SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA
Department of Electronics and Communication Engineering
Academic Year 2020-21, Odd Semester

Study Material

Subject: Elective III- Cryptography and Network Security/Unit-1

Semester: VII **Class: IV Year**

Pre- requisite: Basic knowledge of Digital Communication

Objectives:

- To understand OSI security architecture and classical encryption techniques.
- To acquire fundamental knowledge on the concepts of finite fields and number theory.

Topics to be Covered:

Introduction & Number Theory: Services, Mechanisms and attacks, the OSI security architecture, Network security model, Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography) Finite Fields and Number Theory: Groups, Rings, Fields, Modular arithmetic, Euclid's algorithm. Finite fields, Polynomial Arithmetic, Prime numbers-Fermat's and Euler's theorem, Testing for primality, The Chinese remainder theorem, Discrete logarithms

Chapter Outcome:

After studying this chapter, Student should be able to:

1. Describe the key security requirements of confidentiality, integrity, and availability.
2. Discuss the types of security threats and attacks that must be dealt with
3. Examples of the types of threats and attacks that apply to different categories of computer and network assets.
4. Describe different types of encryption techniques and mathematical models for encryption.

UNIT I - INTRODUCTION & NUMBER THEORY

INTRODUCTION:

Computer security, **cybersecurity** or **information technology security** (**IT security**) is the protection of computer systems and networks from the theft of or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide.

Computer and network security is essentially a battle of wits between a culprit who tries to find holes and the designer or administrator who tries to close them.

Computer security is a series of protocols that a company or an individual follows to ensure information maintains its "ICA" – integrity, confidentiality and availability.

CRYPTOGRAPHY:

Cryptography is the art and science of making a cryptosystem that is capable of providing information security. Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental information security services.

CRYPTOSYSTEM

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as a cipher system.

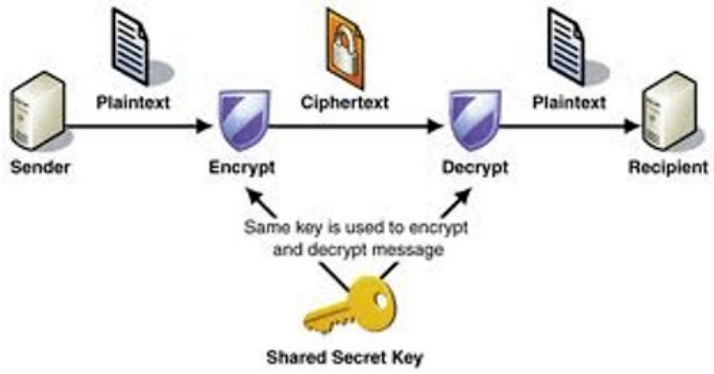


Figure.1. Cryptosystem

Cryptosystem shown in fig.1, is th study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. When transmitting electronic data, the most common use of **cryptography** is to encrypt and decrypt email and other plain-text messages. It reformats and transform our data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

Network Security - measures to protect data during their transmission

Internet Security - measures to protect data during their transmission over a collection of interconnected networks

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

This definition introduces three key objectives that are at the heart of computer security:

Confidentiality (C): Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

Integrity(I): Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.

Availability (A): Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

These three concepts form what is often referred to as the **CIA triad**. The three concepts embody the fundamental security objectives for both data and for information and computing services.

Confidentiality: This term covers two related concepts:

Data1 **confidentiality:** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

Privacy: Assures that individual's control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Integrity: This term covers two related concepts:

Data integrity: Assures that information and programs are changed only in a specified and authorized manner.

System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Availability: Assures that systems work promptly and service is not denied to authorized users. Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture. Two of the most commonly mentioned are as follows:

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

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

THE OSI SECURITY ARCHITECTURE:

To assess effectively the security needs of an organization and to evaluate and choose various security products and policies, the manager responsible for security needs, some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. The OSI security architecture was developed in the context of the OSI protocol architecture by ITU-T.

ITU-T: The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) is a United Nations sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

Recommendation X.800, Security Architecture for OSI, defines a systematic approach. The OSI security architecture is useful to managers as a way of organizing the task of providing security.

The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined

briefly as

- **Security attack**
- **Security mechanism**.
- **Security service**

threat and attack are commonly used to mean more or less the same thing. The definitions taken from RFC 4949, Internet Security Glossary.

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Attack

An attack on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

SECURITY ATTACK:

Any action that compromises the security of information owned by an organization. There are four general categories of attack which are listed below.

Interruption

An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability.

e.g., destruction of piece of hardware, cutting of a communication line or disabling of file management system.

Figure.2a

Interception

An unauthorized party gains access to an asset. This is an attack on confidentiality. Unauthorized party could be a person, a program or a computer. e.g., wiretapping to capture data in the network, illicit copying of files.

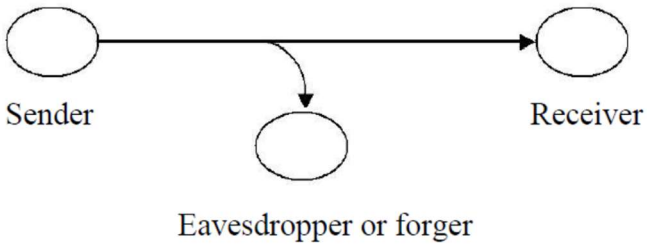


Figure. 2b

Modification

An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity.

e.g., changing values in data file, altering a program, modifying the contents of messages being transmitted in a network.

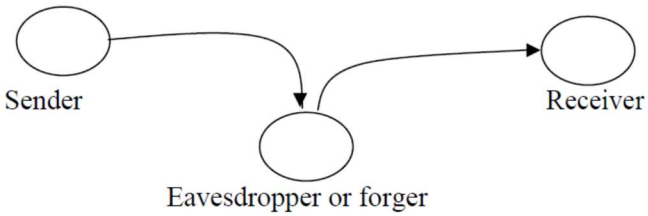


Figure. 2c

Fabrication

An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity.

e.g., insertion of spurious message in a network or addition of records to a file.

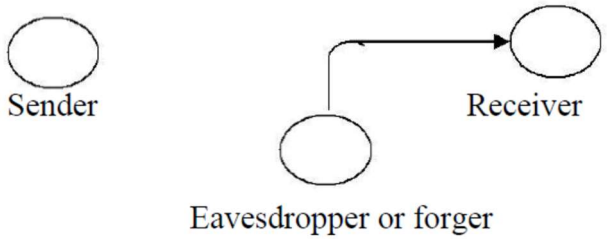


Figure.2d

The attack is majorly classified into two types:

- Active attack
- Passive Attack

PASSIVE ATTACK:

Passive attacks (Fig.3) are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted.

