PRATHYUSHA ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

REGULATION R2021

III YEAR - V SEMESTER

CB3491 CRYPTOGRAPHY AND CYBER SECURITY

CB3491 CRYPTOGRAPHY AND CYBER SECURITY

UNIT I INTRODUCTION TO SECURITY

Computer Security Concepts – The OSI Security Architecture – Security Attacks – Security Services and Mechanisms – A Model for Network Security – Classical encryption techniques: Substitution techniques, Transposition techniques, Steganography – Foundations of modern cryptography: Perfect security – Information Theory – Product Cryptosystem – Cryptanalysis.

UNIT II SYMMETRIC CIPHERS

Number theory – Algebraic Structures – Modular Arithmetic – Euclid‘s algorithm – Congruence and matrices – Group, Rings, Fields, Finite Fields SYMMETRIC KEY CIPHERS: SDES – Block Ciphers – DES, Strength of DES – Differential and linear cryptanalysis – Block cipher design principles – Block cipher mode of operation – Evaluation criteria for AES – Pseudorandom Number Generators – RC4 – Key distribution.

UNIT III ASYMMETRIC CRYPTOGRAPHY

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes – Primality Testing – Factorization – Euler’s totient function, Fermat’s and Euler’s Theorem – Chinese Remainder Theorem – Exponentiation and logarithm ASYMMETRIC KEY CIPHERS: RSA cryptosystem – Key distribution – Key management – Diffie Hellman key exchange -– Elliptic curve arithmetic – Elliptic curve cryptography.

UNIT IV INTEGRITY AND AUTHENTICATION ALGORITHMS

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function: HMAC, CMAC – SHA – Digital signature and authentication protocols – DSS – Schnorr Digital Signature Scheme – ElGamal cryptosystem – Entity Authentication: Biometrics, Passwords, Challenge Response protocols – Authentication applications – Kerberos MUTUAL TRUST: Key management and distribution – Symmetric key distribution using symmetric and asymmetric encryption – Distribution of public keys – X.509 Certificates.

UNIT V CYBER CRIMES AND CYBER SECURITY

Cyber Crime and Information Security – classifications of Cyber Crimes – Tools and Methods –Password Cracking, Keyloggers, Spywares, SQL Injection – Network Access Control – Cloud Security – Web Security – Wireless Security

Text Books:

1. William Stallings, “Cryptography and Network Security – Principles and Practice”, Seventh Edition, Pearson Education, 2017.
2. Nina Godbole, Sunit Belapure, Cyber Security: Understanding Cyber crimes, Computer Forensics and Legal Perspectives, First Edition, Wiley India, 2011.

Reference Books:

1. Behrouz A. Ferouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, 3rd Edition, Tata Mc Graw Hill, 2015.
2. Charles Pfleeger, Shari Pfleeger, Jonathan Margulies, “Security in Computing”, Fifth Edition, Prentice Hall, New Delhi, 2015.

UNIT I INTRODUCTION TO SECURITY

SECURITY POLICIES:

- A **security policy** is a written document in an organization outlining how to protect the organization from threats, including computer **security** threats, and how to handle situations when they do occur.
- A **security policy** must identify all of a company's assets as well as all the potential threats to those assets
- Three main types of policies exist:
- Organizational (or Master) Policy.
- System-specific Policy.
- Issue-specific Policy.

Information security **objectives**

- Integrity—data should be intact, accurate and complete, and IT systems must be kept operational.
- Availability—users should be able to access information or systems when needed.

Structure of a Security Policy:

- Description of the Policy and what is the usage for?
- Where this policy should be applied?
- Functions and responsibilities of the employees that are affected by this policy.
- Procedures that are involved in this policy.
- Consequences if the policy is not compatible with company standards.

1. 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.

ITU-T Recommendation X.800, *Security Architecture for OSI*, defines such 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 was developed in the context of the OSI protocol architecture.

The OSI security architecture focuses on security attacks, mechanisms, and services:

Security attack: Any action that compromises the security of information owned by an organization.

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

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

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 assault 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.

S.No	Threats	Attacks
1.	Threat is an attack tends to be an act which is in process	An attack is a threat that has been executed.
2.	Threat can be either intentional or unintentional	Attack is intentional
3.	Threat is a circumstances that has potential to cause loss or damage	Attack is attempted to cause damage
4.	Threat to information system does not mean information was altered or damaged	Attack might cause alteration of information or damage or obtain information.

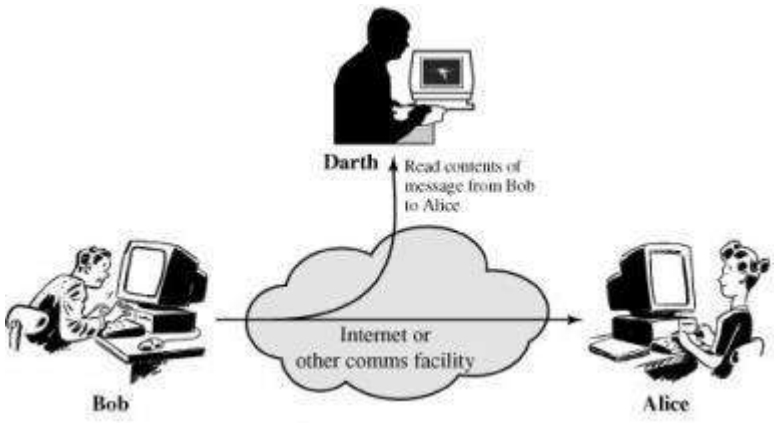
Table 1. Difference between Threats and Attacks

SECURITY ATTACKS:

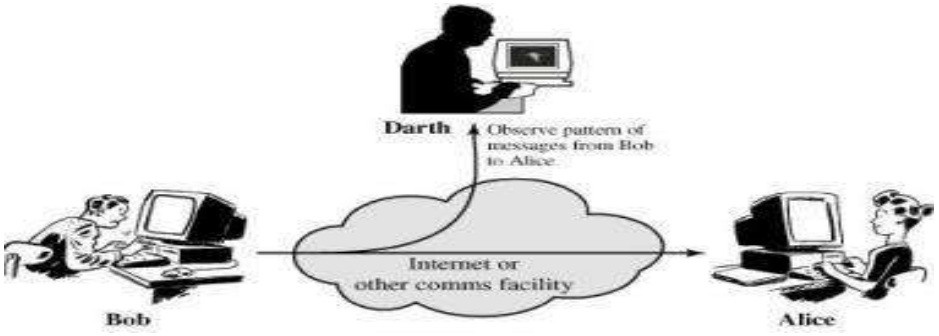
Security attacks could be broadly categorized as

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

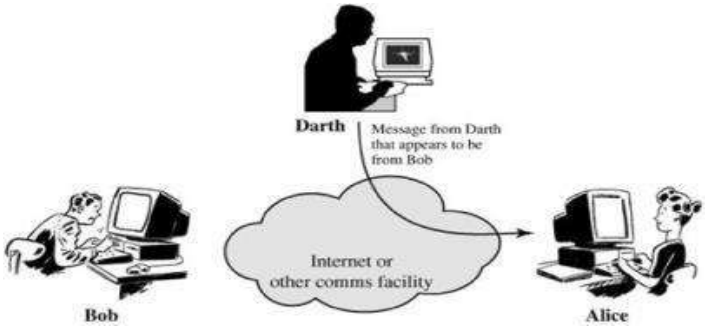


Traffic analysis: If we had encryption protection in place, an opponent might still be able to observe the pattern of the message. The opponent could determine the location and identity of communication hosts and could observe the frequency and length of messages being exchanged. This information might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

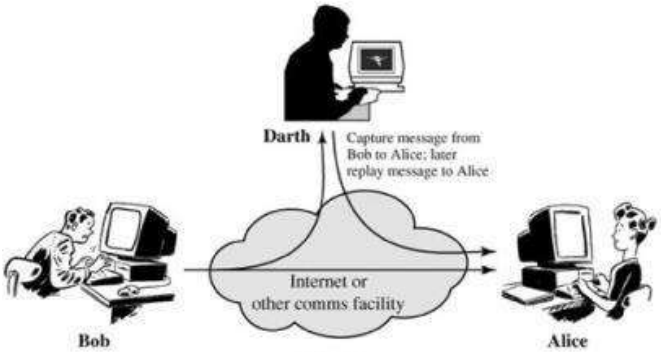


II. Active attacks: These attacks involve some modification of the data stream or the creation of a false stream. It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them. These attacks can be classified in to four categories:

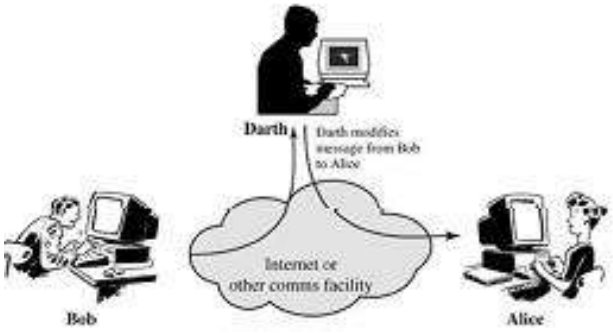
Masquerade (Fabrication)– One entity pretends to be a different entity.



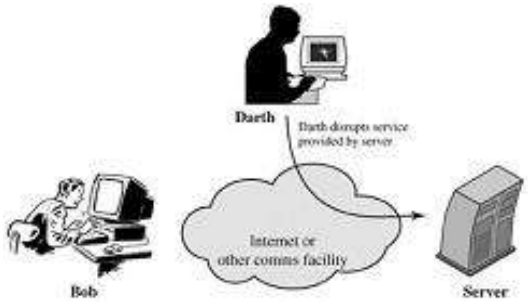
Replay – involves passive capture of a data unit and its subsequent transmission to produce an unauthorized effect.



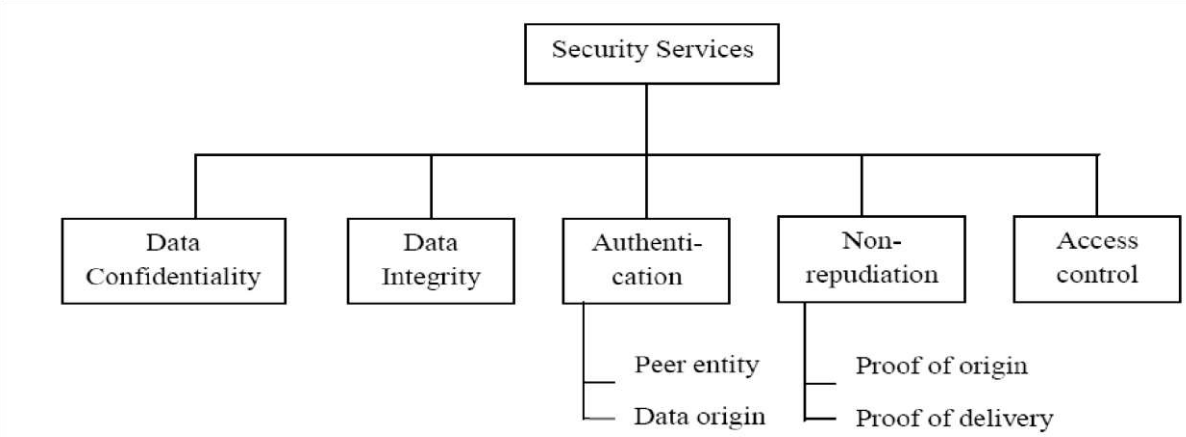
Modification of messages– Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.



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.



SECURITY SERVICES



(i) Authentication: The authentication service is concerned with assuring that a communication is authentic.

Two specific authentication services are defined in X.800:

- **Peer entity authentication:** Provide confidence in the identity of entities connected.
- **Data origin authentication:** Provide assurance that the source of received data is as claimed.

(ii) Access control: Access control is the ability to limit and control the access to host systems and applications.

(iii) Data Confidentiality: Confidentiality is the protection of transmitted data from passive attacks.

- **Connection Confidentiality -** The protection of all user data on a connection
- **Connectionless Confidentiality -** The protection of all user data in a single data block

- **Selective-Field Confidentiality -** The confidentiality of selected fields within the user data on a connection or in a single data block
- **Traffic-Flow Confidentiality -** The protection of the information that might be derived from observation of traffic flows

(iv) Data Integrity: The assurance that data received are exactly as sent by an authorized entity.

- **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.

(v) Non repudiation: 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 MECHANISM

- **Encipherment:**
  It uses mathematical algorithm to transform data into a form that is not readily intelligible. It depends upon encryption algorithm and key
- **Digital signature:**
  Data appended to or a cryptographic transformation of a data unit that is to prove integrity of data unit and prevents from forgery
- **Access control**
  A variety of mechanisms that enforce access rights to resources.
- **Data integrity**
  A variety of mechanism are used to ensure integrity of data unit

- **Traffic padding**
  The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.
- **Notarization**
  The use of a trusted third party to assure certain properties of a data exchange

SECURITY MECHANISM (X.800 Standard)

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.

Table 1.4 Relationship Between Security Services and Mechanisms

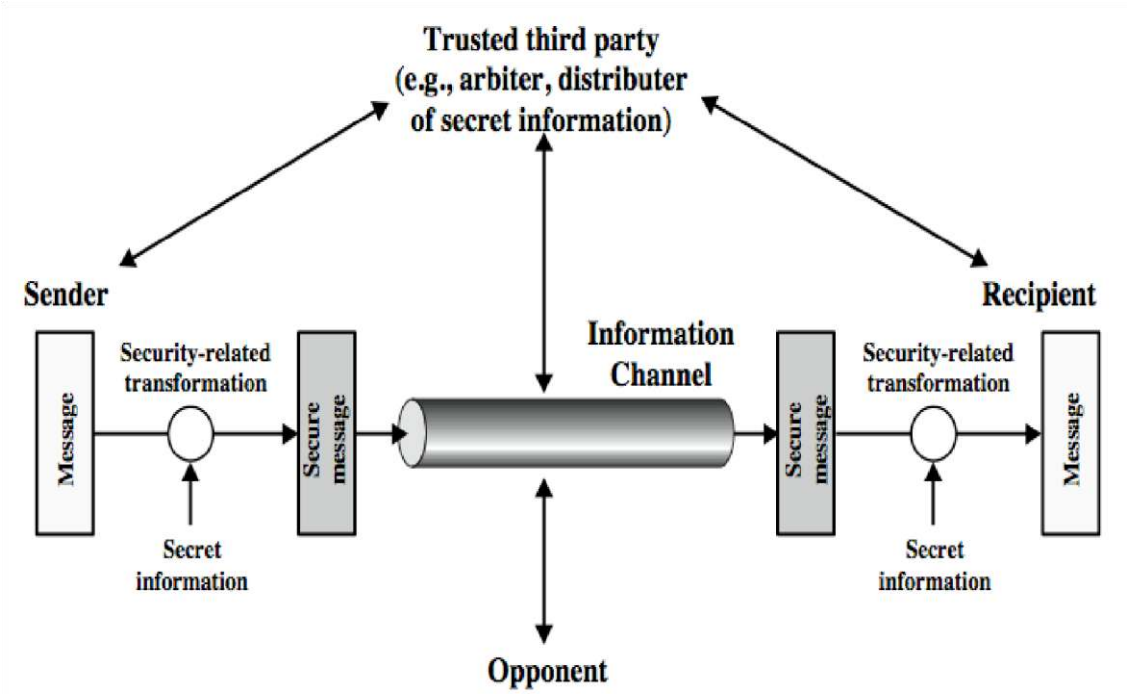
	Mechanism							
Service	**Encipherment**	**Digital Signature**	**Access Control**	**Data Integrity**	**Authentication Exchange**	**Traffic Padding**	**Routing Control**	**Notarization**
Peer Entity Authentication	Y	Y			Y			
Data Origin Authentication	Y	Y						
Access Control			Y					
Confidentiality	Y						Y	
Traffic Flow Confidentiality	Y					Y	Y	
Data Integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

2. A MODEL FOR NETWORK SECURITY

Encryption/Decryption methods fall into two categories.

- Symmetric key
- Public key

In symmetric key algorithms, the encryption and decryption keys are known both to sender and receiver. The encryption key is shared and the decryption key is easily calculated from it. In many cases, the encryption and decryption keys are the same. In public key cryptography, encryption key is made public, but it is computationally infeasible to find the decryption key without the information known to the receiver.



A message is to be transferred from one party to another across some sort of internet. 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.

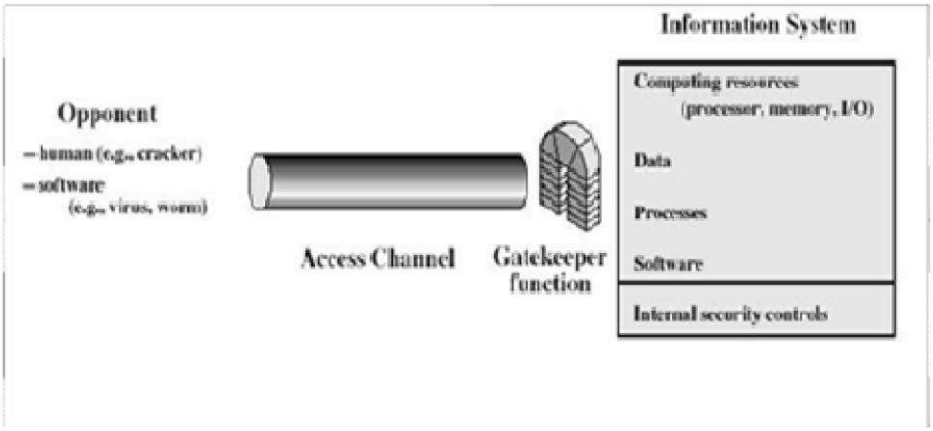
All the techniques for providing security have two components: ○ A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent.

- ○ 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

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

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

MODEL FOR NETWORK ACCESS SECURITY



Using this model requires us to:

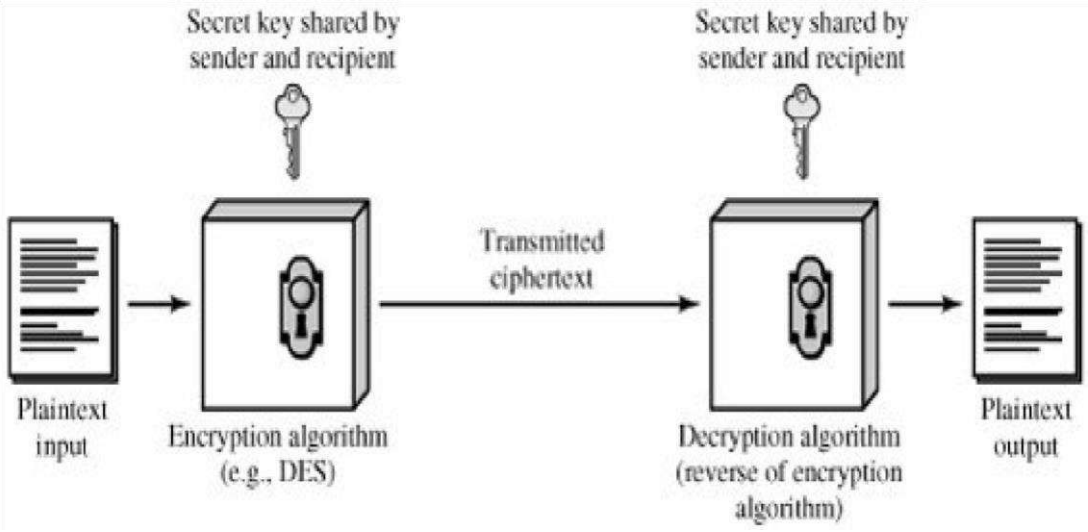
– select appropriate gatekeeper functions to identify users

– implement security controls to ensure only authorized users access designated information or resources

- Trusted computer systems can be used to implement this model

3. CLASSICAL ENCRYPTION TECHNIQUES

Symmetric encryption also referred to as conventional encryption or single-key encryption. Here, the sender and recipient share a common key.



A symmetric encryption scheme has five ingredients

Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.

Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

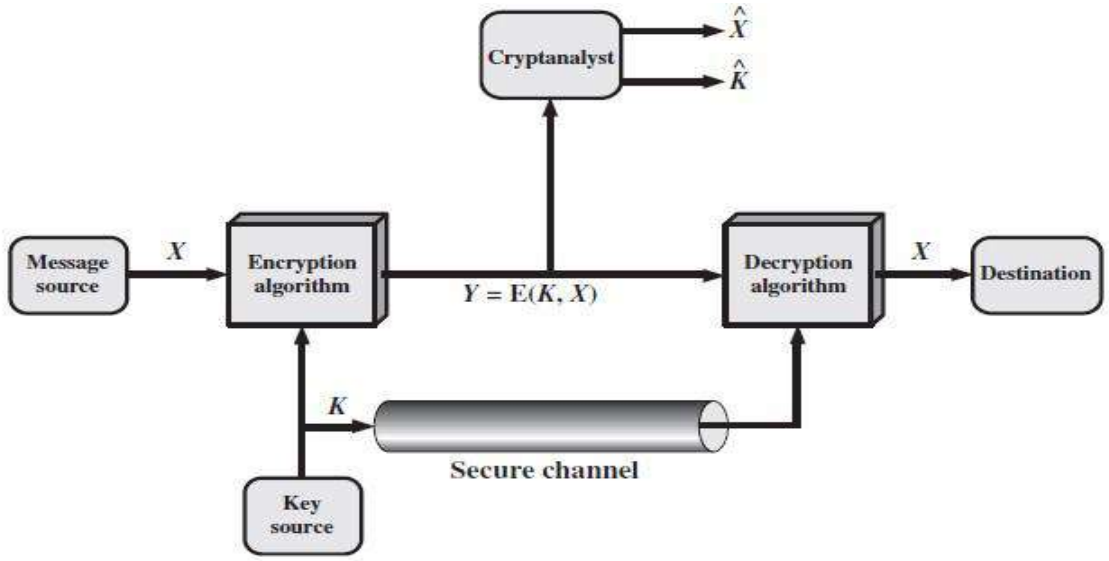
Cipher text: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different cipher texts. The cipher text is an apparently random stream of data and, as it stands, is unintelligible.

Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext.

There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.

It is impractical to decrypt a message on the basis of the cipher text *plus* knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret.



Model of symmetric cryptosystem

A source produces a message in plaintext X

$= [X_1, X_2, ..., X_M]$.

M- elements of X are letters.

For encryption, a key of the form

$K = [K_1, K_2, \ldots, K_J]$ is generated.

If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

With the message X and the encryption key K as input, the encryption algorithm forms the cipher text

$Y = [Y_1, Y_2, \ldots, Y_N]$.

$Y = E(K, X)$

Y- cipher text

E- Encryption algorithm

K- Key

X-Plain text

At the receiver side the transformation:

$X = D(K, Y)$

Y- cipher text

D-Decryption algorithm

K- Key

X-Plain text

If the opponent is interested in only this particular message only, tries to find the message estimate $\hat{X}$. But when the opponent is interested in the current and future messages, tries to find key estimate $\hat{K}$.

Cryptographic systems are generally classified along 3 independent dimensions:

Type of operations used for transforming plain text to cipher text

All the encryption algorithms are based on two general principles:

- **Substitution**, in which each element in the plaintext is mapped into another element
- T**ransposition**, in which elements in the plaintext are rearranged.

The number of keys used

- If the sender and receiver uses same key then it is said to be **symmetric key (or) single key (or) conventional encryption**.
- If the sender and receiver use different keys then it is said to be **public key encryption**.

The way in which the plain text is processed

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

CRYPTANALYSIS AND BRUTE-FORCE ATTACK

There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm and some knowledge of the general characteristics of the plaintext or even some sample plaintext–cipher text pairs.
- **Brute-force attack:** The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.

There are various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst.

Type of Attack	Known to Cryptanalyst
Cipher text Only	• Encryption algorithm • Cipher text
Known Plaintext	• Encryption algorithm • Cipher text • One or more plaintext–cipher text pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Cipher text • Plaintext message chosen by cryptanalyst, together with its corresponding Cipher text generated with the secret key
Chosen Cipher text	• Encryption algorithm • Cipher text • Cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Cipher text

	• Plaintext message chosen by cryptanalyst, together with its corresponding Cipher text generated with the secret key • Cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Encryption algorithms are to be

- **Unconditionally secure**
- **Computationally secure**

An encryption scheme is **unconditionally secure** if the cipher text generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext.

An encryption scheme is said to be **computationally secure**

- If the cost of breaking the cipher exceeds the value of the encrypted information
- If the time required to break the cipher exceeds the useful lifetime of the information.

4. SUBSTITUTION TECHNIQUES

- A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- Substitution ciphers can be categorized as either

  i) Monoalphabetic ciphers or ii) polyalphabetic ciphers.
- In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.
- In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the cipher text is one-tomany.

VARIOUS SUBSTITUTION CIPHERS ARE

(i) Caesar Cipher or Shift Cipher Or Additive Cipher (Monoalphabetic Cipher)

(ii) Playfair cipher

(iii) Hill cipher

(iv) Vignere cipher (Poly alphabetic cipher)

(v) Vernam Cipher or One time pad (Poly alphabetic cipher)

(i)CAESAR CIPHER (OR) SHIFT CIPHER

Caeser cipher was proposed by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

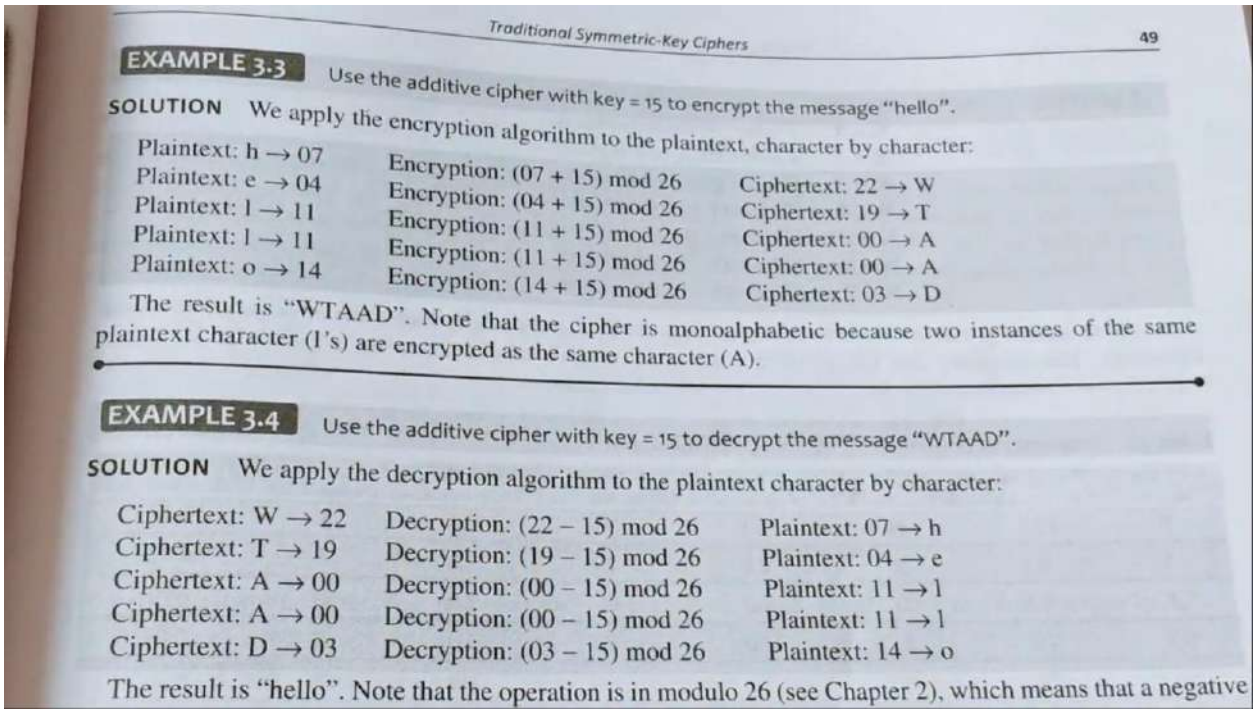
```
plain:  meet me after the toga party
cipher: PHHW PH DIWHU WKH WRJD SDUWB
```

```
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



Traditional Symmetric-Key Ciphers 49

EXAMPLE 3.3 Use the additive cipher with key = 15 to encrypt the message "hello".

SOLUTION We apply the encryption algorithm to the plaintext, character by character:

Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

The result is "WTAAD". Note that the cipher is monoalphabetic because two instances of the same plaintext character (l's) are encrypted as the same character (A).

EXAMPLE 3.4 Use the additive cipher with key = 15 to decrypt the message "WTAAD".

SOLUTION We apply the decryption algorithm to the plaintext character by character:

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

The result is "hello". Note that the operation is in modulo 26 (see Chapter 2), which means that a negative

Note that the alphabet is wrapped around, so that letter following 'z' is 'a'.

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

C = **E(3, P = (P+3) mod 26**

Decryption is

P=D(3,c)=(C-3) mod 26

The general Caesar algorithm is

C = E(K, P) = (P + K) mod 26 where

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

The decryption algorithm is simply

P = D(K, C) = (C - K) mod 26

If it is known that a given cipher text is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 26 possible keys.

Cryptanalysis of Caesar Cipher

1. The encryption and decryption algorithms are known
2. There are only 26 possible keys. Hence brute force attack takes place
3. The language of the plaintext is known and easily recognizable

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

Brute-Force Cryptanalysis of Caesar Cipher

(ii) MONOALPHABETIC CIPHER

- Each plaintext letter maps to a different random cipher text letter
- Here, 26! Possible keys are used to eliminate brute force attack

There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext (e.g., non-compressed English text), then the analyst can exploit the regularities of the language.

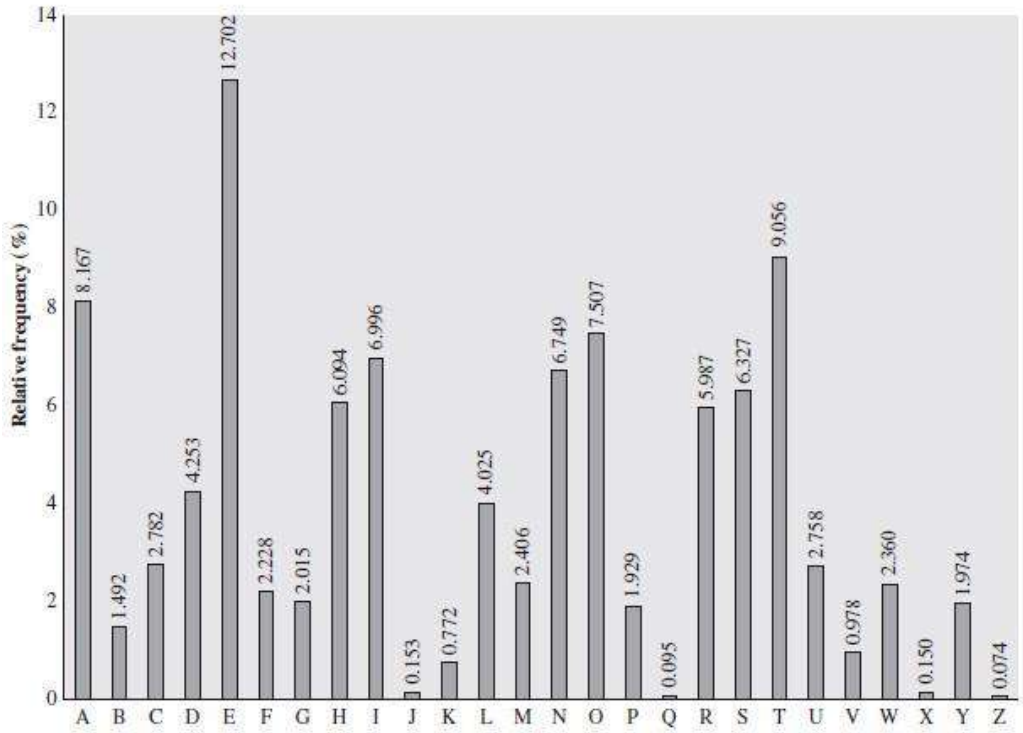
MONOALPHABETIC CIPHER:

- A Single Cipher alphabets for each plain text alphabet is used throughout the process.

- Rather than just shifting the alphabet could shuffle (jumble) the letters arbitrarily
- Relationship between a character in the plain text to a symbol in cipher text is always one to one
- Assume the below table(without the repetition of letter, any letters can be mapped)
  Plain: a b c de f g h I j k l m n o p q r s t u v w x y z
  Cipher: DKVQF IB JWPESCXHTMYAUOLRGZN
- **Example 1(By referring the above table)**
  Plain text : MY NAME
  Cipher Text: CZ XDCF
- From the above example, we say that wherever we use M. The M letter can replaced by C. (i.e) A Single Cipher alphabets for each plain text alphabet is used throughout the process.

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				



Relative frequency of letters in English text

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a        e  e te  a that e e a       a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t   ta t ha e ee  a e  th    t  a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e  e e tat  e   the   t
```

Only four letters have been identified, but already we have quite a bit of the message. Continued analysis of frequencies plus trial and error should easily yield a solution from this point. The complete plaintext, with spaces added between words, follows:

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

(iii) 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. The playfair algorithm is based on the use of 5x5 matrix of letters constructed using a keyword.

Let the keyword be "monarchy".

The matrix is constructed by

- Filling in the letters of the keyword from left to right and from top to bottom
- Duplicates are removed
- Remaining unfilled cells of the matrix is filled with remaining alphabets in alphabetical order.

The matrix is 5x5. It can accommodate 25 alphabets. To accommodate the 26th alphabet I and J are counted as one character.

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

Rules for encryption

- Repeating plaintext letters that would fall in the same pair are separated with a filler letter such as 'x'.
- 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 (or JM, as the encipherer wishes).

Example

Plain text: Balloon

Ba ll oo n

Ba lx lo on

Ba❼I/JB

lx❼SU lo❼PM

on❼NA

Strength of playfair cipher

 Playfair cipher is a great advance over simple mono alphabetic ciphers.

 Since there are 26 letters, 26x26 = 676 diagrams are possible, so identification of individual digram is more difficult.

Frequency analysis is much more difficult.

Disadvantage

Easy to break because it has the structure and the resemblance of the plain text language

(iv) HILL CIPHER

It is a multi-letter cipher. It is developed by Lester Hill. The encryption algorithm takes m successive plaintext letters and substitutes for them m cipher text letters. The substitution is determined by m linear equations in which each character is assigned numerical value (a=0,b=1…z=25). For m =3 the system can be described as follows:

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

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11}K_{12}K_{13} \\ K_{21}K_{22}K_{23} \\ K_{31}K_{32}K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

C=KP mod 26

C and P are column vectors of length 3 representing the cipher and plain text respectively. Consider the message 'ACT', and

$$\begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix}$$

The key below (or GYBNQKURP in letters)

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$$

Thus the enciphered vector is given by:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} = \begin{pmatrix} 67 \\ 222 \\ 319 \end{pmatrix} \equiv \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \pmod{26}$$

which corresponds to a ciphertext of 'POH'

Decryption

Decryption algorithm is done as **$P=K^{-1}C \bmod 26$**

In order to decrypt, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters).

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}^{-1} \equiv \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix}$$

Cipher text of 'POH'

$$\begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \equiv \begin{pmatrix} 260 \\ 574 \\ 539 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} \pmod{26}$$

Now gets us back the plain text 'ACT'

Merits and Demerits

- Completely hides single letter and 2 letter frequency information.
- Easily attacked with known plain text attack

(v)POLYALPHABETIC CIPHERS

Poly alphabetic cipher is a simple technique to improve mono-alphabetic technique.

- There is no fixed substitution
- Each Occurrence of a character may have a different substitute (i.e) we can use more than one substitution for the same letter

Plain Text : MY NAME

Cipher Text: NP OBXZ (here, M Letter can be replaced with N and X. No fixed substitute)

The features are

- A set of related mono-alphabetic substitution rules are used
-

A key determines which particular rule is chosen for a given transformation.

Example: **Vigenere Cipher**

Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of encryption is simple: Given a key letter x and a plaintext letter y, the cipher text is at the intersection of the row labelled x and the column labelled y; in this case, the cipher text is V. To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword.

Key=deceptive

Plain text= we are discovered save yourself

e.g., key = d e c e p t i v e d e c e p t i v e d e c e p t i v e

PT = w e a r e d i s c o v e r e d s a v e y o u r s e l f

CT = ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the cipher text letter in that row determines the column, and the plaintext letter is at the top of that column.

