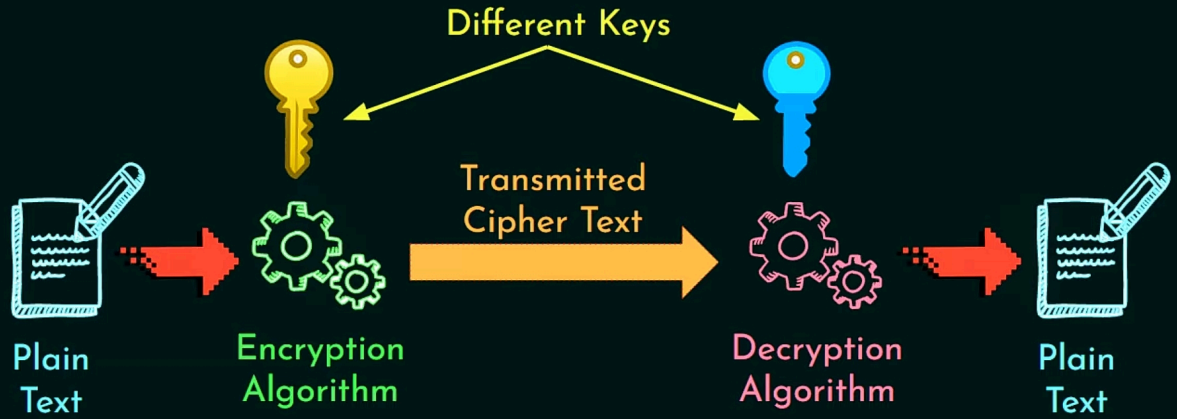
Types of Cryptography

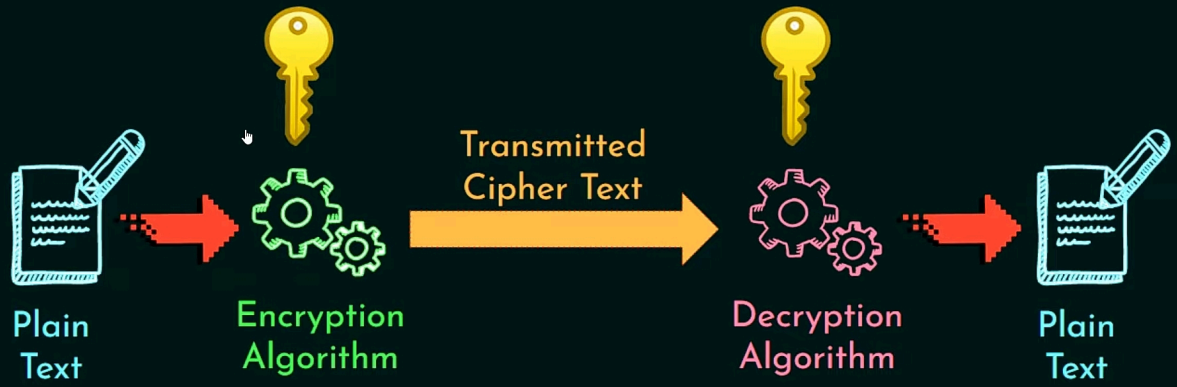
- ★ Symmetric Cryptography (Private Key Cryptography)
- ★ Asymmetric Cryptography (Public Key Cryptography)

Asymmetric Cryptography



Encryption schemes

- ★ Unconditionally secure
- ★ Computationally secure



Computer Security – Definition

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

-

NIST

CIA Triad

- ★ Confidentiality
- ★ Integrity
- ★ Availability

Levels of impact of security breach

- ★ Low
- ★ Medium
- ★ High

CIA Triad

- ★ Confidentiality
- ★ Integrity
- ★ Availability

Additional:

- ★ Authenticity
- ★ Accountability

CIA Triad

- ★ Confidentiality (Example: Account information)
- ★ Integrity (Example: Patient's information)
- ★ Availability (Example: Authentication service)

Threats and Attacks (RFC 2828)

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.

The OSI Security Architecture

Security Attack: Action that compromises the security.

Security Mechanism: Detect, prevent, or recover from a security attack.

Security Service: Enhances the security, counter security attacks, and provide the service.

Security Attacks

- ★ Passive attacks
- ★ Active attacks

Security Services

- ★ Authentication
- ★ Access control
- ★ Data confidentiality
- ★ Data Integrity
- ★ Nonrepudiation

Security Mechanisms

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

Security Attack

★ Action that compromises the security of an individual or an organization.

Types:

1. Passive attack
2. Active attack

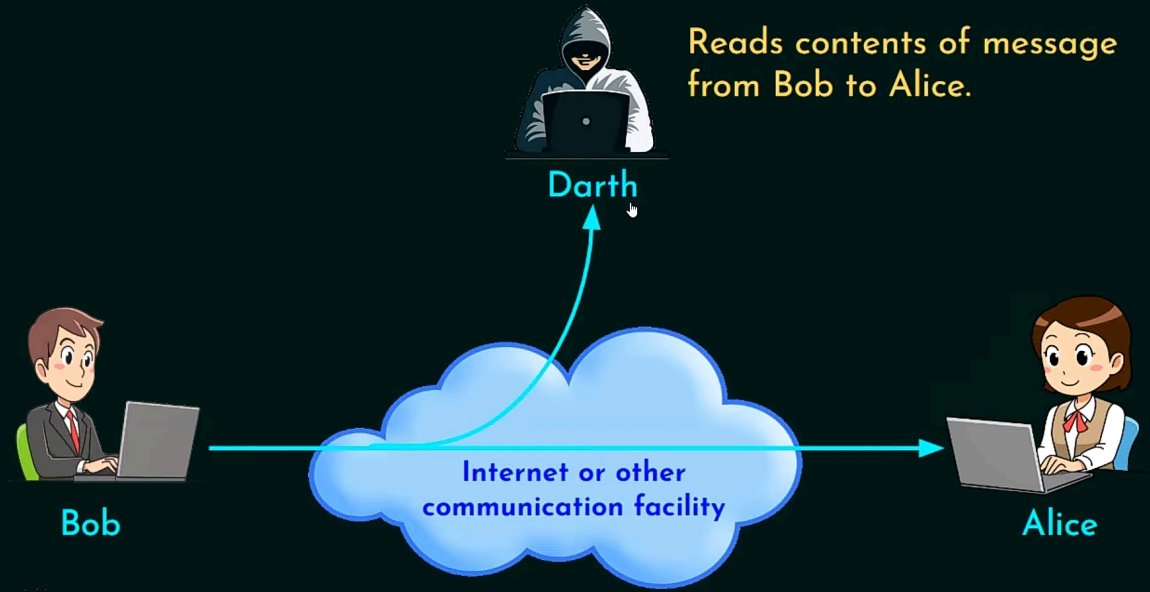
Passive Attack

- ★ Attempts to learn or make use of information from the system.
- ★ Does not affect system resources.
- ★ Eavesdropping or monitoring of transmissions.
- ★ Goal: Obtain information that is being transmitted.

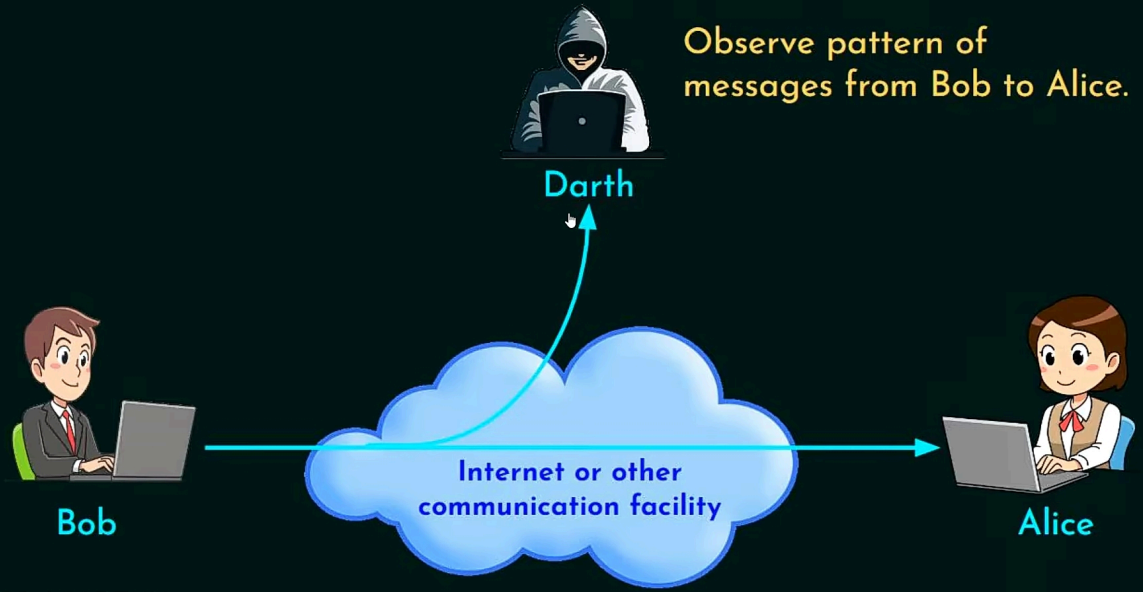
Types:

1. Release of message contents
2. Traffic Analysis

Release of message contents



Traffic Analysis



Active Attack

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

★ Subdivided into four categories

1. Masquerade
2. Replay
3. Modification of messages
4. Denial of Service (DoS).

Masquerade

Darth

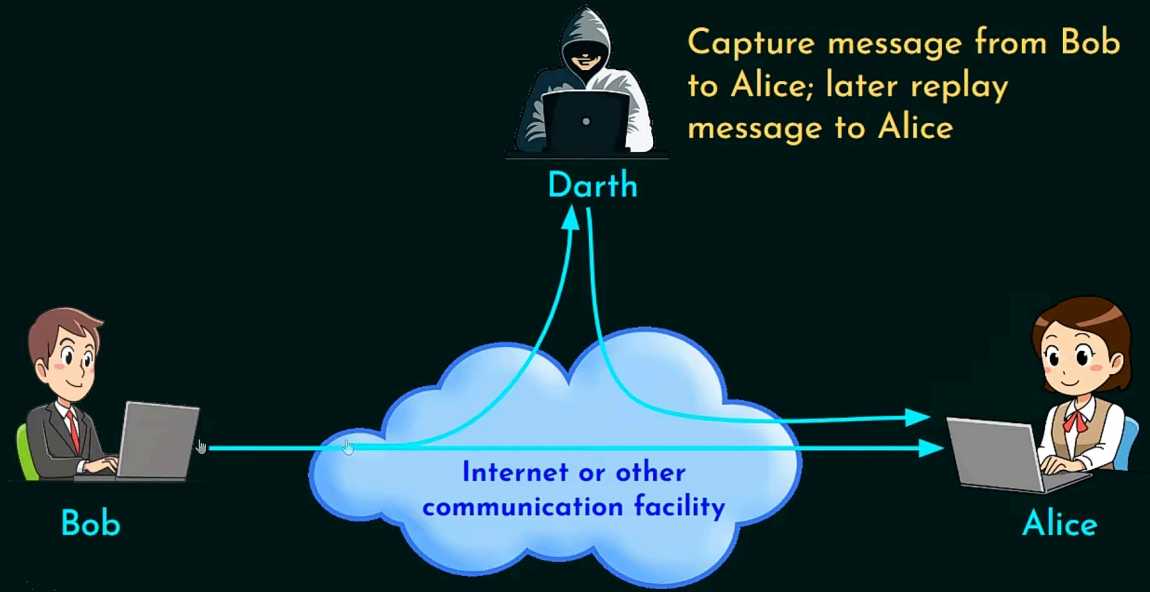
Message from Darth
appears to be from Bob

Bob

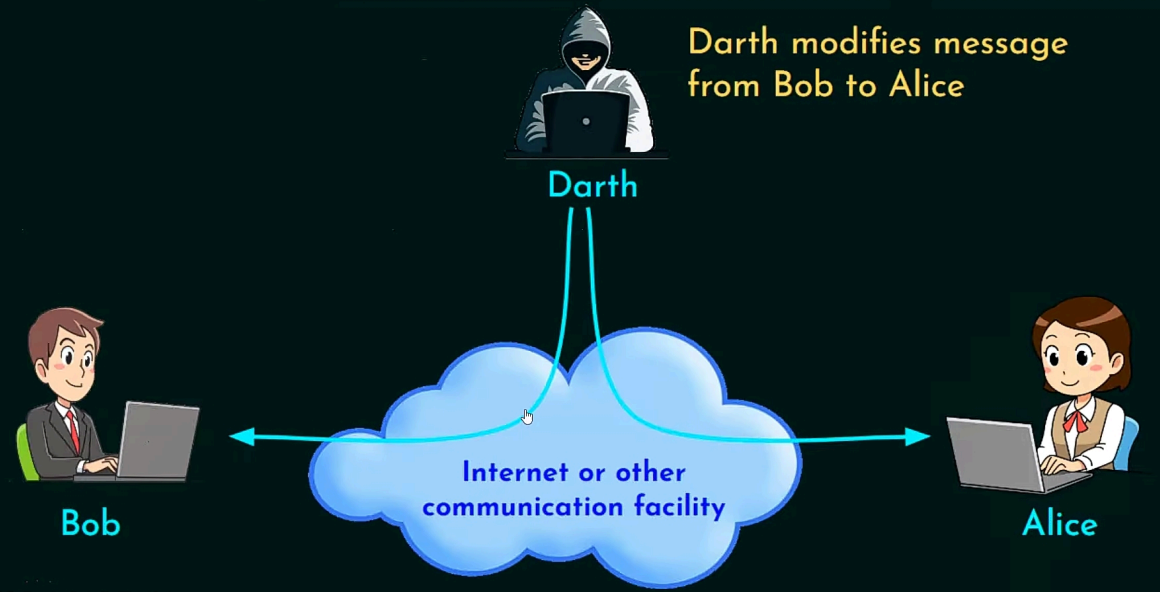
Internet or other
communication facility

Alice

Replay



Modification of message



Denial of Service (DoS)

Darth

Darth Disrupts service provided by server

Bob

Internet or other communication facility

Server

Passive attack Vs Active attack

Passive Attack

- ★ Hard to Detect.
- ★ Neither sender nor receiver is aware of the attack.
- ★ Encryption prevents the success of the passive attacks.
- ★ More emphasis is on prevention than detection.

Active Attack

- ★ Hard to Prevent.
- ★ Difficult to prevent - Physical, software and network vulnerabilities.
- ★ Detect and recover from any disruption or delays.
- ★ If the detection has a deterrent effect, it may also contribute to prevention.

Security Services – Definition

The processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

Security Services

- ★ Authentication
  - Peer entity authentication
  - Data origin authentication

Security Services

- ★ Authentication
  - ○ Peer entity authentication
  - ○ Data origin authentication
- ★ Access control
- ★ Data confidentiality
- ★ Data Integrity
- ★ Nonrepudiation

Security Mechanisms

- ★ Specific security mechanisms
- ★ Pervasive security mechanisms

Specific Security Mechanisms

- ★ Encipherment
- ★ Digital Signature
- ★ Access Control
- ★ Data Integrity
- ★ Authentication Exchange

Specific Security Mechanisms

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

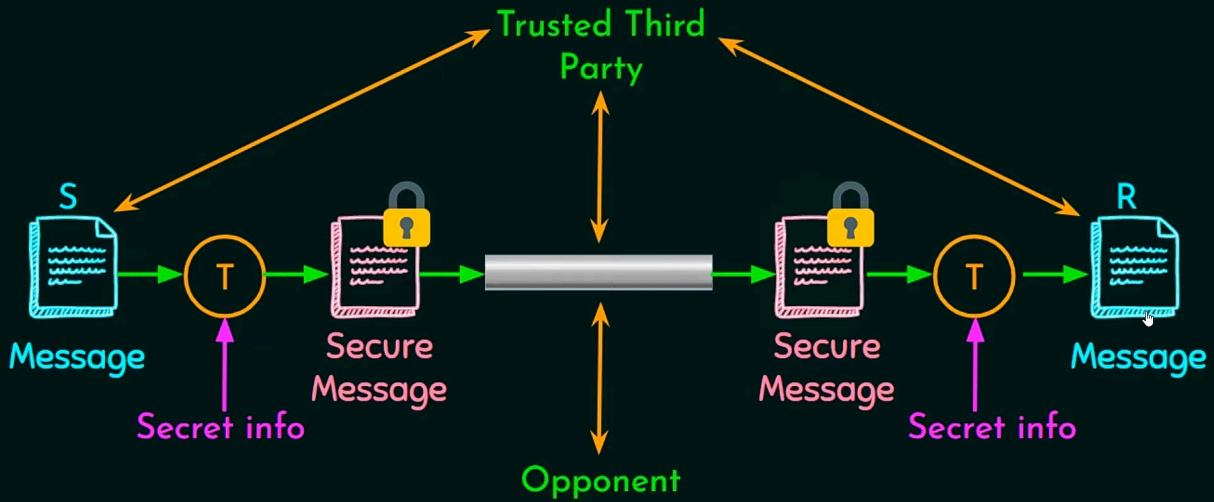
Specific Security Mechanisms

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

Pervasive Security Mechanisms

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

Model for Network Security



Legends used

T - Security related transformation; S - Sender; R - Receiver

Model for Network Security

Four major tasks:

1. Design an algorithm.
2. Generate the secret information.
3. Develop methods for distribution and sharing of information.
4. Specify a protocol.