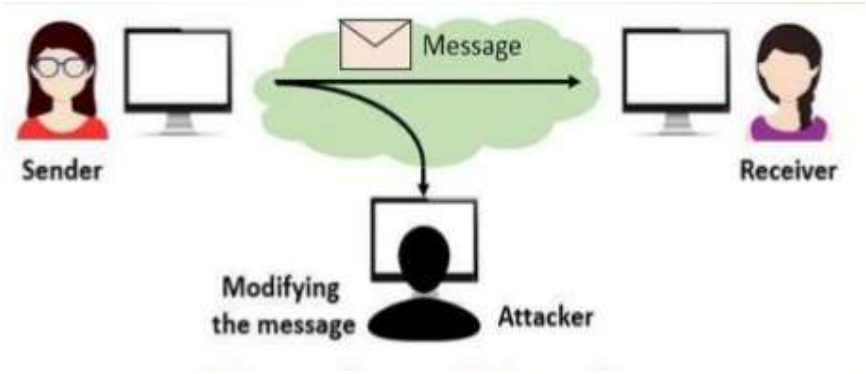


Figure.3

Passive attacks are of two types:

Release of message contents: A telephone conversation, an e-mail message and a transferred file may contain sensitive or confidential information. We would like to prevent the opponent from learning the contents of these transmissions.

Traffic analysis: If we had encryption protection in place, an opponent might still be able to observe the pattern of the message. The opponent could determine the location and identity of communication hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of data. However, it is feasible to prevent the success of these attacks.

ACTIVE ATTACKS:

These attacks involve some modification of the data stream or the creation of a false stream.

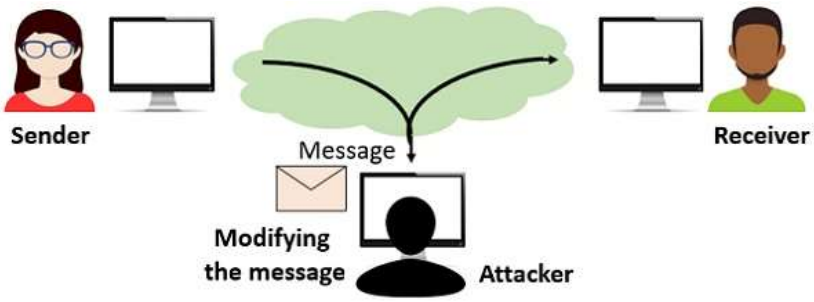


Figure.4

These attacks can be classified in to four categories:

Masquerade – One entity pretends to be a different entity.

Replay – involves passive capture of a data unit and its subsequent transmission to produce an unauthorized effect.

Modification of messages – Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.

Denial of service – Prevents or inhibits the normal use or management of communication facilities. Another form of service denial is the disruption of an entire network, either by disabling the network or overloading it with messages so as to degrade performance. It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

Security mechanism: A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

Cryptanalysis: Cryptanalysis is the study of methods for obtaining the meaning of encrypted information, without access to the secret information that is typically required to do so. Typically, this involves knowing how the system works and finding a secret key. Cryptanalysis is also referred to as codebreaking or cracking the code.

Brute-force attack: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

SECURITY SERVICE: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. X.800 divides these services into five categories

Confidentiality: Ensures that the information in a computer system and transmitted information are accessible only for reading by authorized parties.

Eg., printing, displaying and other forms of disclosure.

Authentication: Ensures that the origin of a message or electronic document is correctly identified, with an assurance that the identity is not false.

Integrity: Ensures that only authorized parties are able to modify computer system assets and transmitted information. Modification includes writing, changing status, deleting, creating and delaying or replaying of transmitted messages.

Non repudiation: Requires that neither the sender nor the receiver of a message be able to deny the transmission.

Access control: Requires that access to information resources may be controlled by or the target system.

Availability: Requires that computer system assets be available to authorized parties when needed.

AUTHENTICATION:

The authentication service is concerned with assuring that a communication is Authentic, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. Two specific authentication services are defined in X.800:

Peer Entity Authentication

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

Data Origin Authentication

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

ACCESS CONTROL

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

DATA CONFIDENTIALITY

The protection of data from unauthorized disclosure. Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified.

Connection Confidentiality: The protection of all user data on a connection.

Connectionless Confidentiality: The protection of all user data in a single data block

AUTHENTICATION

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

Traffic Flow Confidentiality: The protection of the information that might be derived from observation of traffic flows.

DATA INTEGRITY

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

Connection Integrity with Recovery

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

Connection Integrity without Recovery: As above, but provides only detection without recovery.

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

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

Selective-Field Connectionless Integrity: Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

NONREPUDIATION

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

Nonrepudiation, Origin: Proof that the message was sent by the specified party.

Nonrepudiation, Destination: Proof that the message was received by the specified party.

SECURITY MECHANISMS

One of the most specific security mechanisms in use is cryptographic techniques. Encryption or encryption-like transformations of information are the most common means of providing security.

SPECIFIC SECURITY MECHANISMS

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

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

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

Access Control: A variety of mechanisms that enforce access rights to resources

Data Integrity: A variety of mechanisms used to assure the integrity of a data unit or stream of data units

Authentication Exchange: A mechanism intended to ensure the identity of an entity by means of information exchange.

Traffic Padding: The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

Routing Control: Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

Notarization: The use of a trusted third party to assure certain properties of a data exchange.

PERVASIVE SECURITY MECHANISMS

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

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

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

Event Detection: Detection of security-relevant events.