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

EXAMPLE 2 :

```
key:            deceptivedeceptivedeceptive
plaintext:      wearediscoveredsaveyourself
ciphertext:     ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Expressed numerically, we have the following result.

key	3	4	2	4	15	19	8	21	4	3	4	2	4	15
plaintext	22	4	0	17	4	3	8	18	2	14	21	4	17	4
ciphertext	25	8	2	21	19	22	16	13	6	17	25	6	21	19

key	19	8	21	4	3	4	2	4	15	19	8	21	4
plaintext	3	18	0	21	4	24	14	20	17	18	4	11	5
ciphertext	22	0	21	25	7	2	16	24	6	11	12	6	9

Strength of Vigenere cipher o There are multiple ciphertext letters for each plaintext letter. o Letter frequency information is obscured

(vi) VERNAM CIPHER or ONE-TIME PAD

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:

$$C_i = P_i \oplus K_i$$

C_i - ith binary digit of cipher text P_i - i^{th} binary digit of plaintext K_i - ith binary digit of key

$\oplus$ – exclusive OR operation

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:

$$P_i = C_i \oplus K_i$$

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

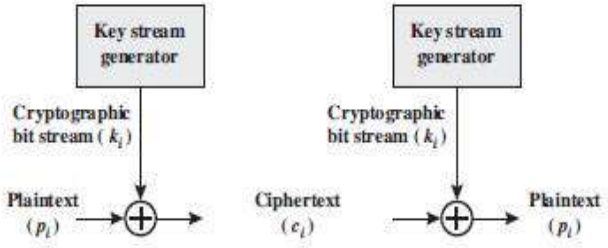


Figure 2.7 Vernam Cipher

Advantages

- It is unbreakable since cipher text bears no statistical relationship to the plaintext

- Not easy to break **Drawbacks**
- Practically impossible to generate a random key as to the length of the message
- The second problem is that of key distribution and key protection.

Due to the above two drawbacks, one time pad is of limited use and is used for low band width channel which needs high security.

5. TRANSPOSITION TECHNIQUES

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. i) **Rail Fence Cipher**

It 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 Cipher text MEATECOLOSETTHSHOHUE **ii)**

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 x

CT = ESOTCUEEHMHLAHSTOETOX

Demerits

- Easily recognized because the frequency is same in both plain text and cipher text.
- Can be made secure by performing more number of transpositions.

6. STEGANOGRAPHY

In Steganography, the plaintext is hidden. The existence of the message is concealed. For example, the sequence of first letters of each word of the overall message spells out the hidden message. Various other techniques have been used historically; some examples are the following:

- **Character marking:** Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at 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 up in front of a light.

- **Typewriter correction ribbon:** Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

Drawback

- It requires a lot of overhead to hide a relatively few bits of information.
- Once the system is discovered, it becomes virtually worthless

MODERN CRYPTOGRAPHY:

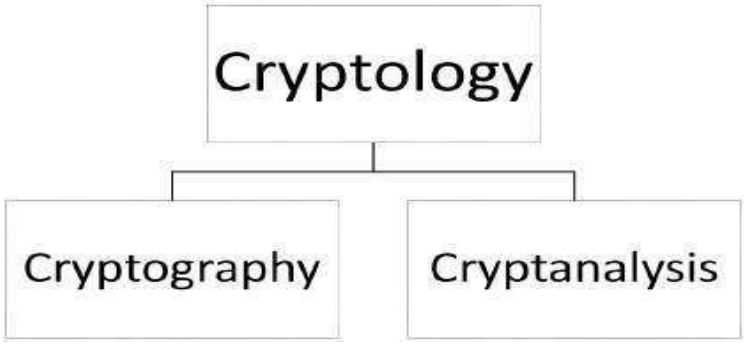
Modern cryptography is the cornerstone of computer and communications security. Its foundation is based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory.

Difference between Classical Cryptography and Modern Cryptography:

Classic Cryptography	Modern Cryptography
It manipulates traditional characters, i.e., letters and digits directly.	It operates on binary bit sequences.
It is mainly based on 'security through obscurity'.	It relies on publicly known mathematical algorithms for coding the information.
It requires the entire cryptosystem for communicating confidentially.	Modern cryptography requires parties interested in secure communication to possess the secret key only.

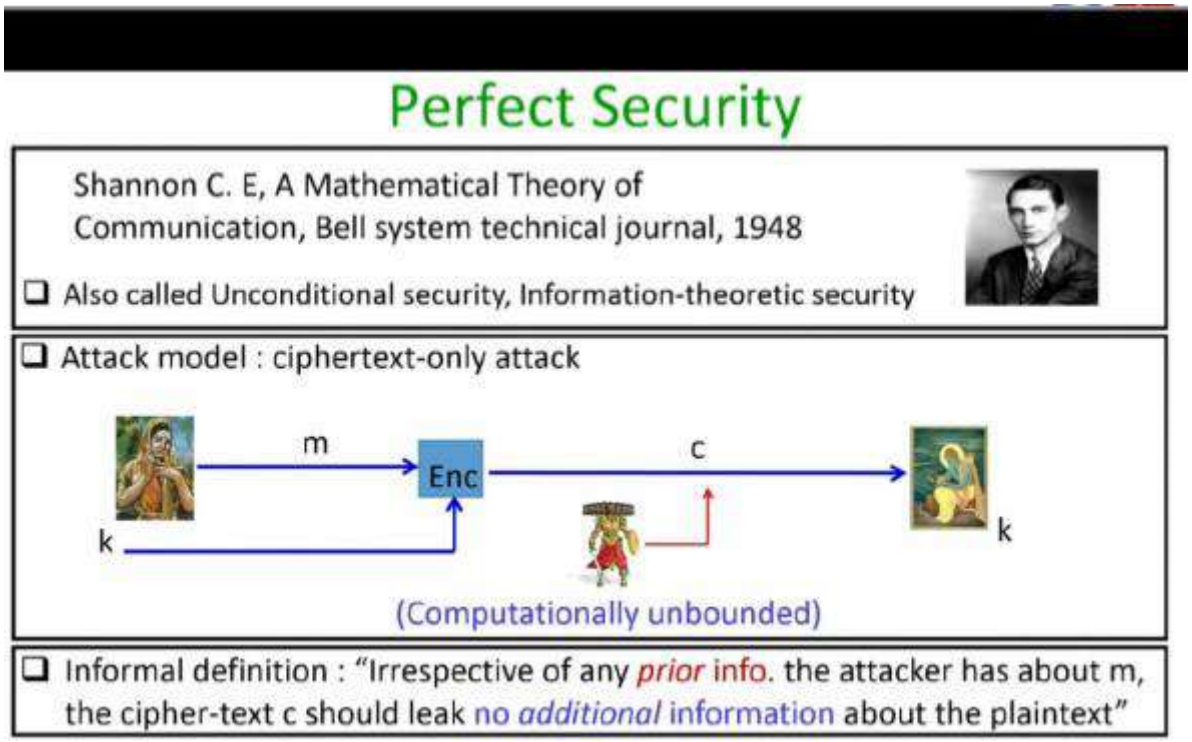
Cryptology:

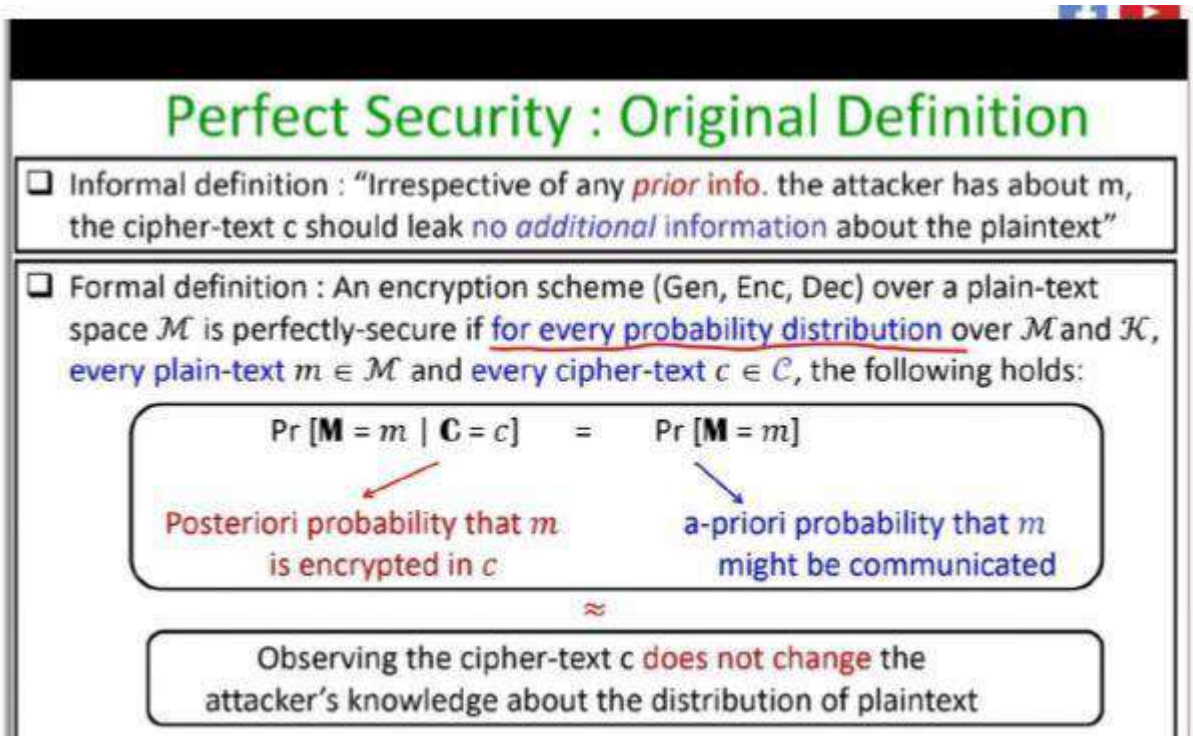
- Cryptography is the art and science of making a cryptosystem that is capable of providing information security.
- The art and science of breaking the cipher text is known as Cryptanalysis
- Cryptology is the study of codes, both creating and solving them.

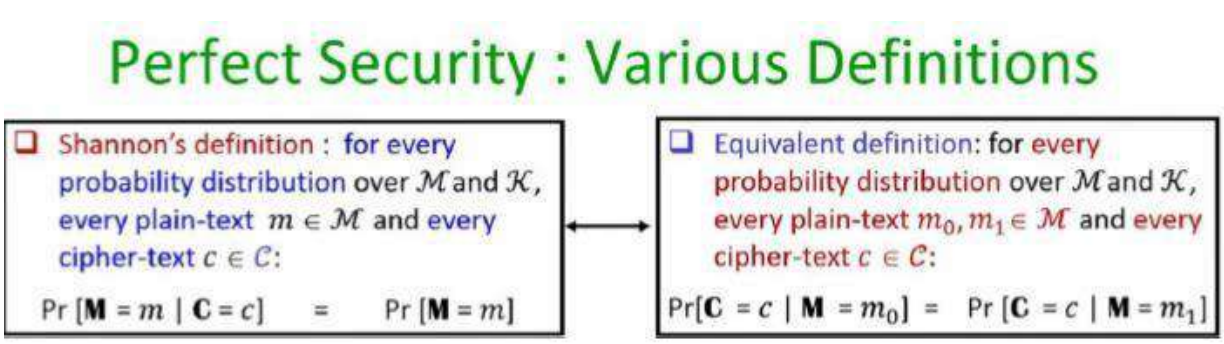


Perfect Security:

Perfect Secrecy (or Information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext. In effect this means that, no matter how much ciphertext you have, it does not convey anything about what the plaintext and key were.







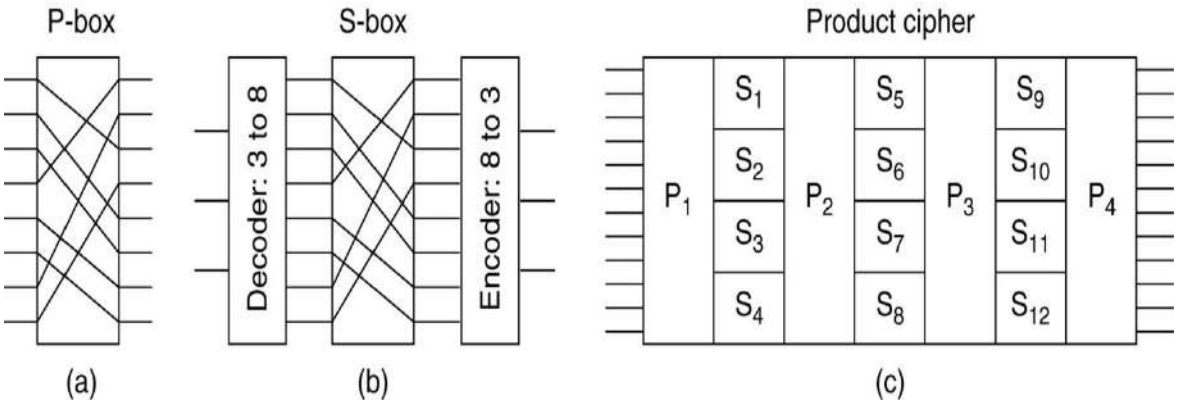
Interpretation:
1)Probability of knowing a plain text remains the same before and after seeing the cipher text
2)Probability distribution of cipher text is independent of Plain text.
Perfect secrecy has nothing to do with plaintext distribution. Thus a crypto-scheme will achieve perfect secrecy irrespective of the language used in the plaintext.

Information theory:

- Concepts, methods and results from coding theory and information theory are widely used in cryptography and cryptanalysis. See the article ban (unit) for a historical application.
- Information theory is also used in information retrieval, intelligence gathering, gambling, statistics, and even in musical composition.
- A key measure in information theory is Entropy. Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process.

- In terms of Cryptography, entropy must be supplied by the cipher for injection into the plaintext of a message so as to neutralise the amount of structure that is present in the unsecure plaintext message.

Product Cryptosystem:



Basic elements of product ciphers.
(a) P-box. (b) S-box. (c) Product.

Cryptanalysis:

Cryptanalysis is the study of ciphertext, ciphers and cryptosystems with the aim of understanding how they work and finding and improving techniques for defeating or weakening them.

UNIT II SYMMETRIC KEY CRYPTOGRAPHY

SYMMETRIC KEY CRYPTOGRAPHY: Algebraic structures - Modular arithmetic-Euclid‟s algorithm-Congruence and matrices - Groups, Rings, Fields- Finite fields **SYMMETRIC KEY CIPHERS:** SDES – Block cipher Principles of DES – Strength of DES – Differential and linear cryptanalysis - Block cipher design principles – Block cipher mode of operation – Evaluation criteria for AES – Advanced Encryption Standard - RC4 – Key distribution.

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.

MODULAR ARITHMETIC

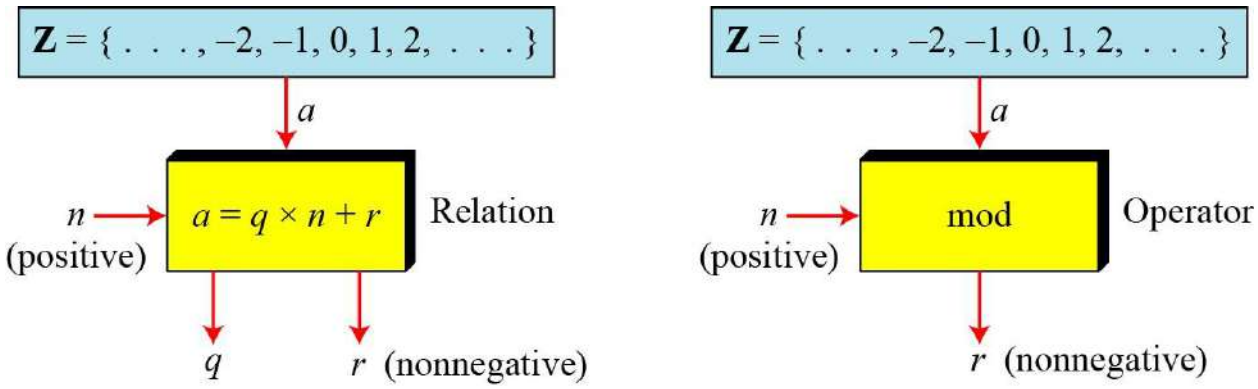
If *a* is an integer and *n* is a positive integer, we define *a* mod *n* to be the remainder when *a* is divided by *n*. The integer *n* is called the **modulus**.

$$a = qn + r \qquad 0 \le r < n;$$

$$q = \lfloor a/n \rfloor$$

The division relationship (a = q × 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.

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



Examples

Find the result of the following operations:

a. 27 mod 5 **b. 36 mod 12**

Solution

a. Dividing 27 by 5 results in r = 2
b. Dividing 36 by 12 results in r = 0.

CONGRUENCE

Two integers a and b are said to be congruent modulo n if

a (mod n)≡ b (mod n) a ≡ b (mod n) 73 ≡ 4 mod 23

Properties of Congruences

Congruences have the following properties:

1. a≡ b (mod n) if n|(a-b)
2. a≡ b (mod n) implies b≡ a(mod n)
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

To show that two integers are congruent, we use the congruence operator (≡). For example, we write:

$$2 \equiv 12 \pmod{10} \qquad 13 \equiv 23 \pmod{10}$$
$$3 \equiv 8 \pmod{5} \qquad 8 \equiv 13 \pmod{5}$$

Example:

Perform the following operations (the inputs come from Zn): a.
Add 7 to 14 in Z15.
b. Subtract 11 from 7 in Z13.
c. Multiply 11 by 7 in Z20.

Solution

$$(14 + 7) \bmod 15 \rightarrow (21) \bmod 15 = 6$$
$$(7 - 11) \bmod 13 \rightarrow (-4) \bmod 13 = 9$$
$$(7 \times 11) \bmod 20 \rightarrow (77) \bmod 20 = 17$$

MODULAR ARITHMETIC OPERATIONS

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$

2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$

3. $[(a \bmod n) * (b \bmod n)] \bmod n = (a * b) \bmod n$

Example: 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

RELATIVELY PRIME

Two integers are **relatively prime,** if their only common positive integer factor is 1.
8 and 15 are relatively prime because
Positive divisors of 8 are 1,2,4,8
Positive divisors of 15 are 1, 3, 5, 15 Therefore,
common positive factor=1.

EUCLIDEAN ALGORITHM

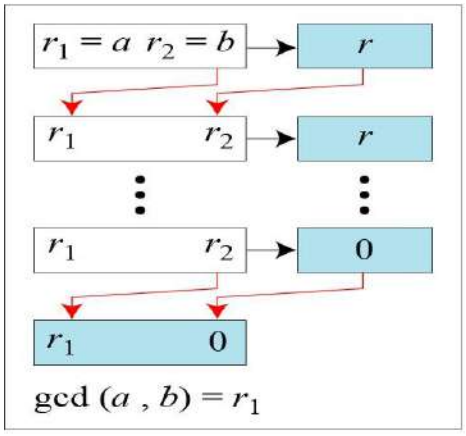
Euclidean algorithm is a simple procedure for determining the greatest common divisor of two positive integers.

The positive integer *c* is said to be the greatest common divisor of *a* and *b* if

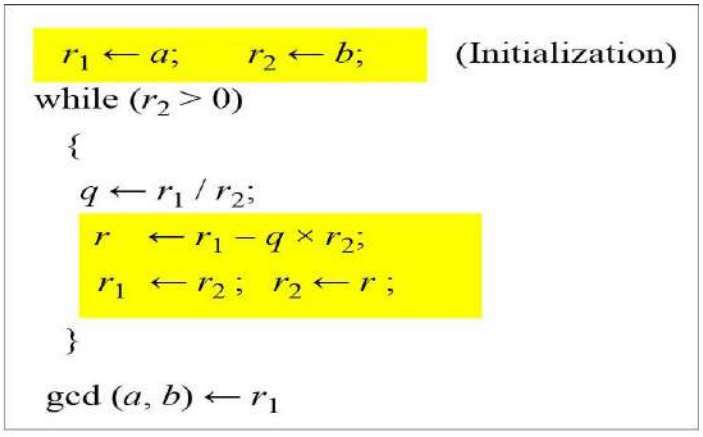
1. *c* is a divisor of *a* and of *b*.
2. Any divisor of *a* and *b* is a divisor of *c*.

Fact 1: gcd (a, 0) = a

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



a. Process



b. Algorithm

EUCLID(*a*, *b*)

1. A← *a*; B←*b*
2. if B = 0 **return** A = gcd(*a*, *b*)
3. R = A mod B
4. A←B
5. B←R
6. goto 2

Euclidean Algorithm Revisited
For any integers *a, b,* with *a* ≥ *b* ≥ 0, gcd(*a, b*) = gcd(*b, a* mod *b*)

Example 1 gcd(55, 22) = gcd(22, 55 mod 22) = gcd(22, 11) = 11 gcd(18, 12) = gcd(12, 6) = gcd(6, 0) = 6 gcd(11, 10) = gcd(10, 1) = gcd(1, 0) = 1

Example 2 Find the greatest common divisor of 2740 and 1760. Solution:
We have gcd (2740, 1760) = 20.

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	

Example 3 Find the greatest common divisor of 25 and 60. Solution
: We have gcd (25, 65) = 5.

q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

Euclidean Algorithm	
Calculate	**Which satisfies**
$r_1 = a \bmod b$	$a = q_1 b + r_1$
$r_2 = b \bmod r_1$	$b = q_2 r_1 + r_2$
$r_3 = r_1 \bmod r_2$	$r_1 = q_3 r_2 + r_3$
•	•
•	•
•	•
$r_n = r_{n-2} \bmod r_{n-1}$	$r_{n-2} = q_n r_{n-1} + r_n$
$r_{n+1} = r_{n-1} \bmod r_n = 0$	$r_{n-1} = q_{n+1} r_n + 0$ $d = \gcd(a, b) = r_n$

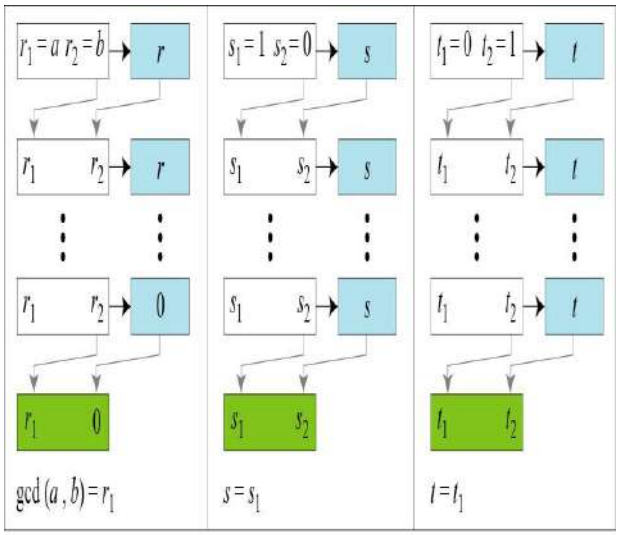
Recursive function: If (b=0)
then return a; else return
Euclid(b, a mod b);

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*.



a. Process

$r_1 \leftarrow a$; $r_2 \leftarrow b$;
$s_1 \leftarrow 1$; $s_2 \leftarrow 0$;
$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;
(Initialization)

while ($r_2 > 0$)
{
$q \leftarrow r_1 / r_2$;

$r \leftarrow r_1 - q \times r_2$;
$r_1 \leftarrow r_2$; $r_2 \leftarrow r$;
(Updating *r*'s)

$s \leftarrow s_1 - q \times s_2$;
$s_1 \leftarrow s_2$; $s_2 \leftarrow s$;
(Updating *s*'s)

$t \leftarrow t_1 - q \times t_2$;
$t_1 \leftarrow t_2$; $t_2 \leftarrow t$;
(Updating *t*'s)

}
gcd (*a* , *b*) $\leftarrow r_1$; $s \leftarrow s_1$; $t \leftarrow t_1$

b. Algorithm

Example1: Given a = 161 and b = 28, find gcd (a, b) and the values of s and t.

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	

Solution

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

POLYNOMIAL ARITHMETIC

A **polynomial** of degree n (integer $n \geq 0$) is an expression of the form

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$$

where the a_i – *coefficients*

$$f(x) = \sum_{i=0}^{n} a_i x^i; \quad g(x) = \sum_{i=0}^{m} b_i x^i; \quad n \geq m$$

Addition is defined as

$$f(x) + g(x) = \sum_{i=0}^{m} (a_i + b_i)x^i + \sum_{i=m+1}^{n} a_i x^i$$

Multiplication is defined as

$$f(x) \times g(x) = \sum_{i=0}^{n+m} c_i x^i$$

where

$$c_k = a_0 b_k + a_1 b_{k-1} + \cdots + a_{k-1} b_1 + a_k b_0$$

Eg.: Let $f(x) = x^3 + x^2 + 2$ and $g(x) = x^2 - x + 1$, where S is the set of integers. Then
$f(x) + g(x) = x^3 + 2x^2 - x + 3$ $f(x) - g(x) = x^3 + x + 1$ $f(x) * g(x) = x^5 + 3x^2 - 2x + 2$

Example 1 Find gcd[$a(x)$, $b(x)$] for $a(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ and $b(x) = x^4 + x^2 + x + 1$.
Euclidean algorithm to compute the greatest common divisor of two polynomials gcd[$a(x)$, $b(x)$] = gcd[$b(x)$, $a(x)$ mod $b(x)$]

=gcd(b(x),r1(x))

=gcd[r1(x),b(x) mod r1(x)]

$$
\begin{array}{r}
x^2 + x \\
x^4 + x^2 + x + 1 \overline{)\, x^6 + x^5 + x^4 + x^3 + x^2 + x + 1} \\
\underline{x^6 \quad + x^4 + x^3 + x^2} \\
x^5 \quad + x + 1 \\
\underline{x^5 \quad + x^3 + x^2 + x} \\
x^3 + x^2 \quad + 1
\end{array}
$$

This yields $r_1(x) = x^3 + x^2 + 1$ and $q_1(x) = x^2 + x$.
Then, we divide $b(x)$ by $r_1(x)$.

$$
\begin{array}{r}
x + 1 \\
x^3 + x^2 + 1 \overline{)\, x^4 \quad + x^2 + x + 1} \\
\underline{x^4 + x^3 \quad + x} \\
x^3 + x^2 \quad + 1 \\
\underline{x^3 + x^2 \quad + 1}
\end{array}
$$

This yields $r_2(x) = 0$ and $q_2(x) = x + 1$.
Therefore, $\gcd[a(x), b(x)] = r_1(x) = x^3 + x^2 + 1$.

MULTIPLICATIVE INVERSE

It is easy to find the multiplicative inverse of an element in GF(p) for small values of p *by* constructing a multiplication table, such as shown in Table and the desired result can be read directly. However, for large values of p, this approach is not practical.

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

(a) Addition modulo 7

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

(b) Multiplication modulo 7

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

(c) Additive and multiplicative inverses modulo 7

If a and b are relatively prime, then b has a multiplicative inverse modulo a. That is, if gcd(a, b) = 1, then b has a multiplicative inverse modulo a. That is, for positive integer $b < a$, there exists a b-1 < a such that $bb^{-1} = 1 \bmod a$.

If a is a prime number and $b < a$, then clearly a and b are relatively prime and have a greatest common divisor of 1. We now show that we can easily compute b-1 using the extended Euclidean algorithm.

Finding the

Polynomial Arithmetic Modulo $(x^3 + x + 1)$

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

(a) Addition

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

(b) Multiplication

Multiplicative Inverse of a polynomial

CONGRUENCE AND MATRICES

For a positive integer n, two integers a and b are said to be congruent modulo n (or a is congruent to b modulo n), if a and b have the same remainder when divided by n (or equivalently if a − b is divisible by n). It can be expressed as a ≡ b mod n.

Matrices: ***A matrix of size l X m***

$$\text{Matrix } \mathbf{A}: \quad \overset{l \text{ rows}}{\overset{\overbrace{\qquad\qquad}^{m \text{ columns}}}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{l1} & a_{l2} & \cdots & a_{lm} \end{bmatrix}}}$$

Examples of matrices

$$\begin{bmatrix} 2 & 1 & 5 & 11 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} \quad \begin{bmatrix} 23 & 14 & 56 \\ 12 & 21 & 18 \\ 10 & 8 & 31 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Row matrix | Column matrix | Square matrix | **0** | **I**

Operations and Relations: **Addition and subtraction of matrices**

$$\begin{bmatrix} 12 & 4 & 4 \\ 11 & 12 & 30 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} + \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

$$\mathbf{C = A + B}$$

$$\begin{bmatrix} -2 & 0 & -2 \\ -5 & -8 & 10 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} - \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

$$\mathbf{D = A - B}$$

Multiplication of a row matrix by a column matrix:

$$\underset{\mathbf{C}}{\begin{bmatrix} 53 \end{bmatrix}} = \underset{\mathbf{A}}{\begin{bmatrix} 5 & 2 & 1 \end{bmatrix}} \times \underset{\mathbf{B}}{\begin{bmatrix} 7 \\ 8 \\ 2 \end{bmatrix}}$$

In which: $53 = 5 \times 7 + 2 \times 8 + 1 \times 2$

Multiplication of a 2 × 3 matrix by a 3 × 4 matrix

$$\underset{\mathbf{C}}{\begin{bmatrix} 52 & 18 & 14 & 9 \\ 41 & 21 & 22 & 7 \end{bmatrix}} = \underset{\mathbf{A}}{\begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix}} \times \overset{\mathbf{B}}{\begin{bmatrix} 7 & 3 & 2 & 1 \\ 8 & 0 & 0 & 2 \\ 1 & 3 & 4 & 0 \end{bmatrix}}$$

Scalar multiplication

$$\mathbf{B} \qquad\qquad \mathbf{A}$$

$$\begin{bmatrix} 15 & 6 & 3 \\ 9 & 6 & 12 \end{bmatrix} = 3 \times \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix}$$

The determinant of a square matrix A of size m × m denoted as det (A) is a scalar calculated recursively as shown below:

1. If $m = 1$, $\det(\mathbf{A}) = a_{11}$
2. If $m > 1$, $\det(\mathbf{A}) = \sum_{i=1\ldots m} (-1)^{i+j} \times a_{ij} \times \det(\mathbf{A}_{ij})$

Where $\mathbf{A}_{ij}$ is a matrix obtained from $\mathbf{A}$ by deleting the ith row and jth column.

Calculating the determinant of a 2 × 2 matrix based on the determinant of a 1 × 1 matrix

$$\det\begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det[4] + (-1)^{1+2} \times 2 \times \det[3] \longrightarrow 5 \times 4 - 2 \times 3 = 14$$

or $$\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} \times a_{22} - a_{12} \times a_{21}$$

Calculating the determinant of a 3 × 3 matrix

$$\det\begin{bmatrix} 5 & 2 & 1 \\ 3 & 0 & -4 \\ 2 & 1 & 6 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det\begin{bmatrix} 0 & -4 \\ 1 & 6 \end{bmatrix} + (-1)^{1+2} \times 2 \times \det\begin{bmatrix} 3 & -4 \\ 2 & 6 \end{bmatrix} + (-1)^{1+3} \times 1 \times \det\begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}$$

$$= (+1) \times 5 \times (+4) \quad + \quad (-1) \times 2 \times (24) \quad + \quad (+1) \times 1 \times (3) = -25$$

Cryptography uses residue matrices: matrices where all elements are in Z_n. A residue matrix has a multiplicative inverse if gcd (det(A), n) = 1.

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 7 & 2 \\ 1 & 4 & 7 & 2 \\ 6 & 3 & 9 & 17 \\ 13 & 5 & 4 & 16 \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} 15 & 21 & 0 & 15 \\ 23 & 9 & 0 & 22 \\ 15 & 16 & 18 & 3 \\ 24 & 7 & 15 & 3 \end{bmatrix}$$

$$\det(\mathbf{A}) = 21 \qquad \det(\mathbf{A}^{-1}) = 5$$

GROUPS, RINGS, AND FIELDS

Groups, rings, and fields are the fundamental elements of a branch of mathematics known as abstract algebra, or modern algebra.

GROUPS

A **group** *G*, sometimes denoted by {*G*, •}, is a set of elements with a binary operation denoted by • that associates to each ordered pair (*a, b*) of elements in *G* an element (*a* • *b*) in *G*, such that the following axioms are obeyed:

(A1) Closure: If *a* and *b* belong to *G*, then *a* • *b* is also in *G*.
(A2) Associative: *a*•(*b*•*c*) = (*a*•*b*)•*c* for all *a*, *b*, *c* in *G*.
(A3) Identity element: There is an element e in *G* such that *a* • e = e • *a* = *a* for all *a* in *G*.
(A4) Inverse element: For each *a* in *G*, there is an element a^{-1} in *G* such that $a \bullet a^{-1} = a^{-1} \bullet a$ = e.
If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**. A group is said to be **abelian** if it satisfies the following additional condition:
(A5) Commutative: *a* • *b* = *b* • *a* for all *a*, *b* in *G*.
A group *G* is **cyclic** if every element of *G* is a power a^k (*k* is an integer) of a fixed element *a* ε *G*. The element *a* is said to **generate** the group *G* or to be a **generator** of G. A cyclic group is always abelian and may be finite or infinite.

RINGS