General approaches

1. Cryptanalysis
2. Brute-force attack

Cryptanalysis

- ★ Cryptanalytic attacks - Based on info know to the cryptanalyst.
- ★ Most difficult : Ciphertext only (Not even encryption algorithm)
- ★ Types of cryptanalytic attacks:
  1. Ciphertext Only
  2. Known Plaintext
  3. Chosen Plaintext
  4. Chosen Ciphertext
  5. Chosen Text

Type of Attack	Known to cryptanalyst
Ciphertext Only	★ Encryption Algorithm ★ Ciphertext
Known Plaintext	★ Encryption Algorithm ★ Ciphertext ★ One or more PT-CT pairs formed with secret key
Chosen Plaintext	★ Encryption Algorithm ★ Ciphertext ★ PT message chosen by cryptanalyst, together with its CT generated with the secret key
Chosen Ciphertext	★ Encryption Algorithm ★ Ciphertext ★ CT chosen by cryptanalyst, together with its corresponding decrypted PT generated with the secret key
Chosen Text	★ Chosen Plaintext and Chosen Ciphertext

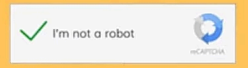
Brute–force attack

- ★ Trying every possible key.
- ★ Until an intelligible translation of the ciphertext into plaintext is obtained.
- ★ Guessing.
- ★ Exhaustive key search.
- ★ Software Tools that can perform brute-force attack.

Aircrack-ng	DaveGrohl	John the ripper
Cain and Abel	Hashcat	Rainbowcrack
Crack	Hydra	Ophcrack

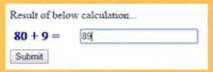
CAPTCHA

Text-based Captcha



ReCAPTCHA

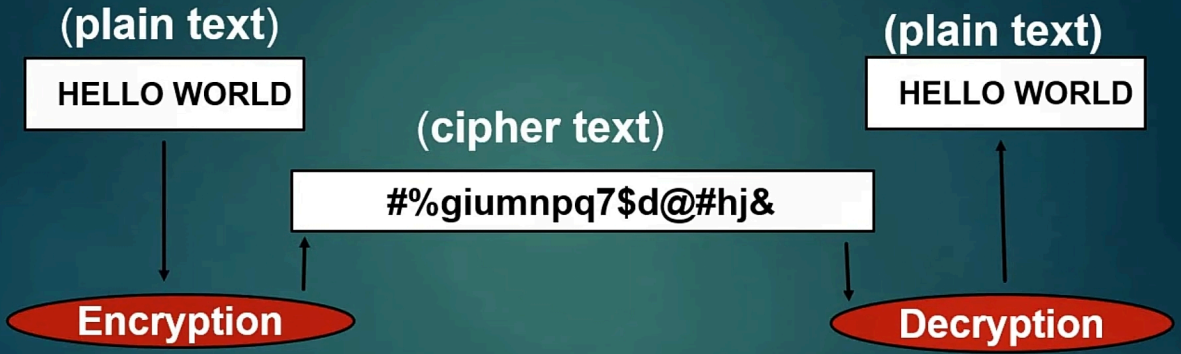
3D Captcha



Mathematical Captcha

Image-based Captcha

CONVENTIONAL SECURITY MODEL



Conventional Encryption

- Conventional encryption requires the translation of plaintext messages into ciphertext messages that can only be decrypted by the intended recipient. A hidden key to be used in encrypting and decrypting is agreed upon by both sender and recipient. The hidden key is usually conveyed by methods of public key encryption.

Figure 1: Flow Diagram

http://wiki.cas.mcmaster.ca/

Classical Encryption Techniques

1. Substitution Technique
2. Transposition Technique

Substitution Technique

★ Letters are replaced by other letters or symbols.

Example:

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 → M

b → X

x → Z

g → A

Plaintext : bag

Ciphertext : XMA

Transposition Technique

- ★ Applying some sort of permutation on the plaintext letters.
- ★ Plaintext: NESO
- ★ Ciphertext: ESON, SONE, ONES, ENOS . . . .

Classical Encryption Technique

Substitution	Transposition
❖ Caesar Cipher	❖ Rail Fence
❖ Monoalphabetic Cipher	❖ Row Column Transposition
❖ Playfair Cipher	
❖ Hill Cipher	
❖ Polyalphabetic Cipher	
❖ One-Time Pad	

Steganography

- ★ Conceal the existence of the message.
- ★ Hiding the message.
- ★ Not an encryption scheme.
- ★ Cryptography renders the message unintelligible to outsiders by various transformations of the text.
- ★ Example: Simply encrypt correct reading exactly twice.

Steganography

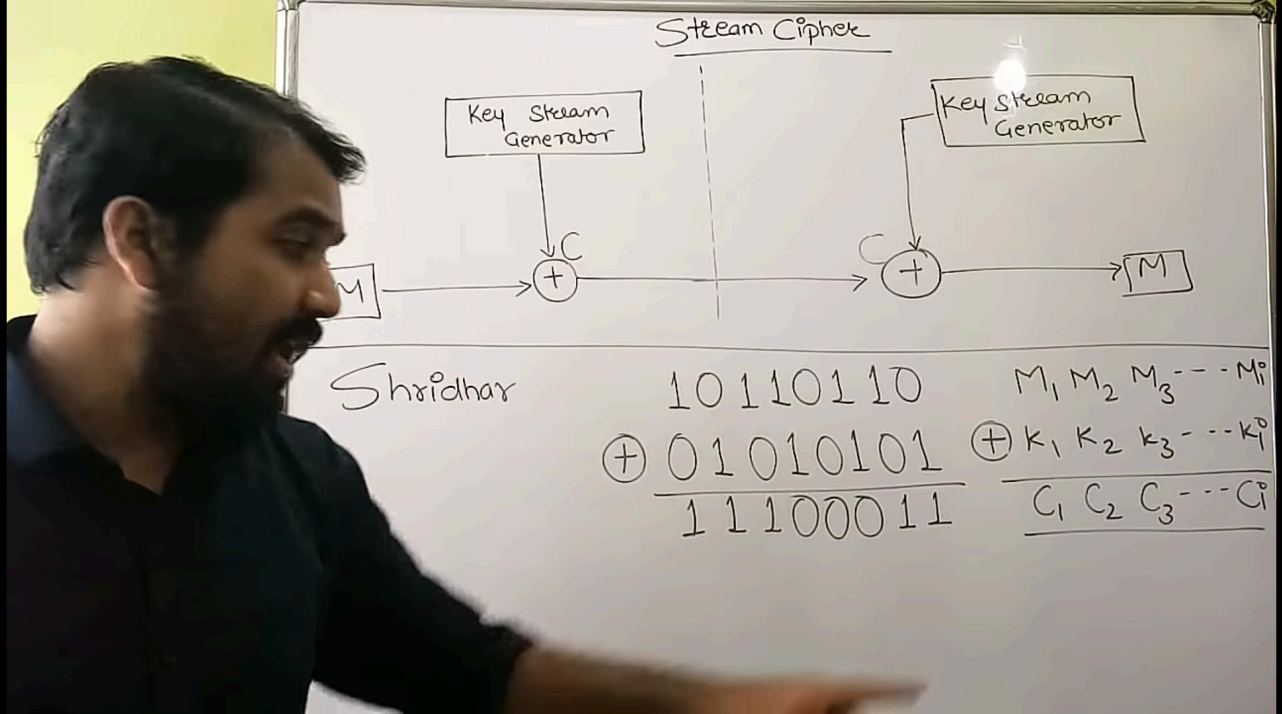
- ★ Character marking.
- ★ Invisible ink.
- ★ Pin punctures.
- ★ Typewriter color ribbon.

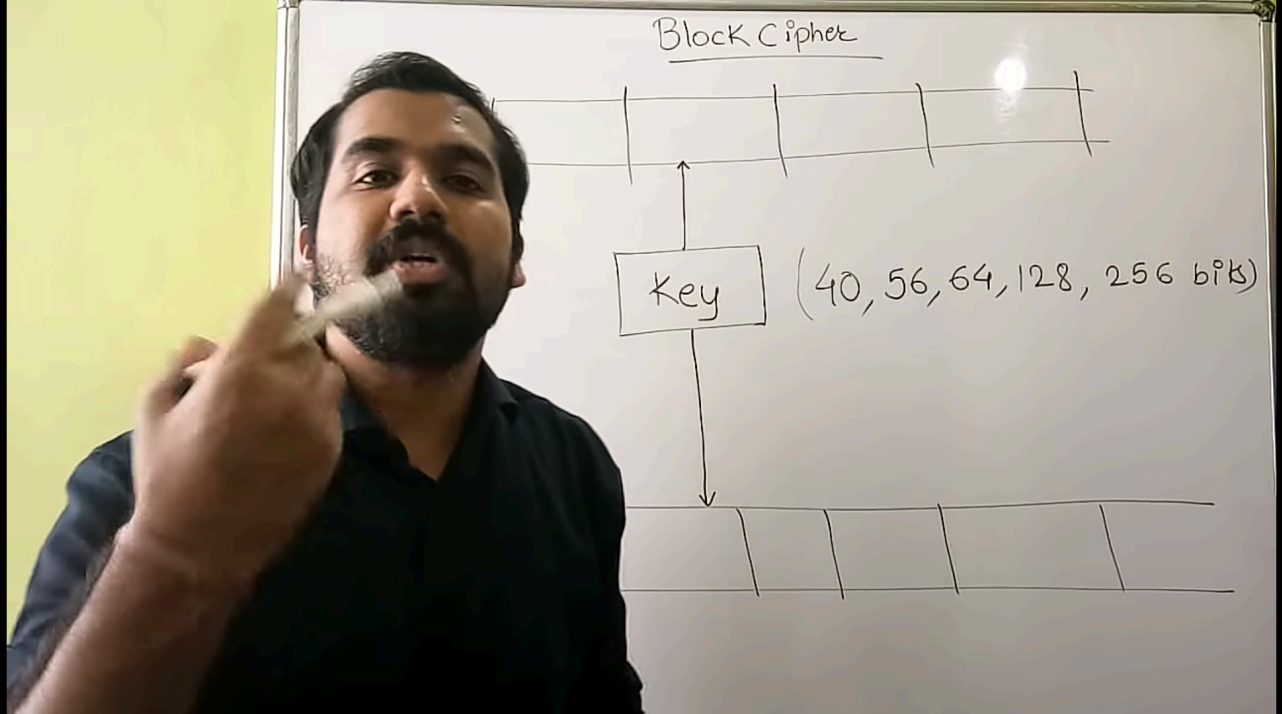
Drawbacks

- ★ Lot of overhead.
- ★ Once the system is discovered, it becomes virtually worthless.

Note:

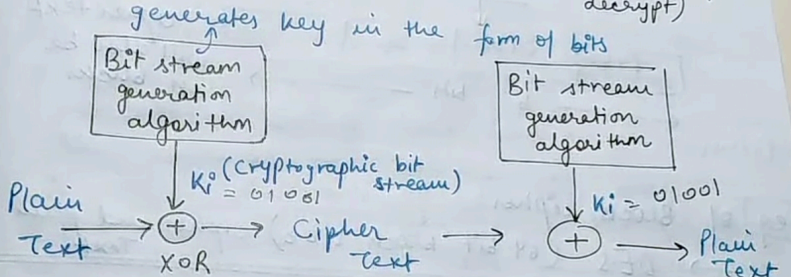
Alternatively, a message can be first encrypted and then hidden using steganography.


Stream Cipher
Key Stream Generator
Key Stream Generator
M
C
C
M
Shridhar
10110110
⊕ 01010101
11100011
M1 M2 M3 --- Mi
⊕ K1 K2 K3 --- Ki
C1 C2 C3 --- Ci


Block Cipher
Key
(40, 56, 64, 128, 256 bits)

stream one bit or 1 byte at a time.

* It is a symmetric key cipher. (ie 1 key for encrypt + decrypt)

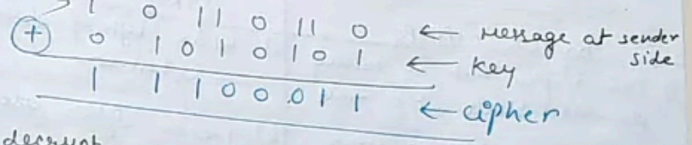


eg

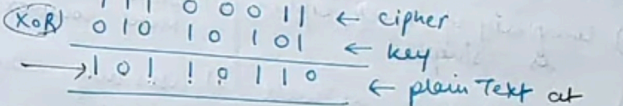
	1	0	1	1	0	1	1	0	← message at sender side
⊕	0	1	0	1	0	0	1	1	← key
	1	1	1	0	0	1	0	1	← cipher

To decrypt,

	1	1	1	0	0	1	0	1	← cipher
XOR	0	1	0	1	0	0	1	1	← key



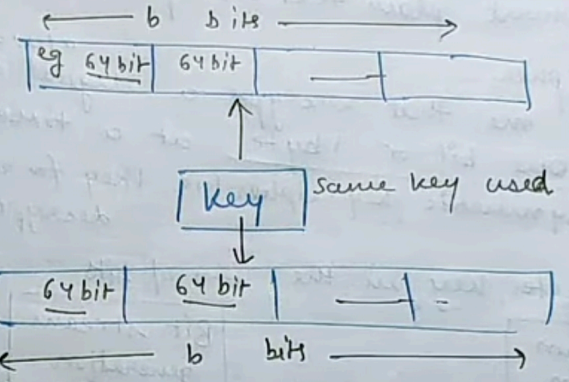
To decrypt,



2) Block cipher

* In this, a block of plain text is treated as a whole and used to produce the ciphertext of equal length.
* Typically, a block size of 64 and 128 bits is used.
* symmetric key cipher (1 key used only).

* key will be applied on each block.



eg of Block cipher

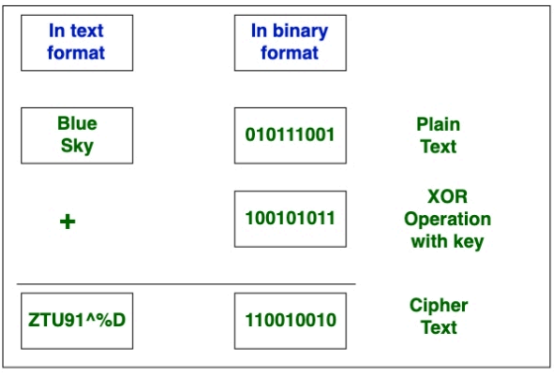
→ DES (64 bit block size) cipher and plain Text.

Difference between Block Cipher and Stream Cipher

Prerequisite – Block cipher modes of operation

Block Cipher and **Stream Cipher** belongs to the symmetric key cipher. These two block ciphers and stream cipher are the methods used for converting the plain text into ciphertext.

The main difference between a **Block cipher** and a **Stream cipher** is that a block cipher converts the plain text into cipher text by taking plain text's block at a time. While stream cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.



Stream Cipher

Let's see the difference between them:

S.NO	Block Cipher	Stream Cipher
1.	Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.
2.	Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
3.	The complexity of block cipher is simple.	While stream cipher is more complex.
4.	Block cipher Uses confusion as well as diffusion.	While stream cipher uses only confusion.
5.	In block cipher, reverse encrypted text is hard.	While in-stream cipher, reverse encrypted text is easy.
6.	The algorithm modes which are used in block cipher are ECB (Electronic Code Book) and CBC (Cipher Block Chaining).	The algorithm modes which are used in stream cipher are CFB (Cipher Feedback) and OFB (Output Feedback).
7.	Block cipher works on transposition techniques like rail-fence technique, columnar transposition technique, etc.	While stream cipher works on substitution techniques like Caesar cipher, polygram substitution cipher, etc.
8.	Block cipher is slow as compared to a stream cipher.	While stream cipher is fast in comparison to block cipher.