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

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

NETWORK SECURITY MODEL:

A model for a network security is shown in the below figure. 5

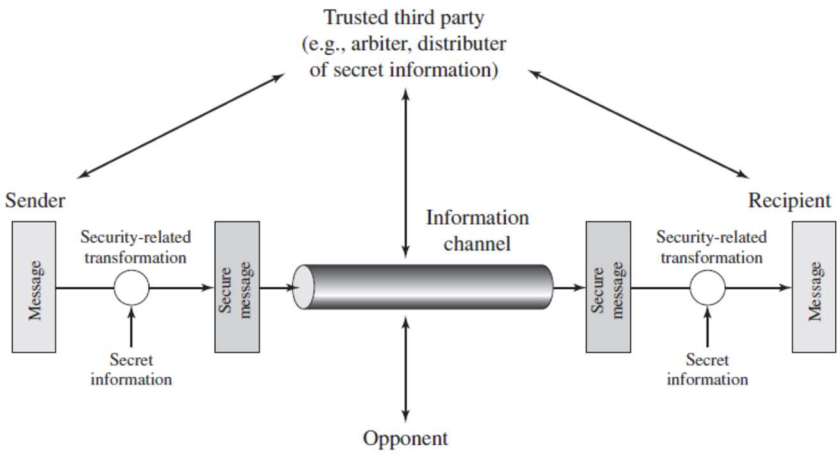


Figure.5 Network Security Model

A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to

take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

• A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.

• Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

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

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

However, there are other security-related situations of interest that do not neatly fit this model but are considered. A general model of these other situations is illustrated in Figure.6 which reflects a concern for protecting an information system from unwanted access.

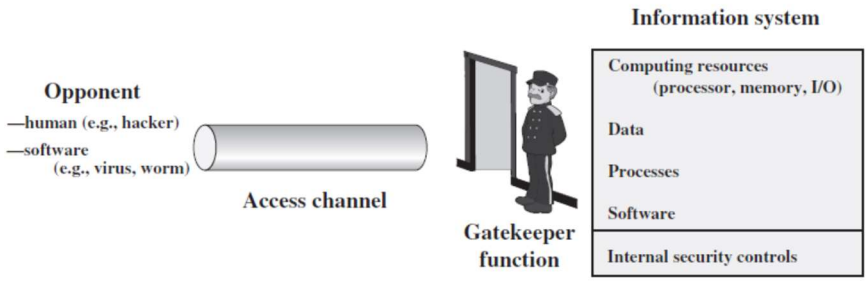


Figure.6 Network Access Security Model

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers. Programs can present two kinds of threats:

Information access threats: Intercept or modify data on behalf of users who should not have access to that data.

Service threats: Exploit service flaws in computers to inhibit use by legitimate users.

Classical Encryption Techniques: A SYMMETRIC CIPHER MODEL:

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the development of public key encryption in the 1970s.

Some basic terminologies used:

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

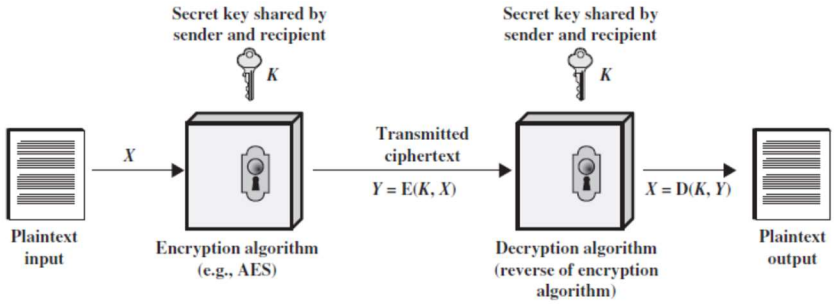


Fig.7 Simplified Model of Symmetric Encryption

A symmetric encryption scheme has five ingredients

A symmetric encryption scheme has five ingredients (Fig.7). Here the original message, referred to as plaintext, is converted into apparently random nonsense, referred to as cipher text. The encryption process consists of an algorithm and a key.

The key is a value independent of the plaintext. Changing the key changes, the output of the algorithm. Once the cipher text is produced, it may be transmitted. Upon reception, the cipher text can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption.

The security depends on several factors. First, the encryption algorithm must be powerful enough that it is impractical to decrypt a message on the basis of cipher text alone. Beyond that, the security depends on the secrecy of the key, not the secrecy of the algorithm.

Two requirements for secure use of symmetric encryption:

- A strong encryption algorithm
- A secret key known only to sender / receiver
- $Y = EK(X)$
- $X = DK(Y)$

assume encryption algorithm is known implies a secure channel to distribute key

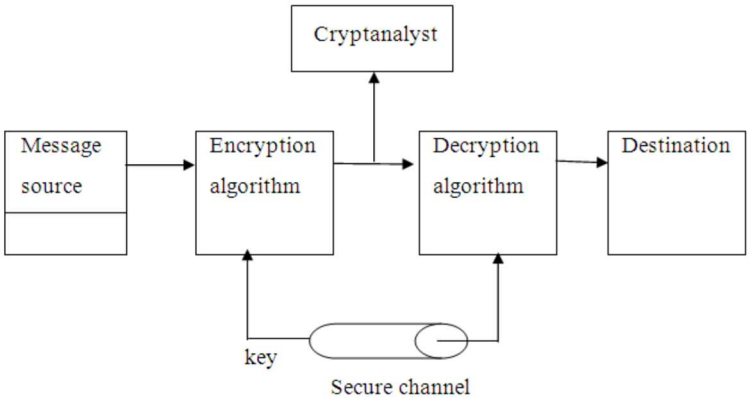


Fig.8. conventional cryptosystem

A source produces a message in plaintext, X = [X1, X2, …, XM] where M are the number of letters in the message. A key of the form K = [K1, K2, …, KJ] is generated. If the key is generated at the source, then it must be provided to the destination by means of some secure channel. With the message X and the encryption key K as input, the encryption algorithm forms

the cipher text Y = [Y1, Y2, …, YN]. This can be expressed as Y = EK(X)

The intended receiver, in possession of the key, is able to invert the transformation: X = DK(Y)

An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both. It is assumed that the opponent knows the encryption and decryption algorithms. If the opponent is interested in only this particular message, then the focus of effort is to recover X by generating a plaintext estimate. Often if the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate.

Substitution Encryption Techniques:

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

- **(i)Caesar cipher (or) shift cipher**
- The earliest known use of a substitution cipher and the simplest was by Julius Caesar.
- The Caesar Cipher is a type of **shift cipher**. Shift Ciphers work by using the modulo operator to encrypt and decrypt messages. The Shift Cipher has a **key K**, which is an **integer from 0 to 25**. We will only share this key with people that we want to see our message
- The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.
- e.g., Plain text: pay more mone Cipher text: SDB PRUH PRQHB
- Note that the alphabet is wrapped around, so that letter following „z‟ is „a‟.
- Note that the alphabet is wrapped around, so that the letter following Z is A.
- We can define the transformation by listing all possibilities, as follows:

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

- Let us assign a numerical equivalent to each letter:

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

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

For Encrypt each plaintext letter p, substitute the cipher text letter c such that

C = E(p) = (p+3) mod 26,

a shift may be any amount, so that general Caesar algorithm is

C = E (p) = (p+k) mod 26

where k takes on a value in the range 1 to 25.