A **ring** *R*, sometimes denoted by {*R*, +, *}, is a set of elements with two binary operations, called *addition* and *multiplication*,6 such that for all *a*, *b*, *c* in *R* the following axioms are obeyed.
(A1–A5) *R* is an abelian group with respect to addition; that is, *R* satisfies axioms A1 through A5.
(M1) Closure under multiplication: If *a* and *b* belong to *R*, then *ab* is also in *R*.
(M2) Associativity of multiplication: *a*(*bc*) = (*ab*)*c* for all *a*, *b*, *c* in *R*.

(M3) Distributive laws: $a(b + c) = ab + ac$ for all a, b, c in R.

$(a + b)c = ac + bc$ for all a, b, c in R.

A ring is said to be **commutative** if it satisfies the following additional condition:

(M4) Commutativity of multiplication: $ab = ba$ for all a, b in R.

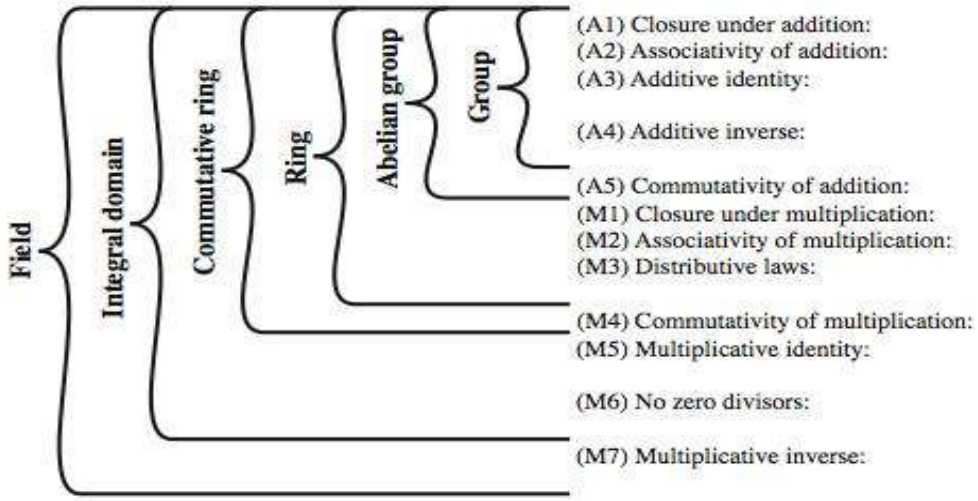
An **integral domain**, which is a commutative ring that obeys the following axioms.

(M5) Multiplicative identity: There is an element 1 in R such that $a1 = 1a = a$ for all a in R.

(M6) No zero divisors: If a, b in R and $ab = 0$, then either $a = 0$ or $b = 0$.

FIELDS

A **field** F, sometimes denoted by $\{F, +, *\}$, is a set of elements with two binary operations, called *addition* and *multiplication*, such that for all a, b, c in F the following axioms are obeyed. **(A1–M6)** F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6. **(M7) Multiplicative inverse:** For each a in F, except 0, there is an element a-1 in F such that $aa^{-1} = (a^{-1})a = 1$



FINITE (GALOIS) FIELDS

- Finite fields play a key role in cryptography.
- Finite field is a field that contains a finite number of elements
- It Can show number of elements in a finite field **must** be a power of a prime p^n
- known as Galois fields, denoted $GF(p^n)$
- In particular often use the fields:
  n=1 then we say as GF(p), or p=2 then we say as GF(2^n).
- GF(p) is the set of integers {0,1, … , p-1} with addition & multiplication modulo p.
- This forms a "well-behaved" finite field.

DATA ENCRYPTION STANDARD

Introduction:

The most widely used private key block cipher, is the Data Encryption Standard (DES).

DES encrypts data in 64-bit blocks using a 56-bit key.

IBM developed Lucifer cipher

- ★ by team led by Feistel in late 60's
- ★ used 64-bit data blocks with 128-bit key

- then redeveloped as a commercial cipher with input from National Security Agency (NSA) and others.
- IBM submitted their revised Lucifer which was eventually accepted as the DES. **Over View of DES:**

The overall scheme for DES encryption is illustrated:

Plain text must be 64 bits in length and key length is 56 bits in length.

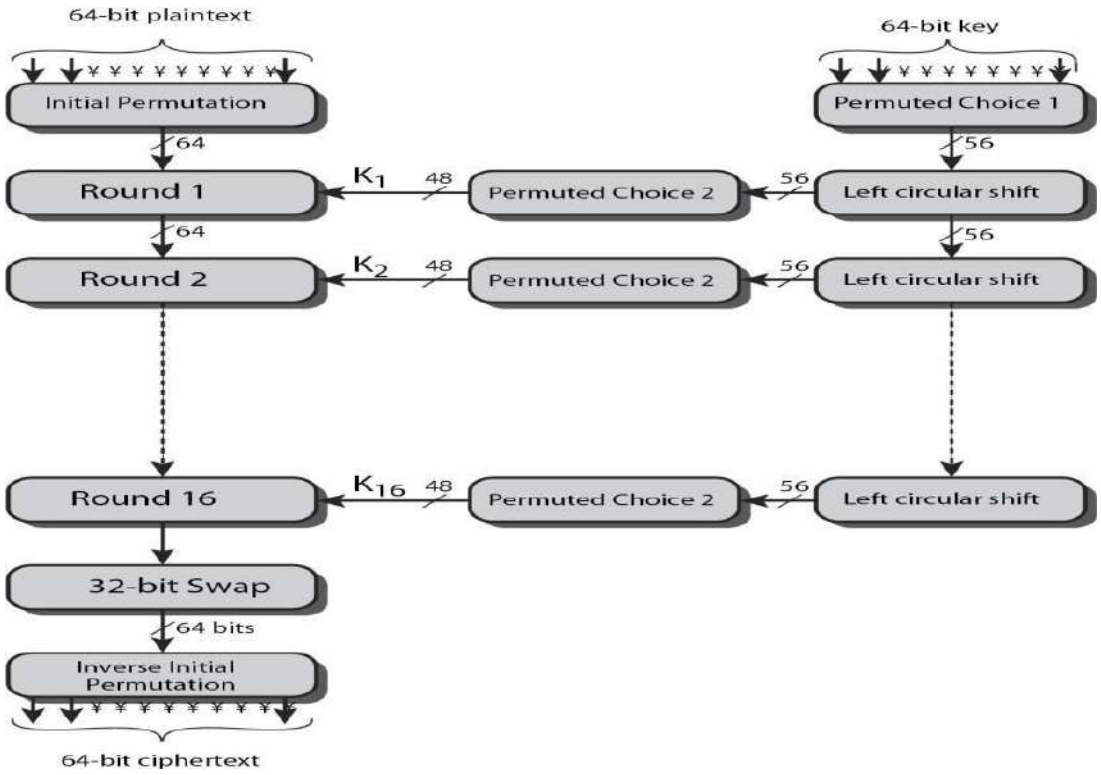
In the figure given below, the left side shows the basic process for enciphering a 64-bit data block which consists of:

- an initial permutation (IP) which shuffles the 64-bit input block
- 16 rounds of a complex key dependent round function involving substitutions & permutations
- a final permutation, being the inverse of IP

The right side shows the handling of the 56-bit key and consists of:

- an initial permutation of the key (PC1) which selects 56-bits out of the 64-bits input, in two 28-bit halves
- 16 stages to generate the 48-bit subkeys using a left circular shift and a permutation of the two 28-bit halves

Fig 1.1 Overview of DES



The main phases in the left hand side of the above figure i.e. processing of the plain text are

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

Table 1: Actual 64 bit order

Initial Permutation (IP): The plaintext block undergoes an initial permutation. 64 bits of the block are permuted. For example the actual input message order can be as follows according to their bit positions:

An example initial Permutation obtained by just writing the even numbered columns first in the row wise and odd numbered columns later in the row wise.

Table 2: Initial Permutation

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

A Single Round Transformation:

- On the right hand side part of the figure, the usage of the 56 bit key is shown. Initially the key is passed through a permutation function.
- Now for each of the 16 iterations, a new subkey (Ki) is produced by combination of a left circular shift and a permutation function which is same for each iteration.
- A different subkey is produced because of repeated shifting of the key bits.
- The left and right halves of each 64 bit intermediate value are treated as separated 32-bit quantities labeled L (left) and R (Right).
- The overall processing at each iteration is given by following steps, which form one round in an S-P network.

Li = Ri-1.

Ri = L i-1 F(R⊕1, Ki)

Where Function F can be described as **P (S(E(R(i-1)⊕ K(i)))**

The following figure shows a closer view of algorithms for a single iteration. The 64bit permuted input passes through 16 iterations, producing an intermediate 64-bit value at the conclusion of each iteration.

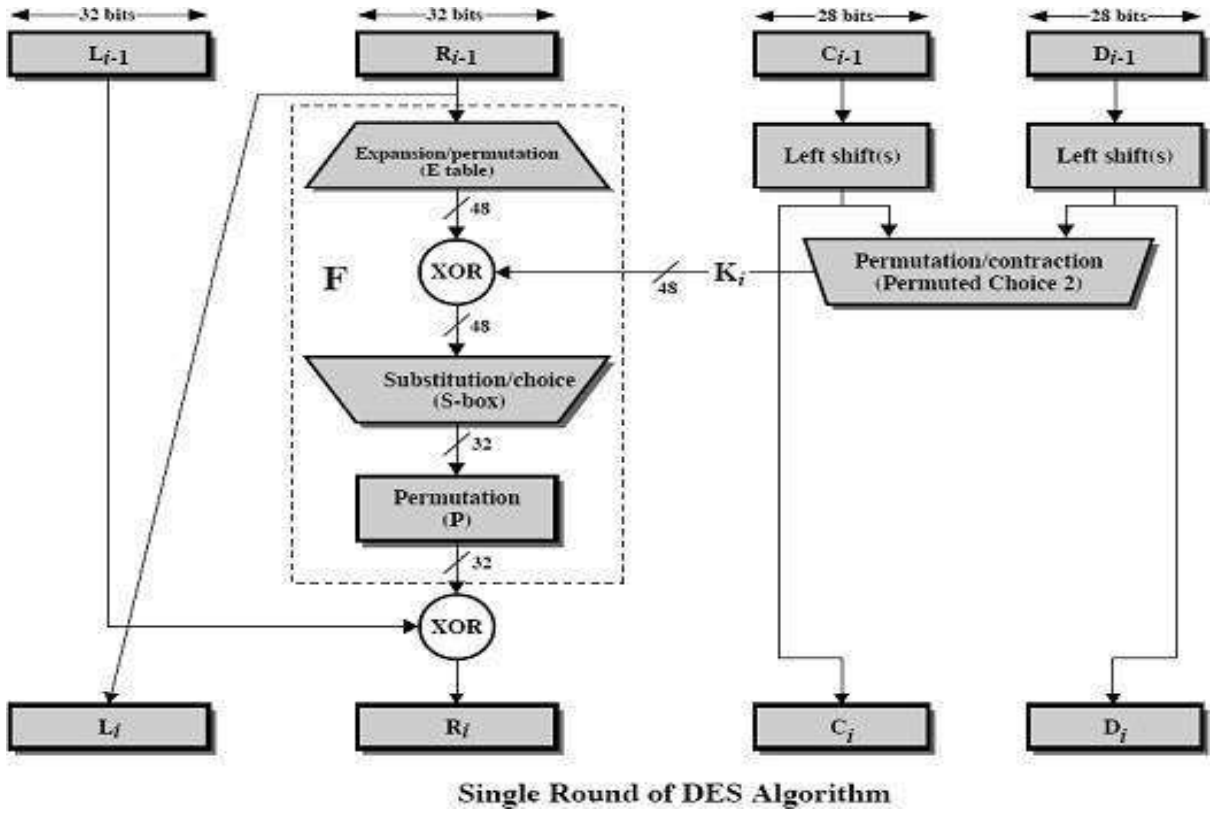


Fig 1.2 Single Round Function A

Complex Transformation (F):

This again contains two sub transformations i) Expansion transformation (E)

ii) Substitution transformation (S)

i) Expansion transformation

- 64 bit permuted block undergoes 16 rounds of complex transformation.
- Subkeys are used in each of the 16 iterations.
- The round Key Ki is 48 bits.
- The R input is 32 bits. The R input is first expanded to 48 bit by using a table that defines a permutation plus an expansion that involves duplication of 16 of R bits.
- The Expansion is as follows: The middle elements belong to the 32 bit R input and Left and right 8 elements are the duplications of the R input, to make as 48 bit.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table 3: Expansion Permutation

ii) Substitution transformation

The **Role of the S-Box** in the transformation (F) is as follows,

The substitution consists of a set of 8 S-Boxes, each accepts 6 input and produce 4 output bits.

These transformations are defined as follows:

- The first and last bits of 6bits used to specify the row, and the rest of the 4 bits are used to specify the column of the specific box. The data in the specified position is used to replace.
- The first and last bits of the input to box Si form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for Si.
- The middle four bits select one of the sixteen columns. The decimal value in the cell selected by the row and column is then converted to its 4-bit representation to produce the output.

Table 4: Substitution Table (S-Box)

	Col 0	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10	Col 11	Col 12	Col 13	Col 14	Col 15
Row0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
Row1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
Row2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
Row3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

For example:

In S1, for input **011100**, the row is 01 (row 1) and the column is 1100 (column 12). The value in row 1, column 12 is 9, so the output is 1001.

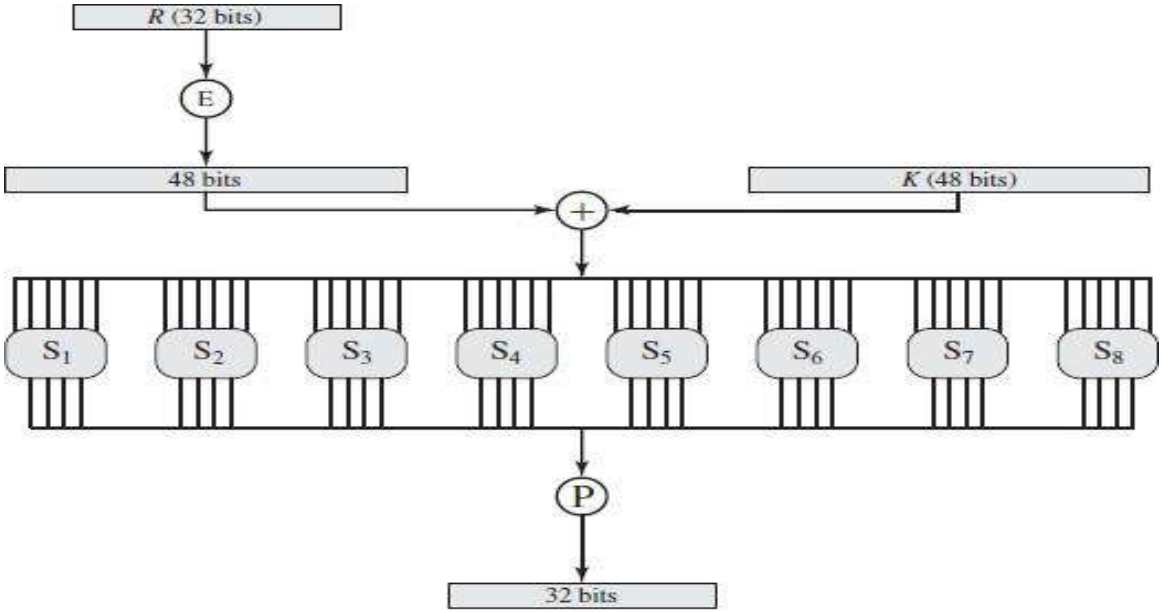


Figure 3.7 Calculation of F(R, K)

The outer two bits of each group select one of four possible substitutions (one row of an S-box). Then a 4-bit output value is substituted for the particular 4-bit input (the middle four input bits). The 32-bit output from the eight S-boxes is then permuted, so that on the next round, the output from each S-box immediately affects as many others as possible.

Inverse Initial Permutation (IP^{-1}): The 64 bit output undergoes a permutation that is inverse of the initial permutation.

Table 5: Final Permutation

(b) Inverse Initial Permutation (IP^{-1})

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

Sub Key Generation Algorithm: A 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64; **every eighth bit is ignored, as indicated by lack of shading** in Table 6. The key is first subjected to a permutation governed by a table labelled Permuted Choice One, indicated in Table 7.

The resulting 56-bit key is then treated as two 28-bit quantities, labelled C0 and D0. **At each round, Ci-1 and Di-1 are separately subjected to a circular left shift or (rotation) of 1 or 2 bits, as governed by Table 9.** These shifted values serve as input to the next round. They also serve as input to the part labelled Permuted Choice Two, represented in Table 8, which produces a 48-bit output that serves as input to the function .

Table 6: Input Key

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

Table 7: Permuted Choice-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18

10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Table 8: Permuted Choice-2

14	17	11	24	1	5	3	28
15	6	21	10	23	18	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Table 9: Schedule of Left Shifts

Round Numbers	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

.

DES Decryption:

The DES decryption uses the same algorithm as encryption, except the application of sub key in the reverse order.

The Avalanche Affect of DES:

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext. This is referred to as the avalanche effect.

Strength of DES – Key Size

1. The Key

- **The level of security provided by DES in two areas: key size and the nature of the algorithm.**
- **With a key length of 56 bits, there are 2^56 possible keys, which is approximately 7.2*10^16 keys. Thus a brute-force attack appeared impractical**. However DES was finally and definitively proved insecure in July 1998. W

- It is important to note that there is more to a key-search attack than simply running through all possible keys. Unless known plaintext is provided, the analyst must be able to recognize plaintext as plaintext. Clearly must now consider alternatives to DES, the most important of which are AES and triple DES.

2. **The Nature of the Encryption algorithm**

The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration.

These techniques utilize some deep structure of the cipher by gathering information about encryptions so that eventually you can recover some/all of the sub-key bits, and then exhaustively search for the rest if necessary.

Generally these are statistical attacks which depend on the amount of information gathered for their likelihood of success. Attacks of this form include ***differential cryptanalysis. Linear cryptanalysis, and related key attacks***.

3. **The Timing Attack**

A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs. The AES analysis process has highlighted this attack approach, and showed that it is a concern particularly with smartcard implementations, though DES appears to be fairly resistant to a successful timing attack.

TRIPLE DES Double

DES:

The simplest form of multiple encryption has two encryption stages and two keys, Fig 2.1 Given a plaintext P and two encryption keys K1 andK2, ciphertext is generated as

$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$$

Decryption requires that the keys be applied in reverse order:

$$P = \mathrm{D}(K_1, \mathrm{D}(K_2, C))$$

For DES, this scheme apparently involves a key length of 56 X 2=112 bits, resulting in a dramatic increase in cryptographic strength. But we need to examine the algorithm more closely.

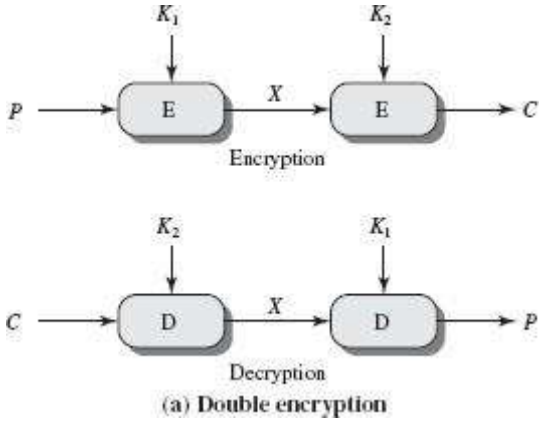


Fig 2.1 Double DES

The first answer to problems of DES is an algorithm called Double DES which includes double encryption with two keys. It increases the key size to 112 bits, which seems to be secure. But, there are some problems associated with this approach. ***Issue of reduction to single stage***:

Suppose it were true for DES, for all 56-bit key values, that given any two keys K1 and K2, it would be possible to find a key K3 such that

$$\mathrm{E}(K_2, \mathrm{E}(K_1, P)) = \mathrm{E}(K_3, P)$$

If this were the case, then double encryption, and indeed any number of stages of multiple encryption with DES, would be useless because the result would be equivalent to a single encryption with a single 56-bit key.

Meet-in-the-middle" attack:

Given a known pair (P, C), the attack proceeds as follows.

First, encrypt P for all 256 possible values of K1. Store these results in a table and then sort the table by the values of X. Next, decrypt C, using all 2^{56} possible values of K2.

As each decryption is produced, check the result against the table for a match. If a match occurs, then test the two resulting keys against a new known plaintext–cipher text pair.

If the two keys produce the correct cipher text, accept them as the correct keys. Test the two keys for the second pair of plaintext-cipher text and if they match, correct keys are found.

Triple DES

Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. 3 DES was developed in 1999 by IBM – by a team led by Walter Tuchman.

3 DES prevents a meet-in-the-middle attack. 3 DES has a 168-bit key and enciphers blocks of 64 bits. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES.

3DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence.

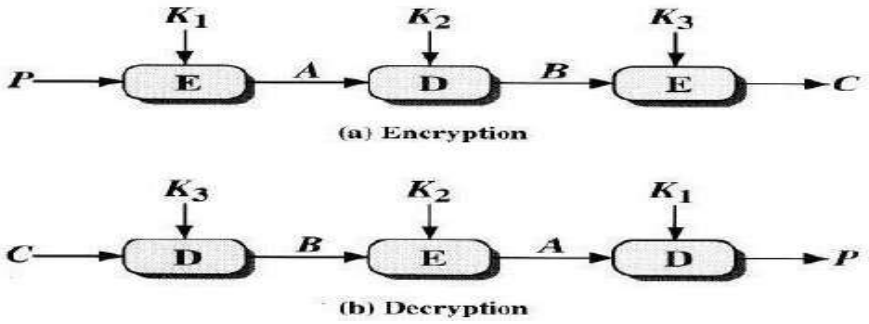


Fig 2.2 Triple DES Encryption and Decryption

C=Ek3[Dk2[Ek1[p]]] Where C= ciphertext, P=plaintext and EK[X] = encryption of X using key K DK[Y] = decryption of Y using key K. Decryption is simply the same operation with the keys reversed.P=Dk1[Ek2[Dk3[c]]]

Triple DES runs three times slower than standard DES, but is much more secure if used properly. With three distinct keys, TDEA has an effective key length of 168 bits making it a formidable algorithm. As the underlying algorithm is DEA, it offers the same resistance to cryptanalysis as is DEA. Triple DES can be done using 2 keys or 3 keys. There is no cryptographic significance to the use of decryption for the second stage of 3DES encryption. Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES: c = Ek1 [D2l [EK3 [P]]] = EK1 [P]

Strength of Triple DES:

If we assume that the cracker would perform 1 million decryptions per 1micro second, then DES would take 10 hours to break. With 128 bit Key It would take 1018 years to break. With 168 bit key Brute force attack is impossible.

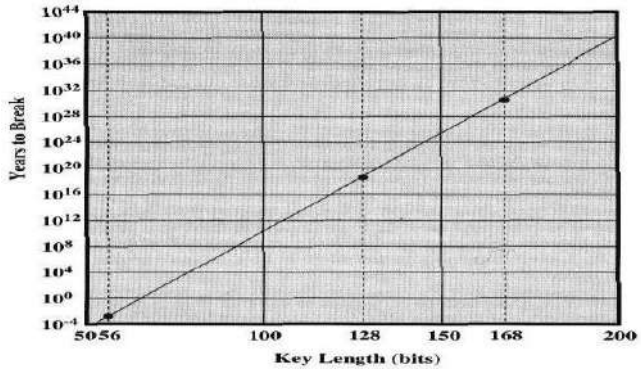


Fig 2.3 Time to break Code

BLOCK CIPHER PRINCIPLES

Any plain text can be transformed into cipher text with following Ciphers

- **Stream Cipher** is the one that encrypts the digital data one bit or one byte at a time to produce the cipher text. Ex:-Autokeyed Vigenere Cipher and Verman Cipher
- **Block Cipher** is the one in which a block of plain text is treated as whole and used to produce a cipher text of equal length. Ex:- DES

Motivations for Feistel Cipher Structure:

- A block cipher operates on a plain text of n-bits to produce the cipher text of nbits. There are 2^n different possible plain text blocks for encryptions to be reversible (Eg: for decryption to be possible), and each must produce **unique cipher text block. Such transformation is called "reversible" or "nonsingular"**
- **If same cipher block has different plain text block such scheme is called "irreversible Transformation"**

Table 1: Reversible and Irreversible Transformations

Reversible Mapping		Irreversible Mapping	
Plain Text	Cipher Text	Plain Text	Cipher Text
00	11	00	11
10	01	10	01
01	00	01	01

In the above example the same cipher block "01" is producing different plain text blocks; this is called "irreversible transformation".

Block Cipher Principles:

Most symmetric block encryption algorithms in current use are based on a structure referred to as a Feistel block cipher.

A block cipher operates on a plaintext block of n bits to produce a cipher text block of n bits. An arbitrary reversible substitution cipher for a large block size is not practical, however, from an implementation and performance point of view.

Feistel points out that what is needed is an approximation to the ideal block cipher system for large n, built up out of components that are easily realizable.

Ideal Block Cipher:

The most general form of the block cipher is Ideal Block Cipher in which a reversible mapping between plain text and cipher text is possible. This allows the maximum number of possible encryption mappings from plain text to cipher text.

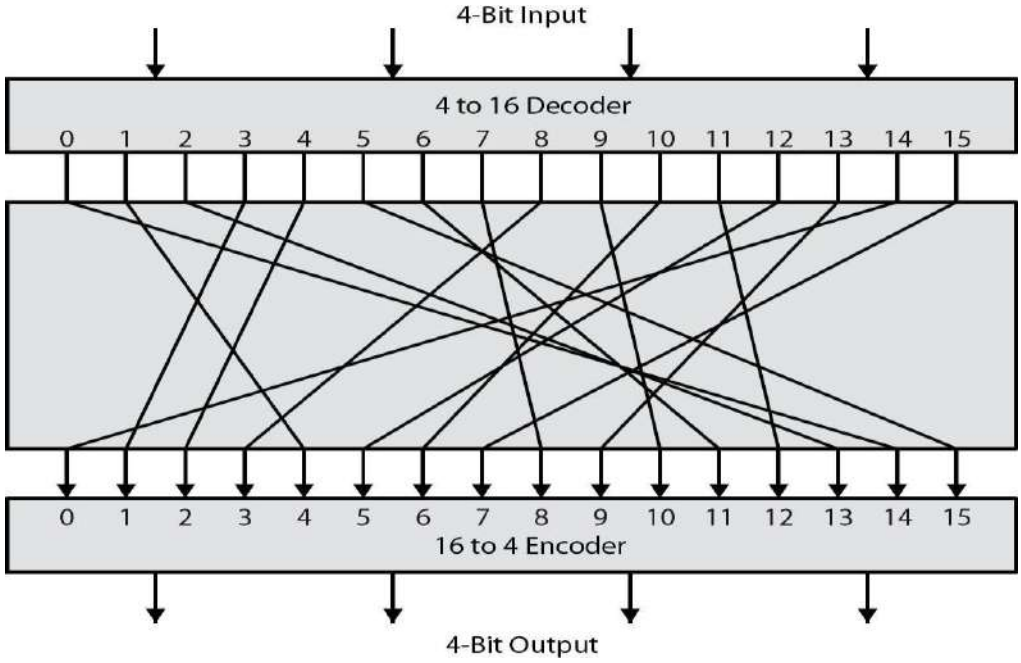


Fig 3.1 General n-bit by n-bit Ideal Block Cipher

Feistel Cipher:

- Horst Feistel, working at IBM Thomas J Watson Research Labs devised a suitable invertible cipher structure in early 70's.
- He proposed that we can actually approximate the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that to produce the final result or product that is cryptographically stronger than any of the component ciphers.
- The structure uses the alternate use of substitutions and permutations, which is proposed by the **Claude Shannon** to develop the product cipher that alternates ***confusion*** and ***diffusion***.

Confusion and Diffusion:

- The terms diffusion and confusion were introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system.
- Every block cipher involves a transformation of a block of plaintext into a block of cipher text, where the transformation depends on the key.
- The mechanism of **diffusion** seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to prevent attempts to deduce the key.
- **Confusion** seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
- So successful are diffusion and confusion in capturing the essence of the desired attributes of a block cipher that they have become the cornerstone of modern block cipher design.

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher** ◆ based on concept of invertible product cipher
- partitions input block into two halves ◆ process through multiple rounds which perform a substitution on left data half ◆ based on round function of right half & ◆ sub key then have permutation swapping halves
- implements Shannon's S-P net concept

The Design Elements of the Feistel Cipher are as follows:

- **Block size** - increasing size improves security, but slows cipher
- **Key size** - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **Number of rounds** –single round offers inadequate security, but multiple rounds increase security, increasing number improves security, but slows cipher
- **Sub key generation** algorithm - greater complexity can make analysis harder, but slows cipher
- **Round function** - greater complexity can make analysis harder, but slows cipher
- **Fast software encryption/decryption –** in many cases encryption is embedded in hardware to increase speed of cipher, more recent concern for practical use
