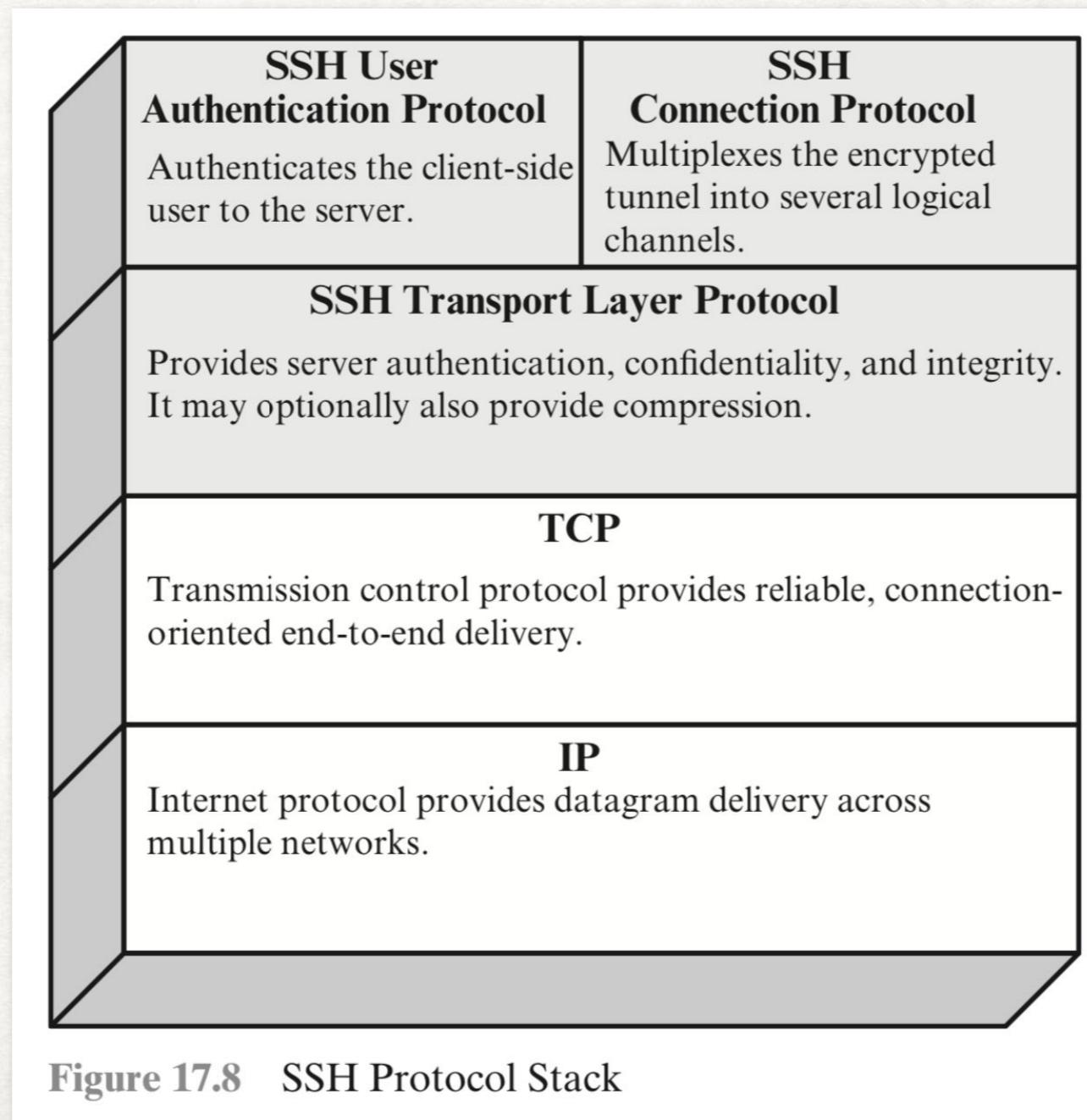
Connection Closure

An HTTP client or server can indicate the closing of a connection by including the following line in an HTTP record: Connection: close. This indicates that the connection will be closed after this record is delivered.