The decryption algorithm is simply P = D(C) = (C-k) mod 26 (or) to Encrypt a message M.

Convert the letter into the number that matches its order in the alphabet starting from 0, and call this number X, (A=0, B=1, C=2, ..., Y=24, Z=25).

Calcúlate: Y = (X + K) mod 26

Convert the number Y into a letter that matches its order in the alphabet starting from 0.

Example:

By using the Shift Cipher with **key K=19** for our message. We encrypt the message **"KHAN",** as follows

ENCRYPTION

	K	H	A	N	
	10	7	0	13	
+	19	19	19	19	
(	29	26	19	32	) mod 26
	3	0	19	6	
	D	A	T	G	

- So, after applying the Shift Cipher with key K=19 our message text "**KHAN**" gave us **cipher text** "**DATG**".
- For every letter in the cipher text **C, convert** the letter into the number that matches its order in the alphabet starting from 0, and call this number Y.
- If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: Simply try all the 25 possible keys.

Monoalphabetic Ciphers:

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. Before proceeding, the term *permutation can be defined*.

A permutation of a finite set of elements *S* is an ordered sequence of all the elements of *S*, with each element appearing exactly once.

For example, if *S* = {a, b, c}, there are six permutations of *S*:

abc, acb, bac, bca, cab, cba

In general, there are *n*! permutations of a set of *n* elements, because the first element can be chosen in one of *n* ways, the second in *n* - 1 ways, the third in *n* – 2 ways, and so on.

plain: a b c d e f g h i j k l m n o p q r s t u v w x y z Caesar cipher: d e f
g h i j k l m n o p q r s T u v w x y z a b c

If, instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 * 1026 possible keys.

This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a mono alphabetic substitution cipher, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.

A countermeasure is to provide multiple substitutes known as homophones, for a single letter. For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone assigned to a letter in rotation or randomly.

Playfair Cipher:

The best-known multiple-letter encryption cipher is the Playfair, which treats diagrams in the plaintext as single units and translates these units into cipher text diagrams

he Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers‟s Have His Carcase

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

In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter.

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

Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.

Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.

Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM

The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 * 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult.

For these reasons, the Playfair cipher was for a long time considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

Hill Cipher:

Another interesting multiletter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929.

The Hill Algorithm

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

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

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

$$c_1c_2c_3 = p_1p_2p_3 \begin{matrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{matrix} mod\ 26$$

or

C = **PK** mod 26

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

Polyalphabetic ciphers

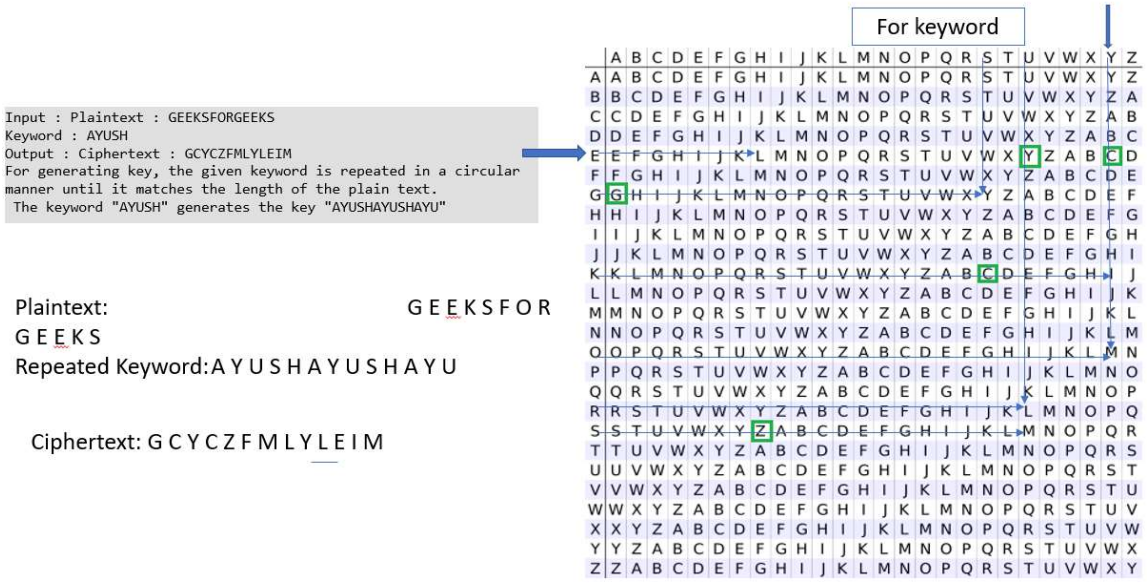
A **polyalphabetic cipher** is any **cipher** based on substitution, using multiple substitution alphabets. The Vigenère **cipher** is probably the best-known example of a **polyalphabetic cipher**.

Difference between monoalphabetic cipher and polyalphabetic cipher:

A **monoalphabetic cipher** is a substitution **cipher** in which the **cipher** alphabet is fixed through the encryption process. ... A **polyalphabetic cipher** is a substitution **cipher** in which the **cipher** alphabet changes during the encryption process.