A modern block cipher is a cipher which encrypts m-bit block of plaintext and decrypts m-bit block of ciphertext. For encryption or decryption, modern block cipher facilitate a K bit key and the decryption algorithm should be inverse of encryption algorithms and for both encryption and decryption similar key is used.

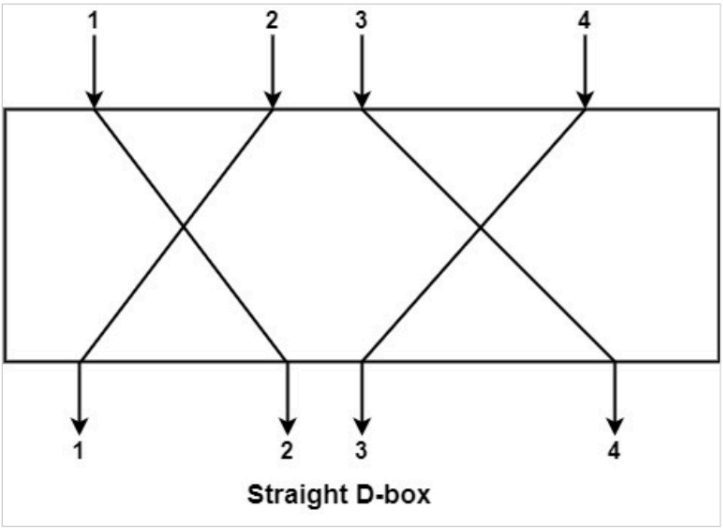
A block cipher works on a plaintext block of n bits to make a cipher text block of n bits. There are possible multiple plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be applicable), each should create a unique cipher text block. Such transformation is known as reversible, or non-singular.

Block cipher modes of operation have been produced to delete the chance of encrypting identical blocks of text the similar method, the ciphertext formed from the previous encrypted block is used to the next block. A block of bits is known as an initialization vector (IV).

There are various components of Modern Block Cipher which are as follows −

- **D-boxes** − A D-box is a permutation box having similar features as traditional transposition ciphers. D-boxes transpose bits. There are three types of D-boxes which are as follows −

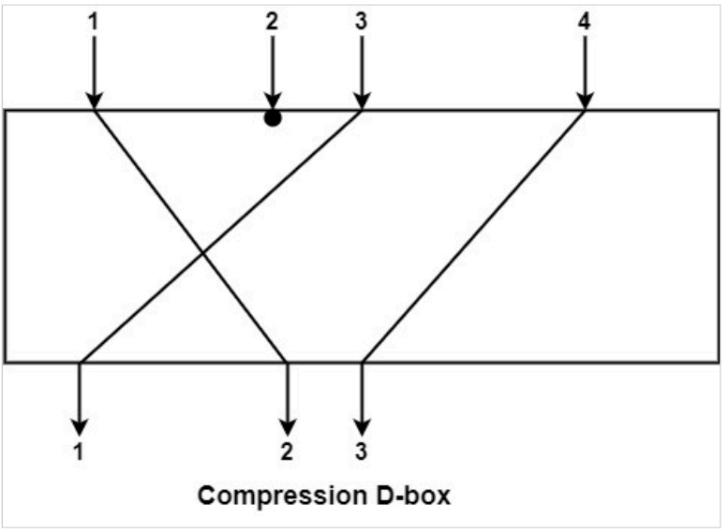
Straight D-box − It creates n inputs, permutes them and supports n outputs. In this, the second input after permutation is the first to be outputted. The first letter in input is permuted to second place, third on fourth place and fourth on third place. There are n! Possible way of mapping in D-box.



Straight D-box

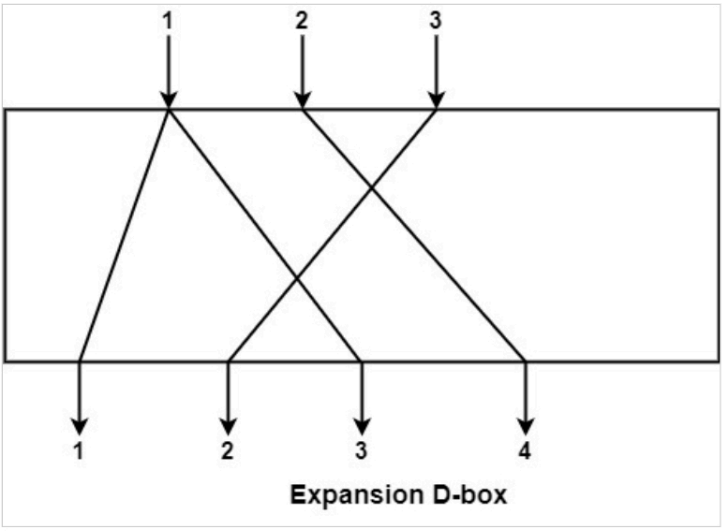
Compression D-box − This is a D-box with n inputs and m outputs, where m<n. There are various inputs are blocked and do not reach the output. Compression D-boxes are used when it is required to permute bits and at the similar time reduce the number of bits for the next stage.

Compression D-box – This is a D-box with n inputs and m outputs, where m<n. There are various inputs are blocked and do not reach the output. Compression D-boxes are used when it is required to permute bits and at the similar time reduce the number of bits for the next stage.



Compression D-box

Expansion D-box – This is a D-box with n inputs and m outputs, where m >n i.e., there are various inputs are connected to more than one output it is used when it is required to transpose bits and the same increase the multiple bits for the next stage.



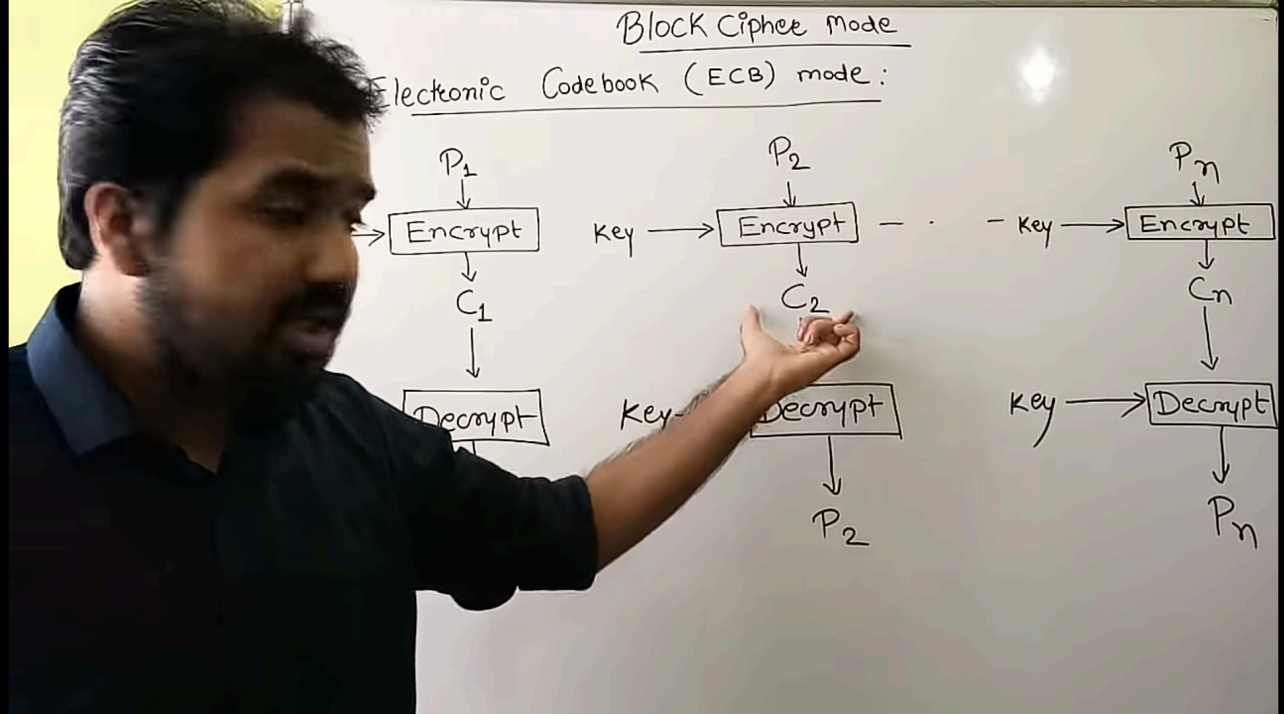
Expansion D-box

- **S-boxes** – These are substitution boxes same to the substitution cipher. The input to an S-box can be a n-bit word but the output can be a m-bit word, where m and n are not essentially the same.
- **Circular Shift** – It can also discovered in modern block ciphers, it can be such as leftshift or right-shift. In the circular left shift, shift each bit in n-bit word with m position to the left and the leftmost m-bits are deleted from the left and become the rightmost bits.

Ginni

Block Ciphee mode

Electronic Codebook (ECB) mode:



Block Cipher Design Principles

Block ciphers are built in the Feistel cipher structure. Block cipher has a specific number of rounds and keys for generating ciphertext. For defining the complexity level of an algorithm few design principles are to be considered.

These are explained as following below :

1. **Number of Rounds** –
   The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.
2. **Design of function F** –
   The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity.

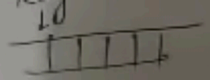
   To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.
3. **Key schedule algorithm** –
   In Feistel Block cipher structure, each round would generate a sub-key for increasing the complexity of cryptanalysis. The Avalanche effect makes it more complex in deriving sub-key. Decryption must be done very carefully to get the actual output as the avalanche effect is present in it.

Article Tags : Computer Networks Network-security

Recommended Articles
1. Difference between Block Cipher and Transposition Cipher
2. Difference between Block Cipher and Stream Cipher
3. Difference between Monoalphabetic Cipher and Polyalphabetic Cipher
4. Difference between Substitution Cipher Technique and Transposition Cipher Technique
5. Block Cipher modes of Operation
6. Firewall Design Principles
7. Design Principles of Security in Distributed System
8. Mobile Forensics - Definition, Uses, and Principles
9. Cryptography and Network Security Principles
10. Principles of Network Applications
11. Transforming a Plain Text message to Cipher Text
12. Vernam Cipher in Cryptography
13. Bifid Cipher in Cryptography
14. Symmetric Cipher Model

Block ciphers Principles:

(design principles)



1. Number of Rounds — 10R, 16R, 20R → harder
2. Design of function F ✓ $f(x) = ax+b$ → Non-linear
3. key schedule Algorithm abcd abc

Block ciphers → modes of operation

ECB, CBC, CFB, OFB, CTR modes

Block ciphers Algorithms

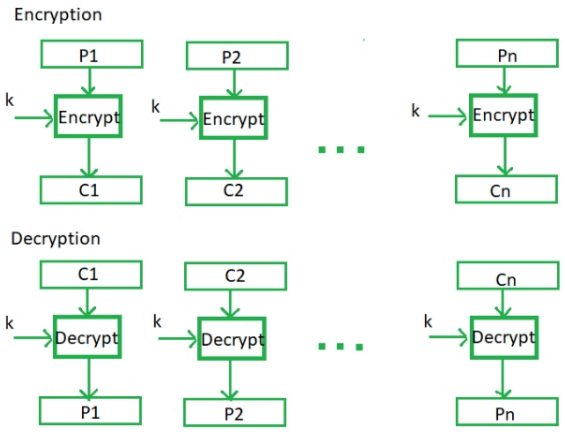
1. DES
2. AES
3. Blowfish

Block Cipher modes of Operation

Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher. **Block cipher** is an encryption algorithm that takes a fixed size of input say *b* bits and produces a ciphertext of *b* bits again. If the input is larger than *b* bits it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

Electronic Code Book (ECB) –
Electronic code book is the easiest block cipher mode of functioning. It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Generally, if a message is larger than *b* bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

Procedure of ECB is illustrated below:



Advantages of using ECB –

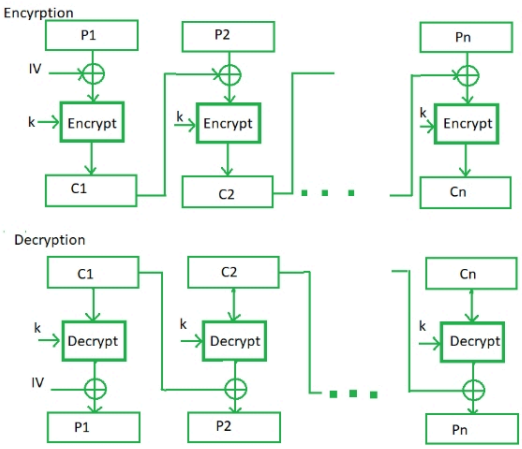
- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

Cipher Block Chaining –
Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

The process is illustrated here:



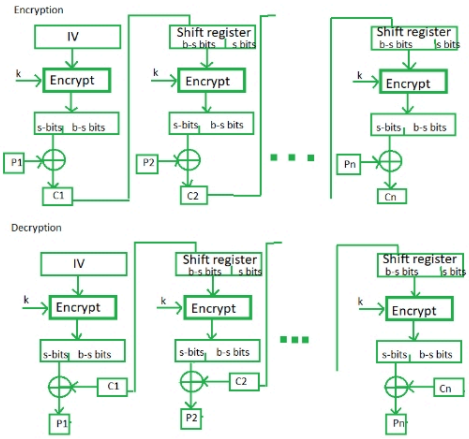
Advantages of CBC –

- CBC works well for input greater than *b* bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.

Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.

Cipher Feedback Mode (CFB) –
In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits.The left-hand side *s* bits are selected along with plaintext bits to which an XOR operation is applied. The result is given as input to a shift register having b-s bits to lhs,s bits to rhs and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithms.



Advantages of CFB –

- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

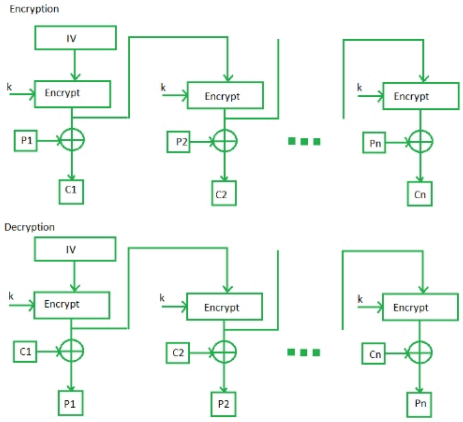
Advantages of CFB –

- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

Disadvantages of using ECB –

- The drawbacks of CFB are the same as those of CBC mode. Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

Output Feedback Mode –
The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases the dependency or relationship of the cipher on the plaintext.



Advantages of OFB –

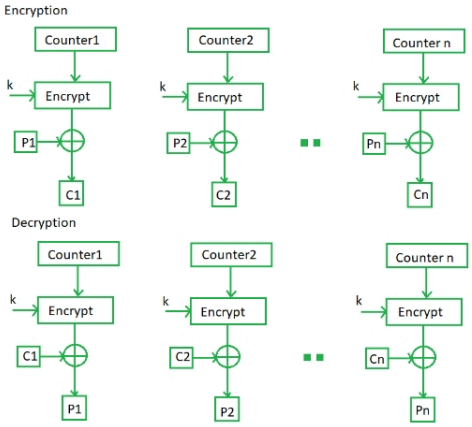
- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.

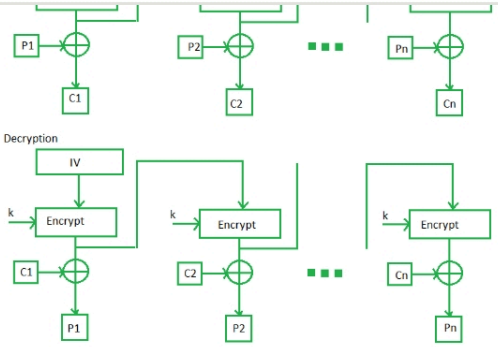
Disadvantages of OFB-

- The drawback of OFB is that, because to its operational modes, it is more susceptible to a message stream modification attack than CFB.

Counter Mode –
The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block. The CTR mode is independent of feedback use and thus can be implemented in parallel.