.

SECURE SHELL (SSH)

The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. SSH also provides a more general client/server capability and can be used for such network functions as file transfer



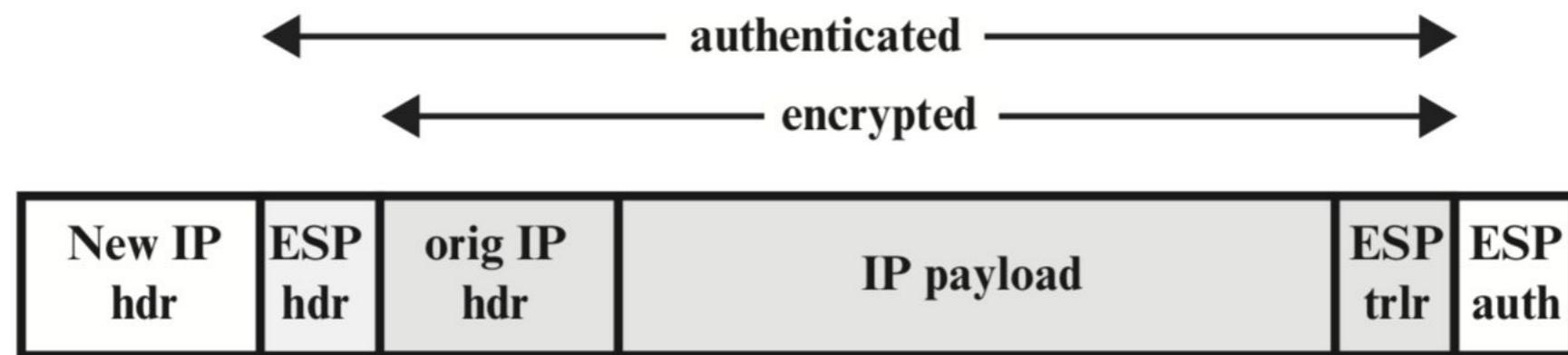
IP SEC

an enterprise can run a secure, private IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises

IP-level security encompasses three functional areas: authentication, confidentiality, and key management.