Vigenere cipher:

- **Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets .The encryption of the original text is done using the *Vigenère square or Vigenère table*.**
- **The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.**
- **At different points in the encryption process, the cipher uses a different alphabet from one of the rows.**
- **The alphabet used at each point depends on a repeating keyword**



The Vigenère cipher can be expressed in the following manner. Assume a sequence of plaintext letters $P = p_0, p_1, p_2, \ldots, p_{n-1}$ and a key consisting of the sequence of letters $K = k_0, k_1, k_2, \ldots, k_{m-1}$, where typically $m < n$. The sequence of ciphertext letters $C = C_0, C_1, C_2, \ldots\ldots, C_{n-1}$ is calculated as follows:

$$C = C_0, C_1, C_2, \ldots, C_{n-1} = E(K, P) = E[(k_0, k_1, k_2, \ldots, k_{m-1}), (p_0, p_1, p_2 \ldots, p_{n-1})]$$

$$= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \ldots\ldots, (p_{m-1} + k_{m-1}) \bmod 26, (p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \ldots\ldots, (p_{2m-1} + k_{m-1}) \bmod 26, \ldots$$

Thus, the first letter of the key is added to the first letter of the plaintext, mod 26, the second letters are added, and so on through the first *m* letters of the plaintext. For the next *m* letters of the plaintext, the key letters are repeated. This process continues until all of the plaintext sequence is encrypted. A general equation of the encryption process is

$C_i = (p_i + k_i \bmod m) \bmod 26$

A general equation for decryption is

$p_i = (C_i - k_i \bmod m) \bmod 26$

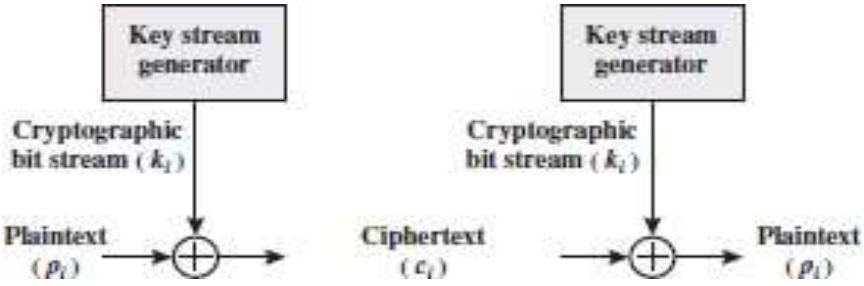
To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword is *deceptive*, the message "we are discovered save yourself" is encrypted as

Key :*deceptivedeceptivedeceptive* plaintext : wearediscoveredsaveyourself

ciphertext : ZICVTWQNGRZGVTWAVZHCQYGLMGJ

The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost.

Vernam Cipher The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it. Such a system was introduced by an AT&T engineer named Gilbert Vernam in 1918.



- The system can be expressed as:

  $ci = pi \oplus ki$

  where

  p_i= *i*th binary digit of plaintext

  k_i= *i*th binary digit of key

c_i= ith binary digit of ciphertext

$\oplus$ s= exclusive-or (XOR) operation

- Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation

One Time Pad Cipher

It is an unbreakable cryptosystem. It represents the message as a sequence of 0s and 1s. this can be accomplished by writing all numbers in binary, for example, or by using ASCII. The key is a random sequence of 0"s and 1"s of same length as the message. Once a key is used, it is discarded and never used again. The system can be expressed as follows:

Ci = Pi Ki

Ci - ith binary digit of cipher text Pi - ith binary digit of plaintext

Ki - ith binary digit of key – exclusive OR opearaiton

Thus, the cipher text is generated by performing the bitwise XOR of the plaintext and the key. Decryption uses the same key. Because of the properties of XOR, decryption simply involves the same bitwise operation:

Pi = Ci Ki

e.g., plaintext = 0 0 1 0 1 0 0 1

Key = 1 0 1 0 1 1 0 0

---

ciphertext = 1 0 0 0 0 1 0 1

Advantage:

 Encryption method is completely unbreakable for a ciphertext only attack.

Disadvantages

 It requires a very long key which is expensive to produce and expensive to transmit.

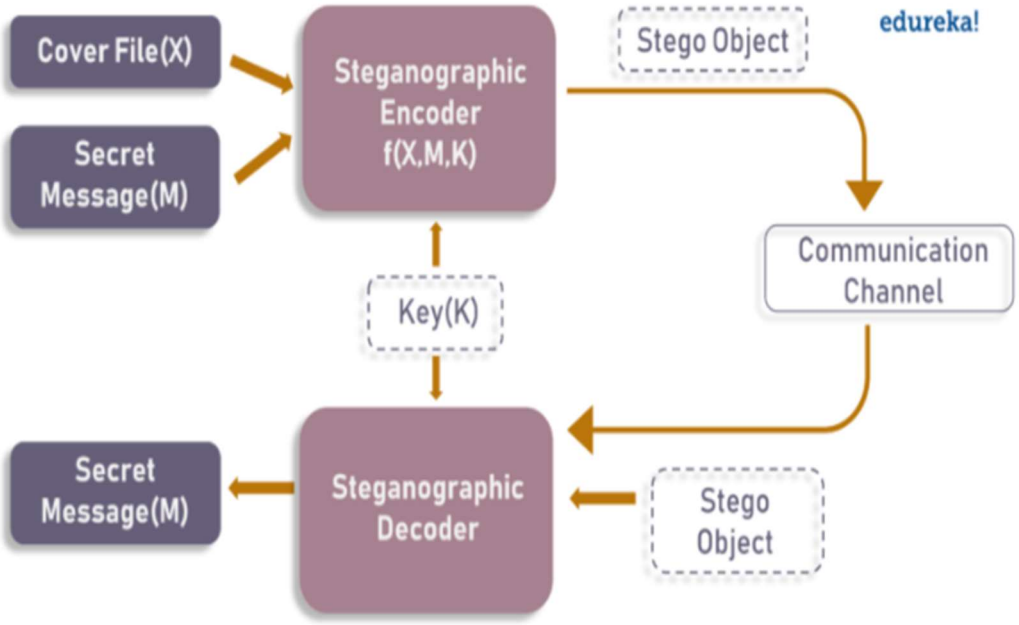
 Once a key is used, it is dangerous to reuse it for a second message; any knowledge on the first message would give knowledge of the second.

STEGANOGRAPHY:

- Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data.