- **Ease of analysis - for easier validation &** there is a great advantage in designing easier algorithm to find all the vulnerabilities, testing of strength

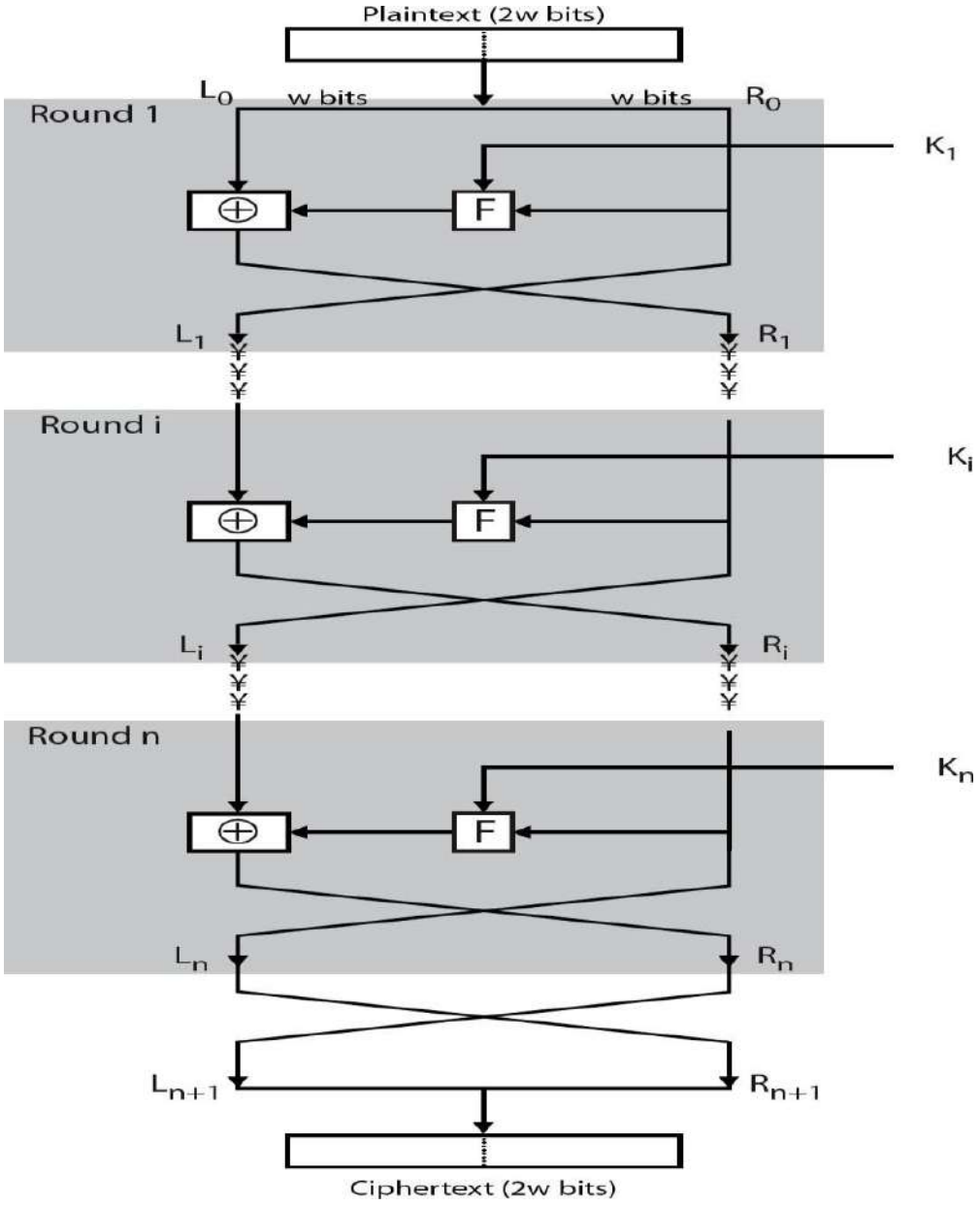


Fig 3.2 Encryption Process

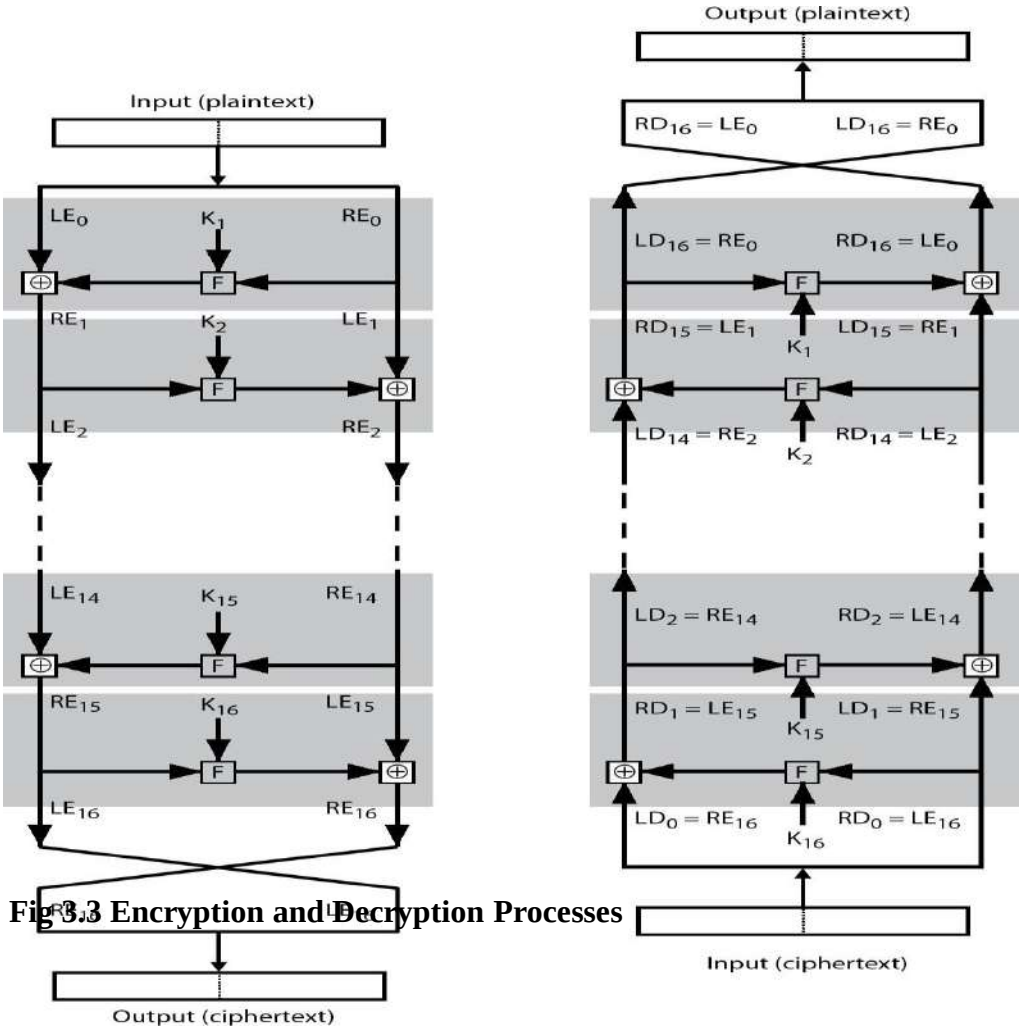


Fig 3.3 Encryption and Decryption Processes

Working procedure of Feistel Cipher:

The plaintext block is divided into two halves, *Lo* and *Ro.*

The two halves of the data pass through *n* rounds of processing and then combine to produce the ciphertext block.

Each round *i* has as inputs *Li-l* and *R;-l'* derived from the previous round, as welI as a subkey Ki, derived from the overall *K.*

In general, the subkeys *Ki* are different from *K* and from each other and are generated from the key by a subkey generation algorithm.

All rounds have the same structure. A substitution is performed on the left half of the data. This is done by applying a *round function* F to the right half of the data and then taking the exclusive-OR (X OR) of the output of that function and the left half of the data.

The round function has the same general structure for each round but is parameterized by the round subkey *Ki.* Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.

The general process of the encryption will be as

follows: LEi= REi-1 REi=LEI
X F(REi-1,Ki)

The Decryption is as

follows: LE16=RE15
RE16=LE15 X F(RE15,K16)

Differential Cryptanalysis

One of the most significant advances in cryptanalysis in recent years is differential cryptanalysis. In this section, we discuss the technique and its applicability to DES. The differential cryptanalysis attack is complex.

The rationale behind differential cryptanalysis is to observe the behavior of pairs of text blocks evolving along each round of the cipher, instead of observing the evolution of a single text block.

Consider the original plaintext block **m** to consist of two halves [**m0, m1**. Each round of DES maps the right-hand input into the left-hand output and sets the right-hand output to be a function of the left-hand input and the sub key for this round. So, at each round, only one new 32-bit block is created. If we label each new block m1 ($2 \leq i \leq 17$), then the intermediate message halves are related as follows:

$$\mathbf{m_{i+1} = m_{i-1} \oplus f(m_i, K_i),\ i = 1, 2, ..., 16}$$

This attack is known as Differential Cryptanalysis because the analysis compares differences between two related encryptions, and looks for a known difference in leading to a known difference out with some (pretty small but still significant) probability. If a number of such differences are determined, it is feasible to determine the subkey used in the function f.

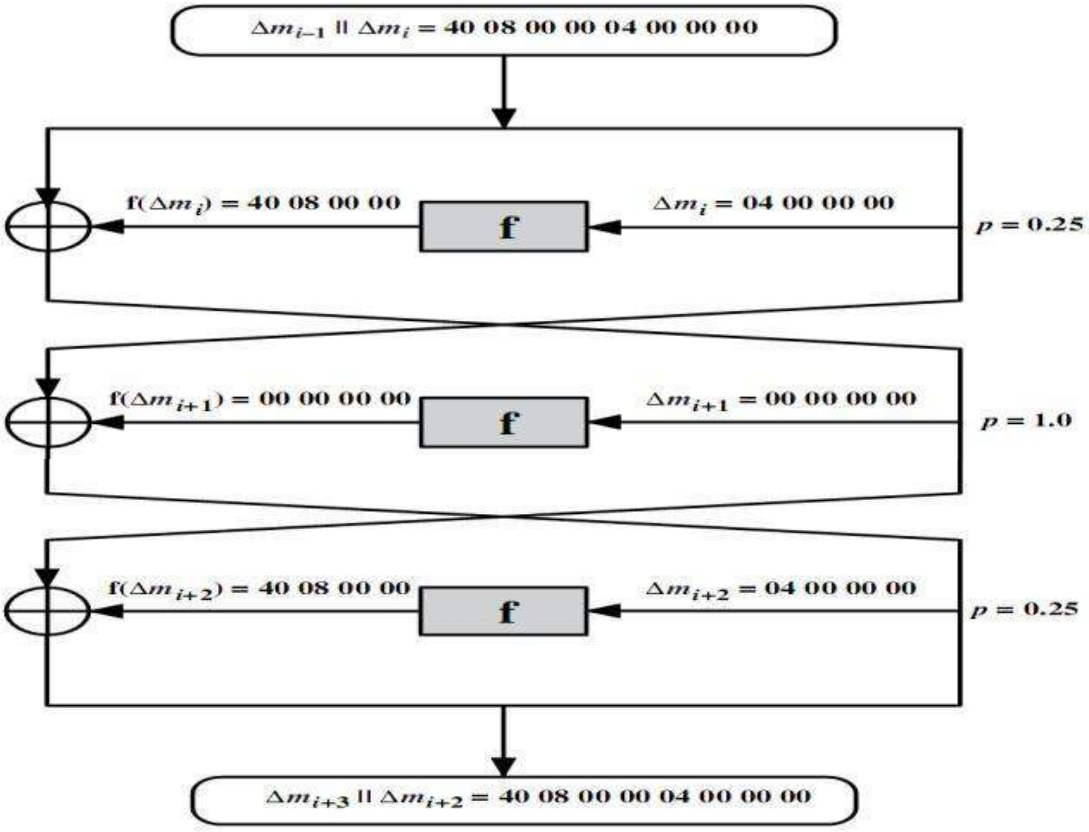


Figure 3.7 Differential Propagation through Three Round of DES (numbers in hexadecimal)

The overall strategy of differential cryptanalysis is based on these considerations for a single round. The procedure is to begin with two plaintext messages m and m' with a given difference and trace through a probable pattern of differences after each round to yield a probable difference for the cipher text. You submit m and m' for encryption to determine the actual difference under the unknown key and compare the result to the probable difference. If there is a match, then suspect that all the probable patterns at all the intermediate rounds are correct. With that assumption, can make some deductions about the key bits. This procedure must be repeated many times to determine all the key bits.

Linear Cryptanalysis

A more recent development is linear cryptanalysis. This attack is based on finding linear approximations to describe the transformations performed in DES. This method can find a DES key given 243 known plaintexts, as compared to 247 chosen plaintexts for differential cryptanalysis. Although this is a minor improvement, because it may be easier to acquire known plaintext rather than chosen plaintext, it still leaves linear cryptanalysis infeasible as an attack on DES.

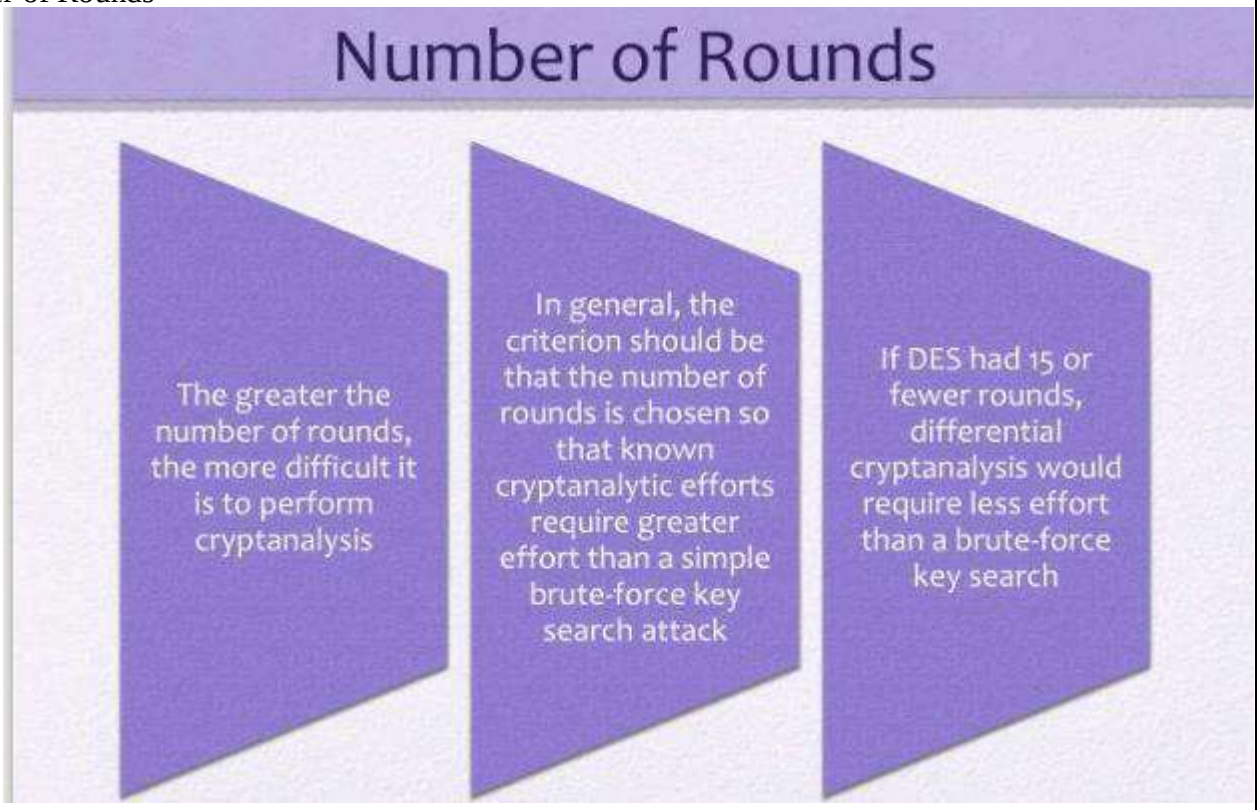
Distinguish between differential and linear cryptanalysis. (May 2012)

- **Linear cryptanalysis** is a general form of cryptanalysis based on finding affine approximations to the action of a cipher. Linear cryptanalysis is one of the most widely used attacks on block ciphers;
- **Differential cryptanalysis** is a general form of cryptanalysis applicable primarily to block ciphers, but also to stream ciphers and cryptographic hash functions. In the broadest sense, it is the study of how differences in information input can affect the resultant difference at the output.
- **Linear cryptanalysis** focuses on statistical analysis against one round of decrypted cipher text. **Differential** analysis focuses on statistical analysis of two inputs and two outputs of a cryptographic algorithm.

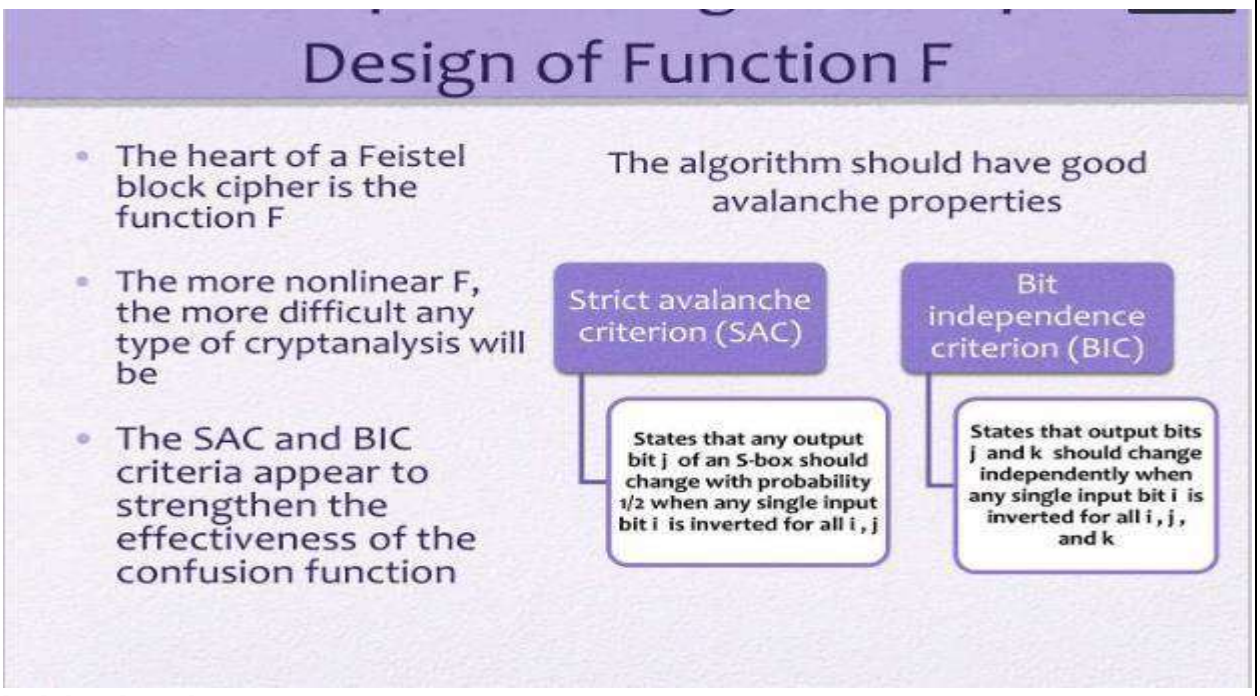
BLOCK CIPHER DESIGN PRINCIPLES

1) Number of Rounds
2) Design of Function F
3) Key Schedule Algorithm

1)Number of Rounds



2)Design of Function F



3)Key Schedule Algorithm

- With any feistel block cipher, the key is used to generate one subkey for each round.
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the diff iculty of deducing individual subkeys and the difficulty of working back to the main key.
- It is suggested that, at a minimum,the key schedule should guarantee key / cipher text strict Avalanche criterion and Bit independence Criterion.

BLOCK CIPHER MODES OF OPERATION

A symmetric block cipher processes one block of data at a time. In the case of DES and 3DES, the block length is 64 bits. For longer amounts of plaintext, it is necessary to break the plaintext into 64 -bit blocks (padding the last block if n ecessary). There are five modes of operations:

- Electronic Codebook Mode
- Cipher Block Chaining Mode
- Cipher Feedback Mode.
- Output Feedback Mode
- Counter Mode

(i) Electronic Codebook Mode:

- The **simplest way** to proceed is what is known as electronic codebook (ECB) mode, in which **plaintext is handled 64 bits at a time and each block of plaintext is encrypted using the same key**.

- The term *codebook* is used because, for a given key, there is a unique ciphertext for every 64-bit block of plaintext.
- Therefore, one can imagine a gigantic codebook in which there is an entry for every possible 64-bit plaintext pattern showing its corresponding ciphertext.

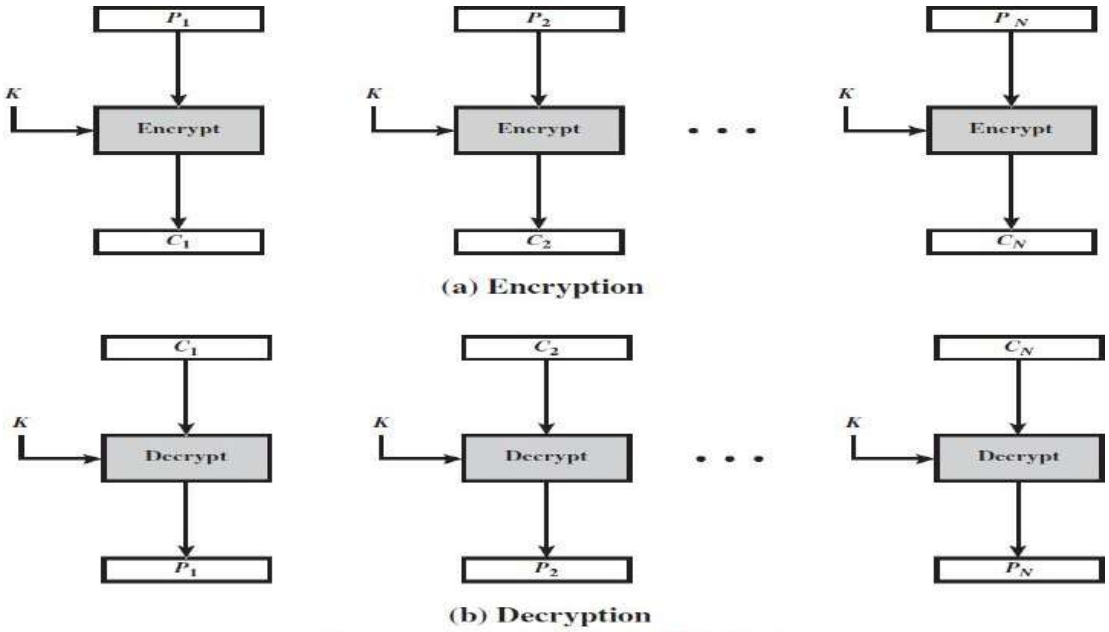


Figure 6.3 Electronic Codebook (ECB) Mode

Advantages and Limitations of ECB:

- Main use is sending a **few blocks of data** .(i.e) If we want to securely send key of DES or AES we can use this mode to send the key securely
- When two messages which have two blocks of plaintexts in common are encrypted with ECB mode the corresponding cipher text blocks will be the same. **Message repetitions may show in ciphertext.**
- Weakness is due to the **encrypted message blocks being independent**

(ii) <u>Cipher Block Chaining Mode (CBC):</u>

- To overcome the problems of repetitions and order independence in ECB, want some way of making the ciphertext dependent on all blocks before it. This is what CBC gives us, by combining the previous ciphertext block with the current message block before encrypting.
- To start the process, use an Initial Value (IV), which is usually well known (often all 0's).
- CBC mode is applicable whenever large amounts of data need to be sent securely, provided that all data is available in advance (eg email, FTP, web etc).

- In effect, we have chained together the processing of the sequence of plaintext blocks, to avoid the **similarities**. The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block. Therefore, repeating patterns of 64 bits are not exposed.
- The Encryption process is as follows:
  Ci=Ek[Ci-1 XOR Pi], where Ek[x] is the encryption of plain text x, using the key K, and XOR is the Exclusive OR operation.
- The Decryption process is as follows: Pi=Dk [Ci] XOR Ci-1

For decryption, each cipher block is passed through the decryption algorithm. The result is XORed with the preceding ciphertext block to produce the plaintext block.

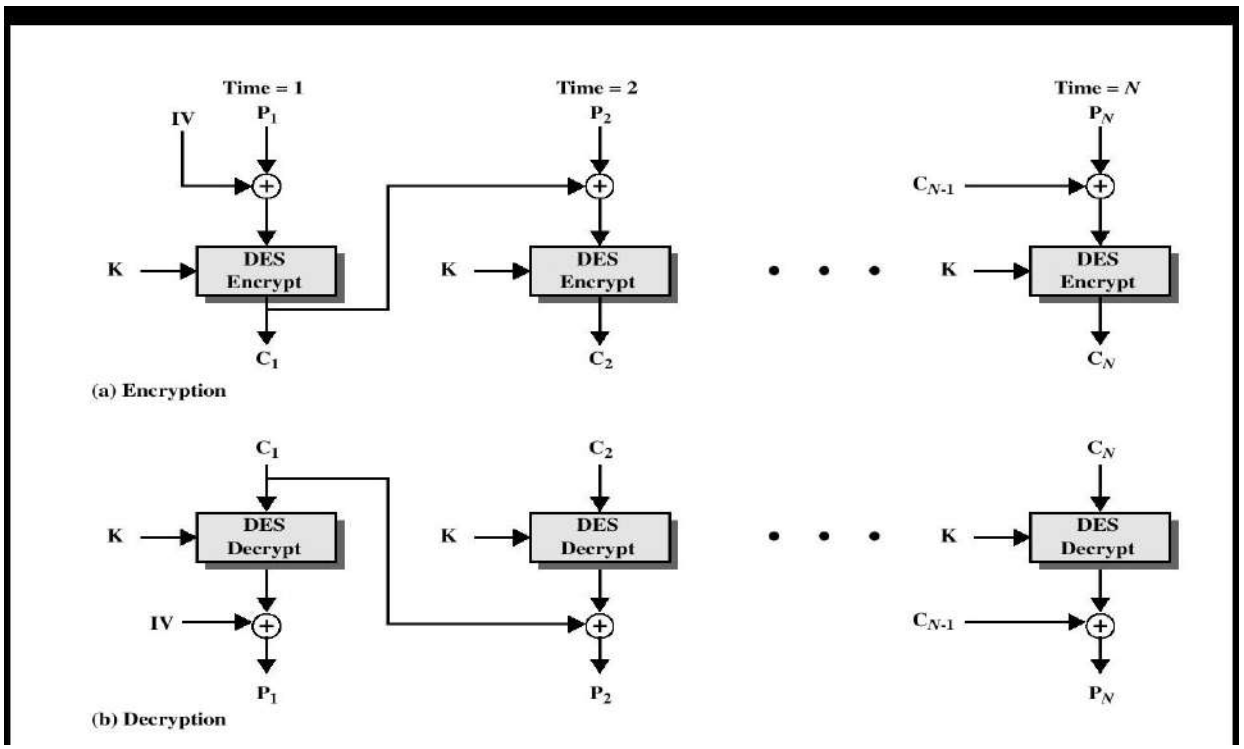


Fig 4.2 Cipher Block Chaining Mode

Advantages and Limitations of CBC:

- A ciphertext block depends on all blocks before it
- Any change to a block affects all following ciphertext blocks
- Need Initialization Vector (IV)
- Padding is done (Last block padded with b bits if it is partial block) **(iii)**

Cipher Feedback Mode (CFB):

It is possible to convert any block cipher into a stream cipher by using the cipher feedback (CFB) mode.

A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character- oriented stream cipher.

First, consider encryption. o The input to the encryption function is a 64-bit shift register that is initially set to some initialization vector (IV).

- The leftmost (most significant) *s* bits of the output of the encryption function are XORed with the first unit of plaintext to produce the first unit of ciphertext , which is then transmitted.
- In addition, the contents of the shift register are shifted left by *s* bits and Cipher text is placed in the rightmost (least significant) *s* bits of the shift register. This process continues until all plaintext units have been encrypted.

Mode Advantages and Limitations of CFB:

Appropriate when data arrives in bits/bytes

- Most common stream mode
- Limitation is need to stall while do block encryption after every n-bits
- Errors propagate for several blocks after the error that if its used over a "noisy" link, then any corrupted bit will destroy values in the current and next blocks

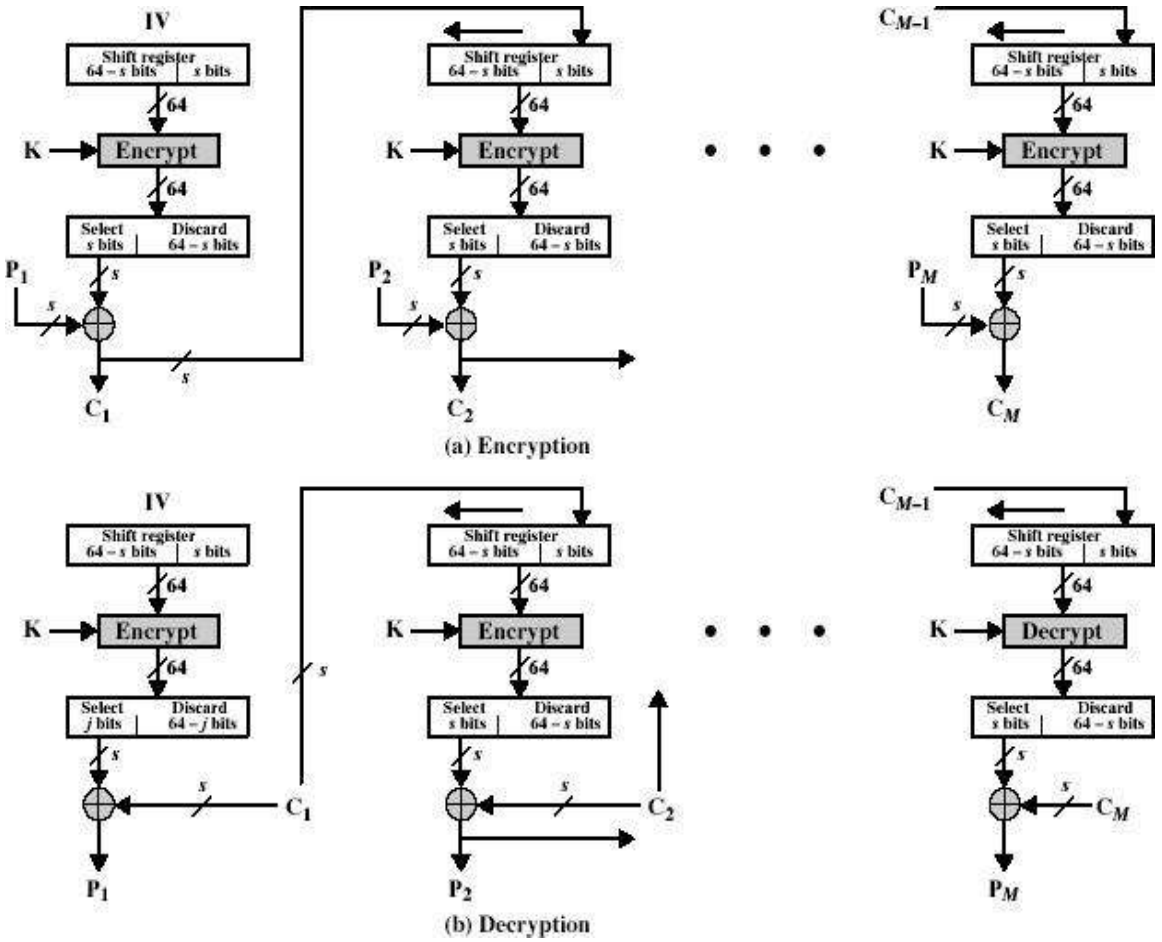


Fig 4.3 Cipher Feedback

(iv) <u>Output Feedback Mode (OFB):</u>

The **output feedback** (OFB) mode is similar in structure to that of CFB.

1) From the below fig, it is the output of the encryption function that is fed back to the shift register in OFB, whereas in CFB, the ciphertext unit is fed back to the become an input for encrypting the next block.

2) OFB mode operates on full blocks of plaintext and ciphertext, whereas CFB operates on an s-bit subset.

3)Like CBC & CFB, OFB uses IV (Must be Nonce- Number used once)

(i.e the IV must be unique to each execution of the encryption algorithm.)

Encryption can be expressed as By rearranging terms, $Cj = Pj\_$ XOR E(K, [Cj-1 XOR Pj-1]) we can demonstrate that decryption works.

$Pj = Cj$ XOR E(K, [Cj-1 XOR Pj-1])

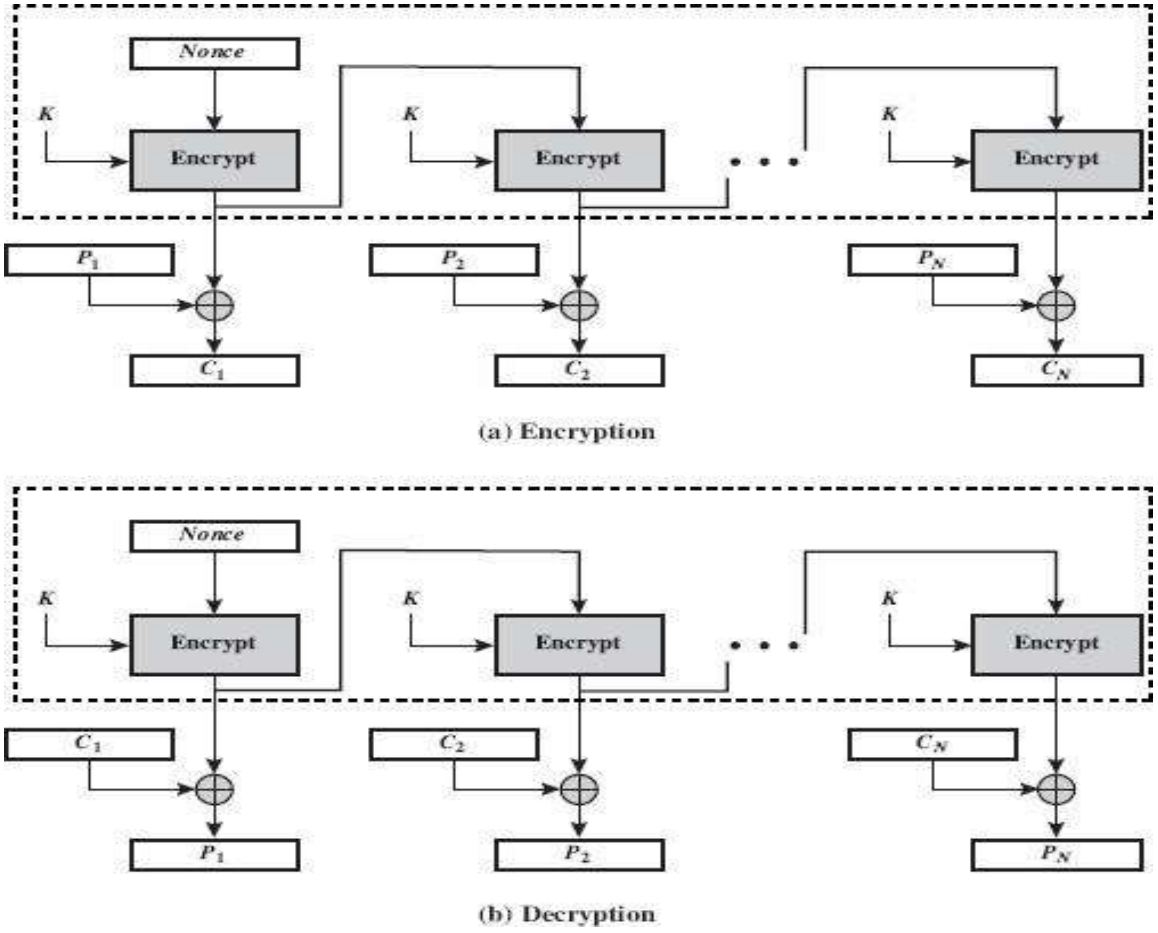


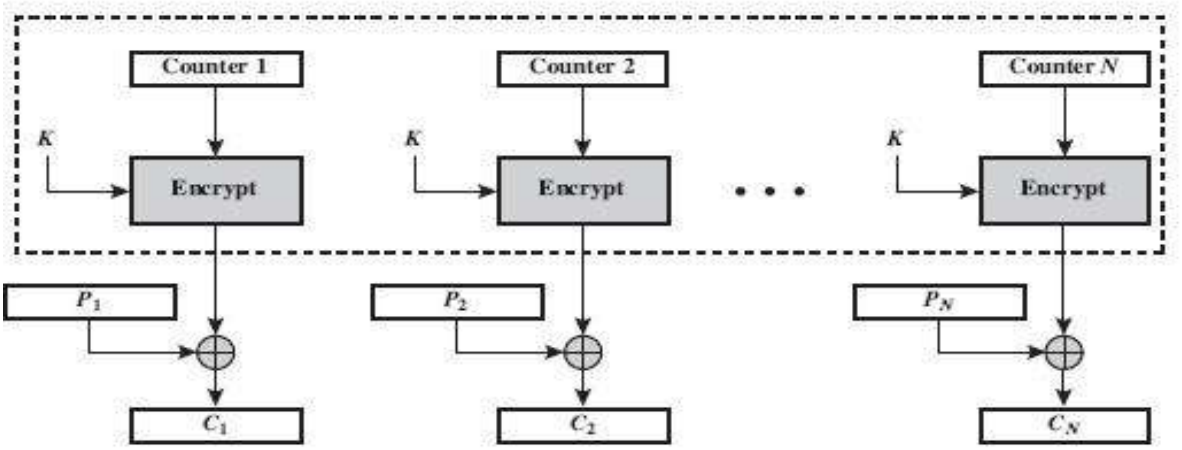
Fig 4.4 Output Feedback Mode

Advantage of OFB Mode: The bit errors in transmission do not propagate (any bit error only affects a single bit)

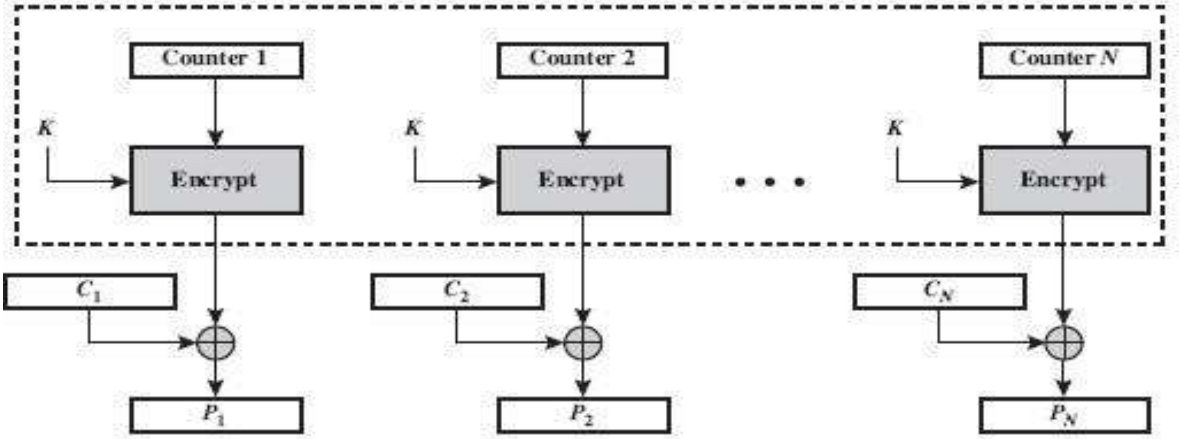
Disadvantage of OFB Mode: The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB.

(v) Counter Mode (CTR):

- Similar to OFB but **encrypts counter value** (hence name) rather than any feedback value
- Must **have a different key & counter value for** every plaintext block (never reused).
- It is being used with applications in ATM (asynchronous transfer mode) network security and IPSec (IP security).
- A counter, equal to the plaintext block size is used.
- Typically the counter is initialized to some value and then incremented by 1 for each subsequent block.



(a) Encryption



(b) Decryption

Fig 4,5 Counter

Mode Advantages and Limitations of CTR:

- Efficiency
  - ★ can do parallel encryptions in h/w or s/w
  - ★ can pre-process in advance of need
  - ★ good for bursty high speed links
- Random access to encrypted data blocks
- Provable security (good as other modes) But must ensure never reuse key/counter values, otherwise could break.

BLOCK CIPHER OPERATION

Table 1 Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

ADVANCED ENCRYPTION STANDARD

AES Evaluation Criteria:

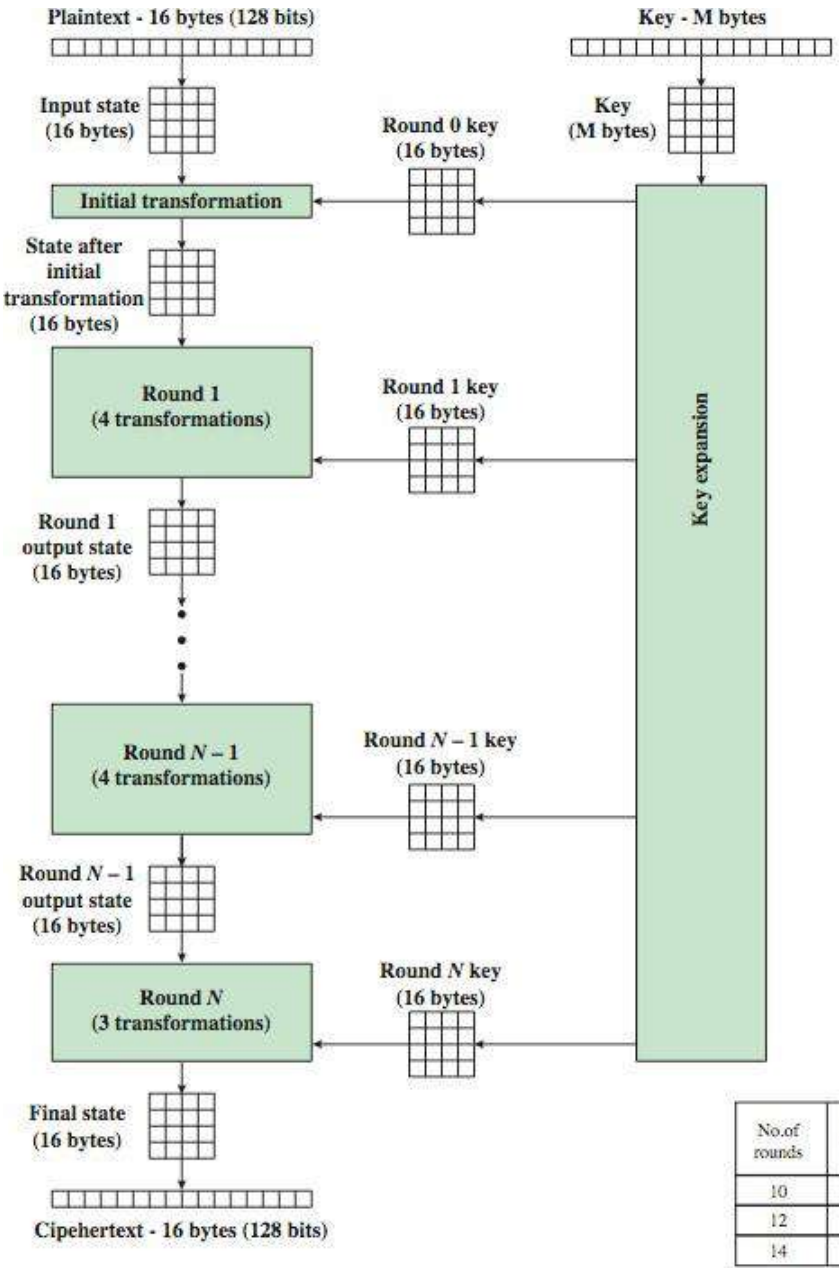
- Initial criteria:
  - ★ Security – effort for practical cryptanalysis
  - ★ Cost – in terms of computational efficiency

- ★ Algorithm & implementation characteristics ➲ Final criteria
- ★ General security
- ★ Ease of software & hardware implementation
- ★ Implementation attacks
- ★ Flexibility (in en/decrypt, keying, other factors) **AES Requirements:**

- Private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES

The AES Cipher:

- ➲ designed by Rijmen-Daemen in Belgium
- ➲ The Advanced Encryption Standard (AES) was published by **NIST** (National Institute of Standards and Technology) in **2001.**
- ➲ has 128/192/256 bit keys, 128 bit data ➲ an **iterative** rather than **feistel** cipher
  - ★ processes data as block of 4 columns of 4 bytes
  - ★ operates on entire data block in every round ➲ designed to be:
  - ★ resistant against known attacks
  - ★ speed and code compactness on many CPUs
  - ★ design simplicity



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

AES ENCRYPTION PROCESS

AESDATASTRUCTURE

Input, state array, and output

The key as a square matrix of bytes expanded into an array of key schedule words. Each word is four bytes, and the total key schedule is 44 words for the 128-bit key.

Key and expanded key

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

Comments about the Overall Structure:

One noteworthy feature of this structure is that it is not a Feistel structure.

The key that is provided as input is expanded into an array of forty-four 32-bit words, w[i]. Four distinct words (128 bits) serve as a round key for each round. Four different stages are used, one of permutation and three of substitution:

Substitute bytes **:** Uses an S-box to perform a byte-by-byte substitution of the block

ShiftRows : A simple permutation

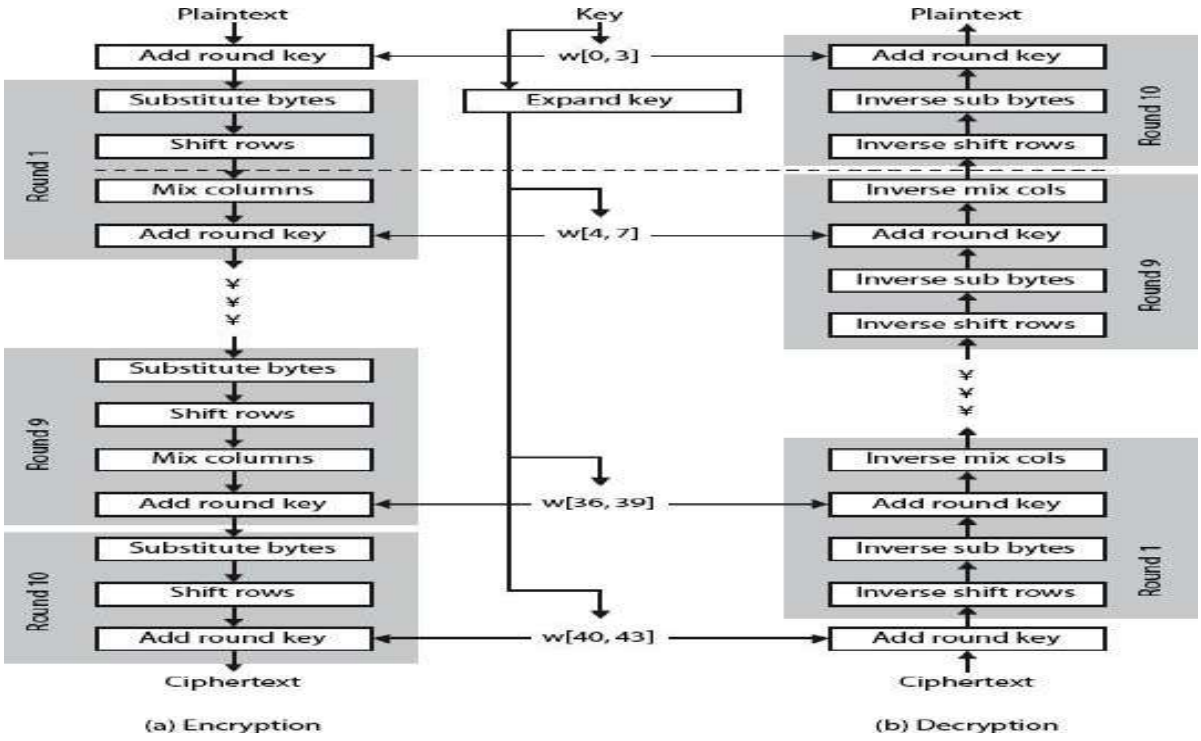
MixColumns **:** A substitution that makes use of arithmetic

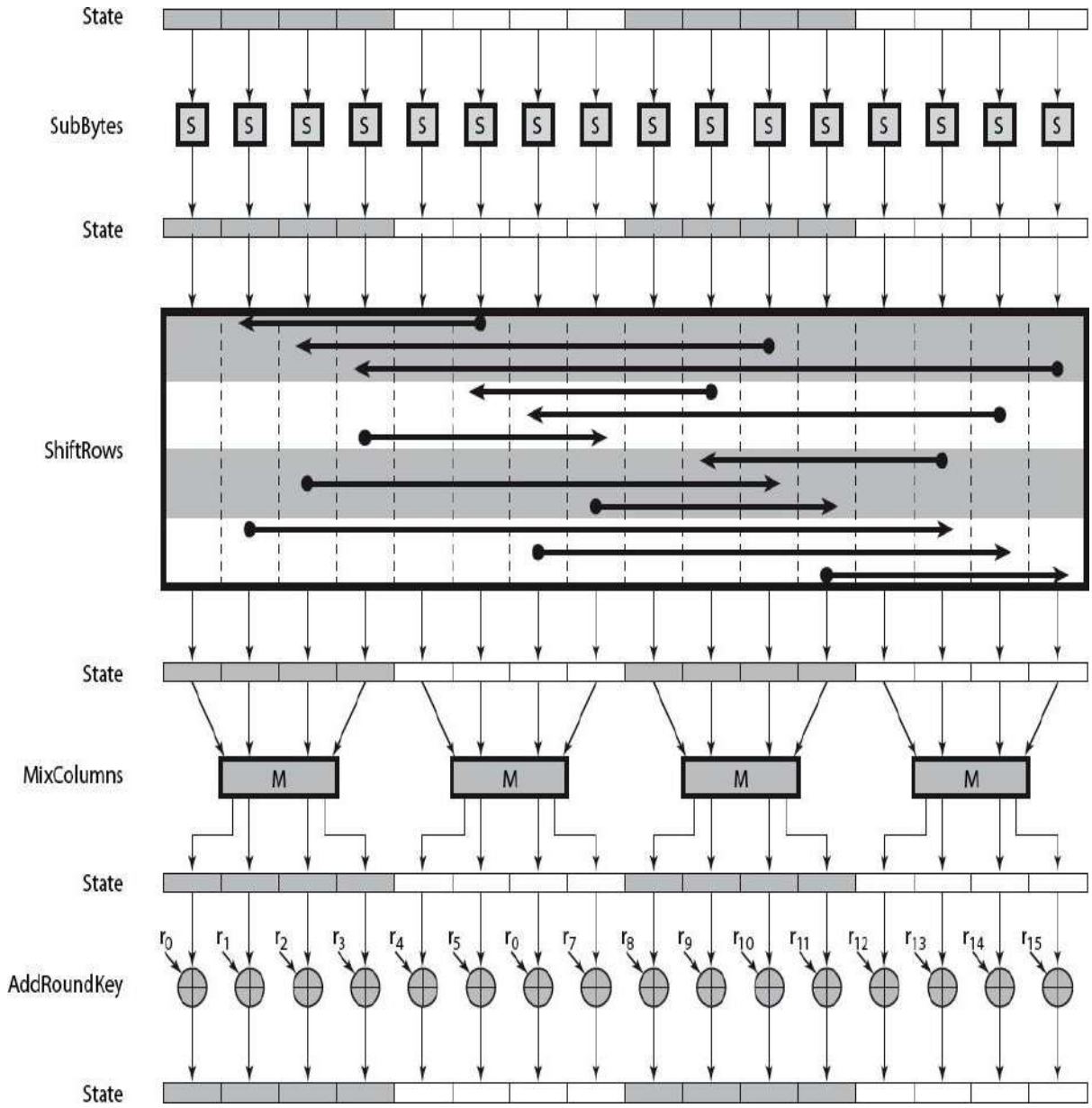
AddRoundKey : A simple bitwise XOR of the current block with a portion of the expanded key

The structure is quite simple. Only the AddRoundKey stage makes use of the key. The other three stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key. Each stage is easily reversible. As with most block ciphers, the decryption algorithm makes use of the expanded key in reverse order. Decryption is not identical to the encryption. This is just because of the structure of the AES.

Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext. The final round of both encryption and decryption consists of only three stages.

AESSTRUCTURE


Plaintext
Add round key
Substitute bytes
Shift rows
Mix columns
Add round key
Round 1
Substitute bytes
Shift rows
Mix columns
Add round key
Round 9
Substitute bytes
Shift rows
Add round key
Round 10
Ciphertext
(a) Encryption
Key
w[0, 3]
Expand key
w[4, 7]
w[36, 39]
w[40, 43]
Plaintext
Add round key
Inverse sub bytes
Inverse shift rows
Round 10
Inverse mix cols
Add round key
Inverse sub bytes
Inverse shift rows
Round 9
Inverse mix cols
Add round key
Inverse sub bytes
Inverse shift rows
Round 1
Add round key
Ciphertext
(b) Decryption

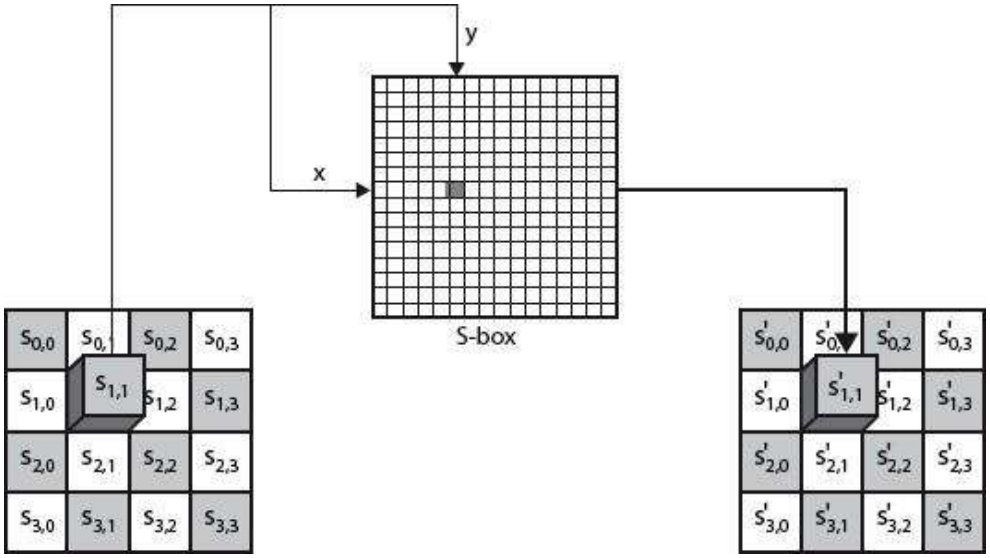


AES ENCRYPTION ROUND

Byte Substitution:

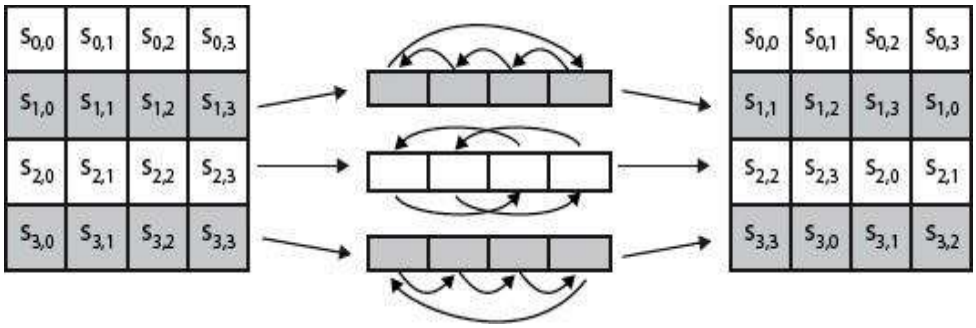
A simple substitution of each byte uses one table of 16x16 bytes containing a permutation of all 256 8-bit values each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)

Eg. byte {95} is replaced by byte in row 9 column 5 which has value {2A}. S-box constructed using defined transformation of values in GF(28) designed to be resistant to all known attacks.



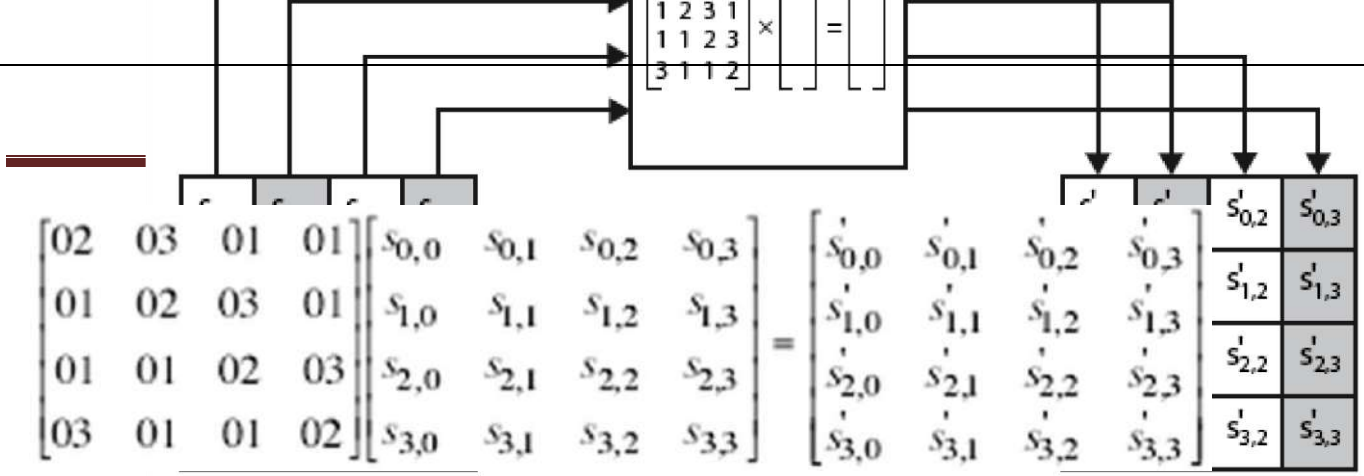
Shift Rows:

A circular byte shift in each; 1st row is unchanged, 2nd row does 1 byte circular shift to left, 3rd row does 2 byte circular shift to left, 4th row does 3 byte circular shift to left decrypt inverts using shifts to right since state is processed by columns, this step permutes bytes between the columns



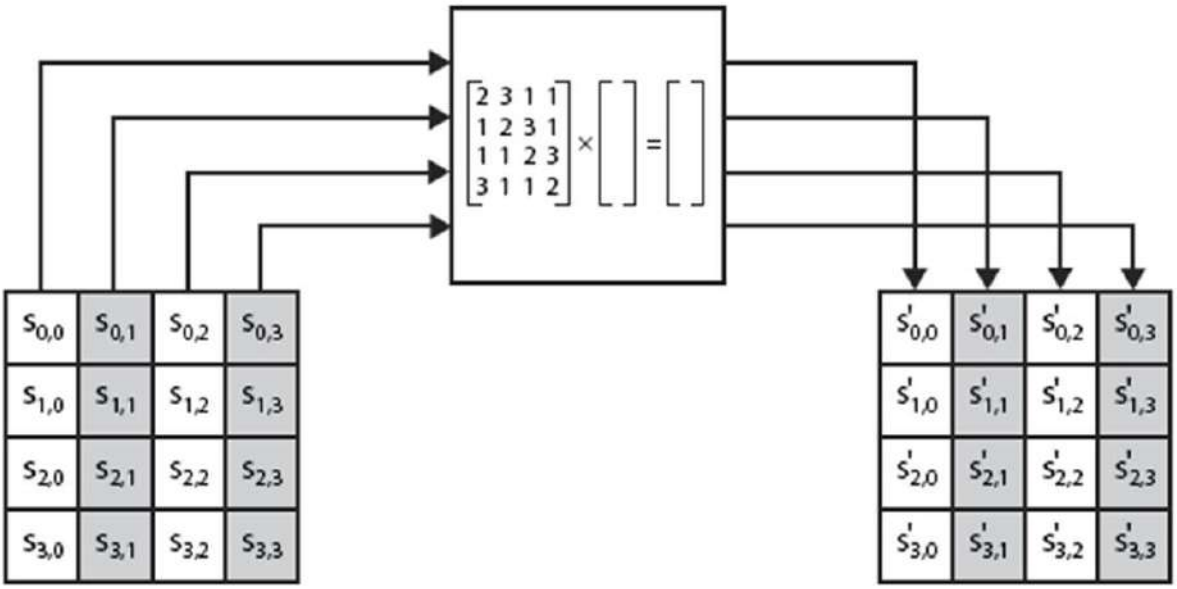
Mix Columns:

Each column is processed separately. Each byte is replaced by a value dependent on all 4 bytes in the column. Effectively a matrix multiplication in GF(28) using prime poly m(x) =x8+x4+x3+x+1



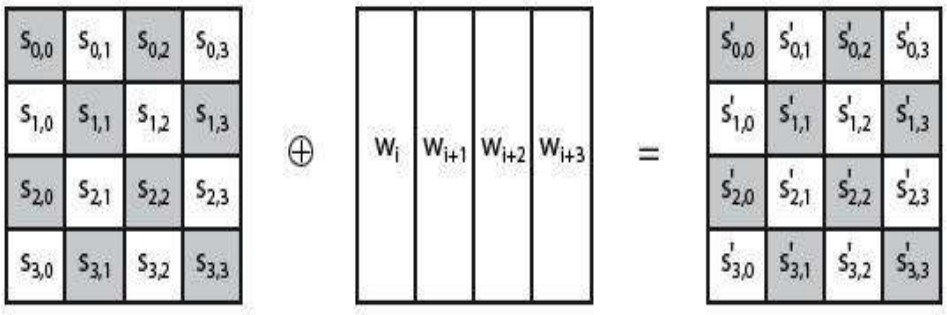
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}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

Mix Columns:



AddRoundKey Transformation

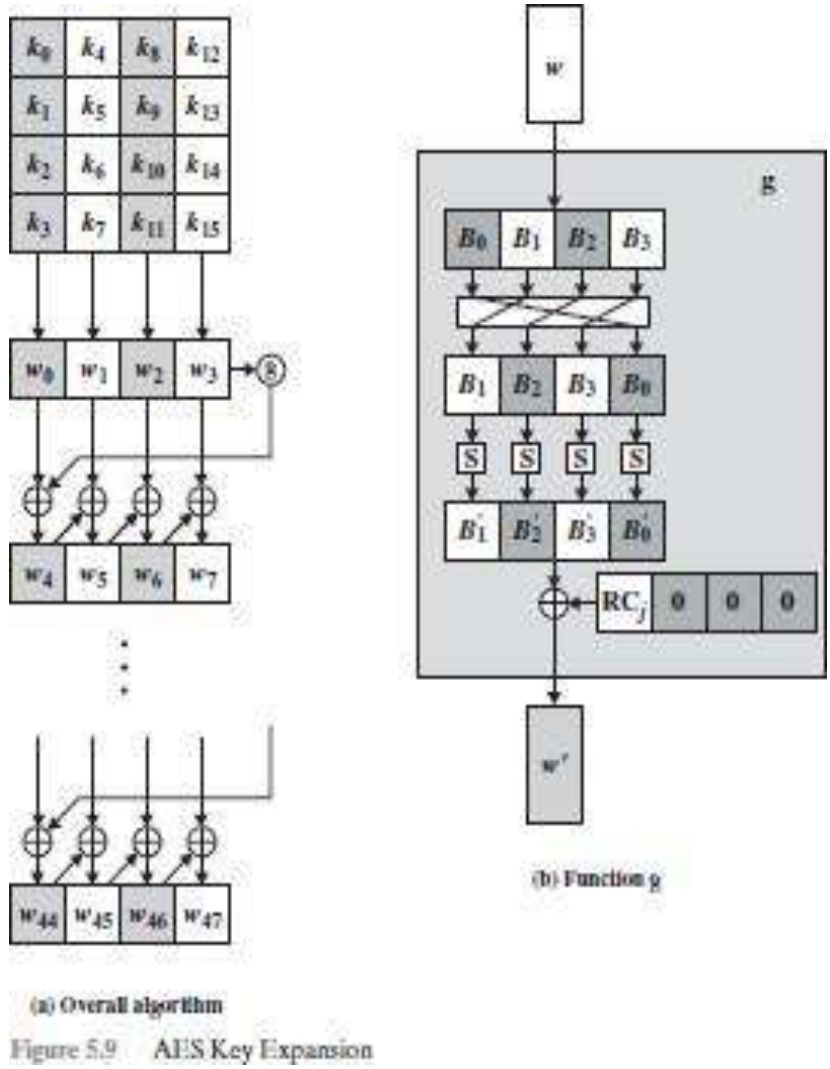
The 128 bits of State are bitwise XORed with the 128 bits of the round key. the operation is viewed as a columnwise operation between the 4 bytes of a State column and one word of the round key; it can also be viewed as a byte-level operation. For example:



47	40	A3	4C		AC	19	28	57		EB	59	8B	1B
37	D4	70	9F		77	FA	D1	5C		40	2E	A1	C3
94	E4	3A	42	⊕	66	DC	29	00	=	F2	38	13	42
ED	A5	A6	BC		F3	21	41	6A		1E	84	E7	D6

AES Key Expansion:

The 128 bit key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added new word depends on the w[i] and w[i-4]. In three out of four cases, a simple XOR is used. For a word whose position in the w array is a multiple of 4, a more complex function is used. The complex function g consists of the following subfunctions.



(a) Overall algorithm

(b) Function g

Figure 5.9 AES Key Expansion

Where R is the called Rotate Constant.

- RotWord (Rotate Word) performs a one-byte circular left shift on a word.
- SubWord(Substitute Word) performs a byte substitution on each byte of its input word, using the S-box
- The result of steps 1 and 2 is XORed with a round constant, R

RC4

- RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security.
- It is a variable key-size stream cipher with byte-oriented operations.
- The algorithm is based on the use of a random permutation.
- RC4 is probably the most widely used stream cipher.
- It is used in the SSL/TLS secure web protocol, & in the WEP & WPA wireless LAN security protocols.
- RC4 was kept as a trade secret by RSA Security, but in September 1994 was anonymously posted on the Internet on the Cypherpunks anonymous remailers list.
- the RC4 key is ued to form a random permutation of all 8-bit values, it then uses that permutation to scramble input info processed a byte at a time.

RC4 Key Schedule

- The RC4 key schedule initialises the state S to the numbers 0..255 After doing this 256 times, the result is a well and truly shuffled array.
- The total number of possible states is 256! - a truly enormous number, much larger even than the 2048-bit (256*8) max key allowed can select. S forms **internal state** of the cipher