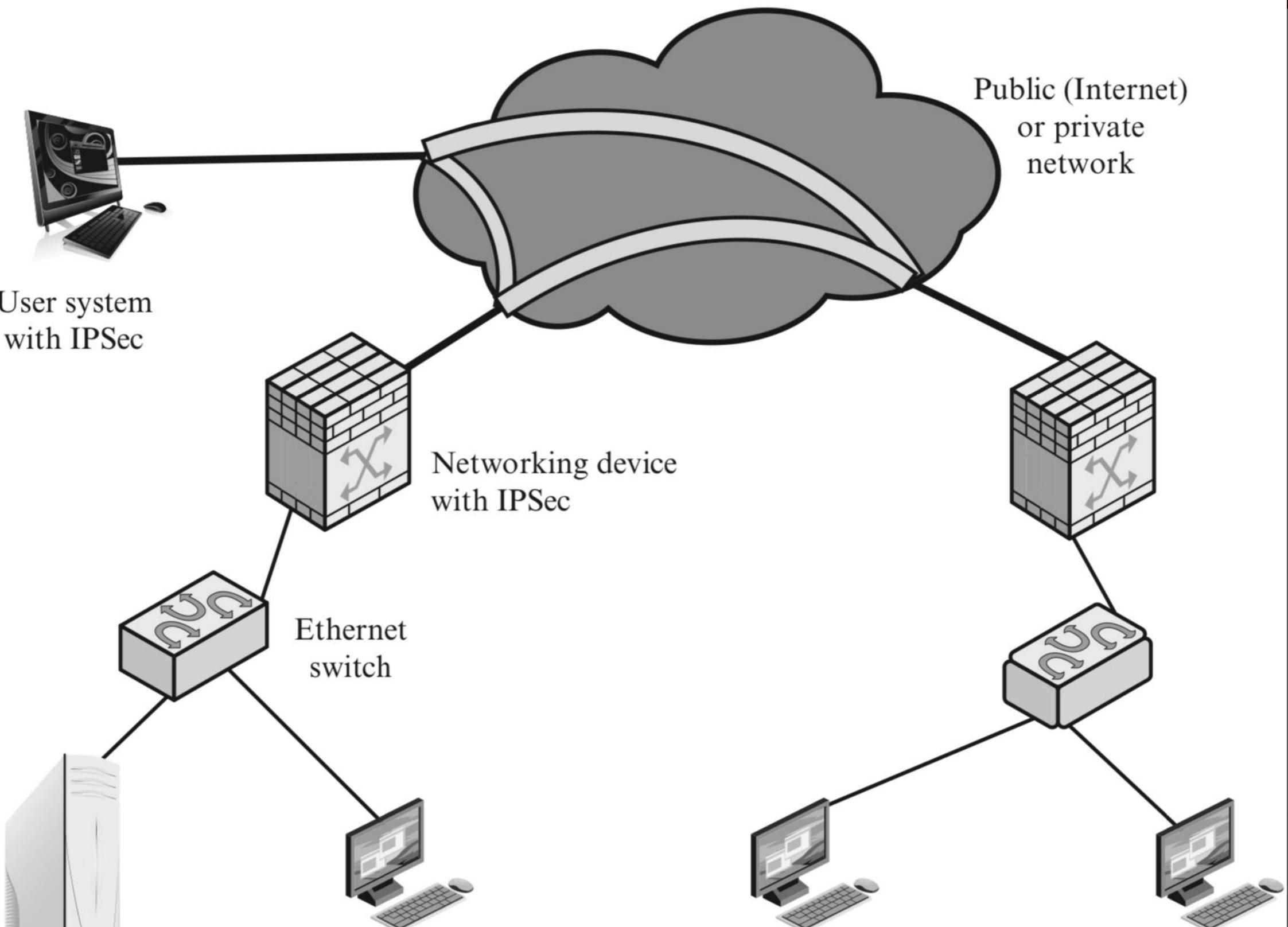
The authentication mechanism assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit. The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The key management facility is concerned with the secure exchange of keys.

- The principal feature of IPsec that enables it to support these varied applications is that it can encrypt and/or authenticate **all** traffic at the IP level



(a) Tunnel-mode format

- Figure (a) shows a simplified packet format for an IPsec option known as tunnel mode, described subsequently. Tunnel mode makes use of an IPsec function, a combined authentication/encryption function called Encapsulating Security Payload (ESP), and a key exchange function
- Figure (b) is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used. These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPsec networking device will typically encrypt all traffic going into the WAN and decrypt traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN



Legend:

Unprotected
IP traffic

IP traffic
protected
by IPsec

Virtual tunnel:
protected
by IPsec

BENEFITS OF IPSEC

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the **overhead of security-related processing**.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the **only means of entrance from the Internet** into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is **transparent** to applications. There is no need to change software on a user or server system when IPsec is implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users.

ROUTING APPLICATIONS

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbour advertisement (a router seeks to establish or maintain a neighbor relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial IP packet was sent.
- A routing update is not forged.

IPSEC ENCOMPASSES THREE FUNCTIONAL AREAS: AUTHENTICATION, CONFIDENTIALITY, AND KEY MANAGEMENT

- Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, Authentication Header (AH); and a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP)
- **Authentication Header (AH):** AH is an extension header to provide message authentication. Because message authentication is provided by ESP, the use of AH is deprecated.
- **Encapsulating Security Payload (ESP):** ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/ authentication.
lists the following IP Sec services:
 - Access control
 - Connectionless integrity
 - Data origin authentication
 - Rejection of replayed packets (a form of partial sequence integrity)
 - Confidentiality (encryption)
 - Limited traffic flow confidentiality

TRANSPORT AND TUNNEL MODES

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.¹ Examples include a TCP or UDP segment or an ICMP packet, all of which operate directly above IP in a host protocol stack.