- It stems from two Greek words, which are *steganos,* means covered and *graphia,* means writing
- Examples,

1. Playing an audio track backwards to reveal a secret message
2. Playing a video at a faster frame rate (FPS) to reveal a hidden image
3. Embedding a message in the red, green, or blue channel of an RGB image
4. Hiding information within a file header or metadata
5. Embedding an image or message within a photo through the addition of digital noise



- As the image depicts, both cover file(X) and secret message(M) are fed into steganographic encoder as input.
- Steganographic Encoder function, f(X,M,K) embeds the secret message into a cover file.
- Resulting Stego Object looks very similar to your cover file, with no visible changes.
- This completes encoding. To retrieve the secret message, Stego Object is fed into Steganographic Decoder.
- Steganography Techniques
- Depending on the nature of the cover object (actual object in which secret data is embedded), steganography can be divided into five types:

1. Text Steganography
2. Image Steganography
3. Video Steganography

4. Audio Steganography
5. Network Steganography

➨ Text Steganography: Text Steganography is hiding information inside the text files. Various techniques used to hide the data in the text are:

- Format Based Method
- Random and Statistical Generation
- Linguistic Method

➨ Image Steganography: Hiding the data by taking the cover object as the image is known as image steganography. There are a lot of ways to hide information inside an image. Common approaches include:

- Least Significant Bit Insertion
- Masking and Filtering
- Redundant Pattern Encoding
- Encrypt and Scatter
- Coding and Cosine Transformation

➨ Audio Steganography: In audio steganography, the secret message is embedded into an audio signal which alters the binary sequence of the corresponding audio file. Different methods of audio steganography include:

- Least Significant Bit Encoding
- Parity Encoding
- Phase Coding
- Spread Spectrum

➨ Video Steganography: In Video Steganography you can hide kind of data into digital video format. Two main classes of Video Steganography include:

- embedding data in uncompressed raw video and compressing it later
- Embedding data directly into the compressed data stream
- Network Steganography (Protocol Steganography): It is the technique of embedding information within network control protocols used in data transmission such TCP, UDP, ICMP etc. For Example, you can hide information in the header of a TCP/IP packet in some fields that are either optional**.**

Example:

(i) the sequence of first letters of each word of the overall message spells out the real (hidden) message.

(ii) Subset of the words of the overall message is used to convey the hidden message.

Various other techniques have been used historically, some of them are:

 Character marking – selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held to an angle to bright light.

Invisible ink – a number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.

 Pin punctures – small pin punctures on selected letters are ordinarily not visible unless the paper is held in front of the light.

 Typewritten correction ribbon – used between the lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

Drawbacks of steganography

 Requires a lot of overhead to hide a relatively few bits of information.

 Once the system is discovered, it becomes virtually worthless.

TRANSPOSITION TECHNIQUES:

All the techniques examined so far involve the substitution of a cipher text symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

Rail fence is simplest of such cipher, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

Plaintext = meet at the school house

To encipher this message with a rail fence of depth 2, we write the message as follows:

m e a t e c o l o s

e t t h s H o h u e

The encrypted message is MEATECOLOSETTHSHOHUE

Row Transposition Ciphers-A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of columns then becomes the key of the algorithm.

e.g., plaintext = meet at the school house

Key = 4 3 1 2 5 6 7

PT = m e e t a t t

h e s c h o o

l h o u s e

CT = ESOTCUEEHMHLAHSTOETO

A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is more complex permutation that is not easily reconstructed.

FINITE FIELDS AND NUMBER THEORY:

- Finite fields have become increasingly important in cryptography.
- A number of cryptographic algorithms rely heavily on properties of finite fields, notably the Advanced Encryption Standard (AES) and elliptic curve cryptography.
- Other examples include the message authentication code CMAC and the authenticated encryption scheme GCM
    - Groups, Rings, Fields, Modular arithmetic, Euclid's algorithm
    - Finite fields Euclid's algorithm
    - Polynomial Arithmetic
    - Prime numbers-Fermat's and Euler's theorem
    - Testing for primality
    - The Chinese remainder theorem
    - Discrete logarithms
- Widely used in cryptography to perform large calculations
- Some basic concepts are
- **Prime Number**: a number that is divisible only by itself and 1 (e.g. 2, 3, 5, 7, 11)
- **Relative Prime Number**: Two integers are **relatively prime** (or coprime) if there is no integer greater than one that divides them both (that is, their greatest common divisor is one). For **example**, 12 and 13, GCD (12,13) = 1→12 and 13 are **relatively prime**, but 12 and 14 are not.,
- Modular

Congruent Modulo

- **Modular :** When we divide two integers we will have an equation that looks like the following:
- A/B=Q remainder R
- A is the dividend

  B is the divisor

  Q is the quotient

  R is the remainder

- Sometimes, we are only interested in what the **remainder** is when we divide A by B. For these cases there is an operator called the modulo operator (abbreviated as mod).
- Using the same A, B, Q, and R as above, we would have: *A* mod *B*=*R*
- We would say this as A *modulo* B *is equal to* R. Where B is referred to as the **modulus**.

Ex. 13/5= 2 remainder of 3 then, **13 mod 5 = 3**

CONGRUENT MODULO:

- Consider two integers a and b
- a and b said to be congruent to n for
- a (mod n) = b (mod n) then
- **a ≡ b (mod n) (OR) a (mod n) = b**
- **example:**
- let a=73, b=4 and n=23
- find a mod n
- 73 mod 23 =4 (reminder of 73/23)
- find b mod n
- since 23 is larger than 4 then,
- 4 mod 23 = 4
- here 73 mod 23 = 4 and 4 mod 23=4, this can be written as
- 73 ≡ 4 (mod 23) ==> a ≡ b (mod n)

Properties of Congruences

Congruences have the following properties:

- **Property 1:** a ≡ b (mod n) if n is multiple of (a-b)
- Example: let a=30, b=10 and n=5
- a-b = 30-10 = 20
- Since 20 is multiple of 5 then 30 ≡ 10 (mod 5)
- **Property 2:** a (mod n) = b (mod n) ==> a ≡ b (mod n)
- **Property 3:** a mod n=b
- and b mod n=c, → b = c (mod n) sub it in a mod n
- then **a mod n= c mod n and a ≡ c (mod n)**
- **Arithmetic Property:** ((a mod n) + (b mod n))mod n= (a+b) mod n **[same for -,*,/]**
- **Commutative Property: (a+b) mod n = (b+a) mod n [same for *]**

- **Associative Property:** ((a + b)+c) mod n = (a+(b + c)) mod n
- **Identity Property:**
- (0+a) mod n = a mod n
- (1 * a) mod n = a mod n

Modular Arithmetic Operations

The (mod n) operator maps all integers into the set of integers {0, 1, c, (n - 1)}. This technique is known as **modular arithmetic**.

Modular arithmetic exhibits the following properties:

[(a mod n) + (b mod n)] mod n = (a + b) mod n

[(a mod n) - (b mod n)] mod n = (a - b) mod n

[(a mod n) × (b mod n)] mod n = (a × b) mod n

First property:

Define (a mod n) = r_a and (b mod n) = r_b. Then we can write $a = r_a + jn$ for some integer j and $b = r_b + kn$ for some integer k.