```
for i = 0 to 255 do
S[i] = i
T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do  j = (j +
S[i] + T[i]) (mod 256)  swap
(S[i], S[j])
```

RC4 Encryption:

To form the stream key for en/decryption (which are identical), RC4 continues to shuffle the permutation array S by continuing to swap each element in turn with some other entry, and using the sum of these two entry values to select another value from the permutation to use as the stream key, which is then XOR'd with the current message byte.

```
i = j = 0
```

for each message byte M_i

i = (i + 1) (mod 256) j =

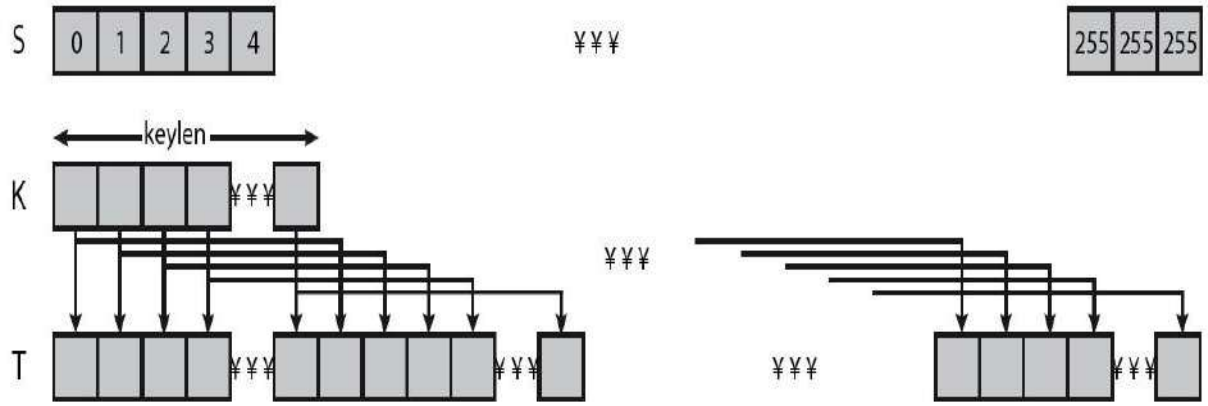
(j + S[i]) (mod 256)

swap(S[i], S[j]) t = (S[i] +

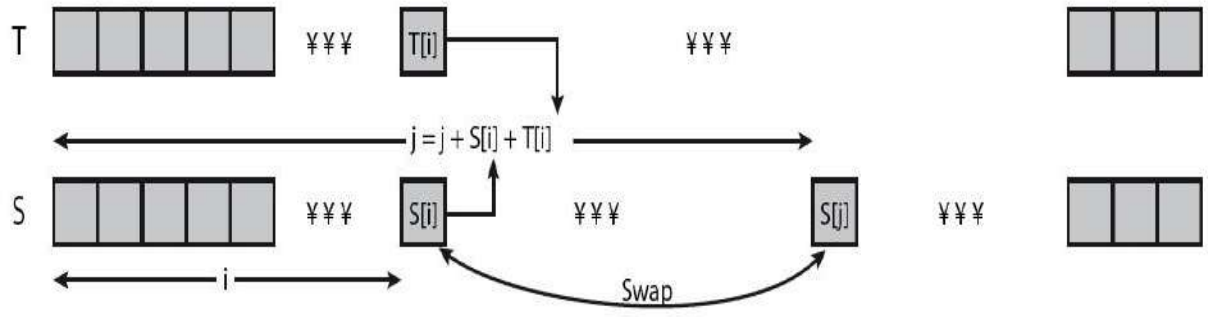
S[j]) (mod 256)

$C_i = M_i$ XOR S[t]

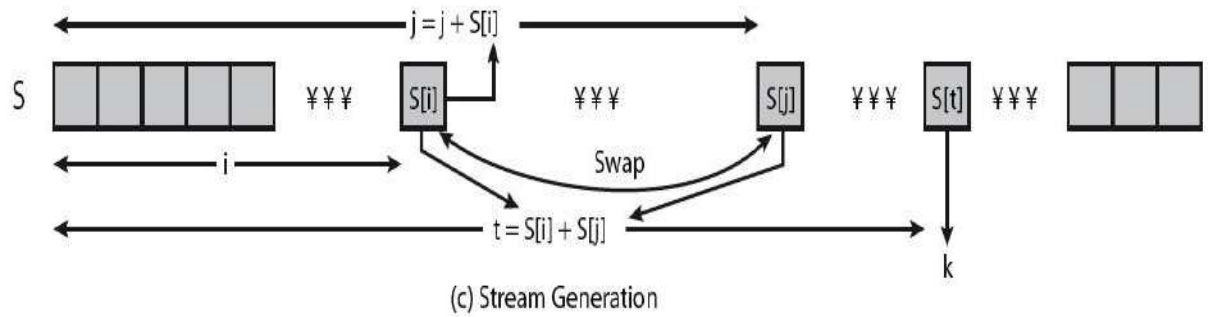
Overview of RC4:



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

RC4 Security:

- A number of papers have been published analyzing methods of attacking RC4, but none of these approaches is practical against RC4 with a reasonable key length, such as 128 bits.
- A more serious problem occurs in its use in the WEP protocol, not with RC4 itself but the way in which keys are generated for use as input to RC4.
- Currently RC4 it's regarded as quite secure, if used correctly, with a sufficiently large key.

KEY DISTRIBUTION:

- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
- Often secure system failure due to a break in the key distribution scheme

Given parties A and B have various key distribution alternatives:

1. A can select key and physically deliver to B
2. Third party can select & deliver key to A & B
3. if A & B have communicated previously can use previous key to encrypt a new key
4. if A & B have secure communications with a third party C, C can relay key between A & B

The strength of any cryptographic system thus depends on the key distribution technique.

For two parties A and B, key distribution can be achieved in a number of ways:

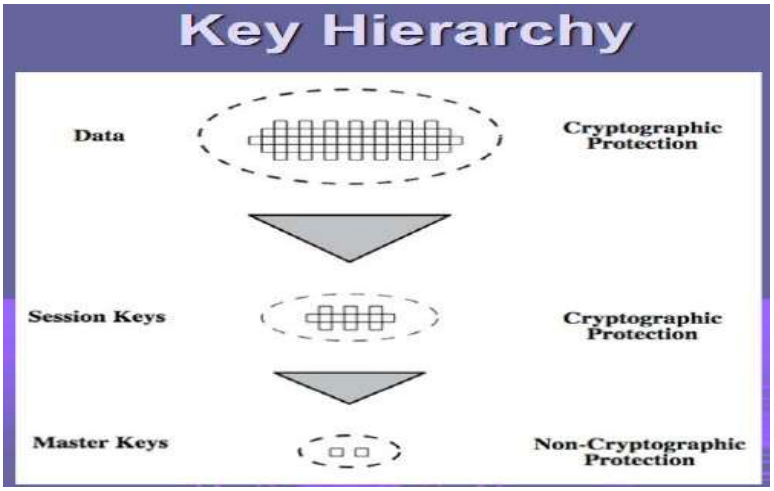
- Physical delivery (1 & 2) is simplest - but only applicable when there is personal contact between recipient and key issuer. This is fine for link encryption where devices & keys occur in pairs, but does not scale as number of parties who wish to communicate grows.
- 3 is mostly based on 1 or 2 occurring first.
- A third party, whom all parties trust, can be used as a **trusted intermediary** to mediate the establishment of secure communications between them (4).
- Must trust intermediary not to abuse the knowledge of all session keys. As number of parties grow, some variant of 4 is only practical solution to the huge growth in number of keys potentially needed.

KEY HIERARCHY:

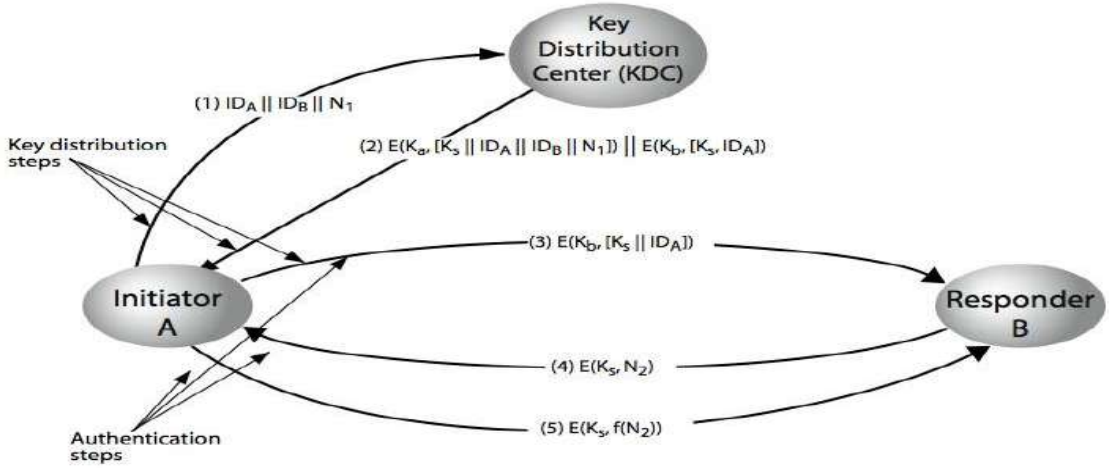
The use of a key distribution center is based on the use of a hierarchy of keys.

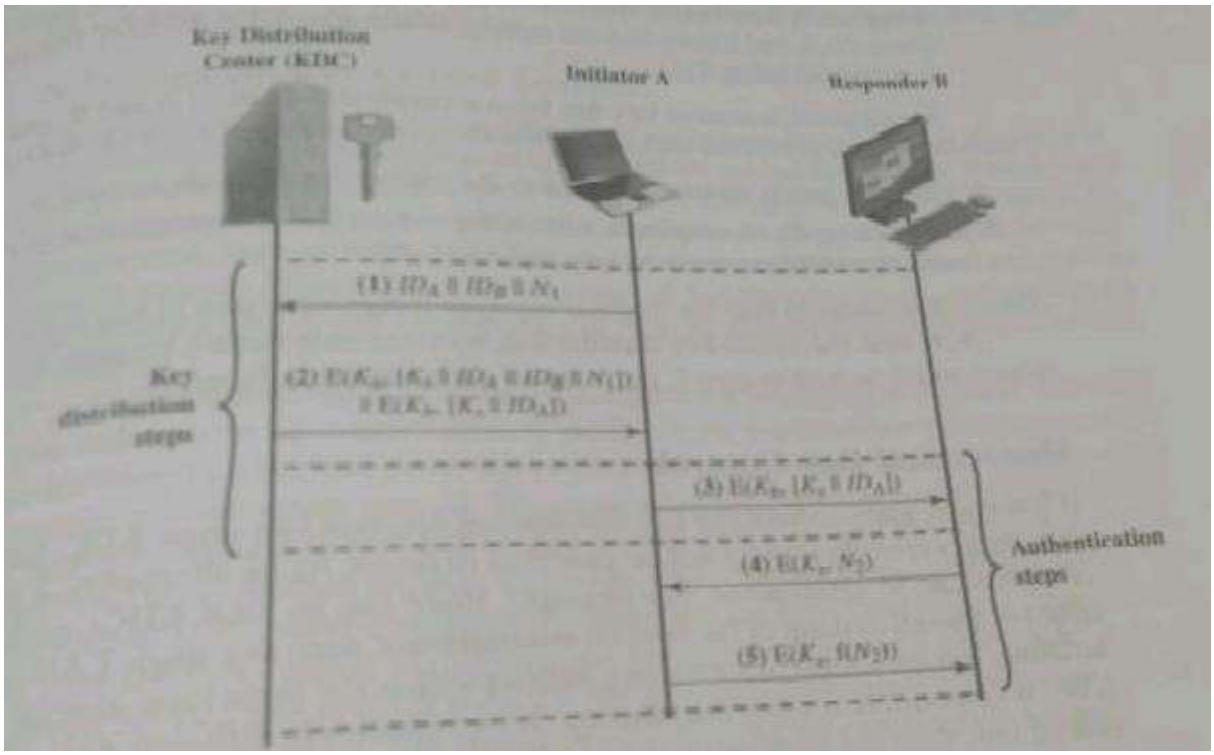
At a minimum, two levels of keys are used:

- A session key, used for the duration of a logical connection;
- A master key shared by the key distribution center and an end system or user and used to encrypt the session key.



Key Distribution Scenario:"Key Distribution Center" (KDC) which shares a unique key with each party (user)





The major issues associated with the use of Key Distribution Centers (KDC's):

- **Hierarchies of KDC's** required for large networks, but must trust each other

  Significant of hierarchical key control. (Nov / Dec 2017)

  - There can be local KDC(Key Distribution Center) responsible for small domain in the large networks.
  - When the two principals are in the same domain the local KDC does the key distribution.
  - When the two principals are in different domain, the local KDC communicates to the global KDC.
  - The key selection can be done by anyone KDC. The number of layers depend upon the network size.

- **session key lifetimes** should be limited for greater security
- use of **automatic key distribution** on behalf of users, but must trust system
- use of **decentralized key distribution**
- **Controlling key usage**

UNIT III PUBLIC KEY CRYPTOGRAPHY

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes – Primality Testing – Factorization – Euler's totient function, Fermat's and Euler's Theorem – Chinese Remainder Theorem – Exponentiation and logarithm

ASYMMETRIC KEY CIPHERS: RSA cryptosystem – Key distribution – Key management – Diffie Hellman key exchange -ElGamal cryptosystem – Elliptic curve arithmetic-Elliptic curve cryptography.

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY

PRIME NUMBER

An integer $p > 1$ is a prime number if and only if its only divisors are ±1 and ±p. Any integer $a > 1$ can be factored in a unique way as

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_t^{a_t}$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each a_i is a positive integer.

Eg, $91 = 7 * 13$
$3600 = 2^4 * 3^2 * 5^2$
$11011 = 7 * 11^2 * 13$

If P is the set of all prime numbers, then any positive integer a can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a_p} \quad \text{where each } a_p \geq 0$$

It is easy to determine the greatest common divisor of two positive integers if we express each integer as the product of primes

Eg $300 = 2^2 * 3^1 * 5^2$
$18 = 2^1 * 3^2$
$\gcd(18, 300) = 2^1 * 3^1 * 5^0 = 6$

The following relationship always holds: If $k = \gcd(a, b)$, then $k_p = \min(a_p, b_p)$ for all p.

TESTING FOR PRIMALITY

For many cryptographic algorithms, it is necessary to select one or more very large prime numbers at random. Thus, we are faced with the task of determining whether a given large number is prime. There is no simple yet efficient means of accomplishing this task.

Miller-Rabin Algorithm

The algorithm due to Miller and Rabin [MILL75, RABI80] is typically used to test a large number for primality.

TEST (*n*)

1. Find integers k, q, with $k > 0$, q odd, so that $(n - 1 = 2^k q)$;
2. Select a random integer a, $1 < a < n - 1$;
3. if $a^q \bmod n = 1$ **then** return("inconclusive");
4. for $j = 0$ **to** $k - 1$ **do**
5. if $a^{2^j q} \bmod n = n - 1$ **then** return("inconclusive");
6. return("composite");

Example 1: Let us apply the test to the prime number $n = 29$.
$(n - 1) = 28 = 2^2(7) = 2^k q$.
First, let us try $a = 10$.
Compute $10^7 \bmod 29 = 17$,
$(10^7)^2 \bmod 29 = 28$, and the test returns inconclusive.
So n is prime number.

FERMAT 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 (also called as Fermat's little Theorem)

Definition:

If P is prime and a is a positive integer not divisible by p, then $\mathbf{a^{p-1} \equiv 1 \pmod p}$

Fermat's little theorem is the basis for the Fermat primality test and is one of the fundamental results of elementary number theory.

This theorem is useful in generating public key in RSA and Primality testing

Proof:

1) Consider the set of positive integers less than *p:*{1,2,3..p-1}
2) Multiply each element by *a, modulo p* to get the set

   X ={ a mod p,2a mod p….(p-1) a mod p}.

   - None of the elements of X is equal to zero because *p* does not divide *a*.
   - No two of the integers in X are equal.
   - We know that (p-1) elements of X are all positive integers with no two elements are equal. So, we can conclude X consists of the set of integers {1,2,……..,p-1} in some order

3) Multiplying the numbers in both sets (p and X) and taking the result mod *p* yields.

$$a * 2a * \ldots * (p-1)a \equiv [(1*2*\ldots*(p-1)] \pmod p$$
$$\{1 * 2 * \ldots * (p-1)\}\ a^{p-1} \equiv [(1*2*\ldots*(p-1)] \pmod p$$
$$(p-1)!\ a^{p-1} \equiv (p-1)! \pmod p$$
$$\mathbf{a^{p-1} \equiv 1 \pmod p}$$

Example

$a = 7, p = 19$ $\quad a^{p-1} \equiv 1 \pmod p.$

- $7^2 = 49 \equiv 11 \pmod{19}$
- $7^4 = 7^2 \times 7^2 = 121 \equiv 7 \pmod{19}$
- $7^8 = 7^4 \times 7^4 = 7 \times 7 = 49 \equiv 11 \pmod{19}$
- $7^{16} \equiv 7^8 \times 7^8 = 11 \times 11 = 121 \equiv 7 \pmod{19}$
- $a^{p-1} = 7^{18} = 7^{16} * 7^2 \equiv 7 * 11 \equiv 1 \pmod{19}$

An alternative form of Fermat's theorem:

If p is prime and a is a positive integer, $\mathbf{a^p \equiv a \pmod p}$

Eg: a=3, p=5
$a^p = 3^5 = 243 \equiv 3 \pmod 5 = a \pmod p$
$a^p \equiv a \pmod p$

Euler's Totient function

Euler's totient function written as $\emptyset(n)$, (called as phi) is defined as the number of positive integers less than n and relatively prime (Co-Prime) to ***n***.

The Properties are as follows

1) ø(1) = 1
2) ø(p)=p-1 for p (p prime)
3) ø(p.q)=(p-1)x(q-1) for p.q (p,q prime)

Suppose that we have two prime numbers *p* and *q*, with *p* not equal to *q*. Then we can show that

n=pq.

ø(n)= ø(pq)= ø(p)* ø(q)=(p-1)*(q-1)

Examples:

1) ø(37) = 36 {ø(p)=p-1 for p (p prime)]
2) ø(21) = ø(3)* ø(7)= (3–1)x(7–1) = 2x6 = 12 where the 12 integers are {1,2,4,5,8,10,11,13,16,17, 19, 20}

[ø(p.q)=(p-1)x(q-1) for p.q (p,q prime)]

Sample Examples

1. What is the value of Φ(13)?
   Because 13 is a prime, Φ (13) = (13 −1) = 12.
2. What is the value of Φ (10)?
   We can use the third rule: Φ (10) = Φ (2) × Φ (5) = 1 × 4 = 4, because 2 and 5 are primes.
3. What is the number of elements in Z14*?
   The answer is Φ (14) = Φ (7) × Φ (2) = 6 × 1 = 6. The members are 1, 3, 5, 9, 11, and 13.

Euler's theorem

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

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

Proof:

The above equation is true, if n is prime, because in that case *ø(n)=(n-1)* and Fermat's theorem holds. However it holds for any integer *n*.

1)Recall that ø(n)is the number of positive integers less than n that are relatively prime to *n*. consider the set of such integers, labeled as follows:

$$R=\{x_1, x_2 \ldots . x_{ø(n)}\}$$

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

2)Now multiply each element by *a modulo n:*

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

3) The set S is a permutation of R, by the following reasons:
1.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.
2. There are no duplicates in S. if $ax_i \bmod n = ax_i \bmod n$, then $x_i = x_j$

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

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

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

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

An alternative form of the theorem is also useful:

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

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

THE CHINESE REMAINDER THEOREM

The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

Let $m_1, m_2 \ldots\ldots m_k$ be integers with $\gcd(m_i, m_j) = 1$, whenever $i \neq j$. Let $a_1, a_2 \ldots\ldots a_k$ be integers, there exists exactly one solution x (mod $m_1, m_2 \ldots\ldots m_k$) to the simultaneous congruences

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$x \equiv a_k \pmod{m_k}$$

If $n_1,n_2,..,n_k$ are positive integers that are pairwise co-prime and $a_1,a_2,\ldots,a_k$ are any integers, then CRT is used to find the values of x that solves the following congruence simultaneously.

Value of $x=(a_1m_1y_1+a_2m_2y_2+\ldots+a_km_ky_k) \bmod M$
Where $M=n_1n_2n_3..n_k$

$m_i=M/n_i$

$m_iy_i=1 \bmod n_i$

Example 1:
Find the solution to the simultaneous equations: $x \equiv 1 \pmod 5$, $x \equiv 2 \pmod 6$, $x \equiv 3 \pmod 7$.
Solution

$M=n_1n_2n_3$

$M=5*6*7=210$

$m_i = M/n_i$

m1=210/5=42

m2=210/6=35

m3=210/7=30

$m_i y_i = 1 \bmod n_i$

$42y_1 = 1 \bmod 5$

$y_1 = 2$

$35y_2 = 1 \bmod 6$

$y_2 = 5$

$30y_3 = 1 \bmod 7$

$y_3 = 2$

$x = (a_1 m_1 y_1 + a_2 m_2 y_2 + a_3 m_3 y_3) \bmod M$

$= ((1*42*2)+(2*35*5)+(3*30*3)) \bmod 210$

$= 193$

Example 2:

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.

$$x = 3 \bmod 7$$
$$x = 3 \bmod 13$$
$$x = 0 \bmod 12$$

If we follow the four steps, we find x = 276. We can check that 276 = 3 mod 7, 276 = 3 mod 13 and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

Example 3:

A bag has contained number of pens if you take out 3 pens at a time 2 pens are left. If you take out 4 pens at a time 1 pen is left and if you take out 5 pens at a time 3 pens are left in the bag. What is the number of pens in the bag.

$x \equiv 2 \bmod 3$

$x \equiv 1 \bmod 4$

$x \equiv 3 \bmod 5$

$a_1 = 2$

$a_2 = 1$

$a_3 = 3$

$n_1 = 3$

$n_2 = 4$

$n_3 = 5$

$M = n_1 n_2 n_3$

$M = 3*4*5 = 60$

$m_i=M/n_i$
m1=60/3=20
m2=60/4=15
m3=60/5=12

$m_iy_i=1 \bmod n_i$
$20y_1=1 \bmod 3$
$Y_1=2 \bmod 3$

$15y_2= 1 \bmod 4$
$y_2=3 \bmod 4$

$12y_3=1 \bmod 5$
$y_3=3 \bmod 5$

$x=(a_1m_1y_1+a_2m_2y_2+ a_3m_3y_3) \bmod M$
=((2*20*2)+(1*15*3)+(3*12*3)) mod 60
=233 mod 60
=53

DISCRETE LOGARITHMS.

Discrete logarithms are fundamental to a number of public-key algorithms. Discrete logarithms are analogous to ordinary logarithms but are defined using modular arithmetic.

Discrete logarithms are fundamental to a number of public-key algorithms, including Diffie-Hellman key exchange and the digital signature algorithm (DSA)

The Powers of an Integer, Modulo *n*

Recall from Euler's theorem that, for every and that are relatively prime,

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

Where $\phi(n)$, Euler's totient function, is the number of positive integers less than and relatively prime to . Now consider the more general expression:

$$a^m = 1 \pmod{n}$$

If a and n are relatively prime, then there is at least one integer m that satisfies Equation, namely M= $\phi(n)$, . The least positive exponent m for which Equation holds is referred to in several ways:

- The order of a(mod n)
- The exponent to which a belongs (mod n)
- The length of the period generated by a

The highest possible exponent to which a number can belong (mod n) is $\phi(n)$. If a number is of this order, it is referred to as a **primitive root** of n .The importance of this notion is that if is a primitive root of , then its powers

$$a, a^2, \ldots, a^{\phi(n)}$$

Logarithms for Modular Arithmetic

A primitive root of a prime number p is one whose powers modulo p generate all the integers from 1 to p - 1. That is, if 'a' is a primitive root of the prime number p, then the numbers

a mod p, a^2 mod p,…, a^{p-1} mod p

are distinct and consist of the integers from 1 through p - 1 in some permutation.

For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$b \equiv a^i \pmod p$ where 0 … i … (p - 1)

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

We denote this value as dloga,p(b)

Note the following:

dlog$_{a,p}(a)$ = 1 because $a^1 \bmod p = a$

dlog$_{a,p}(1)$ = 0 because $a^0 \bmod p = 1 \bmod p = 1$

Calculation of Discrete Logarithms

Consider the equation

$$y = gx \bmod p$$

Given g, x , and p , it is a straightforward matter to calculate y. At the worst, we must perform repeated multiplications, and algorithms exist for achieving greater efficiency.

PUBLIC KEY CRYPTOGRAPHY

Introduction to Public key Cryptography:

- Public key cryptography also called as **asymmetric cryptography**.
- It was invented by whitfield **Diffie** and Martin **Hellman** in 1976. Sometimes this cryptography also called as **Diffie-Helman Encryption**.
- Public key algorithms are based on mathematical problems which admit no efficient solution that are inherent in certain integer factorization, discrete logarithm and Elliptic curve relations.

Public key Cryptosystem Principles:

- The concept of public key cryptography in invented for two most difficult problems of Symmetric key encryption.
  - *The Key Exchange Problem*
  - *The Trust Problem*

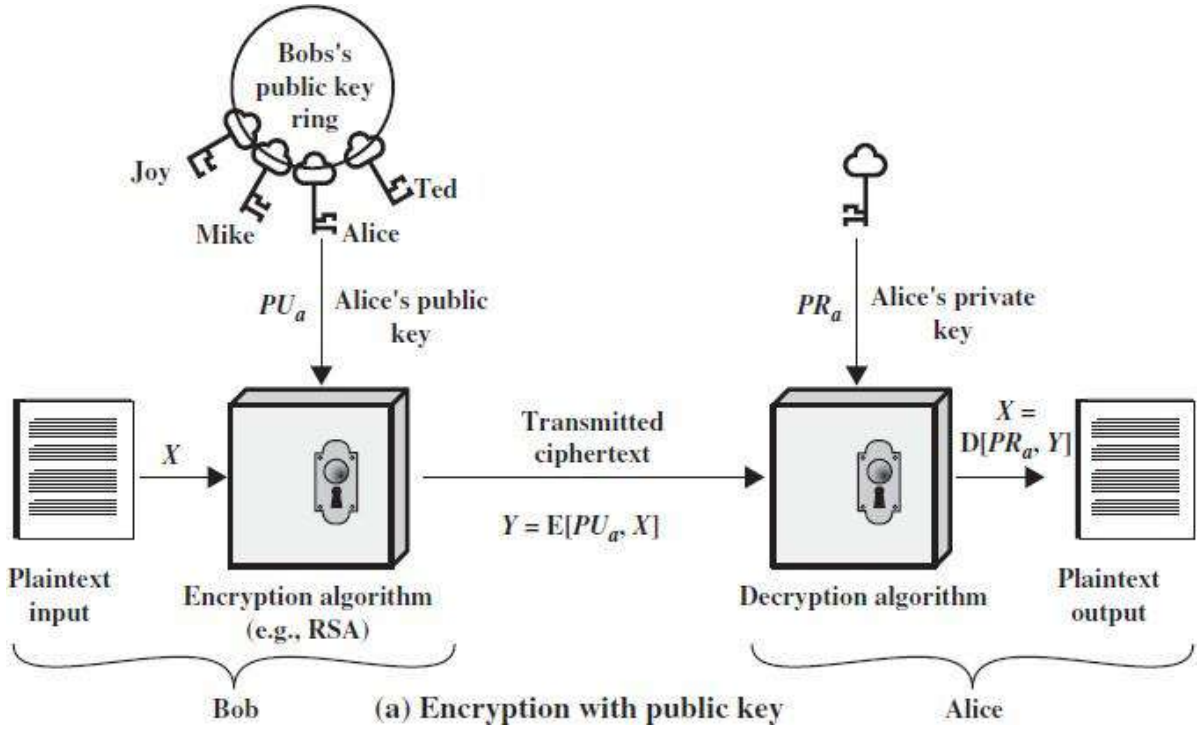
  The Key Exchange Problem: *The key exchange problem arises from the fact that communicating parties must somehow share a secret key before any secure communication can be initiated, and both parties must then ensure that the key remains secret. Of course, direct key exchange is not always feasible due to risk, inconvenience, and cost factors.*

  The Trust Problem: *Ensuring the integrity of received data and verifying the identity of the source of that data can be very important. Means in the symmetric key cryptography system, receiver doesn't know whether the message is coming for particular sender.*

- This public key cryptosystem uses two keys as pair for encryption of plain text and Decryption of cipher text.
- These two keys are names as "**Public key**" and "**Private key**". The private key is kept secret whereas public key is distributed widely.
- A message or text data which is encrypted with the public key can be decrypted only with the corresponding private-key
- This two key system very useful in the areas of confidentiality (secure) and authentication

A **public-key encryption** scheme has six ingredients		
1	**Plaintext**	This is the readable message or data that is fed into the algorithm as input.
2	**Encryption algorithm**	The encryption algorithm performs various transformations on the plaintext.
3	**Public key**	This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input
4	**Private Key**	
5	**Ciphertext**	This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
6	**Decryption algorithm**	This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

Public key cryptography for providing confidentiality (secrecy)



The essential steps are the following.

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

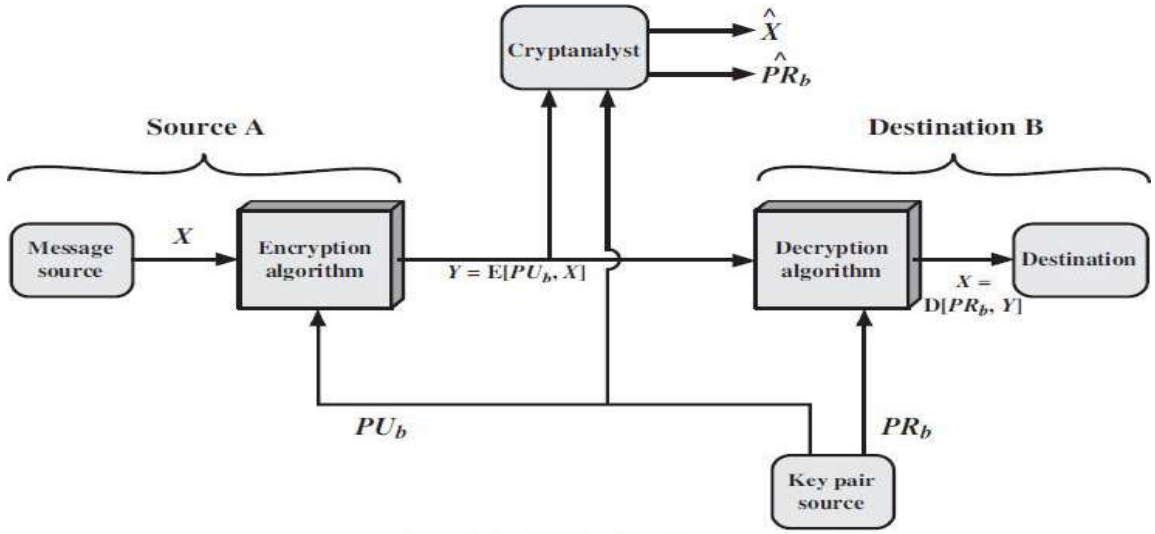


Figure 9.2 Public-Key Cryptosystem: Secrecy

There is some source A that produces a message in plaintext $\boldsymbol{X = [X1, X2, \ldots, XM]}$.

The $\boldsymbol{M}$ elements of $\boldsymbol{X}$ are letters in some finite alphabet. The message is intended for destination **B**. B generates a related pair of keys: a public key, $\boldsymbol{PU_b}$, and a private key, $\boldsymbol{PR_b}$.

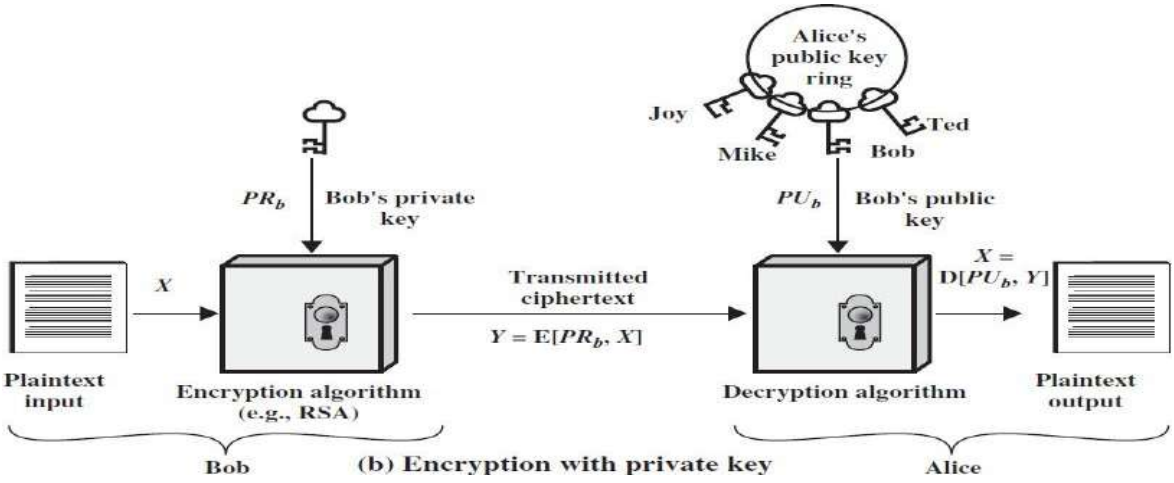
$\boldsymbol{PR_b}$ is known only to B, whereas $\boldsymbol{PU_b}$ is publicly available and therefore accessible by A. With the message X and the encryption key $\boldsymbol{PU_b}$ as input, A forms the ciphertext $Y = [Y1, Y2, \ldots, Y_N]$:

$$Y = \mathrm{E}(PU_b, X)$$

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

$$X = \mathrm{D}(PR_b, Y)$$

<u>Public key cryptography for proving Authentication:</u>



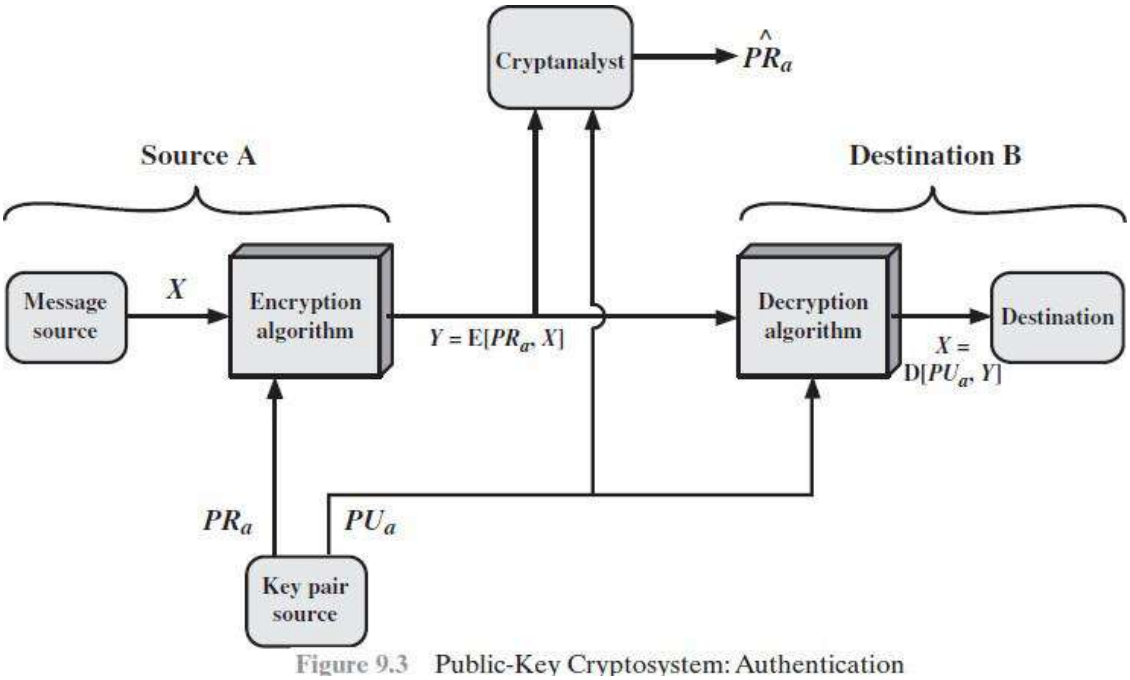


Figure 9.3 Public-Key Cryptosystem: Authentication

The above diagrams show the use of public-key encryption to provide authentication:

$$Y = \mathrm{E}(PR_a, X)$$

$$X = \mathrm{D}(PU_a, Y)$$

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

Public key cryptography for both authentication and confidentiality (Secrecy)

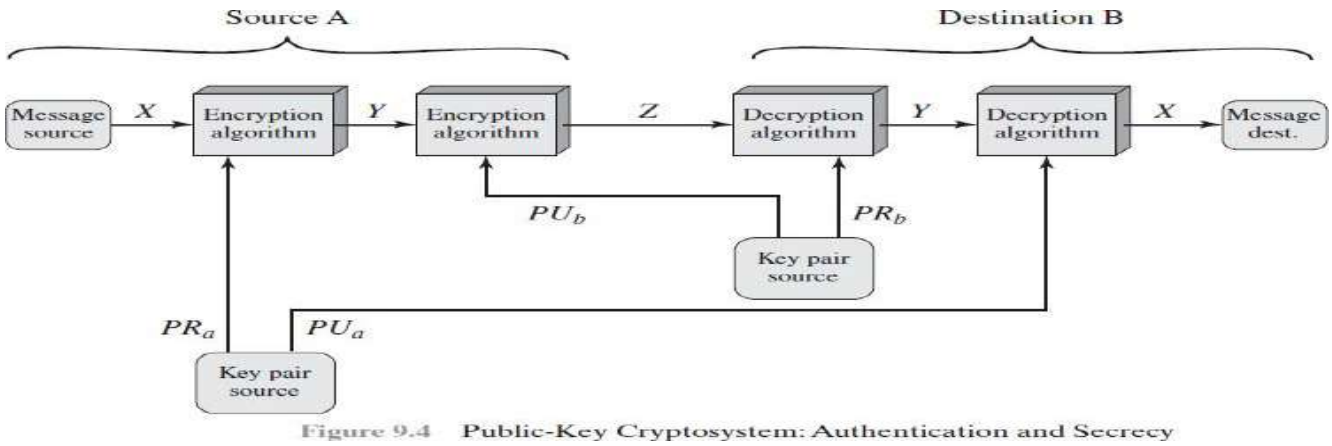


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

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

$$Z = \mathrm{E}(PU_b, \mathrm{E}(PR_a, X))$$

$$X = \mathrm{D}(PU_a, \mathrm{D}(PR_b, Z))$$

In this case, we begin as before by encrypting a message, using the sender's private key. This provides the digital signature. Next, we encrypt again, using the receiver's public key. The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus, confidentiality is provided.

Applications for Public-Key Cryptosystems

The use of **public-key cryptosystems** into three categories

• ***Encryption /decryption***: The sender encrypts a message with the recipient's public key.

• ***Digital signature***: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

• ***Key exchange***: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

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

RSA

- It is the most common public key algorithm.
- This RSA name is get from its inventors first letter (Rivest (R), Shamir (S) and Adleman (A)) in the year 1977.
- The RSA scheme is a block cipher in which the plaintext & ciphertext are integers between 0 and n-1 for some 'n'.
- A typical size for 'n' is 1024 bits or 309 decimal digits. That is, n is less than 2^{1024}

Description of the Algorithm:

- RSA algorithm uses an expression with exponentials.

- In RSA plaintext is encrypted in blocks, with each block having a binary value less than some number n. that is, the block size must be less than or equal to **$\log_2(n)$**
- **RSA** uses two exponents 'e' and 'd' where e is public and d is private.
- Encryption and decryption are of following form, for some PlainText 'M' and CipherText block 'C'

$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

$$M = C^d \bmod = (M^e \bmod n)^{d} \text{ mon } n = (M^e)^d \bmod n = M^{ed} \bmod n$$

- Both sender and receiver must know the value of n.
- The sender knows the value of 'e' & only the reviver knows the value of 'd' thus this is a public key encryption algorithm with a

  Public key PU={e, n}

  Private key PR={d, n}

Requirements:

The RSA algorithm to be satisfactory for public key encryption, the following requirements must be met:

1. It is possible to find values of e, d n such that " $\mathbf{M^{ed} \bmod n = M}$ " for all M<n
2. It is relatively easy to calculate " $\mathbf{M^e \bmod n}$ " and " $\mathbf{C^d \bmod n}$ "for M<n
3. It is infeasible to determine "d" given 'e' & 'n'. The " $\mathbf{M^{ed} \bmod n = M}$ " relationship holds if 'e' & 'd' are multiplicative inverses modulo Ø(n).

   Ø(n) is Euler Totient function

   For p,q primes where p*q and p≠q.

   Ø(n)= Ø(pq)=(p-1)(q-1)

Then the relation between 'e' & 'd' can be expressed as " $\boxed{ed \bmod \emptyset(n)=1}$ "
this is equivalent to saying

$$ed \equiv 1 \bmod \emptyset(n)$$

$$d \equiv e^{-1} \bmod \emptyset(n)$$

That is 'e' and 'd' are multiplicative inverses mod Ø(n).

Note: according to the rules of modular arithmetic, this is true only if 'd' (and 'e') is relatively prime to Ø(n).

Equivalently **gcd(Ø(n), d)=1.**

Steps of RSA algorithm:

Step 1 Select 2 prime numbers p & q

Step 2 Calculate n=pq

Step 3Calculate Ø(n)=(p-1)(q-1)

Step 4 Select or find integer e (public key) which is relatively prime to Ø(n) i.e., e with gcd (Ø(n), e)=1 where 1<e< Ø(n).

Step 5 Calculate "d" (private key) by using following condition. $ed \equiv 1 \bmod \emptyset(n)$

d< Ø(n).

Step 6 Perform encryption by using $C = M^e \bmod n$

Step 7 perform Decryption by using $M = C^d \bmod n$

Example:

1. Select two prime numbers, $p = 17$ and $q = 11$.

2. Calculate $n = pq = 17 \times 11 = 187$.

3. Calculate $\emptyset(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.

4. Select e such that e is relatively prime to $\emptyset(n) = 160$ and less than $\emptyset(n)$; we choose $e = 7$.

5. Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$. The correct value is $d = 23$, because $23 * 7 = 161 = (1 \times 160) + 1$; d can be calculated using the extended Euclid's algorithm

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$.

The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows.

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

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

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

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

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

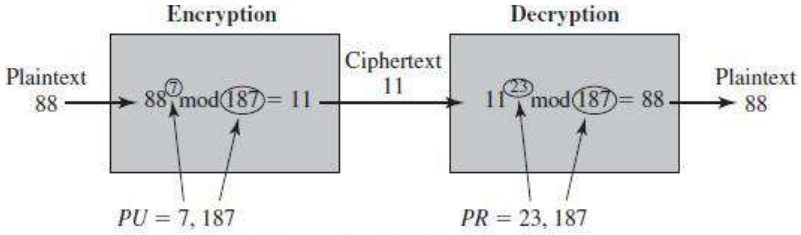


Figure 9.6 Example of RSA Algorithm

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

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

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

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

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

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

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

RSA Attacks

There are four possible approaches to attack the RSA:

- **Brute force:** This involves trying all possible private keys. The defence against this attack is the use of large key space.
- **Mathematical attacks**: There are several approaches, all equivalent in effort to factoring the product of two primes. Three approaches that could be identified of this type are:
    - Factor n into its two prime factors which enables calculation of $\phi(n) = (p - 1) \times (q - 1)$, which in turn enables determination of $d \equiv e^{-1} \pmod{\phi(n)}$.
    - Determine $\phi(n)$ directly, without first determining p and q. Again, this enables determination of $d \equiv e^{-1} \pmod{\phi(n)}$.
    - Determine d directly, without first determining $\phi(n)$.
- **Timing attacks**: These depend on the running time of the decryption algorithm. This attack is alarming for two reasons namely it comes from a completely unexpected direction, and it is a cipher text-only attack. Modular exponentiation algorithm that is accomplished bit by bit, with one modular multiplication performed every iteration and an additional modular multiplication performed for each 1 bit can be used to perform this attack. Simple counter measures could be used to overcome the timing attack. They are
    - ***Constant exponentiation time***: Ensure that all exponentiations take the same amount of time before returning a result. This is a simple fix but does degrade performance.
    - ***Random delay***: Better performance could be achieved by adding a random delay to the exponentiation algorithm to confuse the timing attack. Kocher points out that if defenders don't add enough noise, attackers could still succeed by collecting additional measurements to compensate for the random delays.
    - ***Blinding***: Multiply the cipher text by a random number before performing exponentiation. This process prevents the attacker from knowing what cipher text bits are being processed inside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack.
- **Chosen cipher text attacks (CCAs)**: This type of attack exploits properties of the RSA algorithm. It is defined as an attack in which the adversary chooses a number of ciphertexts and is then given the corresponding plaintexts, decrypted with the target's

private key. Thus, the adversary could select a plaintext, encrypt it with the target's public key, and then be able to get the plaintext back by having it decrypted with the private key. Clearly, this provides the adversary with no new information. Instead, the adversary exploits properties of RSA and selects blocks of data that, when processed using the target's private key, yield information needed for cryptanalysis.

To overcome this simple attack, practical RSA-based cryptosystems randomly pad the plaintext prior to encryption. More sophisticated CCAs are possible and simple padding with a random value is insufficient to provide the desired security. To counter such attacks modifying the plaintext using a procedure known as optimal asymmetric encryption padding will help.

Diffie-Hellman key exchange is the first published public key algorithm, also known as exponential key agreement. And it is based on mathematical principles. The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages.This algorithm itself is limited to exchange of the keys. Security of algorithm depends on computing discrete logarithms values.

KEY MANAGEMENT

There are actually two distinct aspects to the use of public-key cryptography:

- The distribution of public keys
- The use of public-key encryption to distribute secret keys

1. Distribution of Public Keys

There are four different schemes

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

A.Public announcement

Any participant can send his or her public key to any other participant or **broadcast** the key to the community. Uncontrolled Public-Key Distribution

Limitation : Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Authentication is needed to avoid this problem.

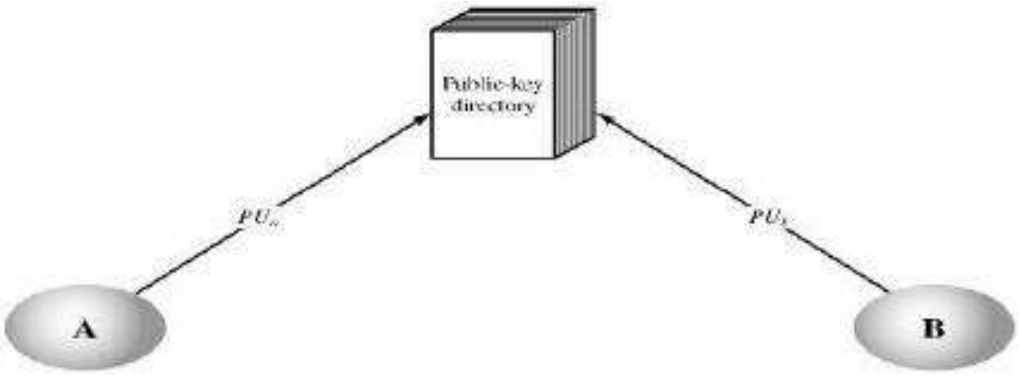
B. Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.

i)The authority maintains a directory with a {name, public key} entry for each participant.

ii)Each participant registers a public key with the directory authority.

iii) Participants could also access the directory electronically.



Limitation :

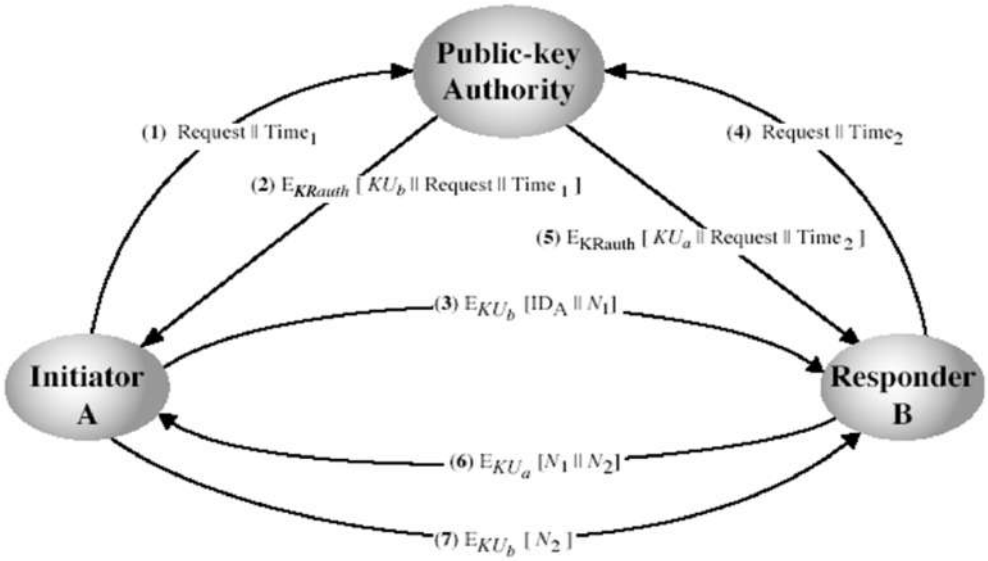
An Adversary may impersonate by stealing the private key of public key directory and falsely send the public key details.

An attacker may attack the records stored in the directory.

C. Public-key authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory.

Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.



i)A sends a time stamped message to the public-key authority containing a request for the current public key of B.

ii) The authority responds with a message that is encrypted using the authority's private key, PRauth. Thus, A is able to decrypt the message using the authority's public key. Therefore, A is assured

that the message originated with the authority. The message includes the following:

- B's public key, PUb which A can use to encrypt messages destined for B
- The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority
- The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B's current public key

iii) A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (IDA) and a nonce (N1), which is used to identify this transaction uniquely.

iv) B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

v) At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange.

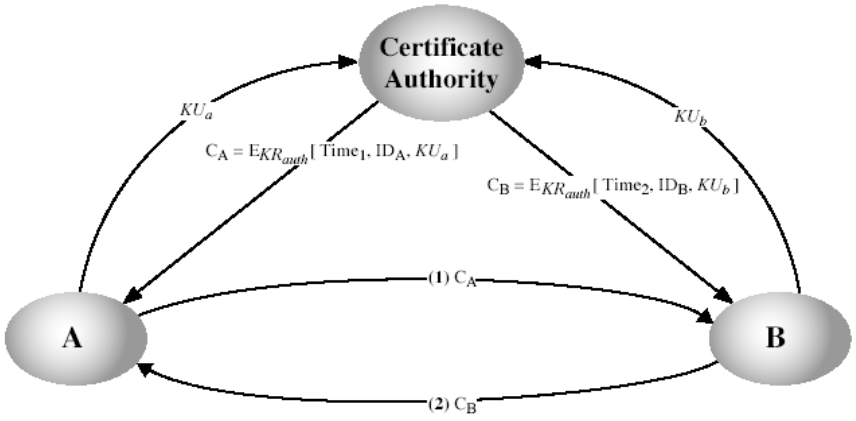
v) B sends a message to A encrypted with PUa and containing A's nonce (N1) as well as a new nonce generated by B (N2) Because only B could have decrypted message (3), the presence of N1 in message (6) assures A that the correspondent is B.

vi) A returns N2, encrypted using B's public key, to assure B that its correspondent is A.

Limitations : Bottleneck may occur in public authority. Tampering of records stored by the authority may take place.

D. Public key certificate

A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.



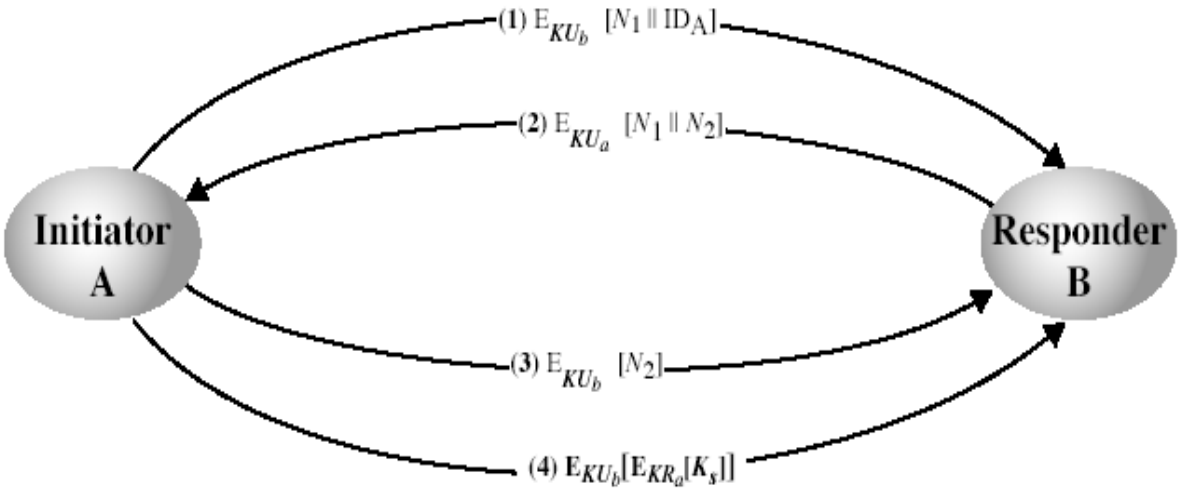
Typically, the third party is a certificate authority, such as a government agency or a financial institution, that is trusted by the user community.

A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate.

2. Secret Key Distribution with Confidentiality and Authentication

i) A uses B's public key to encrypt a message to B containing an identifier of A (IDA) and a nonce (N1), which is used to identify this transaction uniquely.

ii) B sends a message to A encrypted with PUa and containing A's nonce (N1) as well as a new nonce generated by B (N2)



iv) A returns *N*2 encrypted using B's public key, to assure B that its correspondent is A.
A selects a secret key *Ks* and sends *M* = E(*PUb*, E(*PRa*, *Ks*)) to B. Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.

v) B computes D(*PUa*, D(*PRb*, *M*)) to recover the secret key.

Elagamal Cryptographic system

- Public-key cryptosystem related to D-H
- uses exponentiation in a finite field
- with security based difficulty of computing discrete logarithms, as in D-H
- Used in number of standards including DSS (Digital Signature Standard) and S/MIME e-mail standard

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

How K is recovered by the decryption process:

$K = (Y_A)^k \bmod q$ — K is defined during the encryption process

$K = (\alpha^{X_A} \bmod q)^k \bmod q$ — substitute using $Y_A = \alpha^{X_A} \bmod q$

$K = \alpha^{kX_A} \bmod q$ — by the rules of modular arithmetic

$K = (C_1)^{X_A} \bmod q$ — substitute using $C_1 = \alpha^k \bmod q$

Next, using K, we recover the plaintext as

$$C_2 = KM \bmod q$$

$$(C_2 K^{-1}) \bmod q = KMK^{-1} \bmod q = M \bmod q = M$$

Example:

message.

For example, let us start with the prime field GF(19); that is, $q = 19$. It has primitive roots {2, 3, 10, 13, 14, 15}. We choose $\alpha = 10$.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 5$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$.
3. Alice's private key is 5 and Alice's public key is $\{q, \alpha, Y_A\} = \{19, 10, 3\}$.

Suppose Bob wants to send the message with the value $M = 17$. Then:

1. Bob chooses $k = 6$.
2. Then $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$.
3. So
   $C_1 = \alpha^k \bmod q = \alpha^6 \bmod 19 = 11$
   $C_2 = \text{KM} \bmod q = 7 \times 17 \bmod 19 = 119 \bmod 19 = 5$
4. Bob sends the ciphertext (11, 5).

For decryption:

1. Alice calculates $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$.
2. Then K^{-1} in GF(19) is $7^{-1} \bmod 19 = 11$.
3. Finally, $M = (C_2 K^{-1}) \bmod q = 5 \times 11 \bmod 19 = 55 \bmod 19 = 17$.

DIFFIE- HELLMAN KEY EXCHANGE

Algorithm for Diffie-Hellman Key Exchange:

Step 1 two public known numbers q, α

q Prime number

α primitive root of q and α< q.

Step 2 if A & B users wish to exchange a key

a) User A select a random integer X_A<q and computes $Y_A = \alpha^{X_A} \bmod q$

b) User B independently select a random integer X_B <q and computes $Y_B = \alpha^{X_B} \bmod q$

c) Each side keeps the X value private and Makes the Y value available publicly to the outer side.

Step 3 User A Computes the key as $K = (Y_B)^{X_A} \bmod q$

User B Computes the key as $K = (Y_A)^{X_B} \bmod q$

Step 4 two calculation produce identical results

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

$K = (\alpha^{X_B} \bmod q)^{X_A} \bmod q$ (We know that $Y_B = \alpha^{X_B} \bmod q$)

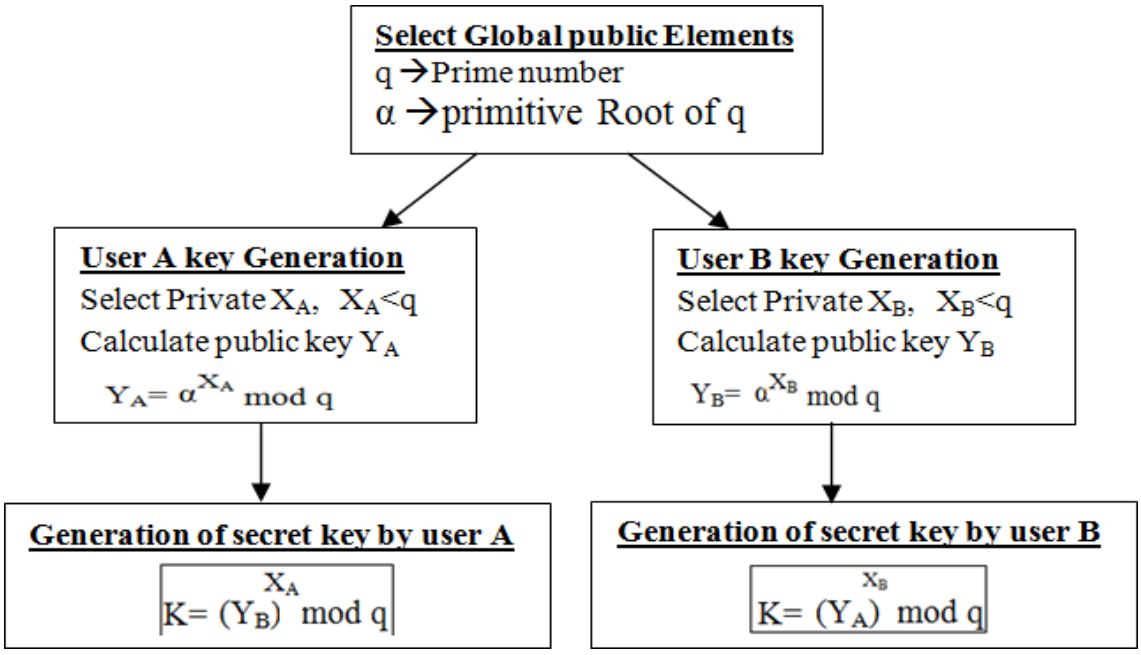
$= (\alpha^{X_B})^{X_A} \bmod q$

$= (\alpha^{X_A})^{X_B} \bmod q$

$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$

$= (Y_A)^{X_B} \bmod q$ (We know that $Y_A = \alpha^{X_A} \bmod q$)

The result is that the two sides have exchanged a secret key.



Example: 1

Here is an example. Key exchange is based on the use of the prime number $q = 353$ and a primitive root of 353, in this case $\alpha = 3$. A and B select secret keys $X_A = 97$ and $X_B = 233$, respectively. Each computes its public key:

A computes $Y_A = 3^{97} \bmod 353 = 40$.

B computes $Y_B = 3^{233} \bmod 353 = 248$.

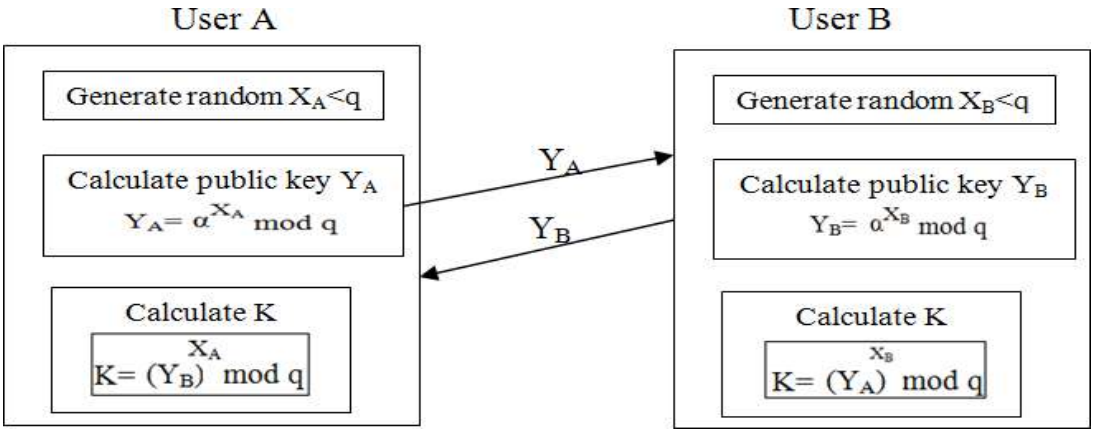
After they exchange public keys, each can compute the common secret key:

A computes $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$.

B computes $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$.

We assume an attacker would have available the following information:

$$q = 353; \alpha = 3; Y_A = 40; Y_B = 248$$



Example2:

Here is an example, taken from. Key exchange is based on the use of the prime number q=71 and a primitive root of 71, in this case $\alpha = 7$. A and B select private keys $X_A = 5$ and $X_B = 12$, respectively. Each computes its public key:

$$Y_A = 7^5 = 51 \bmod 71$$
$$Y_B = 7^{12} = 4 \bmod 71$$

After they exchange public keys, each can compute the common secret key:

$$K = (Y_B)^{X_A} \bmod 71 = 45 = 30 \bmod 71$$
$$K = (Y_A)^{X_B} \bmod 71 = 5112 = 30 \bmod 71$$

From {51, 4}, an attacker cannot easily compute 30.

Example 3:

User A and B exchange the key using Diffie-Hellman algorithm. Assume α=5 q=11 X_A=2 X_B=3. Find the value of Y_A, Y_B and k?

Soln:

$$Y_A = \alpha^{X_A} \bmod q$$
$$= 25 \bmod 11$$
$$= 3$$
$$Y_B = \alpha^{X_B} \bmod q$$
$$= 125 \bmod 11$$
$$= 4$$
$$K = (Y_A)^{X_B} \bmod q$$
$$= 27 \bmod 11$$
$$= 5$$
$$K = (Y_B)^{X_A} \bmod q$$
$$= 16 \bmod 11$$
$$= 5$$

MAN-IN-MIDDLE-ATTACK:

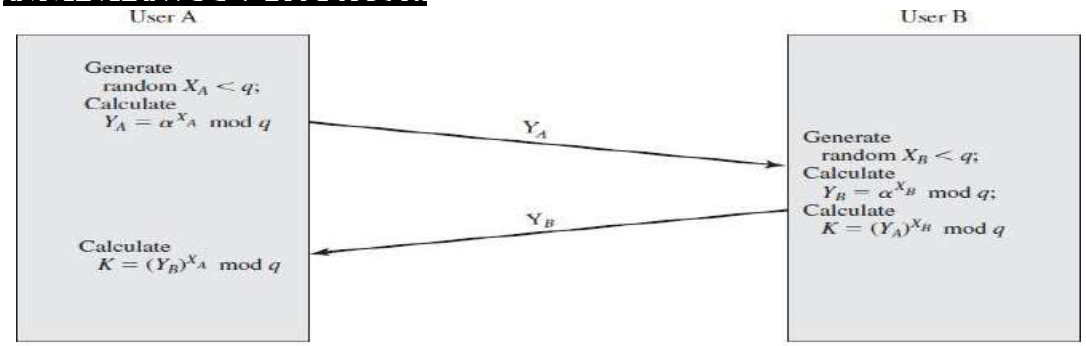


Figure 10.2 Diffie-Hellman Key Exchange

Definition: A man in the middle attack is a form of eavesdropping where communication between two users is monitored and modified by an unauthorized party.

Generally the attacker actively eavesdrops by intercepting (stoping) a public key message exchange.