Its simple implementation is shown below:





Advantages of OFB –

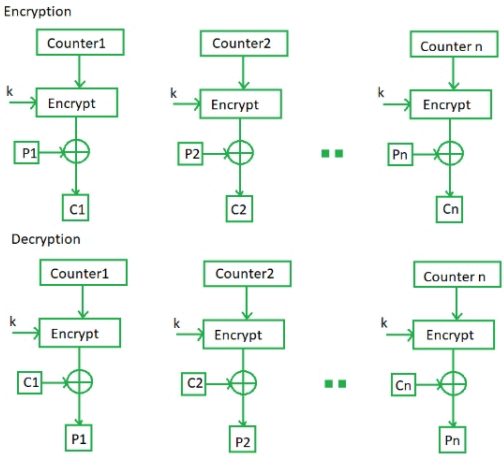
- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.

Disadvantages of OFB-

- The drawback of OFB is that, because to its operational modes, it is more susceptible to message stream modification attack than CFB.

Counter Mode –
The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block. The CTR mode is independent of feedback use and thu can be implemented in parallel.

Its simple implementation is shown below:



Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.
- Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

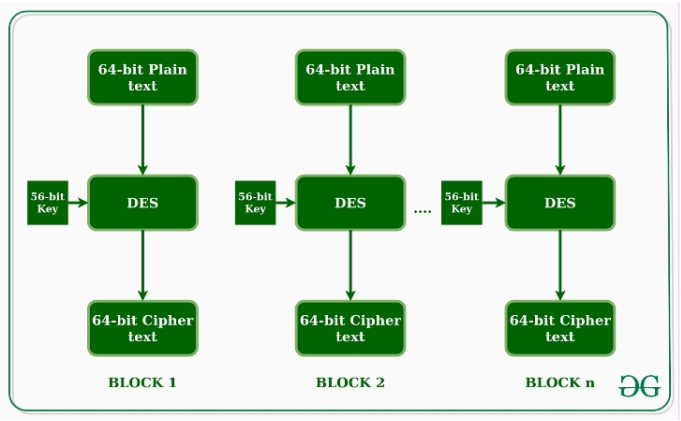
Disadvantages of Counter-

- The fact that CTR mode requires a synchronous counter at both the transmitter and the receiver is a severe drawback. The recovery of plaintext is erroneous when synchronisation is lost.

Data encryption standard (DES) | Set 1

Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of **64 bits** each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is **56 bits**.

The basic idea is shown in the figure:



We have mentioned that DES uses a 56-bit key. Actually, The initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

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

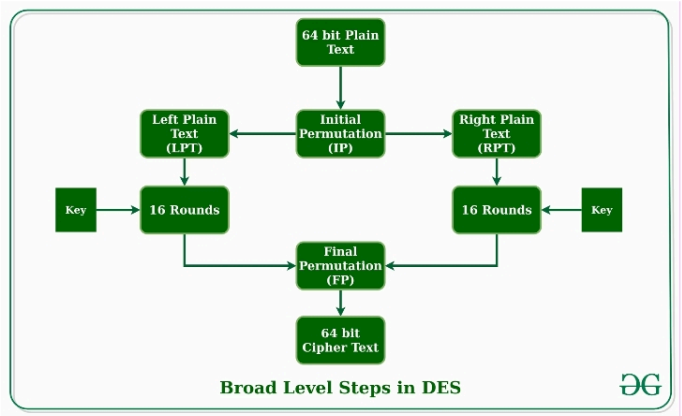
Figure - discording of every 8th bit of original key

Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.

DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition.

Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.
DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

- In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
- The initial permutation is performed on plain text.
- Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
- Now each LPT and RPT go through 16 rounds of the encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
- The result of this process produces 64-bit ciphertext.



Broad Level Steps in DES

Initial Permutation (IP):

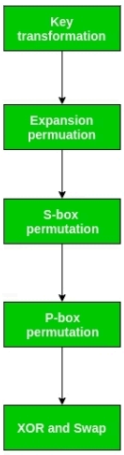
As we have noted, the initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed, as shown in the figure. For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block, and so on.

This is nothing but jugglery of bit positions of the original plain text block. the same rule applies to all the other bit positions shown in the figure.

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	33	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Figure - Initial permutation table

As we have noted after IP is done, the resulting 64-bit permuted text block is divided into two half blocks. Each half-block consists of 32 bits, and each of the 16 rounds, in turn, consists of the broad-level steps outlined in the figure.



Step-1: Key transformation:

We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation. For this, the 56-bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

For example: if the round numbers 1, 2, 9, or 16 the shift is done by only one position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is shown in the figure.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure - number of key bits shifted per round

After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table , we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation.

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

Figure - compression permutation

Because of this compression permutation technique, a different subset of key bits is used in each round. That makes DES not easy to crack.

Step-2: Expansion Permutation:

Recall that after the initial permutation, we had two 32-bit plain text areas called Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called expansion permutation. This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.

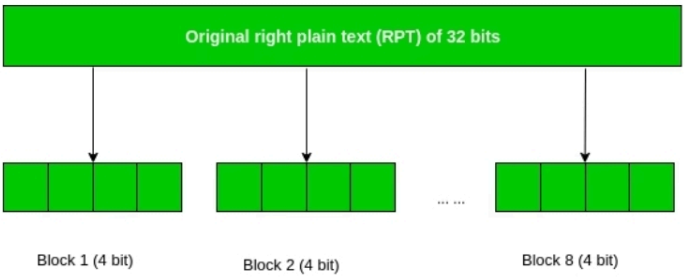


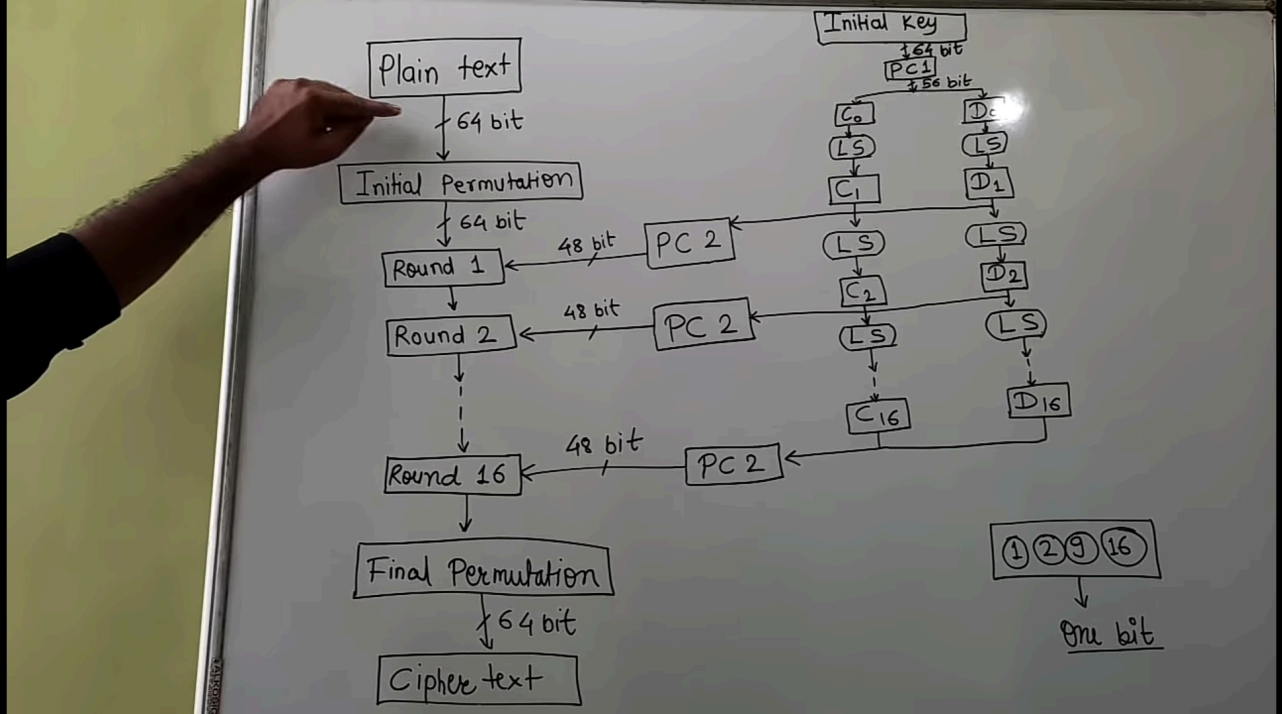
Figure - division of 32 bit RPT into 8 bit blocks

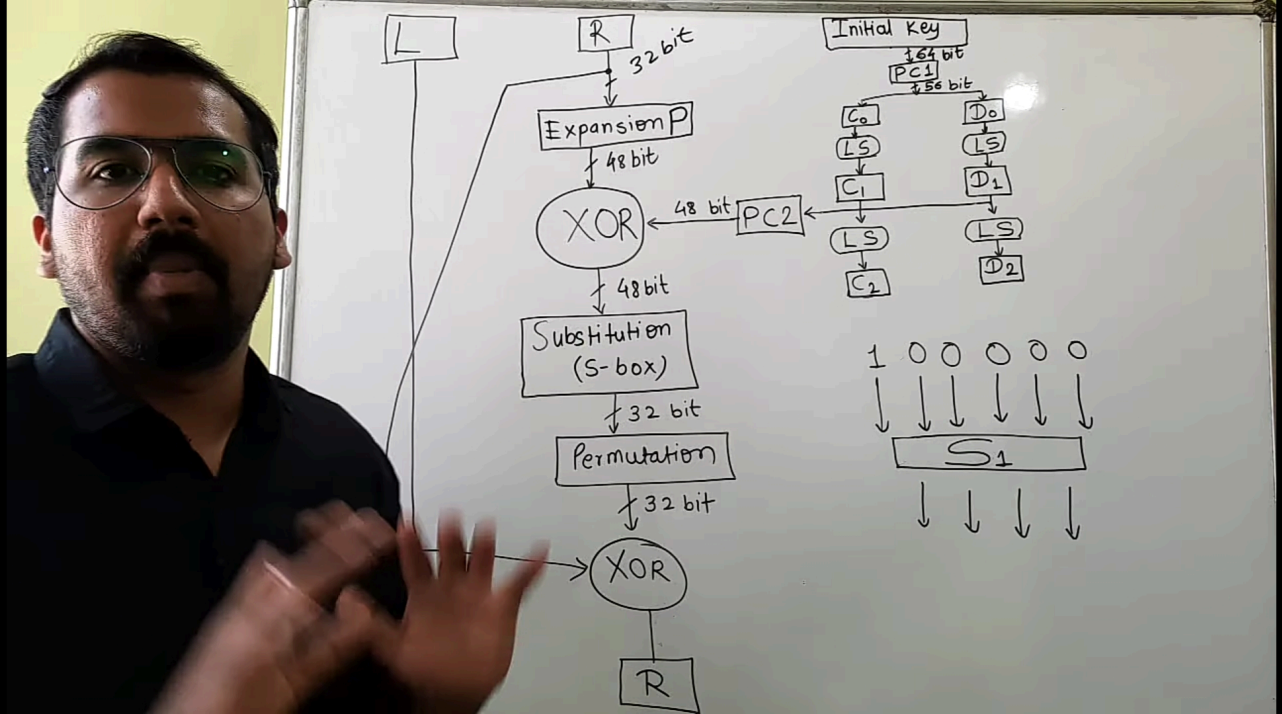
This process results in expansion as well as a permutation of the input bit while creating output. The key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the **32-bit RPT** to **48-bits**. Now the 48-bit key is XOR with 48-bit RPT and the resulting output is given to the next step, which is the **S-Box substitution**.

Javascript | Python

```
// Define DES key and plaintext
const key = "0123456789abcdef";
const plaintext = "Hello, world!";
// Perform DES encryption
const des = new DES(key);
const ciphertext = des.encrypt(plaintext);
// Perform DES decryption
const decrypted = des.decrypt(ciphertext);
// Print results
console.log("Plaintext: ", plaintext);
console.log("Ciphertext: ", ciphertext);
console.log("Decrypted: ", decrypted);
// Define DES class
class DES {
  constructor(key) {
    // Initialize DES with key
    this.key = CryptoJS.enc.Hex.parse(key);
  }

  encrypt(plaintext) {
```


Plain text
64 bit
Initial Permutation
64 bit
Round 1
48 bit
PC 2
Round 2
48 bit
PC 2
Round 16
48 bit
PC 2
Final Permutation
64 bit
Cipher text
Initial Key
64 bit
PC1
56 bit
C_0
D_0
LS
LS
C_1
D_1
LS
LS
C_2
D_2
LS
LS
C_{16}
D_{16}
1 2 9 16
One bit


L
R
32 bit
Expansion P
48 bit
XOR
48 bit
PC2
48 bit
Substitution
(S-box)
32 bit
Permutation
32 bit
XOR
R
Initial Key
64 bit
PC1
56 bit
C0
D0
LS
LS
C1
D1
LS
LS
C2
D2
1 0 0 0 0 0
S1

Strength of Data encryption standard (DES)

Data encryption standard (DES) is a symmetric key block cipher algorithm. The algorithm is based on Feistel network. The algorithm uses a 56-bit key to encrypt data in 64-bit blocks.

There are mainly two categories of concerns about the strength of Data encryption standard. They are:

1. Concerns about the particular algorithm used.
2. Concerns about the usage of key of size 56-bit.

The first concern regarding the algorithm used addresses the possibility of cryptanalysis by making use of the DES algorithm characteristics. A more severe concern is about the length of secret key used. There can be 2^{56} (approximately 7.2×10^{16} keys) possible keys with a key length of 56 bits. Thus, a brute force attack appears to be impractical.

Assuming that on an average one has to search half the key space, to break the cipher text, a system performing one DES encryption per microsecond might require more than thousand years. But, the assumption of one DES encryption per microsecond is too conservative. In July 1998, DES was finally proved to be insecure when the Electronic Frontier Foundation (EFF) had broken a DES encryption. The encryption was broken with the help of a special-purpose "DES cracker" machine. It was reported that the attack took less than 3 days.

Simply running through all possible keys won't result in cracking the DES encryption. Unless known plain text is given, the attacker must be able to differentiate the plain text from other data. Some degree of knowledge about the target plain text and some techniques for automatically distinguishing plain text from garble are required to supplement the brute-force approach. If brute force attack is the only means to crack the DES encryption algorithm, then using longer keys will obviously help us to counter such attacks. An algorithm is guaranteed unbreakable by brute force if a 128- bit key is used.

The differential cryptanalysis, linear cryptanalysis, are examples for statistical attacks on DES algorithm. Few of the important alternatives for DES are AES (Advanced Encryption Standard) and triple DES.

Article Tags : Computer Networks | GATE CS | cryptography | Network-security

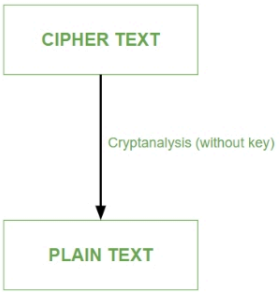
Recommended Articles

1. Data encryption standard (DES) | Set 1
2. Double DES and Triple DES
3. Difference between Software Encryption and Hardware Encryption
4. Simplified Data Encryption Standard Key Generation
5. Simplified Data Encryption Standard | Set 2
6. Advanced Encryption Standard (AES)
7. Difference between AES and DES ciphers
8. Difference Between Data Encryption and Data Compression
9. Granovetter's Strength of Weak Ties in Social Networks
10. Simplified International Data Encryption Algorithm (IDEA)
11. What is Data Encryption?
12. RC4 Encryption Algorithm
13. XOR Encryption by Shifting Plaintext
14. RC5 Encryption Algorithm
15. Image encryption using cellular automata
16. Evolution of Malwares from Encryption to Metamorphism

Differential and Linear Cryptanalysis

Cryptanalysis is the process of transforming or decoding communications from non-readable to readable format without having access to the real key. OR we may say it is the technique of retrieving the plain text of the communication without having access to the key. *Cryptoanalysis is the art, science, or practice of decrypting encrypted messages.* The secret key used for encryption and decoding is considered to be unknown to the cryptologists, mathematicians, and other scientists participating in the process. In contrast to a brute force attack, this form of analysis seeks vulnerabilities in a cryptosystem.

Cryptanalysis frequently comprises a direct evaluation of the cryptosystem in use, which is essentially an advanced concentrated mathematical attempt at decryption utilizing knowledge about the encryption scheme that is already available. They can employ intercepted encrypted messages (ciphertext), intercepted complete, partial, likely, or similar original messages (plaintext), or information (encrypted or original) that is known to be used adaptively in subsequent trials.