Then

(a + b) mod n = (ra + jn + rb + kn) mod n = (ra + rb + (k + j)n) mod n

= (ra + rb) mod n

= [(a mod n) + (b mod n)] mod n

Define (a mod n) = r_a and (b mod n) = r_b. Then we can write $a = r_a + jn$ for some integer j and $b = r_b + kn$ for some integer k.

Then

(a + b) mod n = (ra + jn + rb + kn) mod n = (ra + rb + (k + j)n) mod n

= (ra + rb) mod n

= [(a mod n) + (b mod n)] mod n

Examples of the three properties:

11 mod 8 = 3; 15 mod 8 = 7

[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 =2

(11 + 15) mod 8 =26 mod 8 =2

[(11 mod 8) – (15 mod 8)] mod 8 = -4 mod 8 = 4

(11-15) mod 8 = -4 mod 8 = 4

[(11 mod 8)× (15 mod 8)] mod 8 = 21 mod 8 =5

(11 × 15) mod 8 = 165 mod 8 =5

Exponentiation is performed by repeated multiplication, as in ordinary arithmetic.

To find 11^7 mod 13,

$11^2 = 121 = 4$ (mod 13)

$11^4 = (11^2)^2 = 4^2 = 3$ (mod 13)

$11^7 = 11 \times 4 \times 3 = 132 = 2$ (mod 13)

Thus, the rules for ordinary arithmetic involving addition, subtraction, and multiplication carry over into modular arithmetic. The following table below provides an illustration of modular addition and multiplication modulo 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

(b) Multiplication modulo 8

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(c) Additive and multiplicative inverse modulo 8

Both matrices are symmetric about the main diagonal in conformance to the commutative property of addition and multiplication.

As in ordinary addition, there is an additive inverse, or negative, to each integer in modular arithmetic.

In this case, the negative of an integer x is the integer y such that $(x + y) \bmod 8 = 0$.

To find the additive inverse of an integer in the left-hand column, scan across the corresponding row of the matrix to find the value 0; the integer at the top of that column is the additive inverse; thus, $(2 + 6) \bmod 8 = 0$. Similarly, the entries in the multiplication table are straightforward.

In modular arithmetic mod 8, the multiplicative inverse of x is the integer y

such that $(x\ y) \bmod 8 = 1 \bmod 8$.

FERMAT'S AND EULER'S THEOREM

Two theorems that play important roles in public-key cryptography are Fermat‟s theorem and Euler‟s theorem.

Fermat's Theorem

Fermat‟s theorem states the following: If p is prime and a is a positive integer not divisible by p, then

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

Proof: Consider the set of positive integers less than p: {1, 2, c, p - 1} and multiply each element by a, modulo p, to get the set $X = \{a \bmod p, 2a \bmod p, \text{c}, (p - 1)a \bmod p\}$. None of the elements of X is equal to zero because p does not divide a. Furthermore, no two of the integers in X are equal.

- To see this, assume that $ja \equiv ka \pmod{p}$, where $1 \le j < k \le p - 1$. Because a is relatively prime to p, Eliminate a from both sides of the equation resulting in $j \equiv k \pmod{p}$.
- This last equality is impossible, because j and k are both positive integers less than p. Therefore, (p - 1) elements of X are all positive integers with no two elements equal.
- We can conclude the X consists of the set of integers {1, 2, …..,p - 1} in some order. Multiplying the numbers in both

sets (p and X) and taking the result mod p yields

$$a \times 2a \times \ldots\ldots\ldots \times (p-1)a \equiv [(1 \times 2 \times \ldots\ldots\ldots \times (p-1)] \pmod{p}$$

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

- We can cancel the $(p - 1)!$ term because it is relatively prime to p. This yields Equation, which completes the proof.

$$a = 7, p = 19$$

$$7^2 = 49 \equiv 11 \pmod{19}$$

$$7^4 \equiv 121 \equiv 7 \pmod{19}$$

$$7^8 \equiv 49 \equiv 11 \pmod{19}$$

$$7^{16} \equiv 121 \equiv 7 \pmod{19}$$

$$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \pmod{19}$$

-

- An alternative form of Fermat‟s theorem is also useful: If p is prime and a is a positive integer, then

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

Euler's Theorem

Euler‟s theorem states that for every a and n that are relatively prime:

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

Proof: The above equation is true if n is prime, because in that case,

$\Phi(n) = (n - 1)$ and Fermat‟s theorem holds. However, it also holds for any integer n. $\Phi(n)$ is the number of positive integers less than n that are relatively primeto n.

Consider the set of such integers, labeled as

$R = \{x_1, x_2, \ldots\ldots, x_{\Phi(n)}\}$

That is, each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$.

Now multiply each element by a, modulo n:

$S = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots , (ax_{\phi(n)} \bmod n)\}$

The set S is a permutation of R, by the following line of reasoning:

Because a is relatively prime to n and x_i is relatively prime to n, ax_i must also be relatively prime to n. Thus, all the members of S are integers that are less than n and that are relatively prime to n.

If $ax_i \bmod n = ax_j \bmod n$, then $x_i = x_j$.

Therefore,

$$\prod_{i=1}^{\phi n} (ax_i \bmod n) = \prod_{i=1}^{\phi n} x_i$$

$$\prod_{i=1}^{\phi n} x_i = \sum_{i=1}^{\phi n} (ax_i \bmod n)$$

$$a\phi(n) \times [\prod_{i=1}^{\phi n} x_i \ (mod\ n)]$$

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

which completes the proof. This is the same line of reasoning applied to the proof of Fermat‟s theorem.

a=3; n=10; $\phi(10)= 4a^{\phi(n)} = 3^4 = 81 = 1(\text{mod } 10) = 1(\text{mod n})$
a=2; n=11; $\phi(11)= 10a^{\phi(n)} = 2^{10} = 1024 = 1(\text{mod } 11) = 1(\text{mod n})$

As is the case for Fermat‟s theorem, an alternative form of the theorem is also useful:

$a^{\phi n+1} \equiv a\ mod\ n$

CHINESE REMINDER THEOREM:

One of the most useful results of number theory is the **Chinese remainder theorem** (CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.

The CRT can be stated in several ways. Let

$$M = \prod_{i=1}^{k} m_i$$

where A $Z_{M,}ai$ Z_{mi}, and $a_i = A \bmod m_i$ for 1 i k.