The Diffie- Hellman key exchange is insecure against a "Man in the middle attack".

Suppose user 'A' & 'B' wish to exchange keys, and D is the adversary (opponent). The attack proceeds as follows.

1. 'D' prepares for the attack by generating two random private keys X_{D1} & X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. 'A' transmits 'Y_A' to 'B'
3. 'D' intercepts Y_A and transmits Y_{D1} to 'B'. and D also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.

4. 'B' receives Y_{D1} & calculate $K1 = (Y_{D1})^{X_B} \bmod q$.
5. 'B' transmits 'YB' to 'A"
6. 'D' intercepts 'YB' and transmits YD2 to 'A' and 'D' calculate K1 $K1 = (Y_B)^{X_{D1}} \bmod q$.
7. A receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key *K*1 and Alice and Darth share secret key *K*2. All future communication between Bob and Alice is compromised in the following way.

1. A sends an encrypted message *M*: E(*K*2, *M*) .
2. D intercepts the encrypted message and decrypts it to recover *M*.
3. D sends B $E(K1, M)$ or $E(K1, M')$, where M' is any message. In the first case, D simply wants to eavesdrop on the communication without altering it. In the second case, D wants to modify the message going to B

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.

ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Arithmetic:

A major issue with the use of Public-Key Cryptography, is the size of numbers used, and hence keys being stored. Recently, an alternate approach has emerged, elliptic curve cryptography (ECC), which performs the computations using elliptic curve arithmetic instead of integer or polynomial arithmetic.

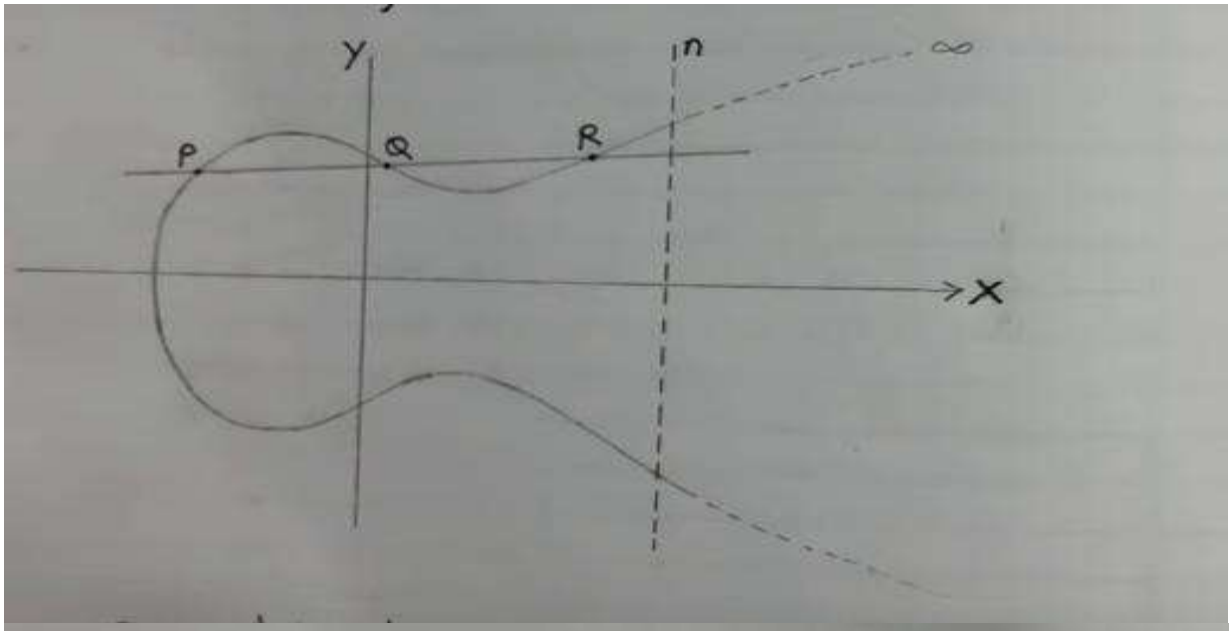
Majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials. It imposes a significant load in storing and processing keys and messages. An alternatives to use elliptic curves; it offers same security with smaller bit sizes.

Real Elliptic Curves

Elliptic Curve Cryptography (ECC)

- It is an asymmetric / public key cryptosystem
- It provides equal security with smaller key size as compared to non ECC algorithms
- It makes use of elliptic curves
- Elliptic curves are defined by some mathematical functions

Ex - $y^2 = x^3 + ax + b$



- symmetric to x-axis
- If we draw a line, it will touch a maximum of 3 points.

ECC Algorithm

ECC Key Exchange

Global Public Elements -

1) $E_q(a, b)$ - Elliptic curve with parameters a, b of q (prime number or an integer of the form 2^m)
2) G - Point on the elliptic curve

User A key generation

Select private key n_A $\quad n_A < n$

Calculate public key P_A $\quad P_A = n_A \times G$

User B key generation

Select private key n_B $\quad n_B < n$

Calculate public key P_B $\quad P_B = n_B \times G$

Calculation of secret key by user A

$$K = n_A \times P_B$$

Calculation of secret key by user B

$$K = n_B \times P_A$$

ECC Encryption

- Let the message be M
- First encode this message M into a point on elliptic curve
  Let this point be P_m

- For encryption, choose a random positive integer k
  The cipher point will be

$$C_m = \{kG, P_m + kP_B\}$$

This point will be sent to the receiver

Decryption

- For decryption, multiply x-coordinate with receiver's secret key

$$kG \times n_B$$

Then subtract $(kG \times n_B)$ from y-coordinate of cipher point

$$P_m + kP_B - (kG \times n_B)$$

we know that $P_B = n_B \times G$

$$\therefore P_m + kP_B - kP_B$$

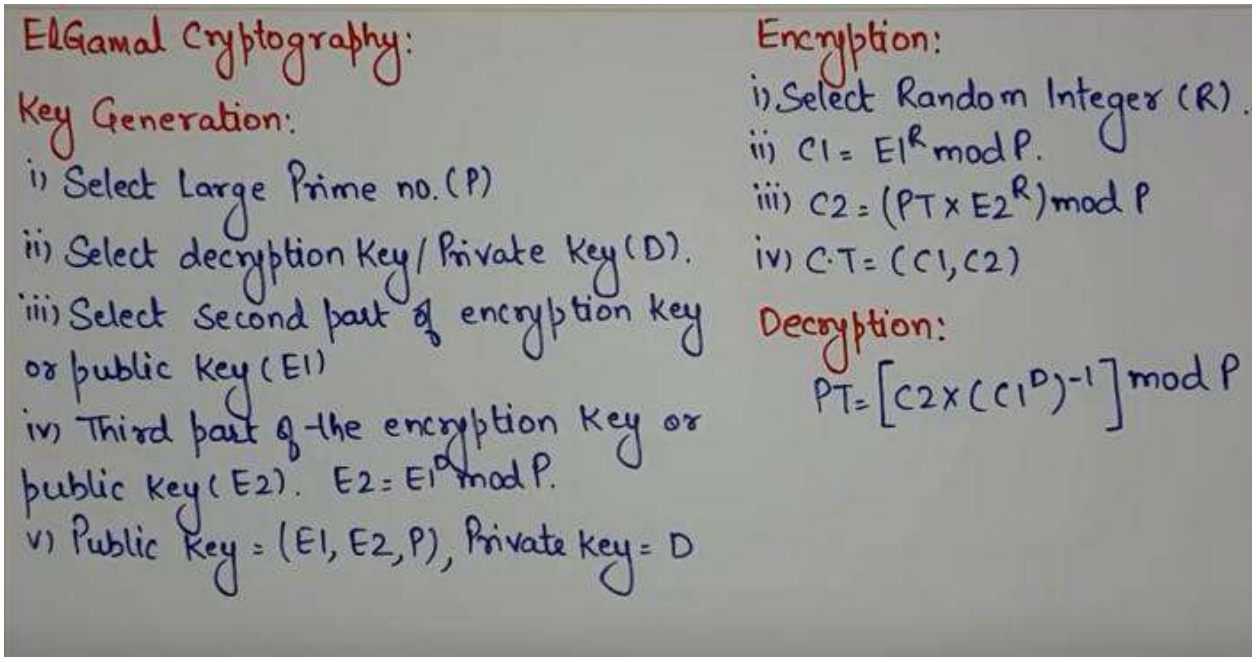
$$= P_m$$

So receiver gets the same point

Comparision of RSA/DSA (Diffie Hellman Algorithm

Key sizes with equivalent security levels

Minimum size (bits) of public keys		
DSA/DH	RSA	ECC
1024	1024	160
2048	2048	224
3072	3072	256
7680	7680	384
15360	15360	512



Elgamal Cryptography: Asymmetruc Key (Encrp-pub key ,Decrypt-Private key)

`Let P=11;D=3;E1=2

$E2=E1^D \bmod p$

$E2= 2^3 \bmod 11$

E2= 8 mod 11

E2=8

Now public key={E1,E2,P}={2,8,11}

Private Key=3

Encryption:

R=4

$C1=E1^R \bmod p$

$C1=2^4 \bmod 11$

C1=16 mod 11=5

Now Pt=7

C2={Pt $E2^R$ mod p}

C2= $7x8^4$ mod 11

C2=28672 mod 11 =6

C2=6

C.T ={5,6}

Decryption

Pt={C2 $(C1^D)^{-1}$ mod p}

Pt={6x$(5^3)^{-1}$)mod 11

First: $(5^3)^{-1}$)mod 11

$(125)^{-1}$)mod 11

125 *X mod 11=1

125*1 mod 11= 121 +4 mod 11=4

125*2 mod 11= 250 mod 11= 242+8 mod11=8

125*3 mod 11=375 mod 11= 374+1 mod 11=1

Pt={6*3 mod 11}

Pt=18 mod 11=7

Pt=7

So the Decrypt and encrypt Ptvalue is same .

UNIT IV MESSAGE AUTHENTICATION AND INTEGRITY

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function and MAC – SHA –Digital signature and authentication protocols – DSS
Entity Authentication: Biometrics, Passwords, Challenge Response protocols- Authentication applications – Kerberos, X.509

MESSAGE AUTHENTICATION

- is a mechanism or service used to **verify the integrity of a message**.
- Message authentication is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)

AUTHENTICATION REQUIREMENT

1. **Disclosure:** Release of message contents to any person or process not possessing the appropriate cryptographic key.
2. **Traffic analysis**: Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection- oriented or connectionless environment, the number and length of messages between parties could be determined.
3. **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity.
4. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.
5. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
6. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.
7. **Source repudiation:** Denial of transmission of message by source.
8. **Destination repudiation:** Denial of receipt of message by destination.

Summary:

Message authentication

✓ A procedure to verify that messages come from the alleged (suspected) source and have not been altered

✓ Message authentication may also verify sequencing and timeliness

Digital signature

✓ *An authentication technique* that also includes measures to counter repudiation by either source or destination

I. AUTHENTICATION FUNCTION

Three classes of functions that may be used to produce an authenticator

1)Message encryption – The **Cipher text** of the entire message serves as its authenticator.

2)Message authentication code(MAC) - A public function of the **message and a secret key** that produces a **fixed-length value** that serves as the authenticator.

3) Hashfunction -A public function that maps a **message** of any length into a **fixed-length hash value**, which serves as the authenticator.

1. **MessageEncryption**

Message encryption by itself can provide a measure of authentication. The analysis differs for symmetric and public-key encryption schemes.

Symmetric Encryption

- Conventional encryption can serve as authenticator
- Conventional encryption provides *authentication* as well as *confidentiality*
- Message *M* transmitted from source A to destination B is encrypted using a secret key *K* shared by A and B. If no other party knows the key, then confidentiality is provided:
- No other party can recover the plaintext of the message.

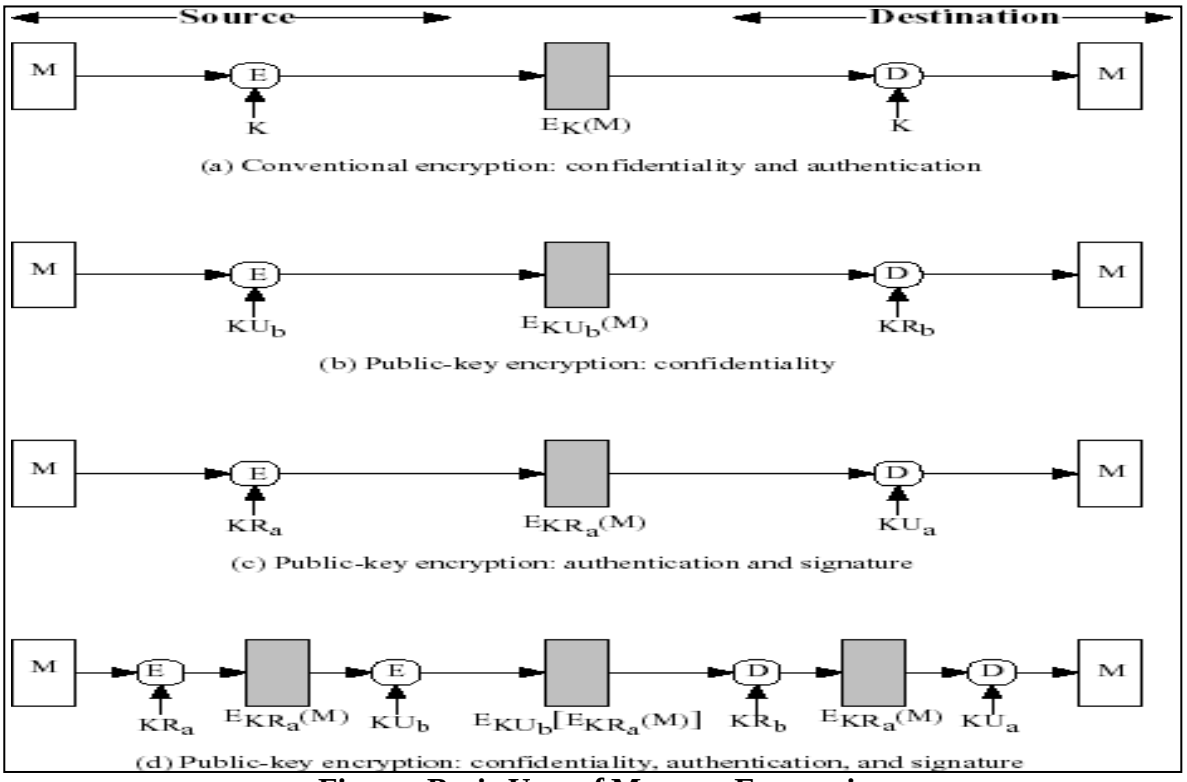


Figure: Basic Uses of Message Encryption

- Given a decryption function D and a secret key *K*, the destination will accept *any* input *X* and produce output $Y = D(K,X)$.
- If *X* is the cipher text of a legitimate message *M* produced by the corresponding encryption function, then *Y* is some plaintext message *M*. Otherwise, *Y* will likely be a meaningless sequence of bits.
- There may need to be some automated means of determining at B whether *Y* is legitimate plaintext and therefore must have come from A.

Public-Key Encryption

- The straightforward use of public-key encryption provides confidentiality but not authentication. The source (A) uses the public key P_{Ub} of the destination (B) to encrypt M. Because only B has the corresponding private key P_{Rb}, only B can decrypt the message.
- To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt.
- To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature, and then using B's public key, which provides confidentiality.
- The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

(a) Conventional (symmetric) Encryption
$A \rightarrow B: E_K[M]$ •Provides confidentiality —Only A and B share K •Provides a degree of authentication —Could come only from A —Has not been altered in transit —Requires some formatting/redundancy •Does not provide signature —Receiver could forge message —Sender could deny message
(b) Public-Key (asymmetric) Encryption
$A \rightarrow B: E_{KU_b}[M]$ •Provides confidentiality —Only B has KR_b to decrypt •Provides no authentication —Any party could use KU_b to encrypt message and claim to be A
$A \rightarrow B: E_{KR_a}[M]$ •Provides authentication and signature —Only A has KR_a to encrypt —Has not been altered in transit —Requires some formatting/redundancy —Any party can use KU_a to verify signature
$A \rightarrow B: E_{KUb}[E_{KRa}(M)]$ •Provides confidentiality because of KU_b •Provides authentication and signature because of KR_a

Table: Confidentiality and Authentication Implications of Message Encryption

2. **<u>Message Authentication Code</u>**
   - Uses a shared secret key to generate a **fixed-size block of data (known as a cryptographic checksum or MAC)** that is appended to the message.

     $$\mathbf{MAC = C_K(M)}$$

     Where *M* is a variable-length Input message, *K* is a Shared secret key, C is MAC function and $C_K(M)$ is the fixed-length authenticator.
   - The MAC is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by recomputing the MAC.
   - A MAC function is similar to encryption. One difference is that the **MAC algorithm need not be reversible**, as it must for decryption. In general, the MAC function is a **many-to-one function**.

Assurances

- Message has not been altered
- Message is from alleged sender
- Message sequence is unaltered (requires internal sequencing)

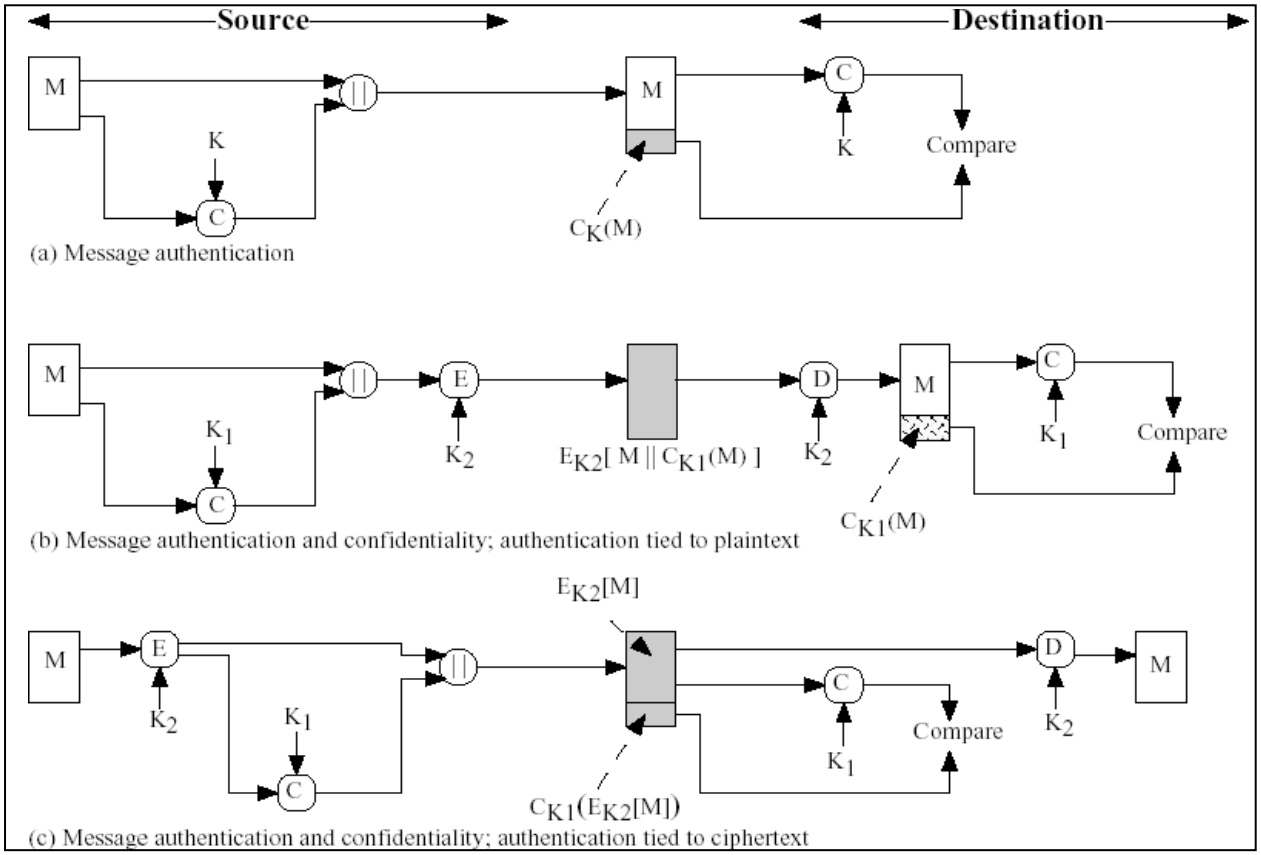


Figure: Basic Uses of MAC

- The process depicted in (Figure a) provides authentication but not confidentiality, because the message as a whole is transmitted in the clear.
- Confidentiality can be provided by performing message encryption either after (Figure b) or before Figure c) the MAC algorithm.
- In both these cases, two separate keys are needed, each of which is shared by the sender and the receiver.
- In the first case, the MAC is calculated with the message as input and is then concatenated to the message. The entire block is then encrypted.
- In the second case, the message is encrypted first.Then the MAC is calculated using the resulting Cipher text and is concatenated to the cipher text to form the transmitted block.
- Note that the MAC does not provide a digital signature because both sender and receiver share the same key.

Table: Basic Uses of Message Authentication Code

(a) $A \rightarrow B: M \,\|\, C_K(M)$ •Provides authentication —Only A and B share K
(b) $A \rightarrow B: E_{K_2}[M \,\|\, C_{K_1}(M)]$ •Provides authentication —Only A and B share K_1 •Provides confidentiality —Only A and B share K_2
(c) $A \rightarrow B: E_{K_2}[M] \,\|\, C_{K_1}(E_{K_2}[M])$ •Provides authentication —Using K_1 •Provides confidentiality —Using K_2

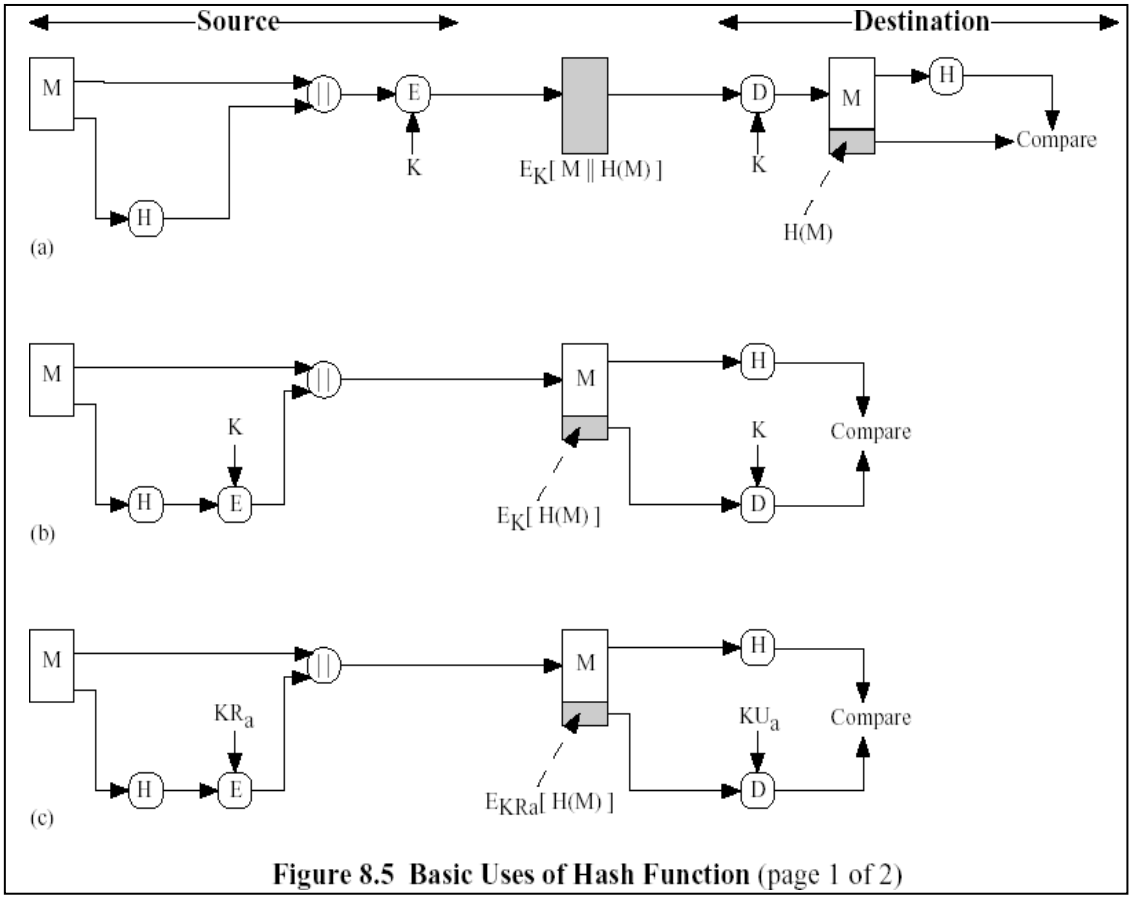


Figure 8.5 Basic Uses of Hash Function (page 1 of 2)

Advantage

- It is cheaper and more reliable to have only one destination responsible for monitoring authenticity.
- Authentication is carried out on a selective basis, messages being chosen at random for checking
- Authentication of a computer program in plaintext is an attractive service.
- Separation of authentication and confidentiality functions affords architectural flexibility
- Separation of authentication check from message use.

3. <u>Hash Function</u>

- **A variation on the message authentication code is the one-way hashfunction**
- **Converts a variable size message M into fixed size hash code H(M) (Sometimes called a message digest)**
- Hash code does not use a key but is a function only of the input message.
- **The hash code (h) is also referred to as a message digest or hash value.**
- The hash code is a function of all the bits of the message and provides an error-detection capability. A change to any bit or bits in the message results in a change to the hash code.

Hash code can be used to provide message authentication, as follows

1. The message plus concatenated hash code is encrypted using symmetric encryption. **E(M ||H)**
2. Only the hash code is encrypted, using symmetric encryption. **M ||E(H)**
3. Only the hash code is encrypted, using public-key encryption and using the sender's private key. M || signed H
4. If confidentiality as well as a digital signature is desired, then the message plus the private-key-encrypted hash code can be encrypted using a symmetric secret key. **E(M || signed H)** gives confidentiality
5. It is possible to use a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value *S*. **M || H(M || S)**
6. Confidentiality can be added to the approach of (e) by encrypting the entire message plus the hash code. E(M || H(M || S))

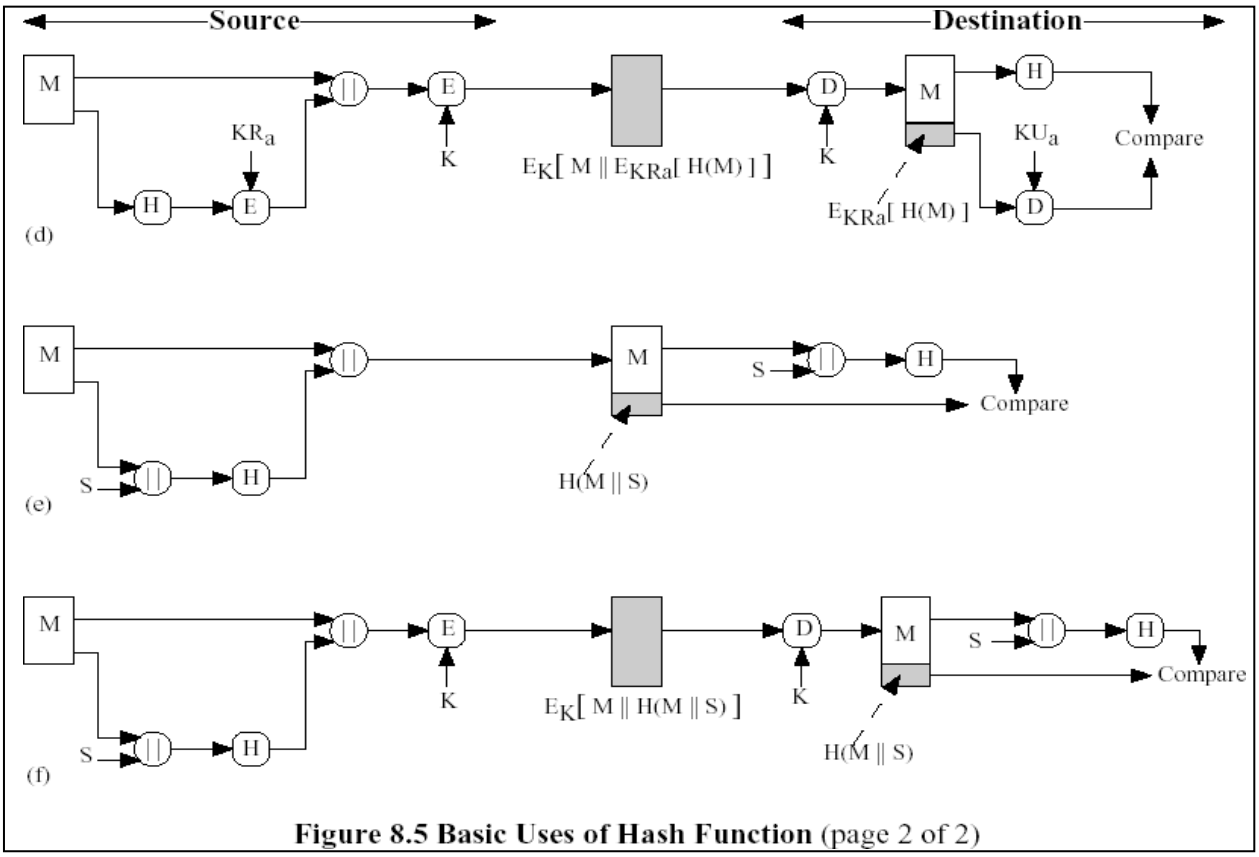


Figure 8.5 Basic Uses of Hash Function (page 2 of 2)

When confidentiality is not required, methods (b) and (c) have an advantage over those that encrypt the entire message in that less computation is required.

Table: Basic Uses of Hash Function H

(a) $A \rightarrow B: E_K[M \| H(M)]$ •Provides confidentiality —Only A and B share K •Provides authentication —H(M) is cryptographically protected	**(d)** $A \rightarrow B: E_K[M \| E_{KRa}[H(M)]]$ •Provides authentication and digital signature •Provides confidentiality —Only A and B share K
(b) $A \rightarrow B: M \| E_K[H(M)]$ •Provides authentication —H(M) is cryptographically protected	**(e)** $A \rightarrow B: M \| H(M \| S)$ •Provides authentication —Only A and B share S
(c) $A \rightarrow B: M \| E_{KRa}[H(M)]$ •Provides authentication and digital signature —H(M) is cryptographically protected —Only A could create $E_{KRa}[H(M)]$	**(f)** $A \rightarrow B: E_K[M \| H(M) \| S]$ •Provides authentication —Only A and B share S •Provides confidentiality —Only A and B share K

Reasons for using Hash

- Encryption software is relatively slow.
- Encryption hardware costs are not negligible.
- Encryption hardware is optimized toward large datasizes.
- Encryption algorithms may be covered by patents

MESSAGE AUTHENTICATION CODES

- A MAC, also known as a cryptographic checksum, is generated by a function C of the form
  T= MAC(K,M)
  Where *M* is a variable-length message, *K* is a secret key shared only by sender and receiver, and MAC(K,*M*) is the fixed-length authenticator, sometimes called a tag.
- The tag is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by recomputing the MAC.

Requirements for MACs:

- Security depends on the **bit length of the key**
- The opponent must resort to a **brute-force attack** using all possible keys.
- On average, such an attack will require $\mathbf{2^{(k-1)}}$ **attempts for a k-bit key**.
- MAC function is a many-to-one function, due to the many-to-one nature of the function.
- Using brute- force methods, how would an opponent attempt to discover a key?
  A number of keys will produce the correct MAC and the opponent has no way of knowing which the correct key is. On average, a total of $2^k/2^n$ = $\mathbf{2^{(k-n)}}$ **keys** will produce a match. Thus, the opponent must iterate the attack:
  - **Round 1**
    - Given: M_1, T_1 = MAC(K, M_1)
    - Compute T_i = MAC(K_i, M_1) for all 2^k keys Number of matches $\approx 2^{(k-n)}$
  - **Round 2**
    - Given: M_2, T_2 = MAC(K, M_2)
    - Compute T_i = MAC(K_i, M_2) for the $2^{(k-n)}$ keys resulting from Round1 Number of matches $\approx 2^{(k-2xn)}$

- If an 80-bit key is used and the MAC is 32 bits long, then the **first round** will produce about 2^{48} possible keys.
- The **second round** will narrow the possible keys to about 2^{16}possibilities.
- The **third round** should produce only a single key, which must be the one used by the sender.
- Brute-force attempt to discover the authentication key is no less effort and may be more effort than that required to discover a decryption key of the same length.
- Other attacks that do not require the discovery of the key are possible.

The MAC function should satisfy the following requirements:

1. If an opponent observes M and MAC(K, M), it should be computationally infeasible for the opponent to construct a message M' such that MAC(K, M') = MAC(K,M).
2. MAC(K, M) should be uniformly distributed in the sense that for randomly chosen messages, M and M', the probability that MAC(K, M) = MAC(K, M') is 2^{-n}, where n is the number of bits in the tag.
3. Let M' be equal to some known transformation on M. That is, M' = f(M). For example, f may involve inverting one or more specific bits. In that case,
   $$Pr[MAC(K, M) = MAC(K, M')] = 2^{-n}.$$

- The first requirement speaks about, an opponent is able to construct a new message to match a given tag, even though the opponent does not know and does not learn the key.
- The second requirement deals with the need to prevent a brute-force attack based on chosen plaintext. That is, if we assume that the opponent does not know K but does have access to the MAC function and can present messages for MAC generation, then the opponent could try various messages until finding one that matches a given tag.

HASH FUNCTIONS

A hash value h is generated by a function H of the form

h = H(M)

- M is a variable-length message, h is a fixed-length hash value, H is a hash function
- The hash value is appended at the source
- The receiver authenticates the message by recomputing the hash value
- Because the hash function itself is not considered to be secret, some means is required to protect the hash value

Requirements for a Hash Function

1. H can be applied to any size datablock
2. H produces fixed-length output
3. H(x) is relatively easy to compute for any given x
4. H is *one-way*, i.e., given h, it is computationally infeasible to find an y s.t. H(y) = h
5. H is *weakly collision resistant*: given x, it is computationally infeasible to find any y! ≠ x s.t. H(y) = H(x)
6. H is *strongly collision resistant*: it is computationally infeasible to find any pair (x,y) s.t. H(x) =H(y)
   - One-way property is essential for authentication
   - Weak collision resistance is necessary to prevent forgery
   - Strong collision resistance is important for resistance to birthday attack

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

Simple Hash Functions

- **Operation of hash functions:**
  - The input is viewed as a sequence of n-bit blocks
  - The input is processed one block at a time in an iterative fashion to produce an n-bit hash function

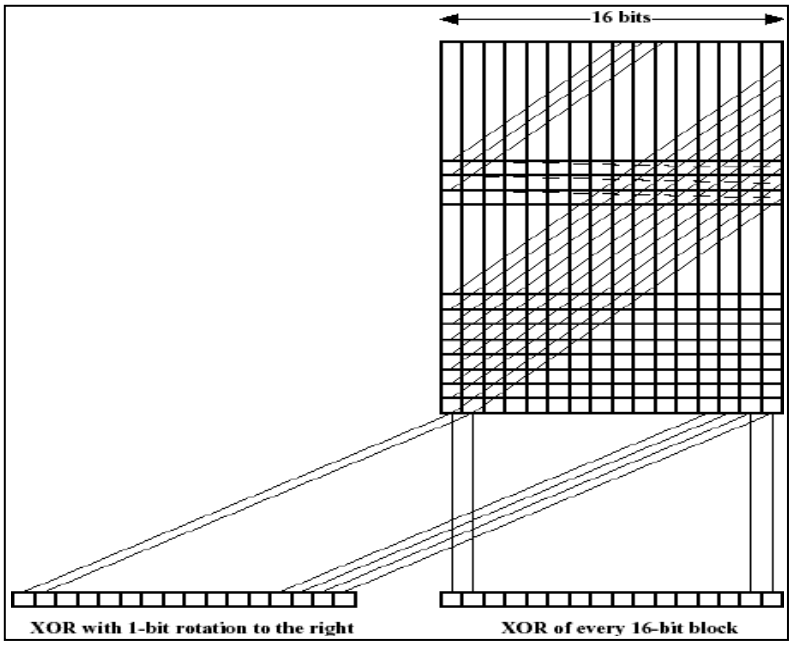
- **Simplest hash function:**
  1) Bitwise XOR of every block
     - $C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$

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

       m = number of n-bit blocks in the input

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

       Known as **longitudinal redundancy check**



Improvement over the simple bitwise XOR

- Initially set the n-bit hash value to zero
- Process each successive n-bit block of data as follows
  - Rotate the current hash value to the left by one bit
  - XOR the block into the hash value

This has the effect of "randomizing" the input more completely and overcoming any regularities that appear in the input. The above Figure illustrates these two types of hash functions for 16-bit hash values.

We can define the scheme as follows:
Given a message consisting of a sequence of 64-bit blocks X_1, X_2,..., X_N, define the hash code C as the block-by-block XOR of all blocks and append the hash code as the final block:

$$C = X_{N+1} = X_1 \oplus X_2 \oplus \ldots \oplus X_N$$

Next, encrypt the entire message plus hash code, using CBC mode to produce the encrypted message Y_1, Y_2,..., Y_{N+1}.
Cipher text of this message can be manipulated in such a way that it is not detectable by the hash code.
For example,

$X_1 = IV \oplus D(K, Y_1)$

$X_i = Y_{i1} \oplus D(K, Y_i)$

$X_{N+1} = Y_N \oplus D(K, Y_{N+1})$

But X_{N+1} is the hash code:

$X_{N+1} = X_1 \oplus X_2 \oplus \ldots \oplus X_N$

$= [IV \oplus D(K, Y_1)] \oplus [Y_1 \oplus D(K, Y_2)] \oplus \ldots \oplus [Y_{N1} \oplus \ldots \oplus D(K, Y_N)]$

Because the terms in the preceding equation can be XORed in any order, it follows that the hash code would not change if the cipher text blocks were permuted.

Birthday Attacks

1. The source, A, is prepared to "sign" a message by appending the appropriate m-bit hash code and encrypting that hash code with A's private key.
2. The opponent generates $2^{m/2}$ variations on the message, all of which convey essentially the same meaning. The opponent prepares an equal number of messages, all of which are variations on the fraudulent message to be substituted for the real one.
3. The two sets of messages are compared to find a pair of messages that produces the same hash code. The probability of success, by the birthday paradox, is greater than 0.5. If no match is found, additional valid and fraudulent messages are generated until a match is made.
4. The opponent offers the valid variation to A for signature. This signature can then be attached to the fraudulent variation for transmission to the intended recipient. **Because the two variations have the same hash code, they will produce the same signature;** the opponent is assured of success even though the encryption key is not known.

Thus, if a 64-bit hash code is used, the level of effort required is only on the order of 2^3. The generation of many variations that convey the same meaning is not difficult. For example, the opponent could insert a number of "space-space-backspace" character pairs between words throughout the document. Variations could then be generated by substituting "space-backspace-space" in selected instances. Alternatively, the opponent could simply reword the message but retain the meaning.

SECURITY OF HASH FUNCTIONS AND MACS

We can group attacks on hash functions and MACs into two categories: **brute-force attacks and cryptanalysis**.

Brute-Force Attacks

The nature of brute-force attacks differs somewhat for hash functions and MACs.

1) Hash Functions

- H is *one-way*, i.e., given h, it is computationally infeasible to find an y s.t. H(y) = h
- H is *weakly collision resistant*: given x, it is computationally infeasible to find any y! ≠ x s.t. H(y) = H(x)
- H is *strongly collision resistant*: it is computationally infeasible to find any pair (x,y) s.t. H(x) =H(y)

H(x) For a hash code of length n, the level of effort required, as we have seen is proportional to the following:

One way	2^n
Weak collision resistance	2^n
Strong collision resistance	$2^{n/2}$

- One-way and weak collision require 2^neffort
- Strong collision requires $2^{n/2}$effort
- If strong collision resistance is required (and this is desirable for a general-purpose secure hash code), $2^{n/2}$ determines the strength of hash code against brute-force attack
- Currently, two most popular hash codes, SHA-1 and RIPEMD-160, provide a 160-bit hash code length

2) Message Authentication Codes

A brute-force attack on a MAC is a more difficult undertaking because it requires known message-MAC pairs.

Given a fixed message x with n-bit hash code h = H(x), a brute-force method of finding a collision is to pick a random bit string y and check if H(y) = H(x). The attacker can do this repeatedly off line.

Whether an off-line attack can be used on a MAC algorithm depends on the relative size of the key and the MAC.

To proceed, we need to state the desired security property of a MAC algorithm, which can be expressed as follows:

Computation resistance: Given one or more text-MAC pairs $[x_i, C(K, x_i)]$, it is computationally infeasible to compute any text-MAC pair $[x, C(K, x)]$ for any new input $x \neq x_i$.

In other words, the attacker would like to come up with the valid MAC code for a given message x. There are two lines of attack possible: Attack the key space and attack the MAC value.

Cryptanalysis

Cryptanalytic attacks on hash functions and MAC algorithms seek to exploit some property of the algorithm to perform some attack other than an exhaustive search. That is, an ideal hash or MAC algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort.

Hash Functions

The hash algorithm involves repeated use of a **compression function**, f, that takes two inputs and produces an n-bit output. At the start of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. Often, b > n; hence the term compression.

The hash function can be summarized as follows:

CV_o = IV = initial n-bit value

CV_i = $f(CV_{i1}, Y_{i1})$ $1 \leq i \leq L$

$H(M)$ = CV_L

where the input to the hash function is a message M consisting of the blocks $Y_o, Y_1, ..., Y_{L1}$.

SECURE HASH ALGORITHM (SHA)

- SHA originally designed by NIST & NSA in 1993
- was revised in 1995 as SHA-1
- based on design of MD4 with key differences
- produces 160-bit hash values

REVISED SECURE HASH FUNCTION:

- adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1

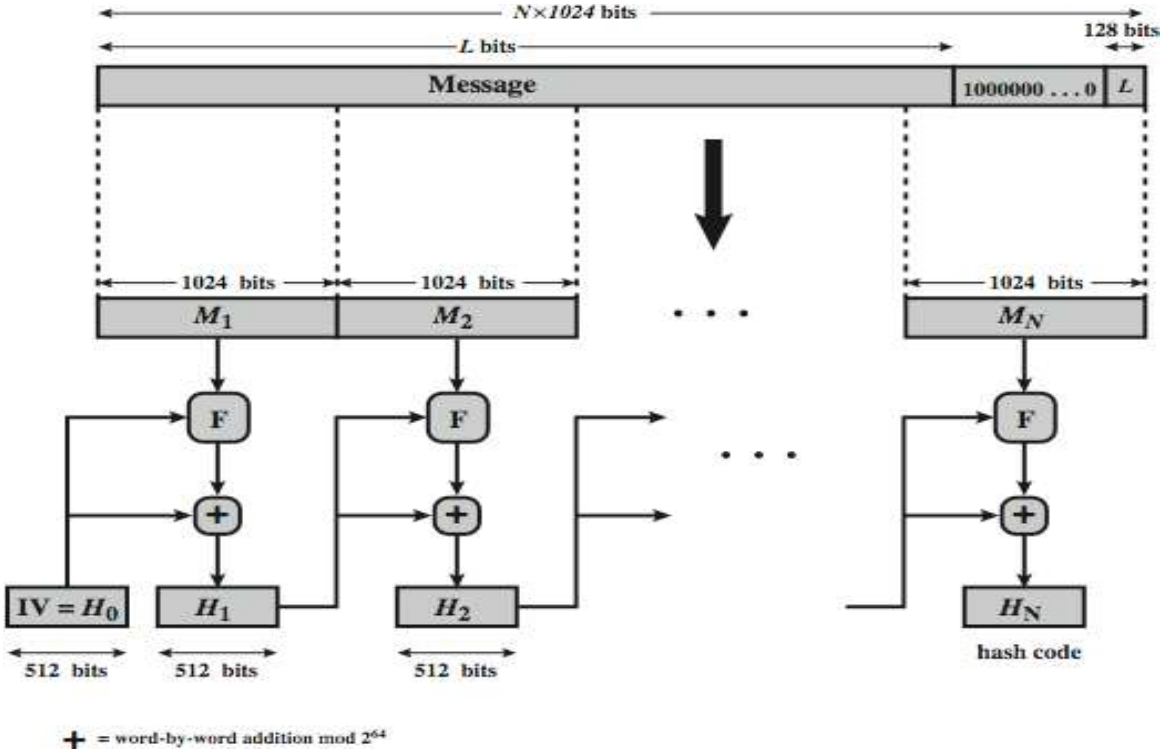
PARAMETERS FOR VARIOUS VERSION OF SHA:

Parameter	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size(in bits)	160	256	384	512
Message size(in bits)	$<2^{64}$	2^{64}	2^{128}	2^{128}
Block size (in bits)	512	512	1024	1024
Word size (in bits)	32	32	64	64
Steps in algorithm	80	64	80	80

SHA-512 logic:

- The input is processed 1024 bits block.
- The algorithm takes as input a message with a maximum length of less than 2^128 bits.
- Produce output is 512 bits message digest.

Algorithm processing Steps: The processing consists of the following steps:
Step 1: Append padding bits
Step 2: Append length
Step 3: Initialize hash buffer
Step 4: Process the message in 1024-bit (128-word) blocks, which forms the heart of the algorithm
Step 5: Output the final state value as the resulting hash



***Step-1: Appending Padding Bits*.** The original message is "padded" (extended) so that its length (in bits) consists of a single 1-bit followed by the necessary number of 0-bits, so that its length is congruent to 896 modulo 1024 (128 bits short of a multiple of 1024)

***Step-2: Append length*:** a block of 64 bits is appended to the message. This block is treated as unsigned 64 bit integers (most significant byte first) and contains the length of the original message.

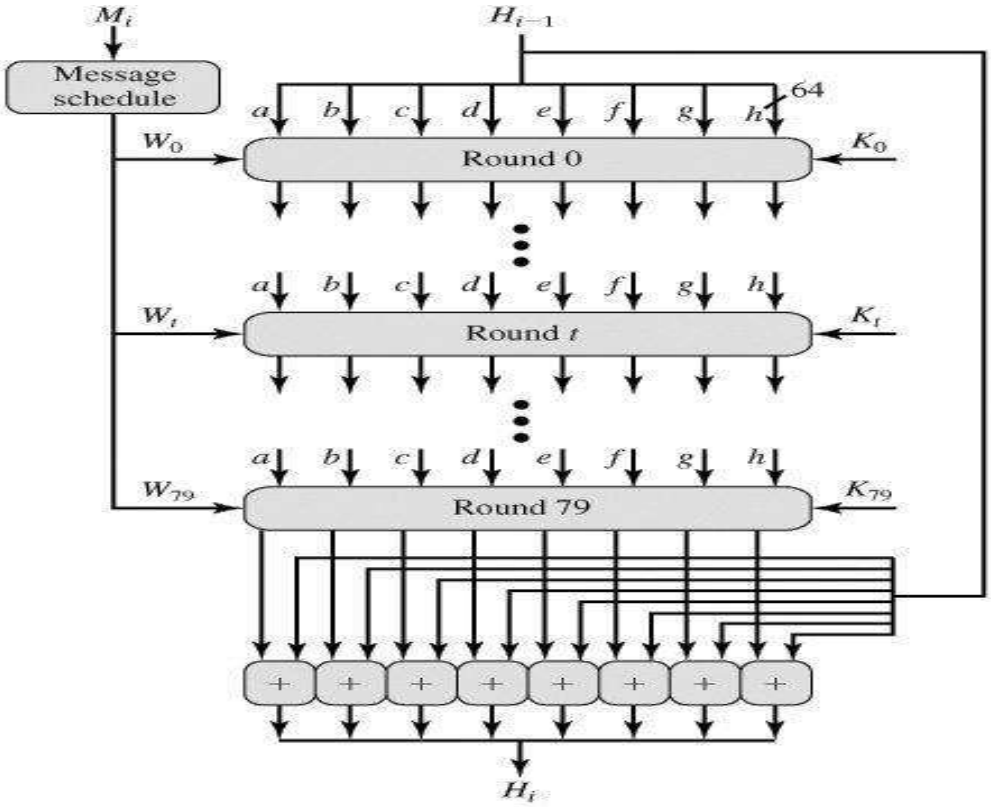
Step-3: Initialize hash buffer: 160 bit buffer is used to hold intermediate and final results of the hash function. This buffer can be represented as eight 64 bit registers (a, b, c, d, e, f, g, h). The registers are initialized to the following 64 bit integers

Word a:6A09E66713BCC908 Word e:BB67AE8584CAA73B
Word b:3C6EF372FE94F82B Word f:A54FF53A5F1D36F1
Word c:510E527FADE682D1 Word g:9B05688C2B3E6C1F
Word d:1F83D9ABFB41BD6B Word h:5BE0CD19137E2179

which are the beginnings, in hexadecimal, of the fractional parts of the square roots of 2, 3, 5, 7, 11, 13, 17, and 19.

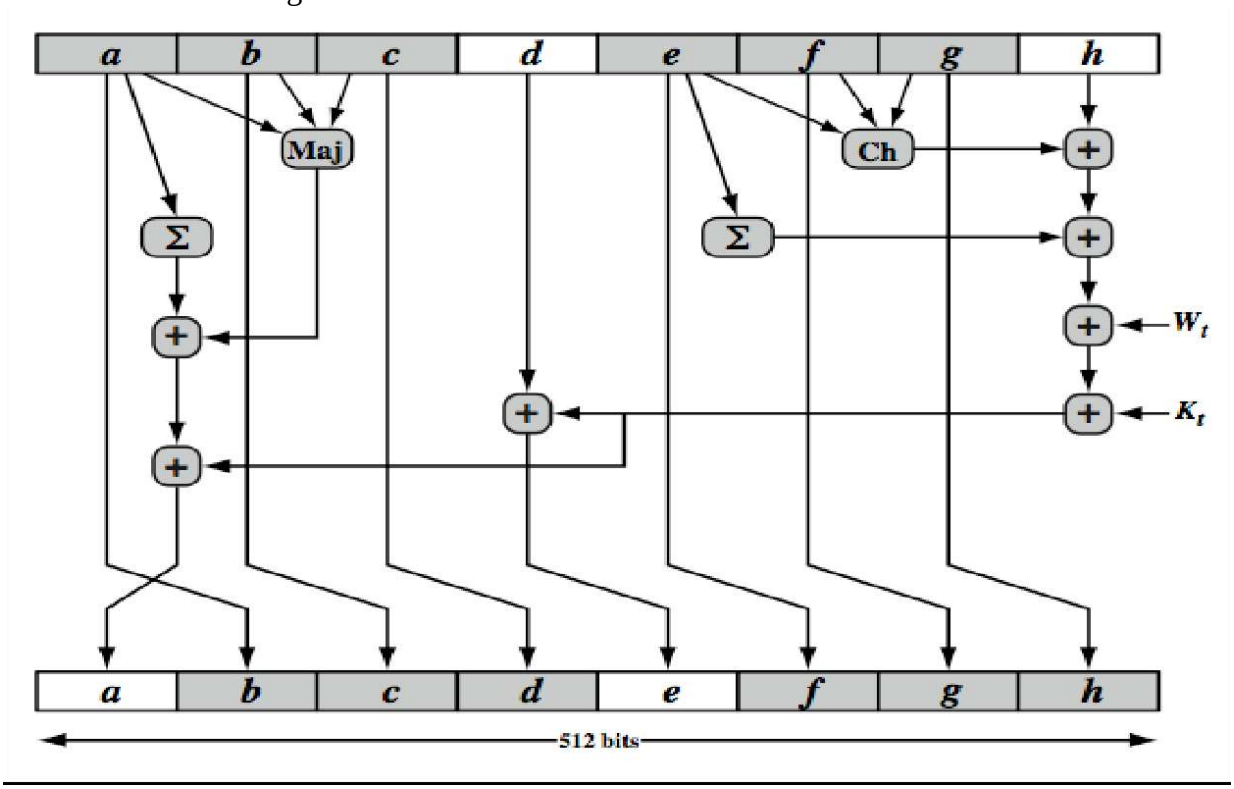
Step 4: Process Message in 512 bits:

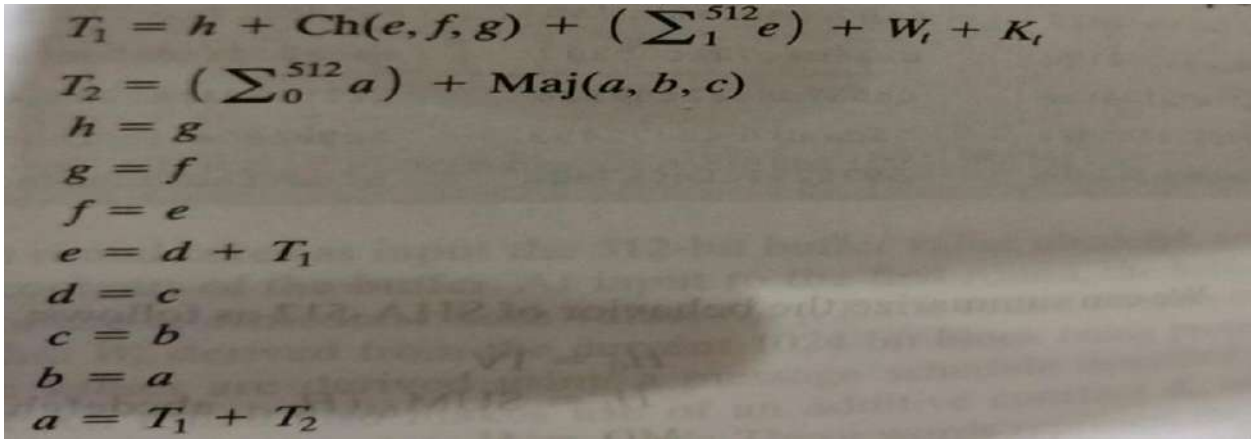
This algorithm consist 4 rounds of 20 steps each. The SHA-512 Compression Function is the heart of the algorithm. In this Step 4, it processes the message in 1024-bit (128-word) blocks, using a module that consists of 80 rounds, labeled F in above Figure. Each round takes as input the 512-bit buffer value, and updates the contents of the buffer. Each round t makes use of a 64-bit value Wt derived using a message schedule from the current 1024-bit block being processed. Each round also makes use of an additive constant Kt, based on the fractional parts of the cube roots of the first eighty prime numbers. The output of the eightieth round is added to the input to the first round to produce the final hash value for this message block, which forms the input to the next iteration of this compression function,



SHA-512 Elementary SHA operation for single round (or) SHA-1 Compression Function:

The logic in each of the 80 steps of the processing of one 512 bit block and the structure of each of the 80 rounds is shown Figure.





$$T_1 = h + \mathrm{Ch}(e,f,g) + \left(\sum_1^{512} e\right) + W_t + K_t$$
$$T_2 = \left(\sum_0^{512} a\right) + \mathrm{Maj}(a,b,c)$$
$$h = g$$
$$g = f$$
$$f = e$$
$$e = d + T_1$$
$$d = c$$
$$c = b$$
$$b = a$$
$$a = T_1 + T_2$$

Each 64-bit word shuffled along one place, and in some cases manipulated using a series of simple logical functions (ANDs, NOTs, ORs, XORs, ROTates), in order to provide the avalanche & completeness properties of the hash function.

The elements are:

Where the elements are:

t=step number; 0<=t<=79

Ch(e,f,g) = (e AND f) XOR (NOT e AND g)

The conditional function: IF e then f else g

Maj(a,b,c) = (a AND b) XOR (a AND c) XOR (b AND c)

The function is true only of the majority (Two or three) of the arguments are true.

∑(a) = ROTR(a,28) XOR ROTR(a,34) XOR ROTR(a,39)

∑(e) = ROTR(e,14) XOR ROTR(e,18) XOR ROTR(e,41)

ROTR(a,n) = Circular right shift(rotation) of the 64 bit argument x by n bits

+ = addition modulo 2^64

Kt = a 64-bit additive constant

Wt = a 64-bit word derived from the current 512-bit input block.

Step-5: Output: After all N 1024 -bit blocks have been processed, the output from the Nth stage is the 512-bit message digest.

The behavior of SHA-512 can be summarized as:

$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, a\ b\ c\ d\ e\ f\ g\ h_i)$

$MD = H_N$

IV→ initialize value of the buffers a b c d e f g h, defined in step3.

a b c d e f g h_i → the output of the last round of processing of the i^{th} message block

N→ the number of blocks in the message

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

MD→ final message digest value

Comparison of MD5 and SHA:

	MD5	**SHA-1**
Message Digest Length	128 bits	160 bits
Basic unit of Processing	512 bits	512 bits
Number of Steps	64 (4 rounds of 16)	80(4 rounds of 20)

Maximum Message Size	∞	264-1 bits
Primitive logical functions	4	4
Additive constants used	64	4
Endian format	Little endian	Big endian

DIGITAL SIGNATURE AND AUTHENTICATION PROTOCOLS

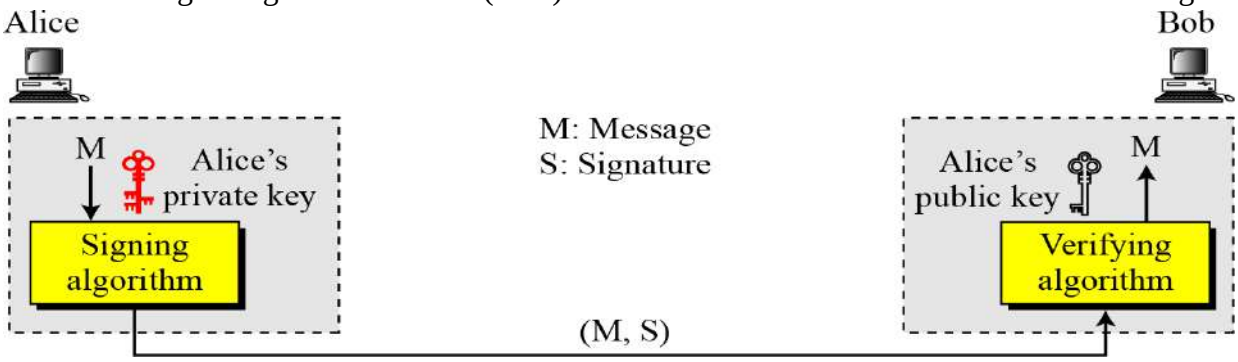
Definition:

A digital signature needs a public-key system. The signer signs with her private key; the verifier verifies with the signer's public key. A digital signature or digital signature scheme is a mathematical scheme for demonstration the authenticity of digital message or document.

Means, a digital signature is an authentication mechanism that enables the creator of a message to attach a code that act as a signature.

This signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

The digital signature standard (DSS) is an NIST standard that uses the secure hash algorithm (SHA).



Properties of Digital Signature

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Digital Signature Requirements

1. The signature must be a bit pattern that depends on the message being signed.
2. The signature must use some information unique to the sender to prevent both forgery and denial.
3. It must be relatively easy to produce the digital signature.
4. It must be relatively easy to recognize and verify the digital signature.
5. It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
6. It must be practical to retain a copy of the digital signature in storage.

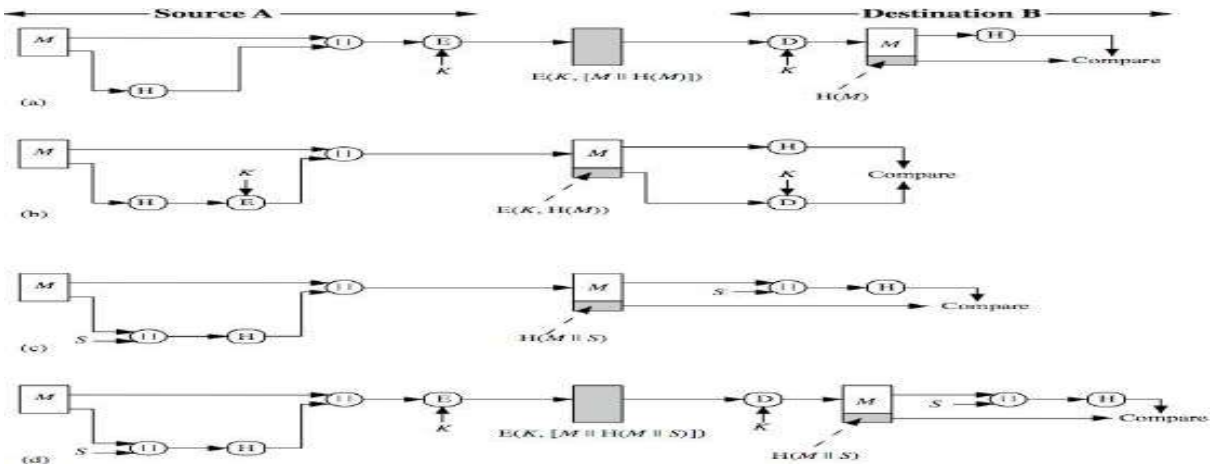
Approaches for Digital Signature

- Direct Digital Signature
- Arbitrated Digital Signature

Direct Digital Signature

The term direct digital signature refers to a digital signature scheme that involves only the communicating parties (source, destination) directly.

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message & signature
- security depends on sender's private-key



Arbitrated Digital Signature

In this every signed message from a sender X to a receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check it origin and content. The message is then dated and sent to Y.

This process is an indication that has been verified to the satisfaction of the arbiter.

By this process, it solves the direct Digital signature problem.

Sender X,

Arbiter A,

Receiver Y,

- X→ construct message M and compute hash value H(M) then X transmitted "M+ Digital Signature" to A.
  Signature consists→ identity "IDx of X +hash value" of all encrypted using K_{XA} (it is common shared key between Sender X and Arbiter A).
- A→ A decrypts the signature & checks the hash value to validate the message. Then transmit it to Y by encryption it with K_{AY} (it is common shared key between Arbiter A and Receiver Y). the message include IDx and M & time Stam.
- Y→ Decrypt it by using K_{AY}

(1) $X \rightarrow A: \ ID_X \| E(PR_x, [ID_X \| E(PU_y, E(PR_x, M))])$

(2) $A \rightarrow Y: E(PR_a, [ID_X \| E(PU_y, E(PR_x, M)) \| T])$

(c) Public-Key Encryption, Arbiter Does Not See Message

Notations:

X=sender M=message

Y=recipient T=time stamp

A=Arbiter PR_X=X's private key

ID_X=ID of X PU_Y=Y's public key PR_A=A's private key

Digital Signature Standard(DSS)

- US Govt approved signature scheme
- Designed by NIST (National Institute of Standards and Technology) & NSA in early 90's
- Published as Federal Information Processing Standard(FIPS 186) in 1991
- revised in 1993, 1996 & then 2000
- Uses the SHA hash algorithm

- The DSS makes use of the Secure Hash Algorithm (SHA) presents a new digital signature technique, the **Digital Signature Algorithm** (DSA).

- creates a 320 bit signature
- with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal & Schnorr schemes

The DSS uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange. Nevertheless, it is a public-key technique.

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature. The signature function also depends on the sender's private key (*PRa*) and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key (PU_G).

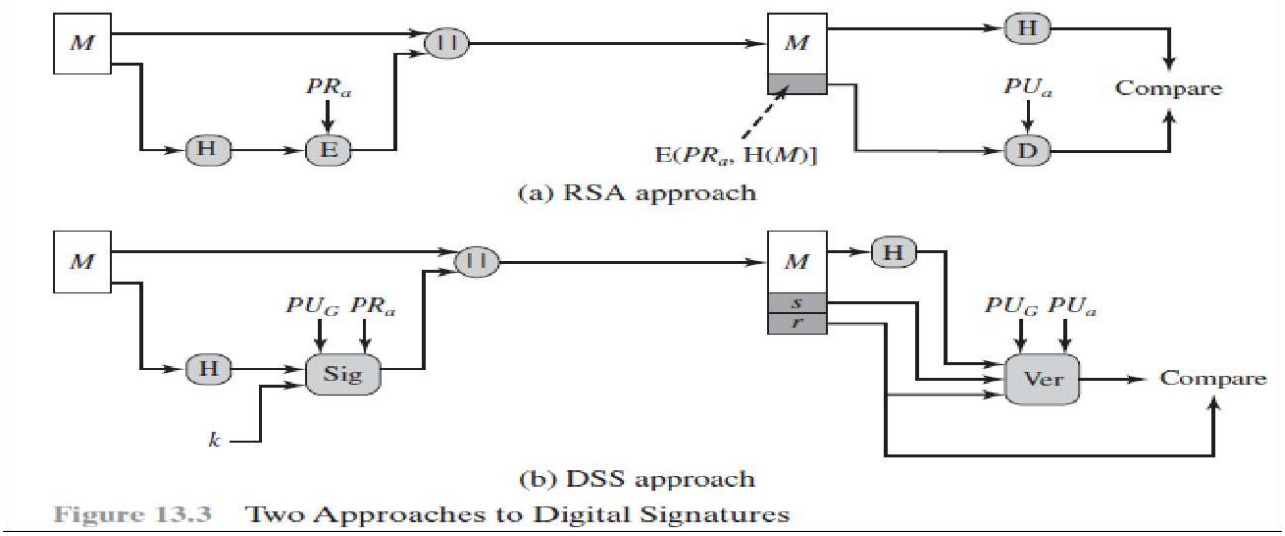


Figure 13.3 Two Approaches to Digital Signatures

Global Public-Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64; i.e., bit length of between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p - 1)$, where $2^{159} < q < 2^{160}$; i.e., bit length of 160 bits

g $= h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p - 1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

Signing

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1} (H(M) + xr)] \bmod q$

Signature = (r, s)

Verifying

$w = (s')^{-1} \bmod q$

$u_1 = [H(M')w] \bmod q$

$u_2 = (r')w \bmod q$

$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

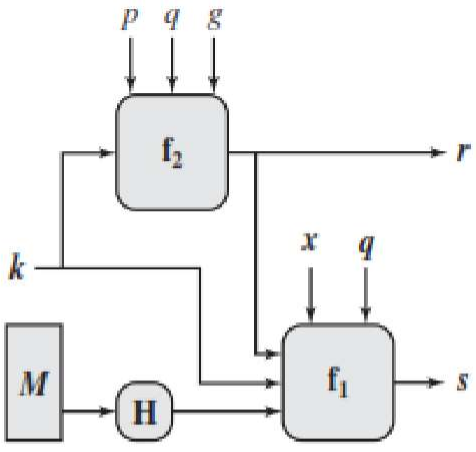
TEST: $v = r'$

M = message to be signed

$H(M)$ = hash of M using SHA-1

M', r', s' = received versions of M, r, s

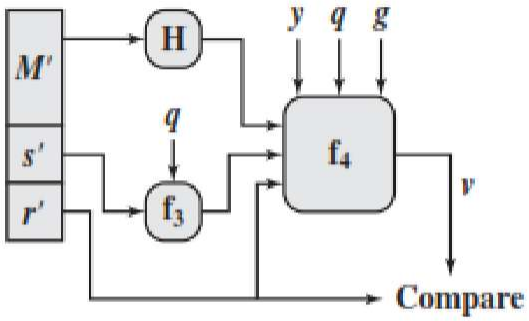
Figure 13.4 The Digital Signature Algorithm (DSA)



$s = f_1(H(M), k, x, r, q) = (k^{-1} (H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(a) Signing



$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q} y^{r'w \bmod q}) \bmod p) \bmod q$

(b) Verifying

Figure 13.5 DSS Signing and Verifying