Process of cryptanalysis

Cryptanalysis is used to break cryptographic security systems and gain access to the contents of the encrypted messages, even if the cryptographic key is unknown.

Types of Cryptanalytic Attacks:

1. **Ciphertext only attack:**

1. In this type of cryptanalytic attack, the attacker has the knowledge of only the ciphertext.
2. The attacker has to detect the plain text using the ciphertext only.
3. This type of attack is not very easy to be implemented.

2. **Known plain text only attack:**

1. In this type of cryptanalytic attack, the attacker has the knowledge of some plain text as well as ciphertext.
2. The attacker tries to decrypt the messages using these two.
3. This type of attack is somewhat easy to implement.

Different Forms of Cryptanalysis:

Cryptanalysis basically has two forms:

Different Forms of Cryptanalysis:

Cryptanalysis basically has two forms:

1. Linear Cryptanalysis:

Linear cryptanalysis is a general type of cryptanalysis based on discovering affine approximations to a cipher's action in cryptography. Block and stream ciphers have both been subjected to attacks. Linear cryptanalysis is one of the two most common attacks against block ciphers, with differential cryptanalysis being the other.

2. Differential Cryptanalysis:

Differential cryptanalysis is a sort of cryptanalysis that may be used to decrypt both block and stream ciphers, as well as cryptographic hash functions. In the widest sense, it is the study of how alterations in information intake might impact the following difference at the output. In the context of a block cipher, it refers to a collection of strategies for tracking differences across a network of transformations, finding where the cipher displays non-random behavior, and using such attributes to recover the secret key (cryptography key).

Difference between Linear Cryptanalysis and Differential Cryptanalysis

S. No.	Linear Cryptanalysis	Differential Cryptanalysis
1.	Linear cryptanalysis was basically invented by Matsui and Yamagishi in the year 1992.	Differential cryptanalysis was first defined in the year 1990 by Eli Biham and Adi Shamir.
2.	Linear cryptanalysis always works on a single bit (one bit at a time).	Differential cryptanalysis can work on multiple bits at a time.
3.	In the case of Linear cryptanalysis, ciphertext attack is a very big disadvantage.	In the case of differential cryptanalysis plain text attack is a very big disadvantage.
4.	The use of linear cryptanalysis is to figure out what is the linear relationship present between some plaintext bits, ciphertext bits, and unknown key bits very easily.	The use of differential cryptanalysis is to get clues about some critical bits, reducing the need for an extensive search.
5.	Subsets of input attributes refer to the internal structures of a single input.	The underlying structure of each individual input is unimportant in this case since the input attributes are differential.
6.	The cryptanalyst decrypts each ciphertext using all available subkeys and analyses the resultant intermediate ciphertext to determine the random outcome for one encryption cycle.	After several encryption rounds, Cryptanalyst analyses the changes in the intermediate ciphertext obtained. The practice of combining assaults is known as differential linear cryptanalysis.
7.	Any random plaintext is selected in Linear Cryptanalysis.	Plaintext is Carefully chosen in Differential Cryptanalysis.
8.	Plaintext is used one by one in linear Cryptanalysis.	Plaintext is used in pairs in Differential Cryptanalysis.
9.	Complexity of attack is low in linear Cryptanalysis.	Complexity of attack is High in Differential Cryptanalysis
10.	Mathematical relation between plaintexts used has Linear approximation (such as a series of XOR operations).	Mathematical relation between plaintexts used has Specific differences (such as XOR).
11.	Goal of the attack is to identify the linear relation between some bits of the plaintext, some bits of the cipher text and some bits of the unknown key.	Goal of the attack is to Identify some bits of the unknown key.

Article Tags : Computer Networks | Computer Subject | Geeks Premier League | Geeks-Premier-League-2022

Simplified International Data Encryption Algorithm (IDEA)

In cryptography, block ciphers are very important in the designing of many cryptographic algorithms and are widely used to encrypt the bulk of data in chunks. By chunks, it means that the cipher takes a fixed size of the plaintext in the encryption process and generates a fixed size ciphertext using a fixed-length key. An algorithm's strength is determined by its key length.

The Simplified **International Data Encryption Algorithm (IDEA)** is a **symmetric key block** cipher that:

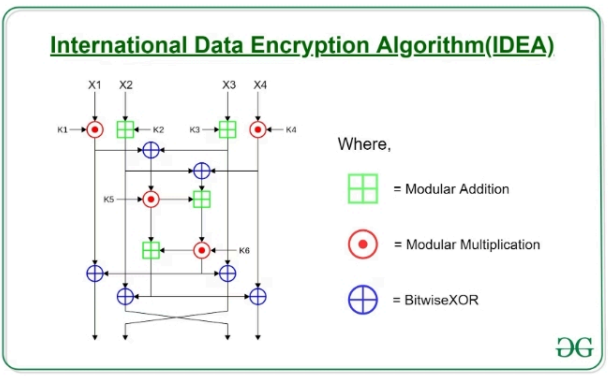
- uses a fixed-length plaintext of **16 bits** and
- encrypts them in **4 chunks of 4 bits** each
- to produce **16 bits ciphertext**.
- The length of the key used is **32 bits**.
- The key is also divided into 8 blocks of 4 bits each.

This algorithm involves a series of 4 identical complete rounds and 1 half-round. Each complete round involves a series of 14 steps that includes operations like:

- Bitwise XOR
- Addition modulo (2^4)
- Multiplication modulo $(2^4)_{+1}$

After 4 complete rounds, the final "half-round" consists of only the first 4 out of the 14 steps previously used in the full rounds. To perform these rounds, each binary notation must be converted to its equivalent decimal notation, perform the operation and the result obtained should be converted back to the binary representation for the final result of that particular step.

Key Schedule: 6 subkeys of 4 bits out of the 8 subkeys are used in each complete round, while 4 are used in the half-round. So, 4.5 rounds require 28 subkeys. The given key, 'K', directly gives the first 8 subkeys. By rotating the main key left by 6 bits between each group of 8, further groups of 8 subkeys are created, implying less than one rotation per round for the key (3 rotations).



	K1	K2	K3	K4	K5	K6
Round 1	1101	1100	0110	1111	0011	1111
Round 2	0101	1001*	0001	1011	1100	1111
Round 3	1101	0110	0111	0111*	1111	0011
Round 4	1111	0101	1001	1101	1100	0110*
Round 4.5	1111	1101	0110	0111		

* denotes a shift of bits

Notations used in the 14 steps:

Symbol	Operation
*	Multiplication modulo $(2^4)_{+1}$
+	Addition modulo (2^4)
^	Bitwise XOR

The 16-bit plaintext can be represented as **X1** || **X2** || **X3** || **X4**, each of size 4 bits. The 32-bit key is broken into 8 subkeys denoted as K1 || K2 || K3 || K4 || K5 || K6 || K7 || K8, again of size 4 bits each. Each round of 14 steps uses the three algebraic operation- Addition modulo (2^4), Multiplication modulo (2^4)+1 and Bitwise XOR. The steps involved are as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4
5. Step 1 ^ Step 3
6. Step 2 ^ Step 4
7. Step 5 * K5
8. Step 6 + Step 7
9. Step 8 * K6
10. Step 7 + Step 9
11. Step 1 ^ Step 9
12. Step 3 ^ Step 9
13. Step 2 ^ Step 10
14. Step 4 ^ Step 10

The input to the next round is Step 11 || Step 13 || Step 12 || Step 14, which becomes X1 || X2 || X3 || X4. This swap between 12 and 13 takes place after each complete round, except the last complete round (4th round), where the input to the final half round is Step 11 || Step 12 || Step 13 || Step 14.

After last complete round, the half-round is as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4

The final output is obtained by concatenating the blocks.

Example:

```
Key: 1101 1100 0110 1111 0011 1111 0101 1001
Plaintext: 1001 1100 1010 1100
Ciphertext: 1011 1011 0100 1011
```

Explanation:
The explanation is only for 1st complete round (the remaining can be implemented similarly) and the last half-round.

- **Round 1:**
  - From the plaintext: **X1 – 1001, X2 – 1100, X3 – 1010, X4 – 1100**
  - From the table above: **K1 – 1101, K2 – 1100, K3 – 0110, K4 – 1111, K5 – 0011, K6 – 1111**
  -

```
(1001(9) * 1101(13))(mod 17) = 1111(15)
(1100(12) + 1100(12))(mod 16) = 1000(8)
(1010(10) + 0110(6))(mod 16) = 0000(0)
(1100(12) * 1111(15))(mod 17) = 1010(10)
(1111(15) ^ 0000(0)) = 1111(15)
(1000(8) ^ 1010(10)) = 0010(2)
(1111(15) * 0011(3))(mod 17) = 1011(11)
(0010(2) + 1011(11))(mod 16) = 1101(13)
(1101(13) * 1111(15))(mod 17) = 1000(8)
(1011(11) + 1000(8))(mod 16) = 0011(3)
(1000(8) ^ 1111(15)) = 0111(7)
(1000(8) ^ 0000(0)) = 1000(8)
(0011(3) ^ 1000(8)) = 1011(11)
(0011(3) ^ 1010(10)) = 1001(9)
```

- **Round 1 Output**: 0111 1011 1000 1001 *(Step 12 and Step 13 results are interchanged)*
- **Round 2:**
  - From Round 1 output: **X1 – 0111, X2 – 1011, X3 – 1000, X4 – 1001**
  - From the table above: **K1 – 0101, K2 – 1001, K3 – 0001, K4 – 1011, K5 – 1100, K6 – 1111**
  - **Round 2 Output**: 0110 0110 1110 1100 *(Step 12 and Step 13 results are interchanged)*
- **Round 3:**
  - From Round 2 Output: **X1 – 0110, X2 – 0110, X3 – 1110, X4 – 1100**
  - From the table above: **K1 – 1101, K2 – 0110, K3 – 0111, K4 – 0111, K5 – 1111, K6 – 0011**
  - **Round 3 Output**: 0100 1110 1011 0010 *(Step 12 and Step 13 results are interchanged)*
- **Round 4:**
  - From Round 3 Output: **X1 – 0100, X2 – 1110, X3 – 1011, X4 – 0010**
  - From the table above: **K1 – 1111, K2 – 0101, K3 – 1001, K4 – 1101, K5 – 1100, K6 – 0110**
  - **Round 4 Output**: 0011 1110 1110 0100 *(Step 12 and Step 13 results are interchanged)*
- **Round 4.5:**
  - From Round 4 Output: **X1 – 0011, X2 – 1110, X3 – 1110, X4 – 0100**
  - From the table above: **K1 – 1111, K2 – 1101, K3 – 0110, K4 – 0111**
  - **Round 4.5 Output**: 1011 1011 0100 1011 *(Step 2 and Step 3 results are **not** interchanged)*
  -

```
(0011(3) * 1111(15))(mod 17) = 1011(11)
(1110(14) + 1101(13))(mod 16) = 1011(11)
(1110(14) + 0110(6))(mod 16) = 0100(4)
(0100(4) * 0111(7))(mod 17) = 1011(11)
```

- **Final Ciphertext** is **1011 1011 0100 1011**

NOTE: For every round except the final transformation, a swap occurs, and the input is given to the next round

generated.

Encryption in IDEA

IDEA derives most of its security from multiple interleaved mathematical operations:

- modular addition
- modular multiplication
- bitwise exclusive-OR (XOR)

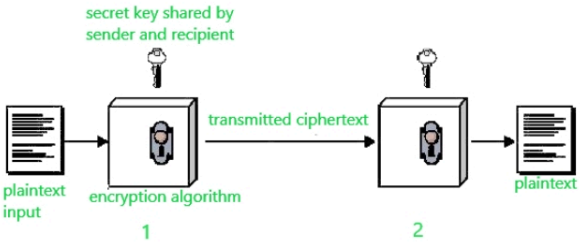
By using a 128-bit key, IDEA encrypts a 64-bit block of plaintext into a 64-bit block of ciphertext. One process partitions the plaintext block into four 16-bit subblocks for each of the eight complete rounds, namely X1, X2, X3 and X4.

Another process produces six 16-bit key subblocks for each of the encryption rounds, namely Z1, Z2, Z3, Z4, Z5 and Z6. For subsequent output transformation, a further four 16-bit key subblocks are required. Thus, from a 128-bit key, a total of 52 16-bit

Conventional Encryption

Conventional encryption is a cryptographic system that uses the same key used by the sender to encrypt the message and by the receiver to decrypt the message. It was the only type of encryption in use prior to the development of public-key encryption.

It is still much preferred of the two types of encryption systems due to its simplicity. It is a relatively fast process since it uses a single key for both encryption and decryption In this encryption model, the sender encrypts plaintext using the receiver's secret key, which can be later used by the receiver to decrypt the ciphertext. Below is a figure that illustrates this concept.



Suppose A wants to send a message to B, that message is called plaintext. Now, to avoid hackers reading plaintext, the plaintext is encrypted using an algorithm and a secret key (at 1). This encrypted plaintext is called ciphertext. Using the same secret key and encryption algorithm run in reverse(at 2), B can get plaintext of A, and thus the message is read and security is maintained.

The idea that uses in this technique is very old and that's why this model is called conventional encryption.

Conventional encryption has mainly 5 ingredients :

1. **Plain text** –
   It is the original data that is given to the algorithm as an input.

2. **Encryption algorithm** –
   This encryption algorithm performs various transformations on plain text to convert it into ciphertext.

3. **Secret key** –
   The secret key is also an input to the algorithm. The encryption algorithm will produce different outputs based on the keys used at that time.

4. Ciphertext –
   It contains encrypted information because it contains a form of original plaintext that is unreadable by a human or computer without proper cipher to decrypt it. It is output from the algorithm.

5. **Decryption algorithm** –

Suppose A wants to send a message to B, that message is called plaintext. Now, to avoid hackers reading plaintext, the plaintext is encrypted using an algorithm and a secret key (at 1). This encrypted plaintext is called ciphertext. Using the same secret key and encryption algorithm run in reverse(at 2), B can get plaintext of A, and thus the message is read and security is maintained.

The idea that uses in this technique is very old and that's why this model is called conventional encryption.

Conventional encryption has mainly 5 ingredients :

1. **Plain text** –
   It is the original data that is given to the algorithm as an input.

2. **Encryption algorithm** –
   This encryption algorithm performs various transformations on plain text to convert it into ciphertext.

3. **Secret key** –
   The secret key is also an input to the algorithm. The encryption algorithm will produce different outputs based on the keys used at that time.

4. Ciphertext –
   It contains encrypted information because it contains a form of original plaintext that is unreadable by a human or computer without proper cipher to decrypt it. It is output from the algorithm.

5. **Decryption algorithm** –
   This is used to run encryption algorithms in reverse. Ciphertext and Secret key is input here and it produces plain text as output.

Requirements for secure use of conventional encryption :

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

Advantages of Conventional Encryption :

1. **Simple** –
   This type of encryption is easy to carry out.

2. **Uses fewer computer resources** –
   Conventional encryption does not require a lot of computer resources when compared to public-key encryption.

3. **Fast** –
   Conventional encryption is much faster than asymmetric key encryption.

Disadvantages of Conventional Encryption Model:

1. Origin and authenticity of the message cannot be guaranteed, since both sender and receiver use the same key, messages cannot be verified to have come from a particular user.
2. It isn't much secured when compared to public-key encryption.
3. If the receiver lost the key, he/she cant decrypt the message and thus making the whole process useless.
4. This scheme does not scale well to a large number of users because both the sender and the receiver have to agree on a secret key before transmission.

Article Tags : Computer Networks | Technical Scripter 2020

Recommended Articles

1. Difference between Software Encryption and Hardware Encryption
2. RC4 Encryption Algorithm

Knowledge about the number and length of messages between nodes may enable an opponent to determine who is taking to whom. This can have obvious implications in a military conflict. Even in commercial applications, traffic analysis may yield information that the traffic generators would like to conceal.

The following types of information that can be derived from a traffic analysis attack: Identities of partners, how frequent the partners are communicating, message pattern. Message length, or quantity of messages that suggest important information is being exchanged, and the events that correlate with special conversation between particular partners.

With the use of traffic patterns a covert channel can be established. A covert channel is a means of communication which transfers information unintended by the designers of the communication facility. The channel is used to transfer information in a way that violates a security policy. An employee may wish to communicate information to an outsider through a secured channel evading detection by the management. The two participants could setup a code in which an apparently legitimate message of a less than a certain length represents binary zero, whereas a longer message represents a binary one. Other such schemes are possible.