The CRT makes two assertions. The mapping of the above equation is a one-to-one correspondence (called a **bijection**) between Z_M and the Cartesian product $Zm1 \times Zm2 \times \ldots \times Zmk$. That is, for every integer A such that $0 \leq A \leq M$, there is a unique k-tuple ($a1$, $a2$,…., ak) with $0 \leq ai < mi$ that represents it, and for every such k-tuple ($a1$, $a2$,….. ak), there is a unique integer A in ZM.

Operations performed on the elements of ZM can be equivalently performed on the corresponding k-tuples by performing the operation independently in each coordinate position in the appropriate system.

FINITE FIELDS

Groups, Rings and Field:

- **Group**: A set of elements that is closed with respect to some operation.
- Closed-> The result of the operation is also in the set
- The operation obeys:
- Obeys associative law: **(a.b).c = a.(b.c)**
- Has identity e: **e.a = a.e = a**
- Has inverses **a-1: a.a-1 = e**
- **Abelian Group**: The operation is commutative

a.b = b.a

- Example: Z8, + modular addition, identity =0

Cyclic Group

Exponentiation: Repeated application of operator

- example: **a3 = a.a.a**
- Cyclic Group: Every element is a power of some fixed element, i.e., **b = ak**

for some a and every b in group a is said to be a generator of the group

- Example: {1, 2, 4, 8} with mod 12 multiplication, the generator is 2.
- 20=1, 21=2, 22=4, 23=8, 24=4, 25=8

Ring:

- A group with two operations: addition and multiplication

 The group is abelian with respect to addition: **a+b=b+a**

 Multiplication and additions are both associative:

a+(b+c)=(a+b)+c

a.(b.c)=(a.b).c

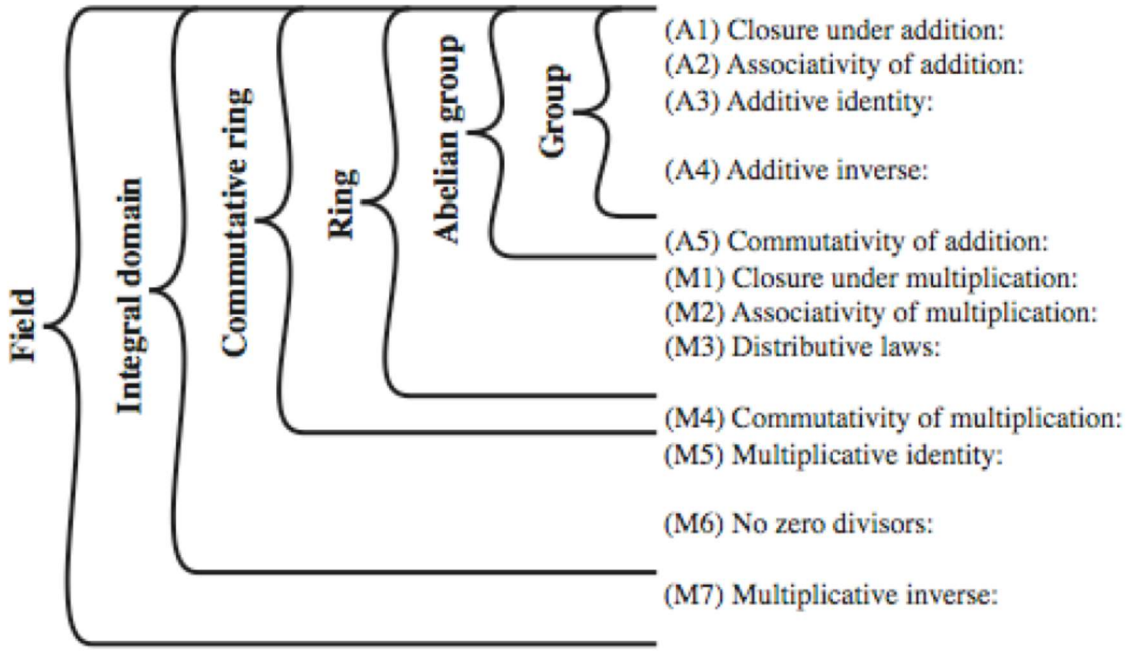
 Multiplication distributes over addition, **a.(b+c)=a.b+a.c**

 Commutative Ring: Multiplication is commutative, i.e., **a.b = b.a**

 Integral Domain: Multiplication operation has an identity and no zero divisors

Field:

An integral domain in which each element has a multiplicative inverse.



Polynomial Arithmetic

$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = \sum a_i x^i$

1. Ordinary polynomial arithmetic:
   - Add, subtract, multiply, divide polynomials,
   - Find remainders, quotient.
   - Some polynomials have no factors and are prime.
2. Polynomial arithmetic with mod p coefficients
3. Polynomial arithmetic with mod p coefficients and mod m(x) operations

Polynomial Arithmetic with Mod 2 Coefficients

- All coefficients are 0 or 1, e.g.,
  let $f(x) = x^3 + x^2$ and $g(x) = x^2 + x + 1$

$f(x) + g(x) = x3 + x + 1$

$f(x)$ x $g(x) = x5 + x2$

- Polynomial Division: $f(x) = q(x)\ g(x) + r(x)$
- can interpret $r(x)$ as being a remainder
- $r(x) = f(x)$ mod $g(x)$
- if no remainder, say $g(x)$ divides $f(x)$
- if $g(x)$ has no divisors other than itself & 1 say it is irreducible (or prime) polynomial
- Arithmetic modulo an irreducible polynomial form a finite field
- Can use Euclid‟s algorithm to find gcd and inverses.

Discrete Logarithm:

The inverse problem to exponential is to find the discreate logarithm of a number modulo P, that is to find i

$$b = a^i \ (mod\ p)$$

Written as

$$i = dlog_a b \ (mod\ p)$$

If a is a primitive root then it always exists, otherwise it may not.

Eg. $x = log_3 4 \ mod\ 13$ has no answer

$x = log_2 \ 3 \ mod\ 13 = 4$ by typing successive power

References

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Behrouz A. Ferouzan, “Cryptography & Network Security”, Tata McGraw Hill, 2007.
3. Man Young Rhee, “Internet Security: Cryptographic Principles”, “Algorithms and Protocols”, Wiley Publications, 2003.
4. Charles Pfleeger, “Security in Computing”, 4th Edition, Prentice Hall of India, 2006.