AUTHENTICATION APPLICATIONS

- It will consider authentication functions
- Its developed to support application-level authentication & digital signatures
    1. Kerberos – a private-key authentication service
    2. X.509 - a public-key directory authentication service

KERBEROS

- Kerberos is an authentication service developed by MIT and is one of the best known and most widely implemented **trusted third party** key distribution systems.
- Provides a centralized authentication server whose function is to authenticate users to servers and servers to users.
- Kerberos relies exclusively on symmetric encryption, making no use of public-key encryption.
- Two versions of Kerberos are in common use.
    - Version 4
    - Version 5

Kerberos Requirements

- **Secure**: A network eavesdropper should not be able to obtain the necessary information to impersonate a user.
- **Reliable**: Kerberos should be highly reliable and should employ a distributed server architecture, with one system able to back up another.
- **Transparent**: The user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- **Scalable**: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

Kerberos Version 4

- A basic third-party authentication scheme
- Have an Authentication Server (AS)
    - Knows the passwords of all users and stores these in a centralized database.
    - AS shares a unique secret key with each server.
    - These keys have been distributed physically or in some other secure manner
    - users initially negotiate with AS to identify self
    - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Have a Ticket Granting server (TGS)
    - issues tickets to users who have been authenticated to AS
    - users subsequently request access to other services from TGS on basis of users TGT

Simple Authentication Dialogue

(1) C →AS: $ID_C||P_C||ID_V$

(2) AS →C: Ticket

(3) C →V : $ID_C||$Ticket

$Ticket = E(K_V, [ID_C||AD_C||ID_V])$

Where

C	= client	ID_V	= identifier of V
AS	= authentication server	P_C	= password of user on C
V	=server	AD_C	= network address of C
ID_C	= identifier of user on C	K_V	= secret encryption key shared by AS and V

- The ticket is encrypted to prevent alteration or forgery.
- The server's ID (ID_V) is included in the ticket so that the server can verify that it has decrypted the ticket properly.
- ID_C is included in the ticket to indicate that this ticket has been issued on behalf of C.

The Version 4 Authentication Dialogue

- Obtain ticket granting ticket from AS
  - Once per session
- Obtain service granting ticket from TGT
  - For each distinct service required
- Client/server exchange to obtain service
  - On every service request

Figure provides a simplified overview of the action.

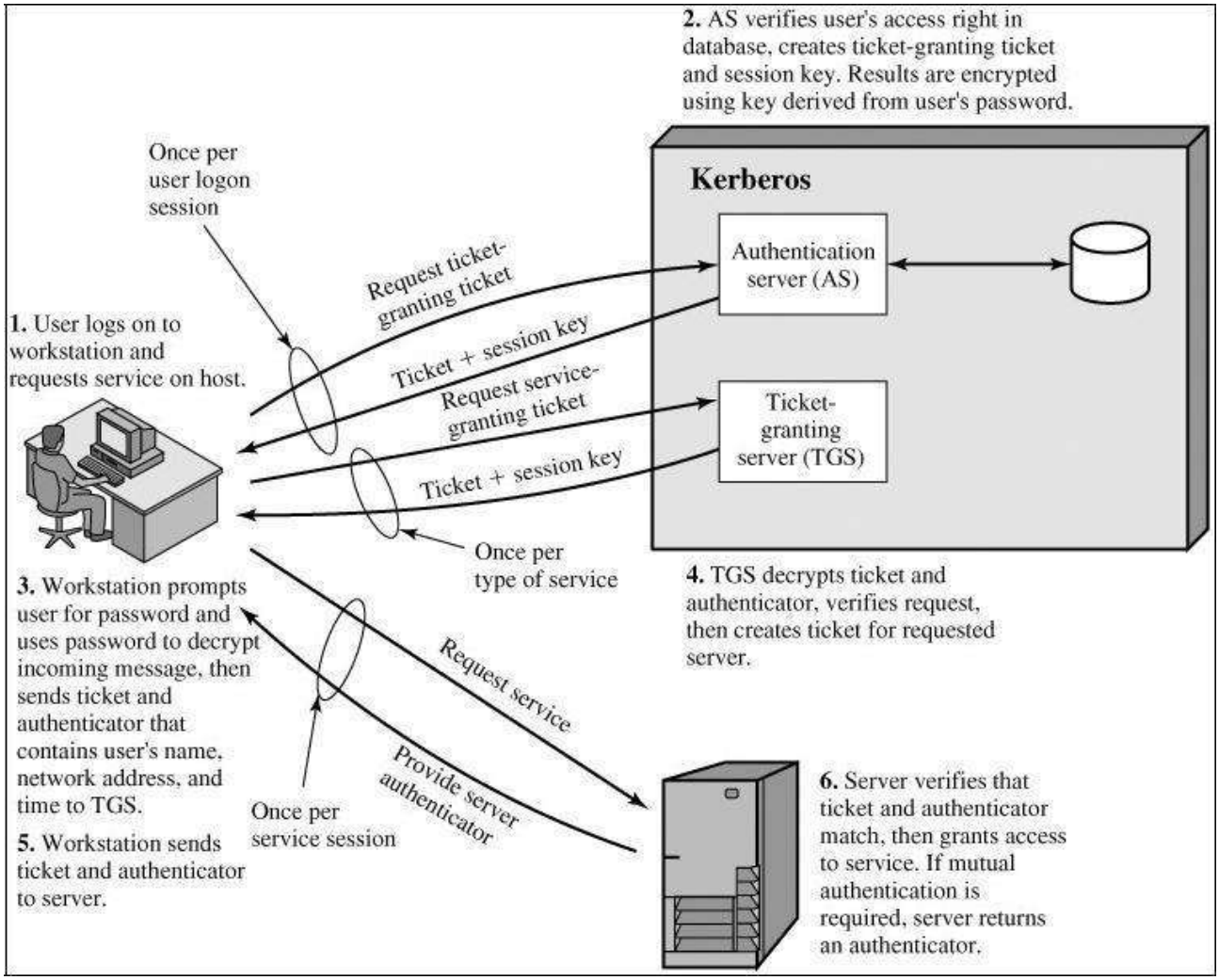


Figure: Overview of Kerberos

- Client sends a message to the AS requesting access to the TGS.
- AS responds with a message, encrypted with a key derived from the user's password (K_C) that contains the ticket.
- Encrypted message also contains a copy of the session key, $K_{C,tgs}$, where the subscripts indicate that this is a session key for C and TGS.
- Session key is inside the message encrypted with K_C, only the user's client can read it.
- Same session key is included in the ticket, which can be read only by the TGS.
- Thus, the session key has been securely delivered to both C and the TGS.
- Message (1) includes a timestamp, so that the AS knows that the message is timely.
- Message (2) includes several elements of the ticket in a form accessible to C. This enables C to confirm that this ticket is for the TGS and to learn its expiration time.

Table: Kerberos Version 4 Message Exchanges

(1) C →AS	$ID_C \| \| ID_{tgs} \| \| TS_1$
(2) AS →C	$E(K_c, [K_{c,tgs} \| \| ID_{tgs} \| \| TS_2 \| \| Lifetime_2 \| \| Ticket_{tgs}])$
	$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| \| ID_C \| \| AD_C \| \| ID_{tgs} \| \| TS_2 \| \| Lifetime_2])$
Authentication Service Exchange to obtain ticket-granting ticket	
(3) C→TGS	$ID_V \| \| Ticket_{tgs} \| \| Authenticator_C$
(4) TGS →C	$E(K_{c,tgs}, [K_{c,v} \| \| ID_V \| \| TS_4 \| \| Ticket_v])$
	$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| \| ID_C \| \| AD_C \| \| ID_{tgs} \| \| TS_2 \| \| Lifetime_2])$ $Ticket_v = E(K_v, [K_{c,v} \| \| ID_C \| \| AD_C \| \| ID_V \| \| TS_4 \| \| Lifetime_4])$ $Authenticator_C = E(K_{c,tgs}, [ID_C \| \| AD_C \| \| TS_3])$
Ticket-Granting Service Exchange to obtain service-granting ticket	
(5) C →V	$Ticket_V \| \| Authenticator_C$
(6) V→C	$E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)
	$Ticket_v = E(K_v, [K_{c,v} \| \| ID_C \| \| AD_C \| \| ID_V \| \| TS_4 \| \| Lifetime_4])$ $Authenticator_C = E(K_{c,v}, [ID_C \| \| AD_C \| \| TS_5])$
Client/Server Authentication Exchange to obtain service	

- The TGS can decrypt the ticket with the key that it shares with the AS. This ticket indicates that user C has been provided with the session key $K_{c,tgs}$. The ticket says, "Anyone who uses $K_{c,tgs}$ must be C."
- The TGS can then check the name and address from the authenticator with that of the ticket and with the network address of the incoming message. If all match, then the TGS is assured that the sender of the ticket is indeed the ticket's real owner.
- The reply from the TGS, in message (4), follows the form of message (2). The message is encrypted with the session key shared by the TGS and C and includes a session key to be shared between C and the server V, the ID of V, and the timestamp of the ticket. The ticket itself includes the same session key.
- C now has a reusable service-granting ticket for V. When C presents this ticket, as shown in message (5), it also sends an authenticator. The server can decrypt the ticket, recover the session key, and decrypt the authenticator.
- If mutual authentication is required, the server can reply as shown in message (6)
- The server returns the value of the timestamp from the authenticator, incremented by 1, and encrypted in the session key. C can decrypt this message to recover the incremented timestamp.

Finally, the client and server share a secret key. This key can be used to encrypt future messages between the two or to exchange a new random session key for that purpose.

Kerberos Realms

Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.
2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.
3. The Kerberos server in each interoperating realm shares a secret key with the server in the other

realm. The two Kerberos servers are registered with each other.

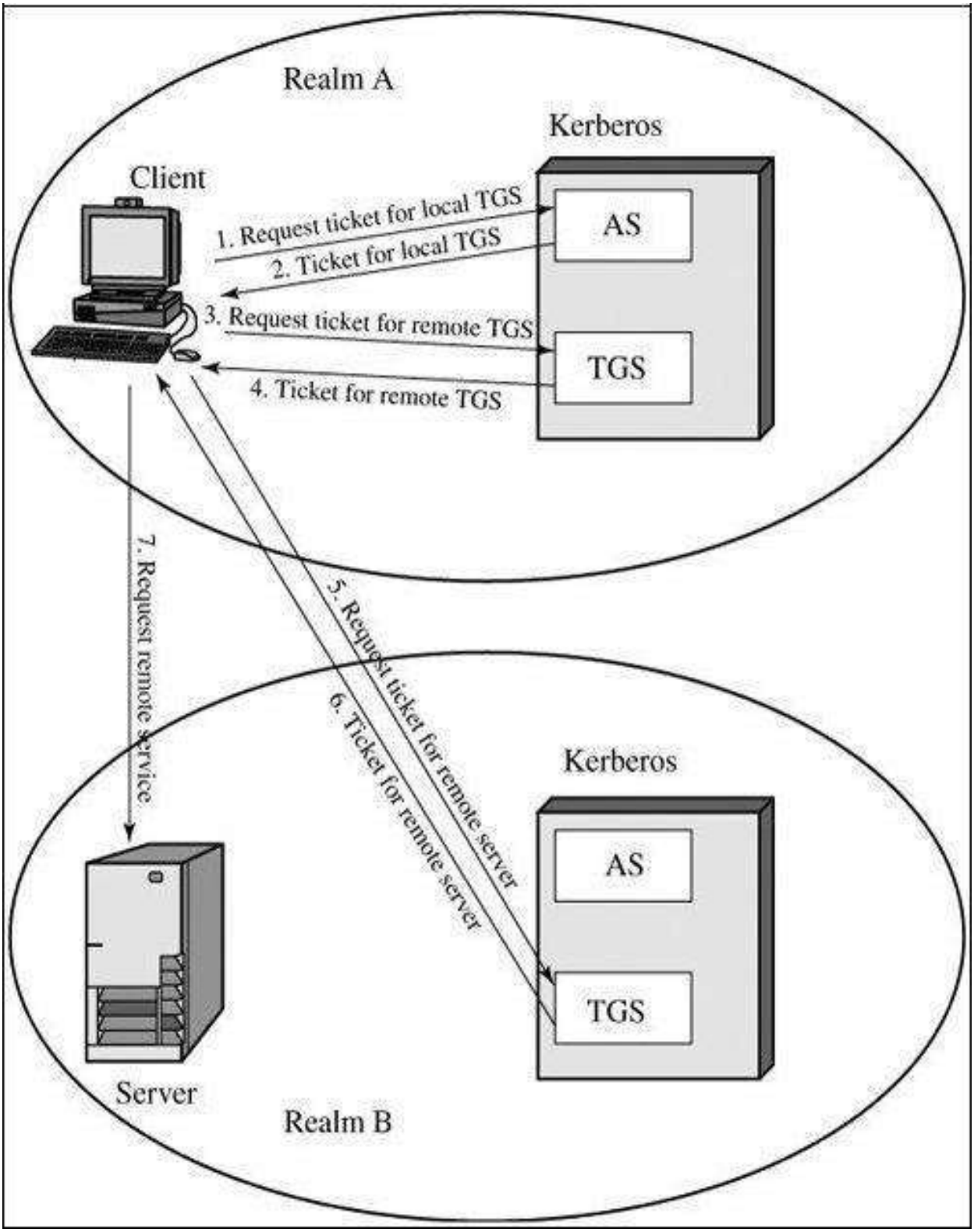


Figure: Request for Service in Another Realm

Such an environment is referred to as a **Kerberos realm**. The concept of *realm* can be explained as follows. A Kerberos realm is a set of managed nodes that share the same Kerberos database.

Kerberos principal, which is a service or user that is known to the Kerberos system. Each Kerberos principal is identified by its principal name. Principal names consist of three parts: **a service or user name, an instance name, and a realm name**

A user wishing service on a server in another realm needs a ticket for that server. The user's client follows the usual procedures to gain access to the local TGS and then requests a ticket-granting ticket for a remote TGS (TGS in another realm). The client can then apply to the remote TGS for a service-granting ticket for the desired server in the realm of the remote TGS.

The ticket presented to the remote server (V_{rem}) indicates the realm in which the user was originally authenticated. The server chooses whether to honor the remote request.

1) C →AS:	$ID_C \| \| ID_{tgs} \| \| TS_1$
(2) AS →C:	$E(K_C, [K_{c,tgs} \| \| ID_{tgs} \| \| TS_2 \| \| Lifetime_2 \| \| Ticket_{tgs}])$
(3) C →TGS:	$ID_{tgsrem} \| \| Ticket_{tgs} \| \| Authenticator_C$
(4) TGS →C:	$E(K_{c,tgs}, [K_{c,tgsrem} \| \| ID_{tgsrem} \| \| TS_4 \| \| Ticket_{tgsrem}])$
(5) C →TGS_{rem}:	$ID_{vrem} \| \| Ticket_{tgsrem} \| \| Authenticator_C$
(6) TGS_{rem} →C:	$E(K_{C,tgsrem}, [K_{c,vrem} \| \| ID_{vrem} \| \| TS_6 \| \| Ticket_{vrem}])$
(7) C →V_{rem}:	$Ticket_{vrem} \| \| Authenticator_C$

Kerberos Version 5

- Developed in mid 1990's
- Specified as internet standard rfc 1510
- Provides improvements over V4

Differences between Versions 4 and 5

1. **Kerberos Version 4 Environmental shortcomings**
    1. **Encryption system dependence:** Version 4 requires the use of **DES**. In version 5, ciphertext is tagged with an encryption type identifier so that **any encryption technique** may be used.
    2. **Internet protocol dependence:** Version 4 requires the use of **Internet Protocol (IP) addresses**. Version 5 **network addresses are tagged with type and length, allowing any network address type to be used.**
    3. **Message byte ordering:** In version 4, the sender of a message employs a **byte ordering** of its own choosing. In version 5, all message structures are defined using **Abstract Syntax Notation One (ASN.1)**
    4. **Ticket lifetime:** Lifetime values in version 4 are encoded in an 8-bit quantity in units of five minutes. Thus, the maximum lifetime that can be expressed is 2^8 x 5 = **1280 minutes**, or a little over 21 hours. In version 5, tickets include an **explicit start time and end time, allowing tickets with arbitrary lifetimes.**
    5. **Authentication forwarding:** Version 4 **does not allow credentials** issued to one client to be forwarded to some other host and used by some other client. Version 5 provides this capability.
    6. **Interrealm authentication:** In version 4, **interoperability among N realms** requires on the order of N^2 **Kerberos-to-Kerberos relationships**, as described earlier. Version 5 supports a method that requires fewer relationships, as described shortly.

2. **Technical Deficiencies**
    1. **Double encryption:** [messages (2) and (4)] that tickets provided to clients are encrypted twice, once with the secret key of the target server and then again with a secret key known to the client. **The second encryption is not necessary and is computationally wasteful.**
    2. **PCBC encryption:** Encryption in version 4 makes use of a nonstandard mode of DES known as **propagating cipher block chaining (PCBC).** Version 5 provides **explicit integrity mechanisms, allowing the standard CBC mode to be used for encryption.**
    3. **Session keys:** Each ticket includes a **session key** that is used by the client to encrypt the authenticator sent to the service associated with that ticket. In version 5, it is possible for a client and server to **negotiate a subsession key, which is to be used only for that one connection.**
    4. **Password attacks:** Both versions **are vulnerable to a password attack**. Version 5 does provide a mechanism known as **preauthentication**, which should make **password attacks more difficult**, but it **does not prevent them.**

SECURITY

The Version 5 Authentication Dialogue

Consider the **authentication service exchange**. Message (1) is a client request for a ticket-granting ticket. As before, it includes the ID of the user and the TGS. The following new elements are added:

- **Realm**: Indicates **realm of user**
- **Options**: Used to request that **certain flags** be set in the returned ticket
- **Times**: Used by the client to request the following time settings in the ticket:
  - **from: the desired start time for the requested ticket**
  - **till: the requested expiration time for the requested ticket**
  - **rtime: requested** renew-till time
- Error! Hyperlink reference not valid.**e**: A random value to be repeated in message (2) to assure that the response is fresh and has not been replayed by an opponent

Now compare the **ticket-granting service exchange** for versions 4 and 5. We see that message (3) for both versions includes an **authenticator, a ticket, and the name of the requested service.** In addition, version 5 includes **requested times and options for the ticket and a nonce**, all with functions similar to those of message (1).

Summary of Kerberos Version 5 Message Exchanges

(1) C →AS	$Options \| \| ID_C \| \| Realm_C \| \| ID_{tgs} \| \| Times \| \| Nonce_1$
(2) AS →C	$Realm_C \| \| ID_C \| \| Ticket_{tgs} \| \| E(K_C, [K_{C,tgs} \| \| Times \| \| Nonce_1 \| \| Realm_{tgs} \| \| ID_{tgs}])$
	$Ticket_{tgs} = E(K_{tgs}, [Flags \| \| K_{C,tgs} \| \| Realm_C \| \| ID_C \| \| AD_C \| \| Times])$
(a) Authentication Service Exchange to obtain ticket-granting ticket	
(3) C →TGS	$Options \| \| ID_V \| \| Times \| \| \| \| Nonce_2 \| \| Ticket_{tgs} \| \| Authenticator_C$
(4) TGS →C	$Realm_C \| \| ID_C \| \| Ticket_V \| \| E(K_{C,tgs}, [K_{C,V} \| \| Times \| \| Nonce_2 \| \| Realm_V \| \| ID_V])$ $Ticket_{tgs} = E(K_{tgs}, [Flags \| \| K_{C,tgs} \| \| Realm_C \| \| ID_C \| \| AD_C \| \| Times])$
	$Ticket_V = E(K_V, [Flags \| \| K_{C,V} \| \| Realm_C \| \| ID_C \| \| AD_C \| \| Times])$ $Authenticator_C = E(K_{C,tgs}, [ID_C \| \| Realm_C \| \| TS_1])$
(b) Ticket-Granting Service Exchange to obtain service-granting ticket	
(5) C →V	$Options \| \| Ticket_V \| \| Authenticator_C$
(6) V →C	$E_{K_{C,V}}[TS_2 \| \| Subkey \| \| Seq\#]$
	$Ticket_V = E(K_V, [Flags \| \| K_{C,V} \| \| Realm_C \| \| ID_C \| \| AD_C \| \| Times])$ $Authenticator_C = E(K_{C,V}, [ID_C \| \| Realm_C \| \| TS_2 \| \| Subkey \| \| Seq\#])$
(c) Client/Server Authentication Exchange to obtain service	

Client/server authentication exchange, several new features appear in version 5. In message (5), the client may request as an option that mutual authentication is required. The authenticator includes several new fields as follows:

- **Subkey**: The client's choice for an encryption key to be used to protect this specific application session. If this field is omitted, the session key from the ticket ($K_{C,V}$) is used.
- **Sequence number**: An optional field that specifies the starting sequence number to be used by the server for messages sent to the client during this session.

Ticket Flags

INITIAL	This ticket was issued using the AS protocol
PRE-AUTHENT	Client was authenticated by the KDC before a ticket was issued.
HW-AUTHENT	Use of hardware expected to be possessed solely by the named client.
RENEWABLE	This ticket can be used to obtain a replacement ticket that expires at a later date.
MAY-POSTDATE	Tells TGS that a postdated ticket may be issued based on this ticket-granting ticket.
POSTDATED	Indicates that this ticket has been postdated.
INVALID	This ticket is invalid and must be validated by the KDC before use.
PROXIABLE	Tells TGS that a new service-granting ticket with a different network address may be issued based on the presented ticket.
PROXY	Indicates that this ticket is a proxy.
FORWARDABLE	Tells TGS that a new ticket-granting ticket with a different network address may be issued based on this ticket-granting ticket.
FORWARDED	Indicates that this ticket has either been forwarded or was issued based on authentication involving a forwarded ticket-granting ticket.

X.509 AUTHENTICATION SERVICE

- ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a directory service
- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- X.509 certificates are widely used and has 3 versions.
- The directory may serve as a repository of public-key certificates of the type.

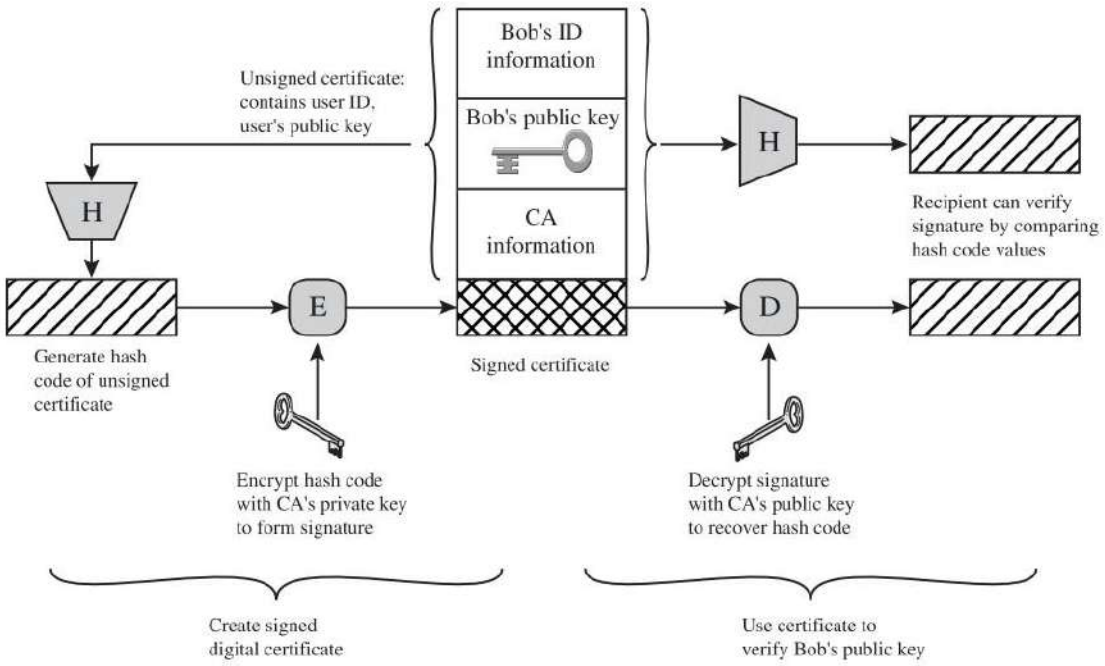


Figure: X.509 Certificate Use

- Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.
- In addition, X.509 defines alternative authentication protocols based on the use of public-key certificates.

- X.509 certificate format is used in S/MIME, IP Security, and SSL/TLS and SET.
- X.509 is based on the use of public-key cryptography and digital signatures. Algorithms not standardised, but RSA recommended.

Certificates

The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

Certificate includes the following elements:

- **Version:** Differentiates among successive versions of the certificate format; the default is version 1.
- **Serial number:** An integer value, unique within the issuing CA, that is unambiguously associated with this certificate.
- **Signature algorithm identifier:** The algorithm used to sign the certificate, together with any associated parameters.
- **Issuer name:** X.500 name of the CA that created and signed this certificate.
- **Period of validity:** Consists of two dates: the first and last on which the certificate is valid.
- **Subject name:** The name of the user to whom this certificate refers.
- **Subject's public-key information:** The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.
- **Issuer unique identifier:** An optional bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.
- **Subject unique identifier:** An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.
- **Extensions:** A set of one or more extension fields. Extensions were added in version 3.
- **Signature:** Covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.

Figure shows the general format of a certificate

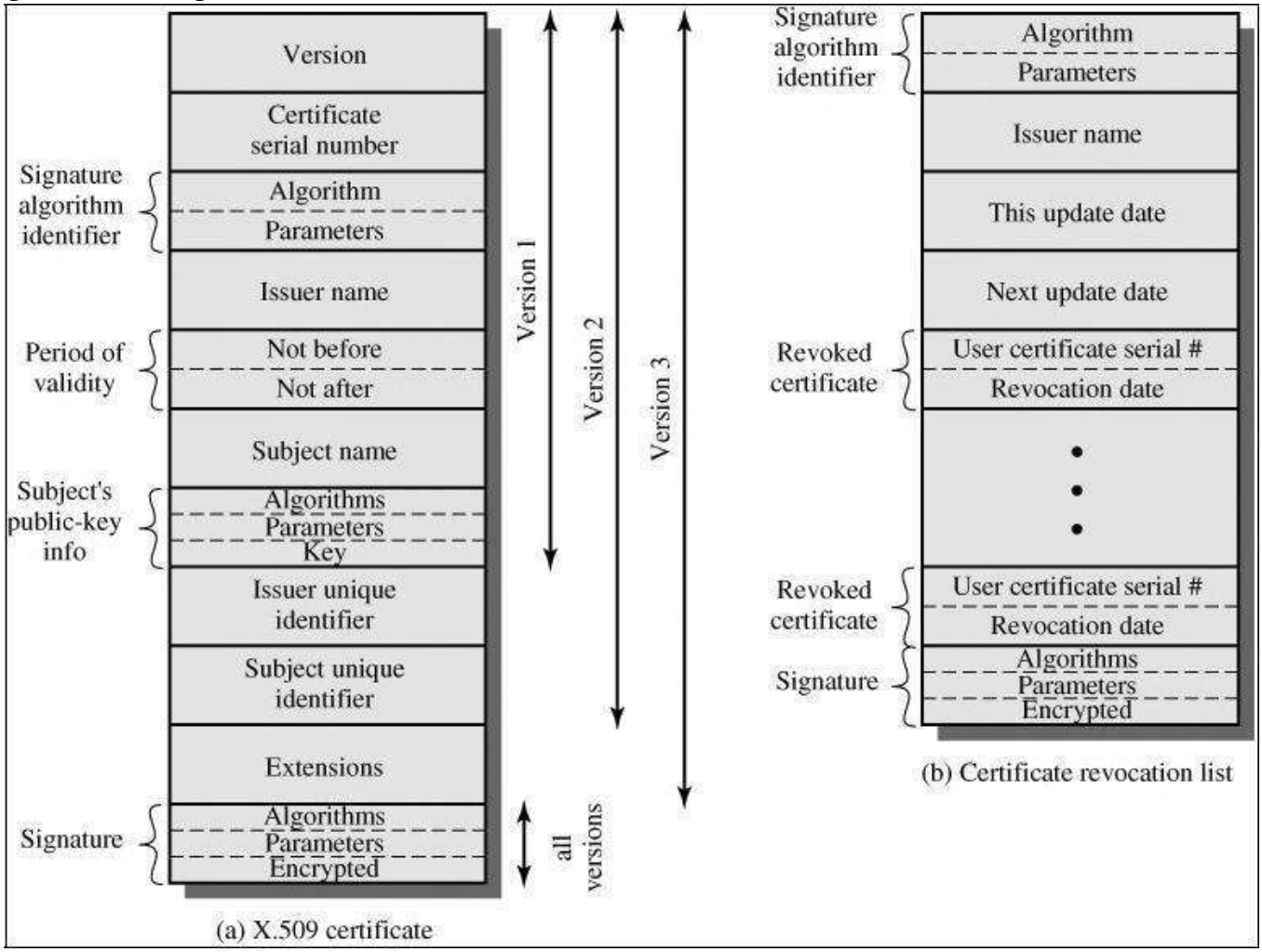


Figure: X.509 Formats

\

The standard uses the following notation to define a certificate:

$CA \ll A \gg = CA \{V, SN, AI, CA, UCA, A, UA, Ap, T^A\}$

where

$Y \ll X \gg$ = the certificate of user X issued by certification authority Y
$Y\{I\}$ = the signing of I by Y. It consists of I with an encrypted hash code appended
V = version of the certificate
SN = serial number of the certificate
AI = identifier of the algorithm used to sign the certificate
CA = name of certificate authority
UCA = optional unique identifier of the CA
A = name of user A
UA = optional unique identifier of the user A
Ap = public key of user A
T^A = period of validity of the certificate

Obtaining a Certificate

User certificates generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verify the user public key that was certified.
- No party other than the certification authority can modify the certificate without this being detected.

Because certificates are unforgettable, they can be placed in a directory without the need for the directory to make special efforts to protect them.

CA Hierarchy

- If both users share a common CA then they are assumed to know its public key
- Otherwise CA's must form a hierarchy
- All these certificates of CAs by CAs need to appear in the directory, and the user needs to know how they are linked to follow a path to another user's public-key certificate.
- X.509 suggests that CAs be arranged in a hierarchy so that navigation is straightforward.
- Use certificates linking members of hierarchy to validate other CA's
- Each CA has certificates for clients (forward) and parent (backward)
    - Each client trusts parents certificates
    - Enable verification of any certificate from one CA by users of all other cas in hierarchy
- The directory entry for each CA includes two types of certificates:
    - **Forward certificates:** Certificates of X generated by other CAs
    - **Reverse certificates:** Certificates generated by X that are the certificates of other CAs

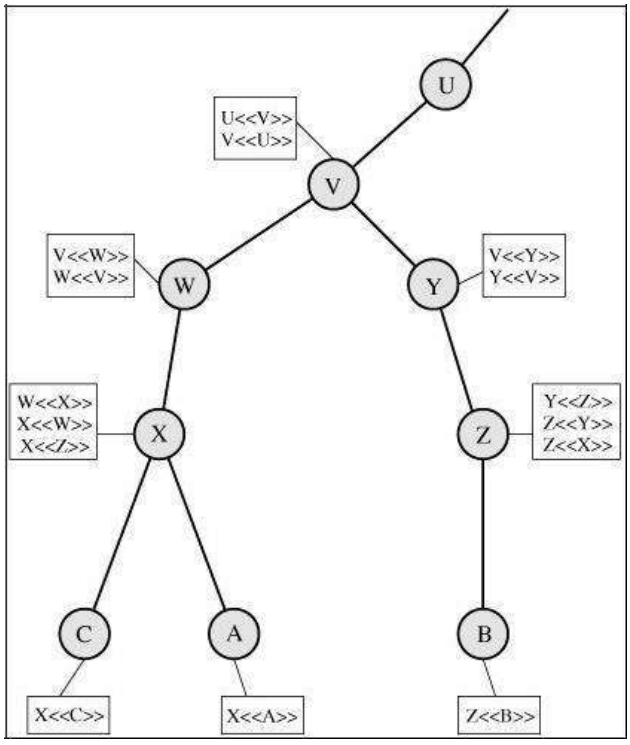


Figure: X.509 Hierarchy: A Example

Track chains of certificates:

A acquires B certificate using chain: X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>
B acquires A certificate using chain: Z<<Y>>Y<<V>>V<<W>>W<<X>>X<<A>>

Certificate Revocation

- Certificates have a period of validity
- May need to revoke before expiry, eg:
    - User's private key is compromised
    - User is no longer certified by this CA
    - CA's certificate is compromised

- CA maintain a list consisting of all revoked but not expired certificates issued by that CA, including both those issued to users and to other CAs.
- Each certificate revocation list (CRL) posted to the directory is signed by the issuer
- When a user receives a certificate in a message, the user must determine whether the certificate has been revoked.
- The user could check the directory each time a certificate is received.
- To avoid the delays associated with directory searches, it is likely that the user would maintain a local cache of certificates and lists of revoked certificates.

Authentication Procedures

- X.509 also includes three alternative authentication procedures
- All these procedures make use of public-key signatures.
- It is assumed that the two parties know each other's public key, either by obtaining each other's certificates from the directory or because the certificate is included in the initial message from each side.

1. One-Way Authentication

One way authentication involves a single transfer of information from one user (A) to another (B), and establishes the following:

1. The identity of A and that the message was generated by A
2. That the message was intended for B
3. The integrity and originality (it has not been sent multiple times) of the message

- Only the identity of the initiating entity is verified in this process, not that of the responding entity.
- Message must include timestamp, nonce, B's identity and is signed by A
- May also be used to convey a session key to B, encrypted with B's public key.

2. Two-Way Authentication

In addition to the three elements just listed, two-way authentication establishes the following elements:

4. The identity of B and that the reply message was generated by B
5. That the message was intended for A
6. The integrity and originality of the reply

- Two-way authentication thus permits both parties in a communication to verify the identity of the other.
- reply includes original nonce from A, also timestamp and nonce from B
- may include additional info for A

3. Three-Way Authentication

- 3 messages (A->B, B->A, A->B) which enables above authentication without synchronized clocks
- has reply from A back to B containing signed copy of nonce from B
- means that timestamps need not be checked or relied upon

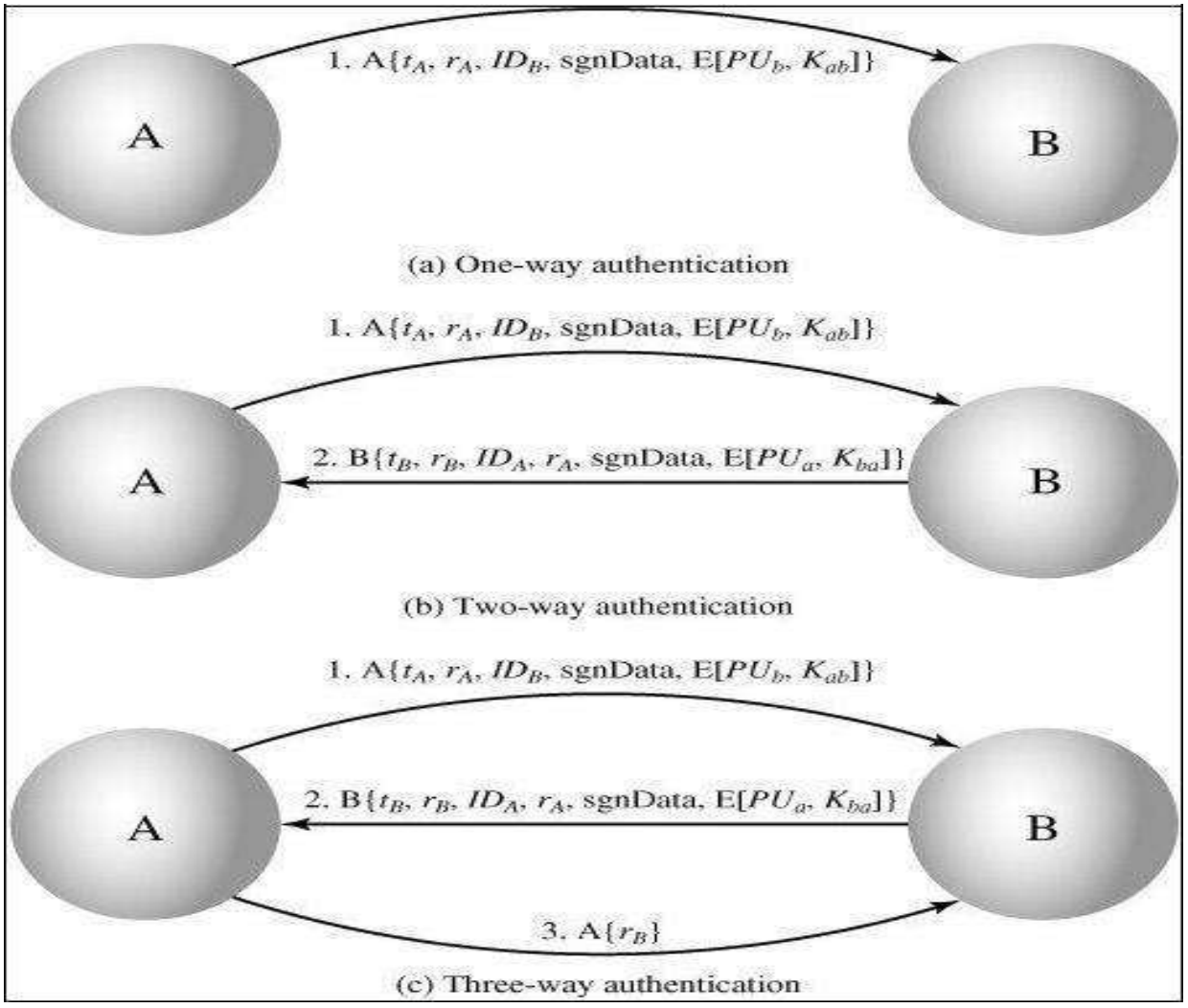


Figure: X.509 Strong Authentication Procedures

X.509 Version 3

The X.509 version 2 format does not convey all of the information that recent design and implementation experience has shown to be needed. The following requirements not satisfied by version 2:

1. The Subject field is inadequate to convey the identity of a key owner to a public-key user. X.509 names may be relatively short and lacking in obvious identification details that may be needed by the user.
2. The Subject field is also inadequate for many applications, which typically recognize entities by an Internet e-mail address, a URL, or some other Internet-related identification.
3. There is a need to indicate security policy information. This enables a security application or function, such as IPSec, to relate an X.509 certificate to a given policy.
4. There is a need to limit the damage that can result from a faulty or malicious CA by setting constraints on the applicability of a particular certificate.
5. It is important to be able to identify different keys used by the same owner at different times. This feature supports key life cycle management, in particular the ability to update key pairs for users and CAs on a regular basis or under exceptional circumstances.

The certificate **extensions** fall into three main categories: key and policy information, subject and issuer attributes, and certification path constraints.

(1) Key and Policy Information

These extensions convey additional information about the subject and issuer keys, plus indicators of certificate policy.. For example, a policy might be applicable to the authentication of electronic data interchange (EDI) transactions for the trading of goods within a given price range.

This area includes the following:

Authority key identifier: Identifies the public key to be used to verify the signature on this certificate or CRL.

Subject key identifier: Identifies the public key being certified.

Key usage: Indicates a restriction imposed as to the purposes for which, and the policies under which, the certified public key may be used.

Private-key usage period: Indicates the period of use of the private key corresponding to the public key. For example, with digital signature keys, the usage period for the signing private key is typically shorter than that for the verifying public key.

Certificate policies: Certificates may be used in environments where multiple policies apply.

Policy mappings: Used only in certificates for CAs issued by other CAs.

(2) Certificate Subject and Issuer Attributes

These extensions support alternative names, in alternative formats, for a certificate subject or certificate issuer and can convey additional information about the certificate subject, to increase a certificate user's confidence that the certificate subject is a particular person or entity. For example, information such as postal address, position within a corporation, or picture image may be required.

The extension fields in this area include the following:

- **Subject alternative name:** Contains one or more alternative names, using any of a variety of forms
- **Subject directory attributes:** Conveys any desired X.500 directory attribute values for the subject of this certificate.

(3) Certification Path Constraints

These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs. The extension fields in this area include the following:

- **Basic constraints:** Indicates if the subject may act as a CA. If so, a certification path length constraint may be specified.
- **Name constraints**: Indicates a name space within which all subject names in subsequent certificates in a certification path must be located.
- **Policy constraints**: Specifies constraints that may require explicit certificate policy identification or inhibit policy mapping for the remainder of the certification path.

UNIT V

INTRODUCTION TO CYBER SECURITY

1.Write Short notes on Cyber crime.

Cyber Crime

Meaning –

Criminal activities carried out by means of computers or the internet.Definition

–

- Cybercrime is defined as a crime where a computer is the object of the crime or isused as a tool to commit an offense.
- A cybercriminal may use a device to access a user's personal information, confidential business information, government information, or disable a device.
- Cybercrime, also called computer crime, the use of a computer as an instrument tofurther illegal ends, such as committing fraud, trafficking in child pornography and intellectual property, stealing identities, or violating privacy.
- Cybercrime, especially through the Internet, has grown in importance as the computer has become central to commerce, entertainment, and government.
- Cyber crime or computer-oriented crime is a crime that includes a computer and anetwork. The computer may have been used in the execution of a crime or it may bethe target.

Cyber crime encloses a wide range of activities, but these can generally be divided into two categories:

a) Crimes that aim computer networks or devices. These types of crimes involve different threats (like virus, bugs etc.) and denial-of-service attacks.

b) Crimes that use computer networks to commit other criminalactivities. These types of crimes include cyber stalking, financial fraud or identity theft.

Origin of the word Cyber Crime

Cyber came from cybernetics. Cybernetics influences game, system, and organizational theory. Cybernetics derived from the Greek kubernētēs which refers to a

pilot or steersman. Related is the Greek word kubernēsis which means "the gift ofgovernance" and applies to leadership.

Who are cyber criminals?

A cybercriminal is an individual who commits cybercrimes, where he/she makes use of the computer either as a tool or as a target or as both.

- Types of Cyber Criminals:

1. Hackers:
   The term hacker may refer to anyone with technical skills, however, it typicallyrefers to an individual who uses his or her skills to achieve unauthorized accessto systems or networks so as to commit crimes.
2. Organized Hackers:
   These criminals embody organizations of cyber criminals, terrorists, and state- sponsored hackers. Cyber criminals are typically teams of skilled criminals targeted on control, power, and wealth. These criminals are extremely organized, and should even give crime as a service. These attackers are usuallyprofoundly prepared and well-funded.
3. Internet stalkers:
   Internet stalkers are people who maliciously monitor the web activity of their victims to acquire personal data. This type of cyber crime is conducted throughthe use of social networking platforms and malware, that are able to track an individual's PC activity with little or no detection.
4. Disgruntled Employees:
   Disgruntled employees become hackers with a particular motive and also commit cyber crimes. It is hard to believe that dissatisfied employees can become such malicious hackers. In the previous time, they had the only option of going on strike against employers. But with the advancement of technology there is increased in work on computers and the automation of processes, it is simple for disgruntled employees to do more damage to their employers and organization by committing cyber crimes. The attacks by such employees bringthe entire system down.

2.Describe the Classification of Cyber Crimes in detail.

Email spoofing

- Email spoofing is a form of cyber attack in which a hacker sends an email that has been manipulated to seem as if it originated from a trusted source.
- For example, a spoofed email may pretend to be from a well-known shopping website, asking the recipient to provide sensitive data, such as a password or credit card number.
- Alternatively, a spoofed email may include a link that installs malware on theuser's device if clicked.
- An example of spoofing is when an email is sent from a false sender address,that asks the recipient to provide sensitive data.
- This email could also contain a link to a malicious website that contains malware.

Spamming

- Spamming is the use of electronic messaging systems like e-mails and other digital delivery systems and broadcast media to send unwanted bulk messagesindiscriminately.
- The term spamming is also applied to other media like in internet forums, instant messaging, and mobile text messaging, social networking spam, junk fax transmissions, television advertising and sharing network spam.
- Spam is any kind of unwanted, unsolicited digital communication that gets sentout in bulk. Often spam is sent via email, but it can also be distributed via textmessages, phone calls, or social media.

Cyber defamation

- The tort of cyber defamation is an act of intentionally insulting, defaming or offending another individual or a party through a virtual medium.
- It can be both written and oral.
- Defamation means giving an "injury to the reputation of a person" resulting from a statement which is false. The term defamation is used in the section 499 of Indian Penal Code, 1860.
- Cyber defamation is also known as internet defamation or online defamation inthe world of internet and its users.
- Cyber defamation is also known as internet defamation or online defamation inthe world of internet and its users.

- Cyber defamation is a new concept but it virtually defames a person through new medium. The medium of defaming the individual's identity is through the help of computers via internet.

Internet time theft

- It refers to the theft in a manner where the unauthorized person uses internethours paid by another person.
- The authorized person gets access to another person's ISP user ID and password,either by hacking or by illegal means without that person's knowledge.
- Basically, Internet time theft comes under hacking. It is the use by anunauthorized person, of the Internet hours paid for by another person.
- **Salami Attack**
- A salami attack is a small attack that can be repeated many times veryefficiently. Thus the combined output of the attack is great.
- In the example above, it refers to stealing the round-off from interest in bankaccounts.
- Even though it is less than 1 cent per account, when multiplied by millions ofaccounts over many months, the adversary can retrieve quite a large amount. Itis also less likely to be noticeable since your average customer would assumethat the amount was rounded down to the nearest cent.

Data Diddling

- Data diddling is a type of cybercrime in which data is altered as it is entered intoa computer system, most often by a data entry clerk or a computer virus.
- Data diddling is an illegal or unauthorized data alteration. Changing data beforeor as it is input into a computer or output.
- Example: Account executives can change the employee time sheet information of employees before entering to the HR payroll application.

Forgery

- Forger" redirects here.

- When a perpetrator alters documents stored in computerized form, the crime committed may be forgery. In this instance, computer systems are the target ofcriminal activity.
- The term forgery usually describes a message related attack against a cryptographic digital signature scheme. That is an attack trying to fabricate a digital signature for a message without having access to the respective signer'sprivate signing key.
- Among the many examples of this crime, taking another's work, whether it be written or visual, such as a artwork, and attempting to distribute it as either yourown or as an original is an example of forgery.
- Likewise, either creating fake documents or producing counterfeit items is considered to be forgery as well.

Web Jacking

- Illegally seeking control of a website by taking over a domain is know as WebJacking.
- Web jacking attack method is one kind of trap which is spread by the attacker to steal the sensitive data of any people, and those people got trappedwho are not aware about cyber security.
- Web jacking attack method is another type of social engineering phishing attack where an attacker create a fake web page of victim website
- An attacker send it to the victim and when a victim click on that link, a messagedisplay on the browser "the site abc.com has move on another address, click here to go to the new location"
- If a victim does click on the link, he/she will redirect on the fake website pagewhere an attacker can ask for any sensitive data such as credit card number, username, password etc.

Emanating from UseNet

- Usenet is a kind of discussion group where people can share views on topic of their interest. The article posted to a newsgroup becomes available to all readersof the newsgroup.
- By its very nature, Usenet groups may carry very offensive, harmful, inaccurateor otherwise inappropriate material, or in some cases, postings that have been mislabeled or are deceptive in another way.

it may block the sender's email address or IP address. This simple action willstop the email bomb by rejecting additional emails from the sender.

Intrusion

- The definition of an intrusion is an unwelcome interruption or a situation wheresomewhere private has an unwelcome visit or addition. When you are having a quiet nap in your backyard and your neighbor's dog comes in uninvited and jumps all over you to wake you up, this is an example of an intrusion.