Network layers are encrypted, reducing the opportunity for traffic analysis. It is still possible in those circumstances for an attacker to assess the amount of traffic on a network and to observe the amount of traffic entering and leaving each end system. Countermeasure to this type of attack is traffic padding.

Traffic padding produces cipher text output continuously, even in the absence of plain text. A continuous random data stream is generated. When plain text is available, it is encrypted and transmitted. When input plaintext is not present, random data are encrypted and transmitted. This makes it impossible for an attacker to distinguish between true data flow and padding and therefore impossible to deduce the amount of traffic.
Traffic padding is essentially a link encryption function. If only end-to-end encryption is employed, then the measures available to the defender are more limited. If encryption is implemented at the application layer, then an opponent can determine transport layer, network-layer addresses and traffic patterns which remain accessible.

Quick Note: Taking the Nonsense out of looking for the right spyware remover

If you really want to take the work out of looking for that right Spyware Protection from a go to the Internet and get a or a Free
Download, In order to prevent your vital information from being ripped from your computer get your Remover Today.

Null messages can be inserted randomly into the stream. These tactics deny opponent knowledge about the amount of data exchanged between end users and obscure the underlying traffic pattern. Encryption can secure network connections to a larger extent.

Key Management in Cryptography

In cryptography, it is a very tedious task to distribute the public and private keys between sender and receiver. If the key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are two aspects for Key Management:

1. Distribution of public keys.
2. Use of public-key encryption to distribute secrets.

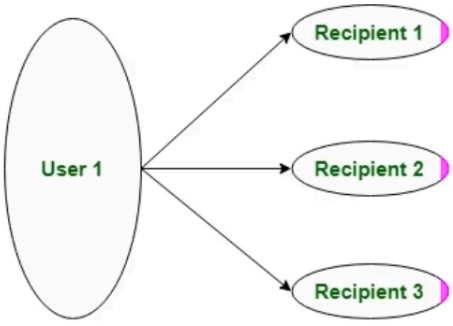
Distribution of Public Key:

The public key can be distributed in four ways:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates.

These are explained as following below:

1. Public Announcement: Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.



Public Key Announcement

2. Publicly Available Directory: In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

3. Public Key Authority: It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

4. Public Certification: This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority's public key.

Key Management in Cryptography

In cryptography, it is a very tedious task to distribute the public and private keys between sender and receiver. If the key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are two aspects for Key Management:

1. Distribution of public keys.
2. Use of public-key encryption to distribute secrets.

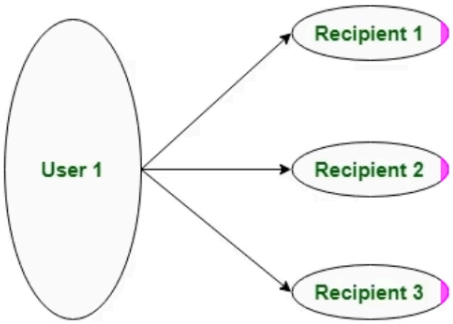
Distribution of Public Key:

The public key can be distributed in four ways:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates.

These are explained as following below:

1. Public Announcement: Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.



Public Key Announcement

2. Publicly Available Directory: In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

3. Public Key Authority: It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

4. Public Certification: This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority's public key.
First sender and receiver both request CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communication.

Article Tags : Computer Networks cryptography

Pseudo Random Number Generator (PRNG)

Pseudo Random Number Generator(PRNG) refers to an algorithm that uses mathematical formulas to produce sequences of random numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers. A PRNG starts from an arbitrary starting state using a **seed state**. Many numbers are generated in a short time and can also be reproduced later, if the starting point in the sequence is known. Hence, the numbers are **deterministic and efficient**.

Why do we need PRNG?

With the advent of computers, programmers recognized the need for a means of introducing randomness into a computer program. However, surprising as it may seem, it is difficult to get a computer to do something by chance as computer follows the given instructions blindly and is therefore completely predictable. It is not possible to generate truly random numbers from deterministic thing like computers so PRNG is a technique developed to generate random numbers using a computer.

How PRNG works?

Linear Congruential Generator is most common and oldest algorithm for generating pseudo-randomized numbers. The generator is defined by the recurrence relation:

$X_{n+1} = (aX_n + c) \bmod m$

where X is the sequence of pseudo-random values

m, $0 < m$ - modulus

a, $0 < a < m$ - multiplier

c, $0 \leq c < m$ - increment

x_0, $0 \leq x_0 < m$ - the seed or start value

We generate the next random integer using the previous random integer, the integer constants, and the integer modulus. To get started, the algorithm requires an initial Seed, which must be provided by some means. The appearance of randomness is provided by performing **modulo arithmetic**..

Characteristics of PRNG

- **Efficient:** PRNG can produce many numbers in a short time and is advantageous for applications that need many numbers
- **Deterministic:** A given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known.Determinism is handy if you need to replay the same sequence of numbers again at a later stage.
- **Periodic:** PRNGs are periodic, which means that the sequence will eventually repeat itself. While periodicity is hardly ever a desirable characteristic, modern PRNGs have a

Group and Abelian Group

Property			Explanation
Abelian Group	Group	A1 - Closure	a, b ∈ G, then (a • b) ∈ G.
		A2 - Associative	a • (b • c) = (a • b) • c for all a, b, c ∈ G.
		A3 - Identity element	(a • e) = (e • a) = a for all a, e ∈ G.
		A4 - Inverse element	(a • a′) = (a′ • a) = e for all a, a′ ∈ G.
	A5 - Commutative		(a • b) = (b • a) for all a, b ∈ G.

Rings

A ring R denoted by {R, +, *}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c ∈ R the following axioms are obeyed:

- Group (A1-A4), Abelian Group(A5).
- Closure under multiplication (M1): If a, b ∈ R then ab ∈ R
- Associativity of multiplication (M2): a (bc) = (ab) c for all a, b, c ∈ R
- Distributive laws (M3) :

  a (b + c) = ab + ac for all a, b, c ∈ R

  (a + b) c = ac + bc for all a, b, c ∈ R

Rings

A ring R denoted by {R, +, *}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c ∈ R the following axioms are obeyed:

- Group (A1-A4), Abelian Group(A5).
- Closure under multiplication (M1): If a, b ∈ R then ab ∈ R
- Associativity of multiplication (M2): a (bc) = (ab) c for all a, b, c ∈ R
- Distributive laws (M3) :

  a (b + c) = ab + ac for all a, b, c ∈ R

  (a + b) c = ac + bc for all a, b, c ∈ R

Note:

Subtraction [a - b = a + (-b)]

Cryptography and Network Security Principles

In present day scenario security of the system is the sole priority of any organisation. The main aim of any organisation is to protect their data from attackers. In cryptography, attacks are of two types such as Passive attacks and Active attacks.

Passive attacks are those that retrieve information from the system without affecting the system resources while active attacks are those that retrieve system information and make changes to the system resources and their operations.

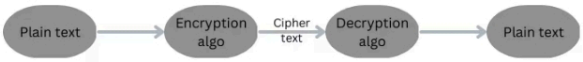


Figure : 1.1

In figure 1.1 it made the text secure by forming it into cipher text using encryption algorithm and further decryption to use it.

The Principles of Security can be classified as follows:

1. **Confidentiality:**
   The degree of confidentiality determines the secrecy of the information. The principle specifies that only the sender and receiver will be able to access the information shared between them. Confidentiality compromises if an unauthorized person is able to access a message.
   For example, let us consider sender A wants to share some confidential information with receiver B and the information gets intercepted by the attacker C. Now the confidential information is in the hands of an intruder C.

2. **Authentication:**
   Authentication is the mechanism to identify the user or system or the entity. It ensures the identity of the person trying to access the information. The authentication is mostly secured by using username and password. The authorized person whose identity is preregistered can prove his/her identity and can access the sensitive information.

3. **Integrity:**
   Integrity gives the assurance that the information received is exact and accurate. If the content of the message is changed after the sender sends it but before reaching the intended

information is in the hands of an intruder C.

2. **Authentication:**
   Authentication is the mechanism to identify the user or system or the entity. It ensures the identity of the person trying to access the information. The authentication is mostly secured by using username and password. The authorized person whose identity is preregistered can prove his/her identity and can access the sensitive information.

3. **Integrity:**
   Integrity gives the assurance that the information received is exact and accurate. If the content of the message is changed after the sender sends it but before reaching the intended receiver, then it is said that the integrity of the message is lost.

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

4. **Non-Repudiation:**
   Non-repudiation is a mechanism that prevents the denial of the message content sent through a network. In some cases the sender sends the message and later denies it. But the non-repudiation does not allow the sender to refuse the receiver.

5. **Access control:**
   The principle of access control is determined by role management and rule management. Role management determines who should access the data while rule management determines up to what extent one can access the data. The information displayed is dependent on the person who is accessing it.

6. **Availability:**
   The principle of availability states that the resources will be available to authorize party at all times. Information will not be useful if it is not available to be accessed. Systems should have sufficient availability of information to satisfy the user request.

7. **Issues of ethics and law**
   The following categories are used to categorize ethical dilemmas in the security system.

   Individuals' right to access personal information is referred to as privacy.
   Property: It is concerned with the information's owner.
   Accessibility is concerned with an organization's right to collect information.

   Accuracy: It is concerned with the obligation of information authenticity, fidelity, and accuracy.

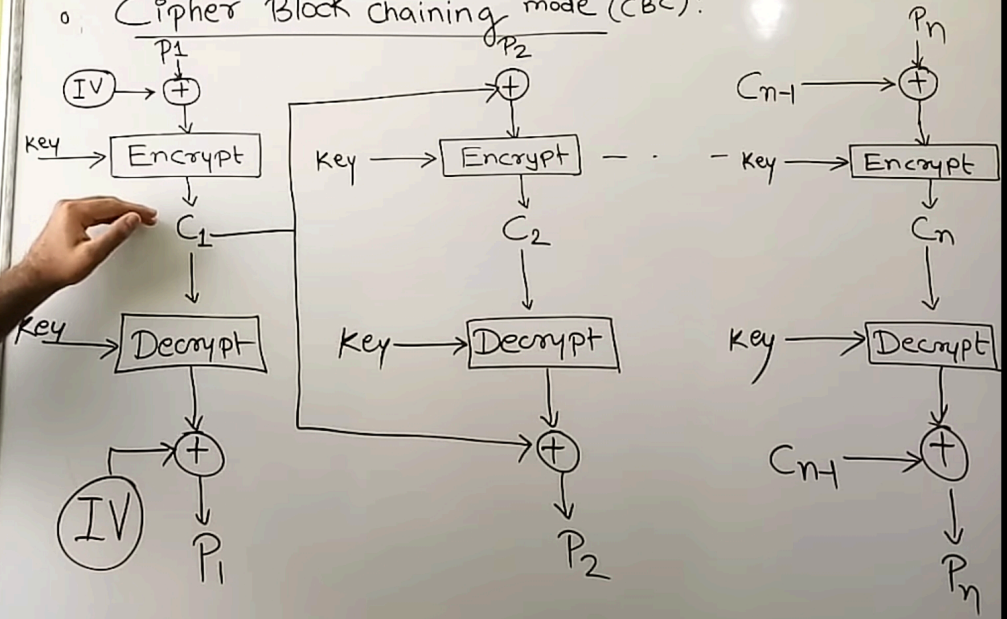
Article Tags : Computer Networks | GATE CS | cryptography | Information-Security | Network-security

Recommended Articles

1. Classical Cryptography and Quantum Cryptography
2. Custom Building Cryptography Algorithms (Hybrid Cryptography)
3. Difference between Network Security and Cyber Security
4. Difference between Application Security and Network Security
5. Difference between Information Security and Network Security
6. Difference between Cryptography and Cyber Security
7. Cybersecurity vs Network Security vs Information Security
8. Difference Between Network Security and Network Administration

Block Ciphee mode

- Cipher Block Chaining mode (CBC):

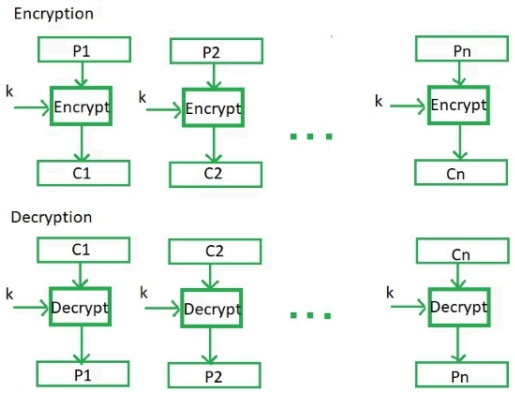


Block Cipher modes of Operation

Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher. **Block cipher** is an encryption algorithm that takes a fixed size of input say *b* bits and produces a ciphertext of *b* bits again. If the input is larger than *b* bits it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

Electronic Code Book (ECB) –
Electronic code book is the easiest block cipher mode of functioning. It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Generally, if a message is larger than *b* bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

Procedure of ECB is illustrated below:



Advantages of using ECB –

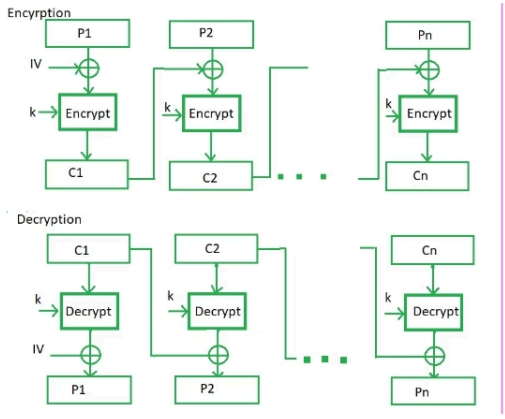
- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

Cipher Block Chaining –
Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

The process is illustrated here:



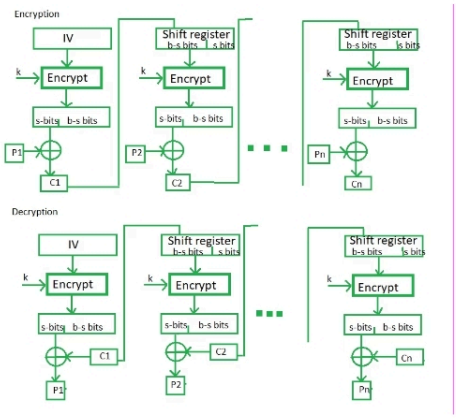
Advantages of CBC –

- CBC works well for input greater than *b* bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.

Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.

Cipher Feedback Mode (CFB) –
In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits.The left-hand side *s* bits are selected along with plaintext bits to which an XOR operation is applied. The result is given as input to a shift register having b-s bits to lhs,s bits to rhs and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithms.



Advantages of CFB –

- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

Disadvantages of using ECB –

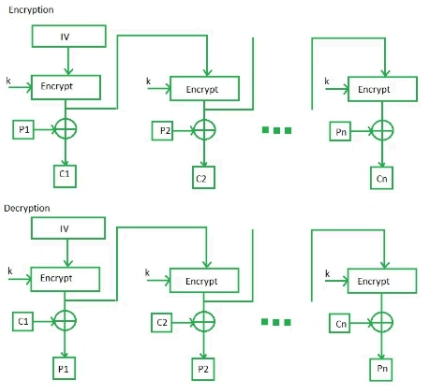
- The drawbacks of CFB are the same as those of CBC mode. Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

Output Feedback Mode –

- The drawbacks of CFB are the same as those of CBC mode. Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

Output Feedback Mode –

The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases the dependency or relationship of the cipher on the plaintext.



Advantages of OFB –

- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.

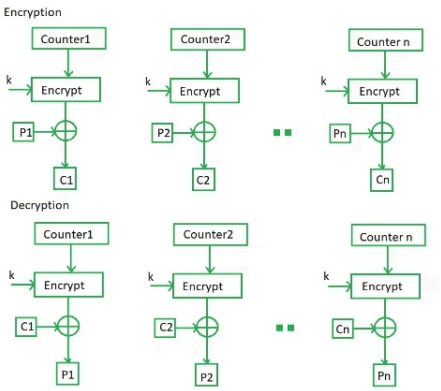
Disadvantages of OFB-

- The drawback of OFB is that, because to its operational modes, it is more susceptible to a message stream modification attack than CFB.