Typically, transport mode is used for end-to-end communication between two hosts (e.g., a client and a server, or two workstations). When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header. For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.

ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header.

TUNNEL MODE

Tunnel mode provides protection to the entire IP packet.

To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new outer IP packet with a new outer IP header.

The entire original, inner, packet travels through a tunnel from one point of an IP network to another; no routers along the way are able to examine the inner IP header.

Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security.

Here is an example of how tunnel mode IPsec operates.

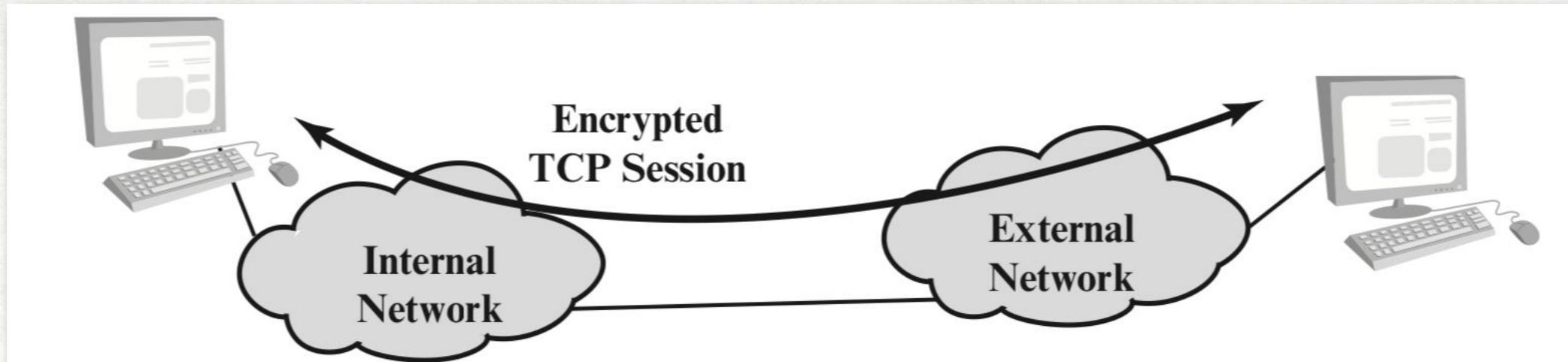
Host A on a network generates an IP packet with the destination address of host B on another network. This packet is routed from the originating host to a firewall or secure router at the boundary of A's network. The firewall filters all outgoing packets to determine the need for IPsec processing.

If this packet from A to B requires IPsec, the firewall performs IPsec processing and encapsulates the packet with an outer IP header. The source IP address of this outer IP packet is this firewall, and the destination address may be a firewall that forms the boundary to B's local network.