- A network intrusion refers to any unauthorized activity on a digital network. Network intrusions often involve stealing valuable network resources and almost always jeopardize the security of networks and/or their data. In order to proactively detect and respond to network intrusions, organizations and their cybersecurity teams need to have a thorough understanding of how network intrusions work and implement network intrusion, detection, and response systems that are designed with attack techniques and cover-up methods in mind.

Password sniffing

- Password Sniffing is a hacking technique that uses a special software application that allows a hacker to steal usernames and passwords simply by observing and passively recording network traffic. This often happens on public WiFi networks where it is relatively easy to spy on weak or unencrypted traffic.
- Password sniffing is an attack on the Internet that is used to steal user names andpasswords from the network. Today, it is mostly of historical interest, as mostprotocols nowadays use strong encryption for passwords. However, it used to be the worst security problem on the Internet in the 1990s, when news of major password sniffing attacks were almost weekly.
- The typical implementation of a password sniffing attack involves gaining access to a computer connected to a local area network and installinga password sniffer on it. The password sniffer is a small program that listens to all traffic in the attached network(s), builds data streams out of TCP/IP packets,

and extracts user names and passwords from those streams that contain protocols that send cleartext passwords.

- The attack can also be performed in switches, routers, and printers. It is common nowadays for attackers to install presence on such devices. They don't run anti-virus and aren't easy to audit. Furthermore, traffic naturally goes through switches and routers, so no extra network packets need to be sent to fool switches into sending traffic of interest to the listening node.

Credit card fraud

- Credit card fraud occurs when an unauthorized person gains access to your information and uses it to make purchases. ... Skimming your credit card, suchas at a gas station pump. Hacking your computer. Calling about fake prizes orwire transfers.
- Here criminals make purchases or obtain cash advances using a credit card account assigned to you. This can occur through one of your existing accounts, via theft of your physical credit card or your account numbers and PINs, or by means of new credit card accounts being opened in your name without your knowledge. Once they're in, thieves then run up charges and stick you and yourcredit card company with the bill.

Identity Theft

- Identity theft is the crime of obtaining the personal or financial information of another person to use their identity to commit fraud, such as making unauthorized transactions or purchases.

3.Write short notes on cyber security

Definition-

Cyber security is the technique of protecting computers, networks, programs and datafrom unauthorized access or attacks that are aimed for exploitation.

OR

Cyber security refers to the measures taken to keep electronic information private and safe from damage or theft. It is also used to make sure electronic devices and data arenot misused.

OR

Cyber security is the body of technology, processes and practices designed to protect network, computers, computer programs and data from attack, damage or unauthorized access.

Cyber Security Threats-

A cyber or cyber security threat is a malicious act that seeks to damage data, steal data, or disrupt digital life in general.

Cyber threats include computer viruses, data breaches, Denial of Service (DoS) attacks and other attack vectors.

Viruses-

- A computer virus is a program which can harm our device and files and infectthem for no further use.
- When a virus program is executed, it replicates itself by modifying other computer programs and instead enters its own coding.
- This code infects a file or program and if it spreads massively, it may ultimatelyresult in crashing of the device.
- Viruses affect your computer by corrupting files, interrupting Internet traffic andtaking over basic functions of your operating system.
- These behaviors can knock a system offline and cause crashes.

- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.

Trojan -

- A Trojan horse is malicious software that is concealed as a useful hostprogram.
- When the host program is run, the Trojan performs a harmful/unwanted
- A Trojan horse, often known as a Trojan, is malicious malware or software that appears to be legal yet has the ability to take control of yourcomputer.
- A Trojan is a computer program that is designed to disturb, steal, or otherwise harm your data or network.

Malwares-

- Malware ("malicious software") is a type of computer program thatinfiltrates and damages systems without the users' knowledge.
- Malware tries to go unnoticed by either hiding or not letting the userknow about its presence on the system.
- You may notice that your system is processing at a slower rate thanusual.

Ransom wares-

- Ransom ware is a type of malware that threatens to publish the victim's personal data or block access to it unless a ransom is paid.
- While some simple ransom ware may lock the system so that it is not difficult for a knowledgeable person to reverse, more advanced malwareuses a technique called crypto viral extortion.
- It encrypts the victim's files, making them inaccessible, and demands a ransom payment to decrypt them.

- Ransom ware attacks are typically carried out using a Trojan as a file that the user is tricked into downloading or opening when it arrives as an email attachment.

Vulnerability-

- In cyber security, a vulnerability is a weakness that can be exploited by cybercriminals to gain unauthorized access to a computer system.
- After exploiting a vulnerability, a cyber attack can run malicious code, install malware and even steal sensitive data.

Security Vulnerability Types

Computer security vulnerabilities can be divided into numerous types based on different criteria—such as where the vulnerability exists, what caused it, or how it could be used. Some broad categories of these vulnerability types include:

- Network Vulnerabilities
- Operating System Vulnerabilities
- Human Vulnerabilities
- Process Vulnerabilities

Network Vulnerabilities

- These are issues with a network's hardware or software that expose it to possible intrusion by an outside party.
- Examples include insecure Wi-Fi access points and poorly-configured firewalls.

Operating System Vulnerabilities

- These are vulnerabilities within a particular operating system that hackers may exploit to gain access to an asset the OS is installed on—or to caused amage.
- Examples include default superuser accounts that may exist in some OS installs and hidden backdoor programs.

Human Vulnerabilities

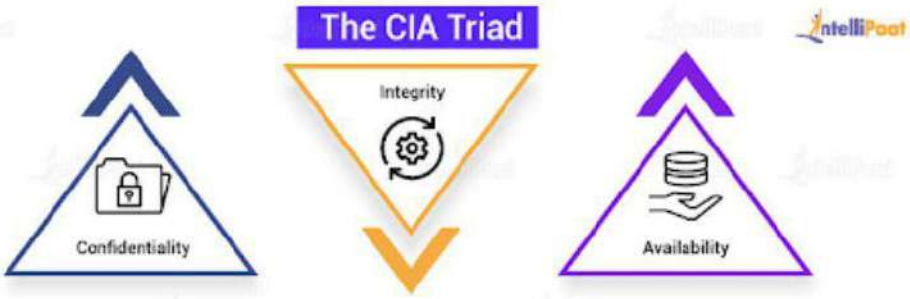
- The weakest link in many cyber security architectures is the humanelement.
- User errors can easily expose sensitive data, create exploitable accesspoints for attackers, or disrupt systems.

CIA Triad-

- **Confidentiality, integrity and availability**, also known as the CIA triad, is a model designed to guide policies for information security within an organization.
- The model is also sometimes referred to as the AIC triad (availability, integrity and confidentiality) to avoid confusion with the Central Intelligence Agency.
- The CIA triad is a common, respected model that forms the basis for thedevelopment of security systems and policies.
- These are used for the identification of vulnerabilities and methods foraddressing problems and creating effective solutions.

Examples of CIA Triad

- The two-factor authentication (debit card with the PIN code) provides confidentiality before authorizing access to sensitive data.
- The ATM and bank software ensure data integrity by maintainingall transfer and withdrawal records made via the ATM in the user'sbank accounting.



Confidentiality

- Confidentiality involves the efforts of an organization to make sure data is kept secret or private.
- To accomplish this, access to information must be controlled to prevent the unauthorized sharing of data—whether intentional or accidental.
- A key component of maintaining confidentiality is making sure that people without proper authorization are prevented from accessing assets important to your business. Conversely, an effective system also ensures that those who need to have access have the necessary privileges.
- For example, those who work with an organization's finances should be able to access the spreadsheets, bank accounts, and other information related to the flow of money. However, the vast majority of other employees—and perhaps evencertain executives—may not be granted access. To ensure these policies are followed, stringent restrictions have to be in place to limit who can see what.

Integrity

- Integrity involves making sure your data is trustworthy and free from tampering.The integrity of your data is maintained only if the data is authentic, accurate,and reliable.
- For example, if your company provides information about senior managers onyour website, this information needs to have integrity. If it is inaccurate, those visiting the website for information may feel your organization is not trustworthy. Someone with a vested interest in damaging the reputation of your organization may try to hack your website and alter the descriptions, photographs, or titles of the executives to hurt their reputation or that of thecompany as a whole.
    - **Availability**
        - Even if data is kept confidential and its integrity maintained, it is often useless unless it is available to those in the organization and the customers they serve.
        - This means that systems, networks, and applications must be functioning as they should and when they should.

4.Write short notes on Tools and Methods used in Cyber Crime

Proxy Server

- It is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.
- A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify andcontrol its complexity.
- Proxies were invented to add structure and encapsulation to distributedsystems.
- Today, most proxies are web proxies, facilitating access to content on theWorld Wide Web and providing anonymity.

A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

- A proxy server that passes requests and responses unmodified is usuallycalled a gateway or sometimes a tunneling proxy.
- A forward proxy is an Internet-facing proxy used to retrieve froma wide range of sources (in most cases anywhere on the Internet).
- A reverse proxy is usually an Internet-facing proxy used as a front-end tocontrol and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load-balancing, authentication, decryption or caching.

Anonymizer

- An anonymizer or an anonymous proxy is a tool that attempts to makeactivity on the Internet untraceable.
- It is a proxy server computer that acts as an intermediary and privacyshield between a client computer and the rest of the Internet.
- It accesses the Internet on the user's behalf, protecting personalinformation by hiding the client computer's identifying information.

There are many reasons for using anonymizers.

- Anonymizers help minimize risk.
- They can be used to prevent identity theft, or to protect search historiesfrom public disclosure.
- Some countries apply heavy censorship on the internet. Anonymizers can help in allowing free access to all of the internet content, but cannot helpagainst persecution for accessing the Anonymizer website itself.
- Furthermore, as information itself about Anonymizer websites arebanned in these countries, users are wary that they may be falling into agovernment-set trap.
- Anonymizers are also used by people who wish to receive objective information with the growing target marketing on the internet and targeted information.

- An anonymizer or an anonymous proxy is a tool that attempts to makeactivity on the Internet untraceable.
- It is a proxy server computer that acts as an intermediary and privacyshield between a client computer and the rest of the Internet.

It accesses the Internet on the user's behalf, protecting personal information by hidingthe client computer's identifying information

Phishing

- Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a genuine (legal) organization to ensnare individuals into providing sensitive data such as personally identifiable information, banking and credit card details, andpasswords.
- The information is then used to access important accounts and can resultin identity theft and financial loss.
- Phishers frequently use emotions like fear, curiosity, urgency, and greedto force recipients to open attachments or click on links.
- Phishing attacks are designed to appear to come from legitimate (legal)companies and individuals.

Viruses-

- A computer virus is a program which can harm our device and files and infectthem for no further use.
- When a virus program is executed, it replicates itself by modifying other computer programs and instead enters its own coding.
- This code infects a file or program and if it spreads massively, it may ultimatelyresult in crashing of the device.
- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.
- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.

Worms-

- Computer worms are similar to viruses in that they replicate themselves and caninflict similar damage.
- Unlike viruses, which spread by infecting a host file, worms are freestandingprograms that do not require a host program or human assistance to propagate.
- Worms don't change programs; instead, they replicate themselves over andover.
- They just eat resources to make the system down.

Keylogger -

- Keyloggers are a form of spyware where users are unaware their actions are being tracked.
- Keyloggers can be used for a variety of purposes; hackers may use them to maliciously gain access to your private information, while employersmight use them to monitor employee activities.
- A keylogger is a tool that captures and records a user's keystrokes. It can record instant messages, email, passwords and any other information you type at any time using your keyboard.

- Keyloggers can be hardware or software.

Trojan -

- A Trojan horse is malicious software that is concealed as a useful hostprogram.
- When the host program is run, the Trojan performs a harmful/unwantedaction.
- A Trojan horse, often known as a Trojan, is malicious malware or software that appears to be legal yet has the ability to take control of yourcomputer.
- A Trojan is a computer program that is designed to disturb, steal, or otherwise harm your data or network.

Backdoor Attack

- A backdoor is a malware type that negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to remotely issue system commands and updatemalware.
- A well-known backdoor example is called FinSpy. When installed on a system, it enables the attacker to download and execute files remotely on the system the moment it connects to the internet, irrespective of the system's physical location. It compromises overall system security.

6.Write short notes on Keyloggers

Keyloggers are a form of spyware where users are unaware their actions are being tracked. Keyloggers can be used for a variety of purposes; hackers may use them tomaliciously gain access to your private information, while employers might use them to monitor employee activities.

Spyware is largely invisible software that gathers information about your computeruse, including browsing. Key loggers are a form of spyware that capture every keystroke you type; they can send this information to remote servers, where log-in information--including your passwords--can be extracted and used.

A keylogger is a tool that captures and records a user's keystrokes. It can record instant messages, email, passwords and any other information you type at any timeusing your keyboard. Keyloggers can be hardware or software.

Spyware is any software that installs itself on your computer and starts covertly monitoring your online behavior without your knowledge or permission. Spywareis a kind of malware that secretly gathers information about a person or organization and relays this data to other parties.

There are two common types of keyloggers. Software and Hardware keyloggers.

- Software Keyloggers.
- Hardware Keyloggers.
- Spear Phishing.
- Drive-by-Downloads.
- Trojan Horse.
- 2-Step Verification.
- Install Anti Malware Software.
- Use Key Encryption Software.

Spyware is mostly classified into four types: adware, system monitors, tracking including web tracking, and trojans; examples of other notorious types include digital rights management capabilities that "phone home", keyloggers, rootkits, andweb beacons.

Keystroke logging, often called keylogging, is the practice of noting (or logging) the keys struck on a keyboard, typically in a covert manner so that the person usingthe keyboard is unaware that such actions are being monitored.

Keystroke logger or keylogger is quicker and easier way of capturing the passwords and monitoring the victims' IT savvy behavior. It can be classified assoftware keylogger and hardware keylogger.

1. Software Keyloggers

Software keyloggers are software programs installed on the computer systems which usually are located between the OS and the keyboard hardware, and every keystroke is recorded. Software keyloggers are installed on a computer system byTrojans or viruses without the knowledge of the user. Cybercriminals always install such tools on the insecure computer systems available in public places andcan obtain the required information about the victim very easily. A keylogger

usually consists of two files that get installed in the same directory: a dynamic linklibrary (DLL) file and an EXEcutable (EXE) file that installs the DLL file and triggers it to work. DLL does all the recording of keystrokes.

2. Hardware Keyloggers

To install these keylogers, physical access to the computer system is required. Hardware keyloggers are small hardware devices. These are connected to the PC and/or to the keyboard and save every keystroke into a file or in the memory of thehardware device. Cybercriminals install such devices on ATM machines to captureATM Cards'PINs. Each keypress on the keyboard of the ATM gets registered by these keyloggers These keyloggers look like an integrated part of such systems; hence, bank customers are unaware of their presence.

3. Antikeylogger

Antikeylogger is a tool that can detect the keylogger installed on the computersystem and also can remove the tool.

Advantages of using antikeylogger are as follows:

- Firewalls cannot detect the installations of keyloggers on the systems; hence, antikeyloggers can detect installations of keylogger.
- This software does not require regular updates of signature bases to workeffectively such as other antivirus and antispy programs if not updated, it does not serve the purpose, which makes the users at risk.
- Prevents Internet banking frauds. Passwords can be easily gained with the help of installing keyloggers.
- It prevents ID theft
- It secures E-Mail and instant messaging/chatting.

4. Spywares

Spyware is a type of malware, that is installed on computers which collects information about users without their knowledge. The presence of Spyware is typically hidden, from the user, it is secretly installed on the user's personal computer. Sometimes, however, Spywares such as keyloggers are installed by theowner of a shared, corporate or public computer on purpose to secretly monitor other users.

It is clearly understood from the term Spyware that it secretly monitors the user.The features and functions of such Spywares are beyond simple monitoring.
Spyware programs collect personal information about the victim, such as the Internet surfing habits/patterns and websites visited. The Spyware can also redirect

Internet surfing activities by installing another stealth utility on the users' computersystem. Spyware may also have an ability to change computer settings, which mayresult in slowing of the Internet connection speeds and slowing of response time that may result into user complaining about the Internet speed connection with Internet Service Providers (ISP).

To overcome the emergence of Spywares that proved to be troublesome for the normal user, anti-Spyware softwares are available in the market. Installation of anti-Spyware has become a common element nowadays from computer securitypractices perspective.

6.Write short notes on SQL Injection

An SQL injection is a type of cyber-attack in which a hacker uses a piece of SQL (Structured Query Language) code to manipulate a database and gain access to potentially valuable information ..... Prime examples include notable attacks against Sony Pictures and Microsoft among others.

SQL injection (SQLi) is a type of cyberattack against web applications that use SQL databases such as IBM Db2, Oracle, MySQL, and MariaDB. As the name suggests, the attack involves the injection of malicious SQL statements to interferewith the queries sent by a web application to its database.

Using SQL injection, a hacker will try to enter a specifically crafted SQL commands into a form field instead of the expected information. The intent is tosecure a response from the database that will help the hacker understand the database construction, such as table names.

DoS and DDoS Attack

A denial-of-service (DoS) attack floods a server with traffic, making a website or resource unavailable. A distributed denial-of-service (DDoS) attack is a DoS attackthat uses multiple computers or machines to flood a targeted resource.

A DoS attack is a denial of service attack where a computer is used to flood a server with TCP and UDP packets. A DDoS attack is where multiple systems target a single system with a DoS attack. The targeted network is then bombardedwith packets from multiple locations. All DDoS = DoS but not all DoS = DDoS

What types of resources are targeted by such DoS attacks? Prevents the authorizeduse of networks, systems, or applications with the help of resources such as memory, bandwidth, CPU, system resources, network connectivity, and diskspace.

There are three main types of DoS attacks:

- Application-layer Flood. In this attack type, an attacker simply floods the service with requests from a spoofed IP address in an attempt to slow or crash the service,illustrated in . ...
- Distributed Denial of Service Attacks (DDoS) ...
- Unintended Denial of Service Attacks.

  - **.** DOS Attack **:**
    A DOS attack is a denial of service attack, in this attack a computer sends massive amount of traffic to a victims computer and shuts it down.Dos attack is a online attack which is used to make the website unavailable for its users when done on a website. This attack make the server of a website down which is connected to internet by sending a large number of traffic toit.
  - **2.** DDOS Attack **:**
    In ddos attack means distributed denial of service in this attack dos attacksare done from many different locations using many systems.

DOS	DDOS
DOS Stands for Denial of service attack.	DDOS Stands for Distributed Denial of service attack.
In Dos attack single system targets the victims system.	In DDos multiple system attacks the victims system..
Victim PC is loaded fromthe packet of data sent from a single location.	Victim PC is loaded from the packet ofdata sent from Multiple location.
Dos attack is slower as compared to ddos.	DDos attack is faster than Dos Attack.
Can be blocked easily as only one system is used.	It is difficult to block this attack as multiple devices are sending packets and

DOS	DDOS

	attacking from multiple locations.

In DOS Attack only single device is used with DOS Attack tools.	In DDos attack Bots are used to attack atthe same time.
DOS Attcaks are Easy to trace.	DDOS Attacks are Difficult to trace.
Volume of traffic in Dos attack is less as comparedto DDos.	DDoS attacks allow the attacker to send massive volumes of traffic to the victim network.
Types of DOS Attacks are: 1. Buffer overflow attacks 2. Ping of Death or ICMP flood 3. Teardrop Attack	Types of DDOS Attacks are: 1. Volumetric Attacks 2. Fragmentation Attacks 3. Application Layer Attacks

8.Write short notes on Password Cracking

Password cracking is the process of attempting to gain Unauthorized access to restricted systems using common passwords or algorithms that guess passwords. In other words, it's an art of obtaining the correct password that gives access to a system protected by an authentication method.

Password cracking refers to various measures used to discover computer passwords. This is usually accomplished by recovering passwords from data stored in, or transported from, a computer system. Password cracking is done by either repeatedly guessing the password, usually through a computer algorithm in which the computer tries numerous combinations until the password is successfully discovered.

Password cracking can be done for several reasons, but the most malicious reason is in order to gain unauthorized access to a computer without the computer owner's awareness. This results in cybercrime such as stealing passwords for the purpose ofaccessing banking information.

Other, nonmalicious, reasons for password cracking occur when someone has misplaced or forgotten a password. Another example of nonmalicious password cracking may take place if a system

9.Write short notes on network access control.

Network access control (NAC), also known as network admission control, is the process of restricting unauthorized users and devices from gaining access to a corporate or private network. NAC ensures

that only users who are authenticated and devices that are authorized and compliant with security policies can enter the network.

As endpoints proliferate across an organization—typically driven by bring-your-own-device (BYOD) policies and an expansion in the use of Internet-of-Things (IoT) devices—more control is needed. Even the largest IT organizations do not have the resources to manually configure all the devices in use. The automated features of a NAC solution are a sizable benefit, reducing the time and associated costs with authenticating and authorizing users and determining that their devices are compliant.

Further, cyber criminals are well aware of this increase in endpoint usage and continue to design and launch sophisticated campaigns that exploit any vulnerabilities in corporate networks. With more endpoints, the attack surface increases, which means more opportunities for fraudsters to gain access. NAC solutions can be configured to detect any unusual or suspicious network activity and respond with immediate action, such as isolating the device from the network to prevent the potential spread of the attack.

Although IoT and BYOD have changed NAC solutions, NAC also serves as a perpetual inventory of users, devices, and their level of access. It serves as an active discovery tool to uncover previously unknown devices that may have gained access to all or parts of the network, requiring IT administrators to adjust security policies.

Further, organizations can choose how NAC will authenticate users who attempt to gain access to the network. IT admins can choose multi-factor authentication (MFA), which provides an additional layer of security to username and password combinations.

Restricting network access also means control of the applications and data within the network, which is normally the target of cyber criminals. The stronger the network controls, the more difficult it will be for any cyberattack to infiltrate the network.

Network access control comes with a number of benefits for organizations:

1. Control the users entering the corporate network
2. Control access to the applications and resources users aim to access
3. Allow contractors, partners, and guests to enter the network as needed but restrict their access
4. Segment employees into groups based on their job function and build role-based access policies
5. Protect against cyberattacks by putting in place systems and controls that detect unusual or suspicious activity
6. Automate incident response

7. Generate reports and insights on attempted access across the organization

10.Write short notes on cloud security.

Cloud security is the set of control-based security measures and technology protection, designed to protect online stored resources from leakage, theft, and data loss. Protection includes data from cloud infrastructure, applications, and threats. Security applications uses a software the same as SaaS (Software as a Service) model.

cloud-based security systems benefit the business by:

- Protecting the Business from Dangers
- Protect against internal threats
- Preventing data loss
- Top threats to the system include Malware, Ransomware, and
- Break the Malware and Ransomware attacks
- Malware poses a severe threat to the businesses.

11.Write short notes on web security.

Web security solution should provide comprehensive protection to users against web-related cyber threats. Some of the essential features of a web security solution include:

URL Filtering: Cybercriminals use a variety of known-bad URLs as part of phishing campaigns or to deliver malware. URL filtering makes it possible to block users from visiting these known-bad and other inappropriate sites and to enforce bandwidth limitations on certain types of sites (such as video streaming).

- **Application Control:** Web security solutions perform traffic inspection at the application layer, which means that they have insight into the application generating the traffic and the data that it contains. This granular visibility makes it possible for web administrators to define application-specific rules to ensure that access to applications and sensitive data is properly controlled both inside and outside of the organization.

- **Data Loss Prevention:** Exfiltration of sensitive and proprietary data can occur in a variety of ways and carries significant costs to an organization. Data loss prevention (DLP) solutions monitor data flows to block potential leakages of sensitive and valuable information.**Antivirus:** Malicious websites are a major delivery vector for malware such as ransomware, trojans, and information

stealers. The antivirus built into a web security solution will inspect all traffic flowing through it to determine if it contains known malware samples identified by unique signatures.

- **SSL Introspection:** A growing percentage of web traffic uses HTTPS, which encrypts the traffic to protect it against eavesdropping. SSL introspection allows an organization's security solutions to inspect this encrypted web traffic, enabling them to detect and block malicious content and data exfiltration.

The web can be a dangerous place, and it poses significant risks to an organization and its employees. A web security solution needs a wide range of features to provide effective protection against these threats.

Benefits of Web Security

A web security solution has deep visibility and granular control over Internet-bound traffic. It inspects traffic at the application layer, providing a better understanding of its function and the data that it contains. These capabilities provide a number of benefits to an organization and its employees, such as:

Malicious Content Protection: Web security blocks known-bad phishing sites and drive-by downloads, and inspects web traffic for malicious content. This helps to protect employees against malware and other threats.

- **Data Security:** DLP solutions monitor movement of an organization's sensitive data. This helps to ensure that sensitive and valuable data is not exposed to unauthorized users.
- **Regulatory Compliance:** Companies need to comply with an ever-increasing number of data protection regulations. Web security solutions help with this by providing increased visibility and control for sensitive and protected data within an organization's possession.
- **Improved Network Performance:** Application control enables network administrators to apply application-specific policies. This allows throttling and blocking of certain sites and traffic, improving the network performance for legitimate business traffic.
- **Secure Remote Work:** Web security solutions enable remote employees to work securely from anywhere. Companies can apply and enforce corporate security policies on employee devices regardless of their location.

12.Write short notes on wireless security.

Wireless Network provides various comfort to end users but actually they are very complex in their working. There are many protocols and technologies working behind to provide a stable connection to users. Data packets traveling through wire provide a sense of security to users as data traveling through wire probably not heard by eavesdroppers.

To secure the wireless connection, we should **focus on** the following areas –

- Identify endpoint of wireless network and end-users i.e., Authentication.
- Protecting wireless data packets from middleman i.e., Privacy.
- Keeping the wireless data packets intact i.e., Integrity.

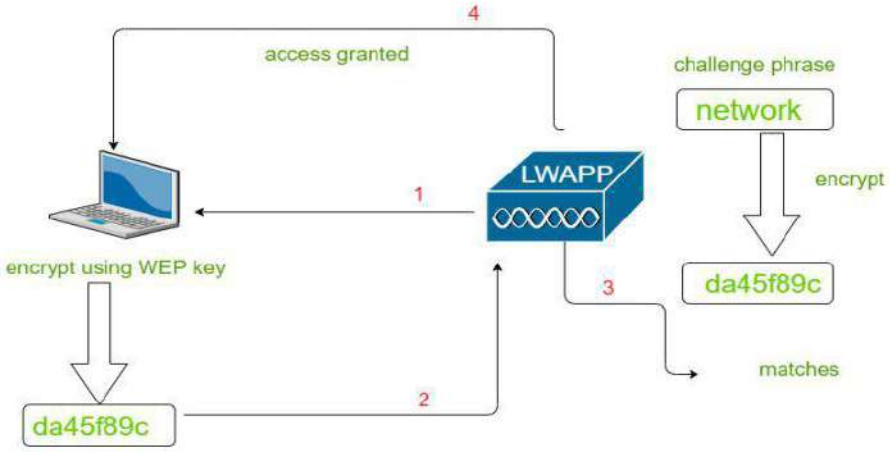
We know that wireless clients form an association with Access Points (AP) and transmit data back and forth over the air. As long as all wireless devices follow 802.11 standards, they all coexist. But all wireless devices are not friendly and trustworthy, some rogue devices may be a threat to wireless security. Rogue devices can steal our important data or can cause the unavailability of the network.

Wireless security is **ensured by** following methods-

- Authentication
- Privacy and Integrity

1. Wired Equivalent Privacy (WEP) :

For wireless data transmitting over the air, open authentication provides no security.
WEP uses the RC4 cipher algorithm for making every frame encrypted. The RC4 cipher also encrypts data at the sender side and decrypt data at the receiving site, using a string of bits as key called WEP key.
WEP key can be used as an authentication method or encryption tool. A client can associate with AP only if it has the correct WEP key. AP tests the knowledge of the WEP key by using a challenge phrase. The client encrypts the phrase with his own key and send back to AP. AP compares the received encrypted frame with his own encrypted phrase. If both matches, access to the association is granted.



Working of WEP Authentication

2. Extensible Authentication Protocol (802.1x/EAP) :

In WEP authentication, authentication of the wireless clients takes place locally at AP. But Scenario gets changed with 802.1x. A dedicated authentication server is added to the infrastructure. There is the participation of three devices –

1. **Supplicant** –
   Device requesting access.
2. **Authenticator** –
   Device that provides access to network usually a Wlan controller (WLC).
3. **Authentication Server** –
   Device that takes client credentials and deny or grant access.

EAP is further of four types with some amendments over each other –

- LEAP
- EAP-FAST
- PEAP
- EAP-TLS