Counter Mode –

The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block. The CTR mode is independent of feedback use and thus can be implemented in parallel.

Its simple implementation is shown below:



Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.
- Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

Disadvantages of Counter-

- The fact that CTR mode requires a synchronous counter at both the transmitter and the receiver is a severe drawback. The recovery of plaintext is erroneous when synchronisation is lost.

Article Tags : Computer Networks

Difference Between Symmetric and Asymmetric Key Encryption

Symmetric Key Encryption: Encryption is a process to change the form of any message in order to protect it from reading by anyone. In Symmetric-key encryption the message is encrypted by using a key and the same key is used to decrypt the message which makes it easy to use but less secure. It also requires a safe method to transfer the key from one party to another.

Asymmetric Key Encryption: Asymmetric Key Encryption is based on public and private key encryption techniques. It uses two different key to encrypt and decrypt the message. It is more secure than the symmetric key encryption technique but is much slower.

Symmetric Key Encryption	Asymmetric Key Encryption
It only requires a single key for both encryption and decryption.	It requires two keys, a public key and a private key, one to encrypt and the other one to decrypt.
The size of cipher text is the same or smaller than the original plain text.	The size of cipher text is the same or larger than the original plain text.
The encryption process is very fast.	The encryption process is slow.
It is used when a large amount of data is required to transfer.	It is used to transfer small amounts of data.
It only provides confidentiality.	It provides confidentiality, authenticity, and non-repudiation.
The length of key used is 128 or 256 bits	The length of key used is 2048 or higher
In symmetric key encryption, resource utilization is low as compared to asymmetric key encryption.	In asymmetric key encryption, resource utilization is high.
It is efficient as it is used for handling large amount of data.	It is comparatively less efficient as it can handle a small amount of data.
Security is less as only one key is used for both encryption and decryption purpose.	It is more secure as two keys are used here- one for encryption and the other for decryption.
The Mathematical Representation is as follows- P = D (K, E(P)) where K –> encryption and decryption key P –> plain text D –> Decryption E(P) –> Encryption of plain text	The Mathematical Representation is as follows- P = D(Kd, E (Ke,P)) where Ke –> encryption key Kd –> decryption key D –> Decryption E(Ke, P) –> Encryption of plain text using encryption key Ke . P –> plain text
Examples: 3DES, AES, DES and RC4	Examples: Diffie-Hellman, ECC, El Gamal, DSA and RSA

Recommended Solve Problems
Solve DSA problems on GfG Practice.

Article Tags : Algorithms | Computer Networks | Difference Between | DSA | cryptography

Recommended Articles

RSA Algorithm in Cryptography

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. **Public Key** and **Private Key.** As the name describes that the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

1. A client (for example browser) sends its public key to the server and requests some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. The client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

Let us learn the mechanism behind the RSA algorithm : >> Generating Public Key:

```
Select two prime no's. Suppose P = 53 and Q = 59.
Now First part of the Public key  : n = P*Q = 3127.
 We also need a small exponent say e :
But e Must be
An integer.
Not be a factor of n.
1 < e < Φ(n) [Φ(n) is discussed below],
Let us now consider it to be equal to 3.
    Our Public Key is made of n and e
```

>> Generating Private Key:

```
We need to calculate Φ(n) :
Such that Φ(n) = (P-1)(Q-1)
      so,  Φ(n) = 3016
    Now calculate Private Key, d :
d = (k*Φ(n) + 1) / e for some integer k
For k = 2, value of d is 2011.
```

Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d = 2011) Now we will encrypt **"HI"** :

Select two prime no's. Suppose **P = 53 and Q = 59.**

Now First part of the Public key : n = P*Q = 3127.

We also need a small exponent say **e :**

But e Must be

An integer.

Not be a factor of n.

1 < e < Φ(n) [Φ(n) is discussed below],

Let us now consider it to be equal to 3.

Our Public Key is made of n and e

>> Generating Private Key:

We need to calculate Φ(n) :

Such that **Φ(n) = (P-1)(Q-1)**

so, Φ(n) = 3016

Now calculate Private Key, **d :**

d = (k*Φ(n) + 1) / e for some integer k

For k = 2, value of d is 2011.

Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d will encrypt **"HI"** :

Convert letters to numbers : H = 8 and I = 9

Thus **Encrypted Data c = $89^e \bmod n$.**

Thus our Encrypted Data comes out to be 1394

Now we will decrypt **1394** :

Decrypted Data = $c^d \bmod n$.

Thus our Encrypted Data comes out to be 89

8 = H and I = 9 i.e. "HI".

Below is the implementation of the RSA algorithm for

$= 4^7 \pmod 3$

$= 4 \pmod 3$

$= 1$

∴ The encrypted ciphertext is : 0 and 1.

Que 2.23. Explain RSA algorithm. Perform encryption and decryption using RSA algorithm for $p = 11, q = 13, e = 7, m = 9$.

AKTU 2015-16, Marks 15

OR

Explain RSA using example.

AKTU 2016-17, Marks 10

Answer

RSA algorithm :

1. The RSA algorithm is asymmetric key cryptographic algorithm.
2. The RSA algorithm is based on the mathematical fact that it is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product.
3. The private and public keys in RSA are made up of 100 or more digits prime numbers.
4. The real challenge in RSA is the selection and generation of the public and private keys.
5. The RSA algorithm is shown as :
   a. Choose two large prime numbers p and q.
   b. Calculate $n = p \times q$.
   c. Select the public key (*i.e.*, the encryption key) e such that it is not a factor of $(p - 1)$ and $(q - 1)$.
   d. Select the private key (*i.e.*, the decryption key) d such that the following equation is true :

$$(d \times e) \bmod (p - 1) \times (q - 1) = 1$$

   e. For encryption, calculate the cipher text C from the plain text M as follows :

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

   f. Send C as the cipher text to the receiver.
   g. For decryption, calculate the plain text C from the cipher text C as follows :

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

Numerical :

Step 1 : $p = 11, q = 13$

Step 2 : $n = p \times q = 11 \times 13 = 143$

Step 3 : Calculate

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

$$= (11 - 1)(13 - 1) = 10 \times 12 = 120$$

Step 4 : Determine d such that $de \equiv 1 \pmod{160}$

$$d = e^{-1} \bmod 160$$

Using extended Euclidean algorithm we calculate d.

q	r_1	r_2	r	t_1	t_2	t
17	120	7	1	0	1	– 17
7	7	1	0	1	– 17	120
	1	0		– 17	120	

$$= -17 \bmod 120$$

$$d = 103$$

Public key = {7, 143}

Private key = {103, 143}

Encryption $(C) = M^e \pmod n$

$M = 9$

$C = 9^7 \bmod 143$

$= [(9^4 \bmod 143) \times (9^2 \bmod 143)$

$(9^1 \bmod 143)] \bmod 143$

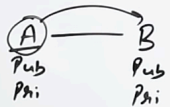
$= (126 \times 81 \times 9) \bmod 143$

$= 91854 \bmod 143$

$= 48$

Decryption $(M) = 13^{103} \bmod 143$

In a RSA cryptosystem a particular A uses two prime nos $p=13$ and $q=17$ to generate his public and private keys. If the Public key of A is 35. Then the ... key of A is _____ (A) 11 (B) 13 (C) 16 (D) 17



...ose two different large random prime no.

...lculate $n = p * q$

...lculate $\phi(n) = (p-1) * (q-1)$

Choose 'e' such that $1 < e < \phi(n)$ $\quad 1 < e < 192$

e is co prime to $\phi(n)$, $\gcd(e, \phi(n)) = 1$

Calculate d, such that $de \equiv 1 \bmod \phi(n)$

Public key 'e' Private key 'd'

$p = 13, \quad q = 17$

$n = p \times q = 13 \times 17 = 221$

$\phi(n) = (p-1) \times (q-1) = 12 \times 16 = 192$

$e = 35 \quad \gcd(35, 192) = 1$

$de = 1 + K\,\phi(n)$

$d = \dfrac{1 + K \cdot \phi(n)}{e} \qquad K = 0, 1, 2 \ldots$

$d = \dfrac{1+0 \times 192}{35} = \dfrac{1}{35} = 0.$

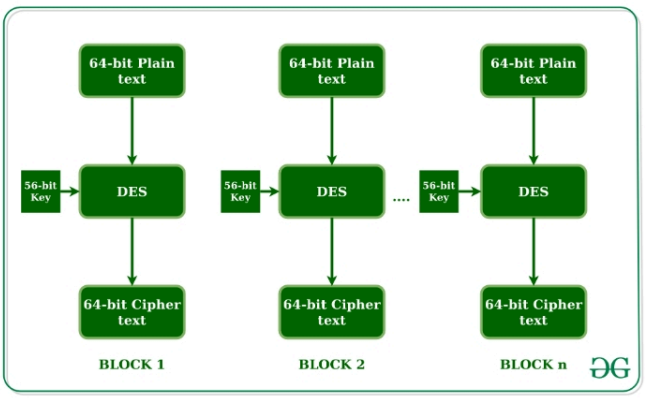
$d = \dfrac{1+1 \times 192}{35} = \dfrac{193}{35} = 5.5$

$d = \dfrac{1+2 \times 192}{35} = \dfrac{385}{35} =$

Data encryption standard (DES) | Set 1

Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of **64 bits** each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is **56 bits**.

The basic idea is shown in the figure:



We have mentioned that DES uses a 56-bit key. Actually, The initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

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

Figure - discording of every 8th bit of original key

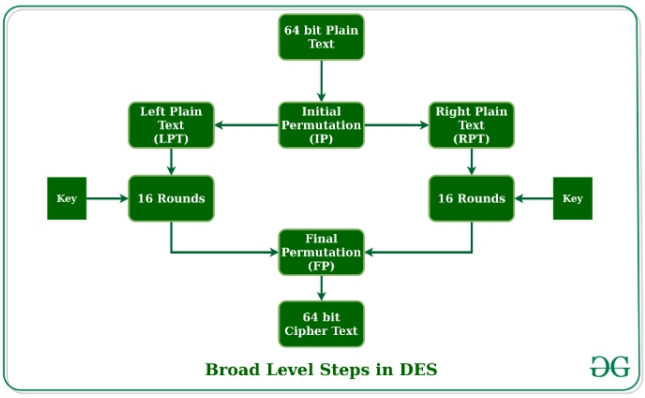
Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.

DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

- In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP)

Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.
DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

- In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
- The initial permutation is performed on plain text.
- Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
- Now each LPT and RPT go through 16 rounds of the encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
- The result of this process produces 64-bit ciphertext.



Broad Level Steps in DES

Initial Permutation (IP):

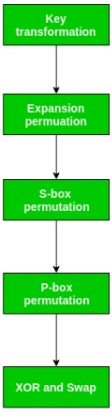
As we have noted, the initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed, as shown in the figure. For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block, and so on.

This is nothing but jugglery of bit positions of the original plain text block. the same rule applies to all the other bit positions shown in the figure.

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	33	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Figure - Initial permutation table

As we have noted after IP is done, the resulting 64-bit permuted text block is divided into two half blocks. Each half-block consists of 32 bits, and each of the 16 rounds, in turn, consists of the broad-level steps outlined in the figure.



Step-1: Key transformation:

We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation. For this, the 56-bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

For example: if the round numbers 1, 2, 9, or 16 the shift is done by only one position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is shown in the figure.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure - number of key bits shifted per round

After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table , we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation.

After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table , we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation.

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

Figure - compression permutation

Because of this compression permutation technique, a different subset of key bits is used in each round. That makes DES not easy to crack.

Step-2: Expansion Permutation:

Recall that after the initial permutation, we had two 32-bit plain text areas called Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called expansion permutation. This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.

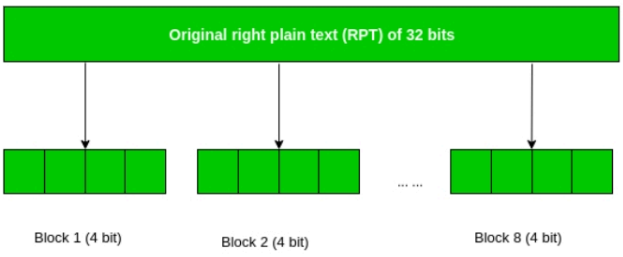
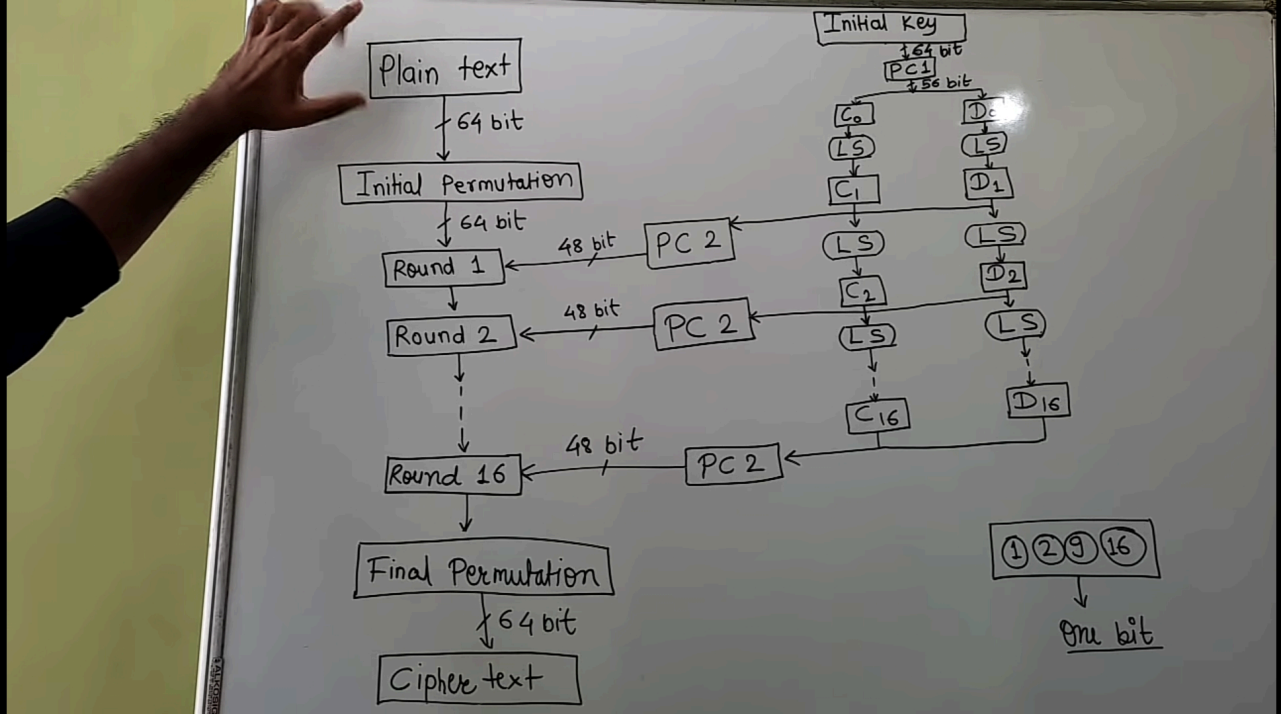