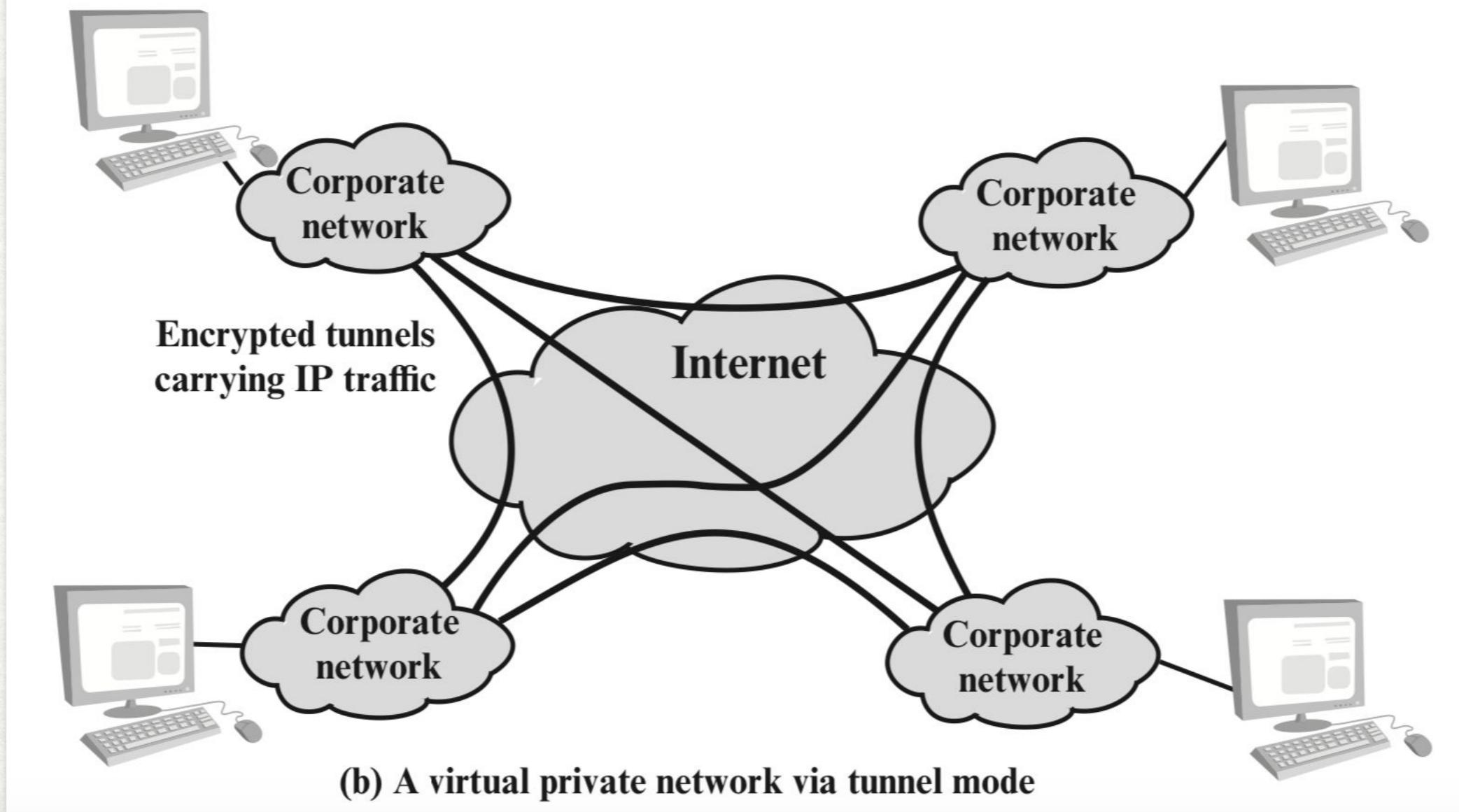
This packet is now routed to B's firewall, with intermediate routers examining only the outer IP header. At B's firewall, the outer IP header is stripped off, and the inner packet is delivered to B.

ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header. AH in tunnel mode authenticates the entire inner IP packet and selected portions of the outer IP header.

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.



(a) Transport-level security



(b) A virtual private network via tunnel mode

Identity management

Identity management (ID management) is the organizational process for identifying, authenticating and authorizing individuals or groups of people to have access to applications, systems or networks by associating user rights and restrictions with established identities.

The main goal of identity management is to ensure that only authenticated users are granted access to the specific applications, systems or IT environments for which they are authorized

This includes control over user provisioning and the process of onboarding new users such as employees, partners, clients and other stakeholders. Identity management also includes control over the process of authorizing system or network permissions for existing users and the offboarding of users who are no longer authorized to access organization systems.

An identity and access management (IAM) system can provide a framework that includes the policies and technology needed to support the management of electronic or digital identities. Many of today's IAM systems use federated identity, which allows a single digital identity to be authenticated and stored across multiple disparate systems.

An IAM system can also be used to deploy single sign-on (SSO) technologies to significantly decrease the number of passwords users need; SSO incorporates a federated-identity approach by using a single login and password to create an authentication token that can be accepted by various enterprise systems and applications.

NON-SSO SCENARIO

- 2 Ask for login info, authenticates user
- 5 Ask for login info, authenticates user