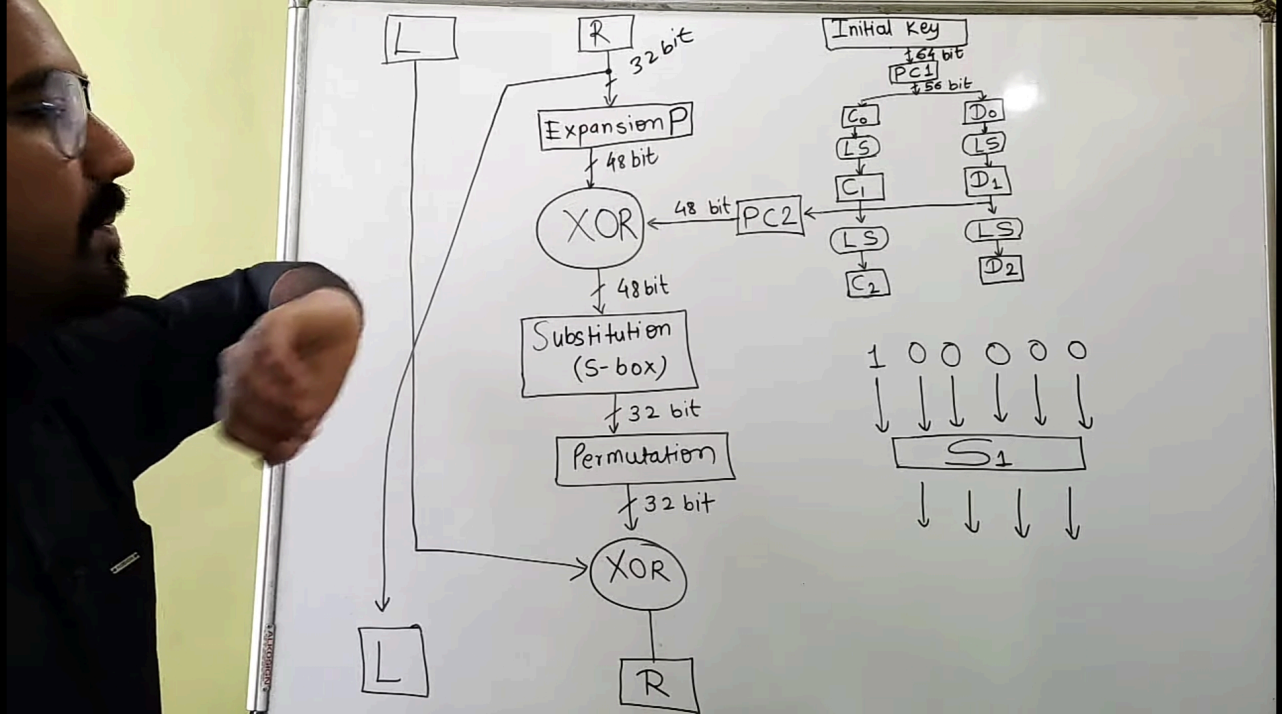
Figure - division of 32 bit RPT into 8 bit blocks

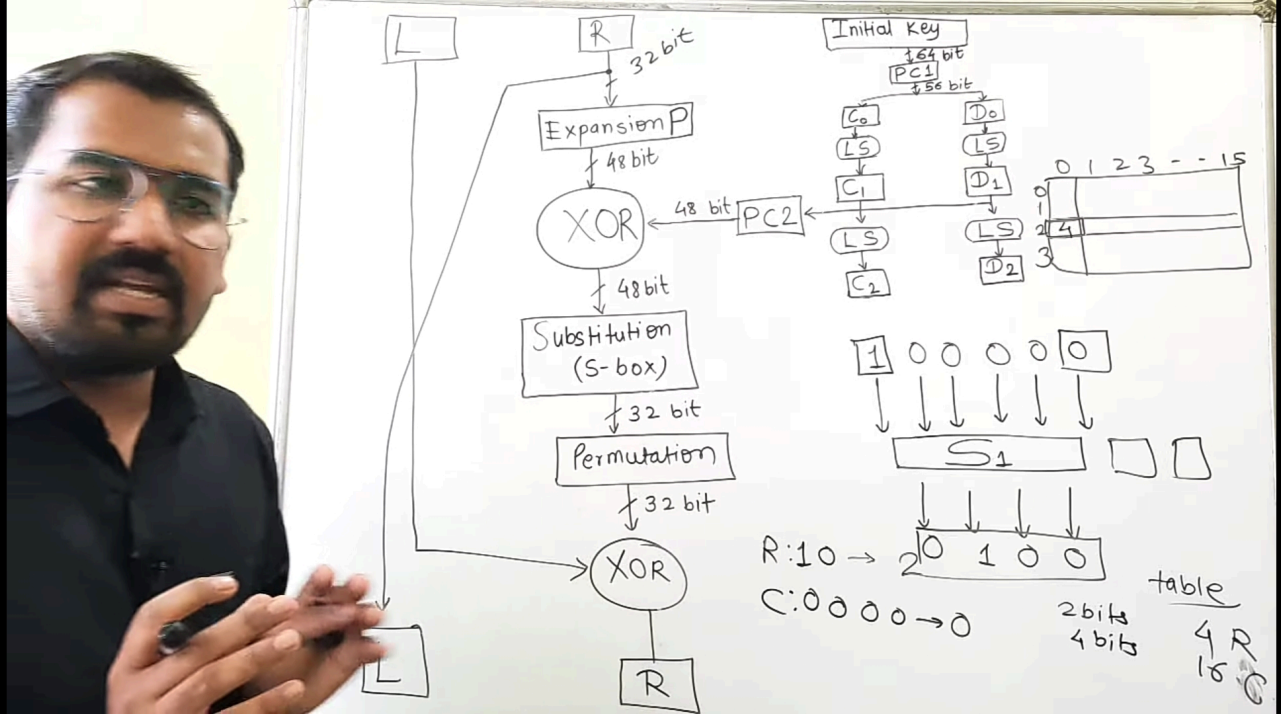
This process results in expansion as well as a permutation of the input bit while creating output. The key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the **32-bit RPT** to **48-bits**. Now the 48-bit key is XOR with 48-bit RPT and the resulting output is given to the next step, which is the **S-Box substitution**.

Javascript | Python

```
// Define DES key and plaintext
const key = "0123456789abcdef";
const plaintext = "Hello, world!";
// Perform DES encryption
const des = new DES(key);
const ciphertext = des.encrypt(plaintext);
```


Plain text
64 bit
Initial Permutation
64 bit
Round 1
48 bit
PC 2
Round 2
48 bit
PC 2
Round 16
48 bit
PC 2
Final Permutation
64 bit
Ciphere text
Initial Key
64 bit
PC1
56 bit
LS
LS
LS
LS
LS
LS
1 2 9 16
One bit


L
R
32 bit
Expansion P
48 bit
XOR
48 bit
PC2
48 bit
Substitution
(S-box)
32 bit
Permutation
32 bit
XOR
L
R
Initial Key
64 bit
PC1
56 bit
C0
D0
LS
LS
C1
D1
LS
LS
C2
D2
1 0 0 0 0 0
S1

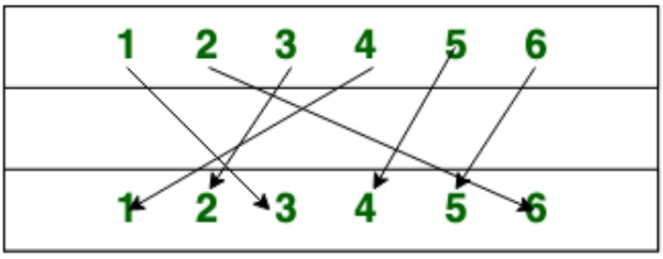

L
R
32 bit
Expansion P
48 bit
XOR
48 bit
PC2
48 bit
Substitution (S-box)
32 bit
Permutation
32 bit
XOR
L
R
Initial Key
64 bit
PC1
56 bit
C0
D0
LS
C1
D1
LS
C2
D2
0 1 2 3 - - 15
0
1
2
3
4
1 0 0 0 0 0
S1
R: 10 → 2
C: 0000 → 0
0 1 0 0
2 bit
4 bit
table
4 R
16 C

Difference between Substitution Cipher Technique and Transposition Cipher Technique

Both **Substitution cipher technique** and **Transposition cipher technique** are the types of Traditional cipher which are used to convert the plain text into cipher text.

Substitution Cipher Technique:
In Substitution Cipher Technique plain text characters are replaced with other characters, numbers and symbols as well as in substitution Cipher Technique, character's identity is changed while its position remains unchanged.

Transposition Cipher Technique:
Transposition Cipher Technique rearranges the position of the plain text's characters. In transposition Cipher Technique, The position of the character is changed but character's identity is not changed.



Transportation Cipher

Difference between Substitution Cipher Technique and Transposition Cipher Technique:

S.NO	Substitution Cipher Technique	Transposition Cipher Technique
1.	In substitution Cipher Technique, plain text characters are replaced with other characters, numbers and symbols.	In transposition Cipher Technique, plain text characters are rearranged with respect to the position.
2.	Substitution Cipher's forms are: Mono alphabetic substitution cipher and poly alphabetic substitution cipher.	Transposition Cipher's forms are: Key-less transposition cipher and keyed transposition cipher.
3.	In substitution Cipher Technique, character's identity is changed while its position remains unchanged.	While in transposition Cipher Technique, The position of the character is changed but character's identity is not changed.
4.	In substitution Cipher Technique, The letter with low frequency can detect plain text.	While in transposition Cipher Technique, The Keys which are nearer to correct key can disclose plain text.
5.	The example of substitution Cipher is Caesar Cipher.	The example of transposition Cipher is Rail Fence Cipher.

Rail Fence Cipher – Encryption and Decryption

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Rail Fence algorithm.
The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.
Examples:

```
Encryption
Input :  "GeeksforGeeks "
Key = 3
Output : GsGsekfrek eoe
Decryption
Input : GsGsekfrek eoe
Key = 3
Output :  "GeeksforGeeks "

Encryption
Input :  "defend the east wall"
Key = 3
Output : dnhaweedtees alf  tl
Decryption
Input : dnhaweedtees alf  tl
Key = 3
Output : defend the east wall

Encryption
Input : "attack at once"
Key = 2
Output : atc toctaka ne
Decryption
Input : "atc toctaka ne"
Key = 2
Output : attack at once
```

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

For example, if the message is “GeeksforGeeks” and the number of rails = 3 then cipher is prepared as:

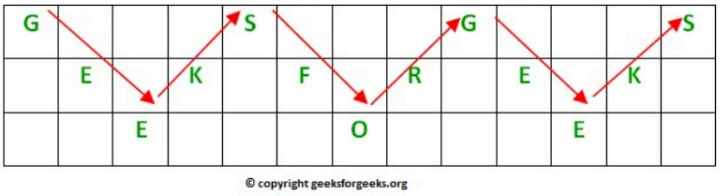
Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

For example, if the message is “GeeksforGeeks” and the number of rails = 3 then cipher is prepared as:



∴Its encryption will be done row wise i.e. GSGSEKFREKEOE

Decryption

Caesar Cipher in Cryptography

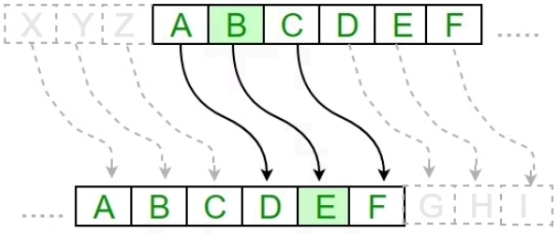
The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down.
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,..., Z = 25. Encryption of a letter by a shift n can be described mathematically as.

$$E_n(x) = (x + n) mod\ 26$$

(Encryption Phase with shift n)

(Decryption Phase with shift n)



Examples :

```
Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Shift: 23
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE
Shift: 4
Cipher: EXXEGOEXSRGI
```

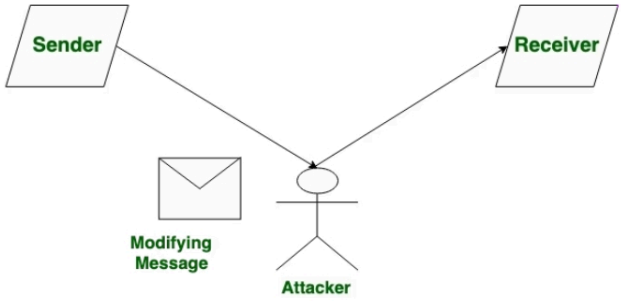
Advantages:

- Easy to implement and use thus, making suitable for beginners to learn about encryption.
- Can be physically implemented, such as with a set of rotating disks or a set of cards, known as a scytale, which can be useful in certain situations.
- Requires only a small set of pre-shared information.
- Can be modified easily to create a more secure variant, such as by using a multiple shift values or keywords.

Disadvantages:

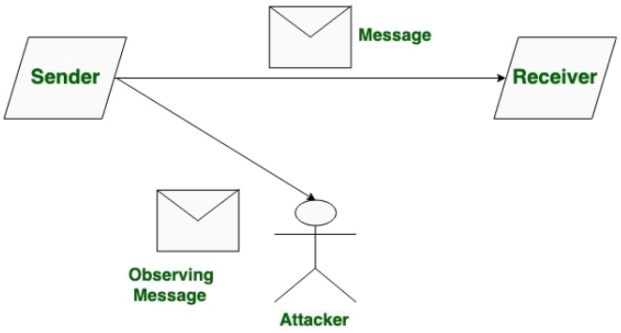
Difference between Active Attack and Passive Attack

Active Attacks: Active attacks are the type of attacks in which, The attacker efforts to change or modify the content of messages. Active Attack is dangerous to Integrity as well as availability. Due to active attack system is always damaged and System resources can be changed. The most important thing is that, In an active attack, Victim gets informed about the attack.



Active Attack

Passive Attacks: Passive Attacks are the type of attacks in which, The attacker observes the content of messages or copies the content of messages. Passive Attack is a danger to Confidentiality. Due to passive attack, there is no harm to the system. The most important thing is that In a passive attack, Victim does not get informed about the attack.



Passive Attack

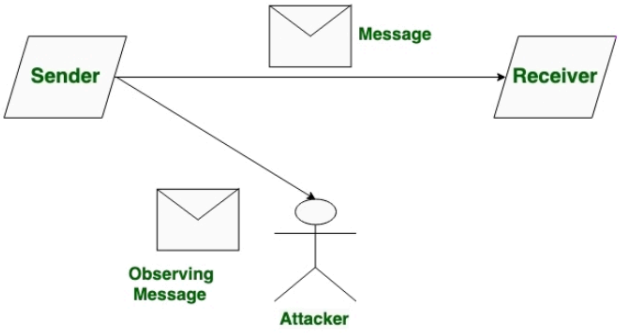
Prerequisite – Types of Security attacks | Active and Passive attacks

Difference between Active Attack and Passive Attack:

Active Attack	Passive Attack
In an active attack, Modification in information takes place.	While in a passive attack, Modification in the information does not take place.
Active Attack is a danger to **Integrity** as well as **availability**.	Passive Attack is a danger to **Confidentiality**.
In an active attack, attention is on prevention.	While in passive attack attention is on detection.
Due to active attacks, the execution system is	

Active Attack

Passive Attacks: Passive Attacks are the type of attacks in which, The attacker observes the content of messages or copies the content of messages. Passive Attack is a danger to Confidentiality. Due to passive attack, there is no harm to the system. The most important thing is that In a passive attack, Victim does not get informed about the attack.



Passive Attack

Prerequisite – Types of Security attacks | Active and Passive attacks

Difference between Active Attack and Passive Attack:

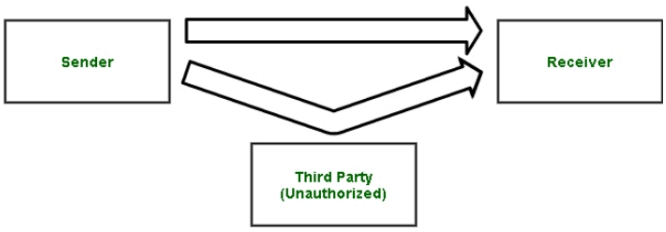
Active Attack	Passive Attack
In an active attack, Modification in information takes place.	While in a passive attack, Modification in the information does not take place.
Active Attack is a danger to **Integrity** as well as **availability**.	Passive Attack is a danger to **Confidentiality**.
In an active attack, attention is on prevention.	While in passive attack attention is on detection.
Due to active attacks, the execution system is always damaged.	While due to passive attack, there is no harm to the system.
In an active attack, Victim gets informed about the attack.	While in a passive attack, Victim does not get informed about the attack.
In an active attack, System resources can be changed.	While in passive attack, System resources are not changing.
Active attack influences the services of the system.	While in a passive attack, information and messages in the system or network are acquired.
In an active attack, information collected through passive attacks is used during execution.	While passive attacks are performed by collecting information such as passwords, and messages by themselves.
An active attack is tough to restrict from entering systems or networks.	Passive Attack is easy to prohibit in comparison to active attack.
Can be easily detected.	Very difficult to detect.
The purpose of an active attack is to harm the ecosystem.	The purpose of a passive attack is to learn about the ecosystem.
In an active attack, the original information is modified.	In passive attack original information is Unaffected.
The duration of an active attack is short.	The duration of a passive attack is long.
The prevention possibility of active attack is High	The prevention possibility of passive attack is low.
Complexity is High	Complexity is low.

Article Tags : Computer Networks | Difference Between | Information-Security

Replay Attack

Data has become very important to us in recent times. Safety and Security of data is of paramount importance. There are several confidential and sensitive information, which we cannot risk getting into wrong hands. However, sometimes an unauthorized person gets access to our information. Any action by an unauthorized person or hacker which poses a threat on the integrity, confidentiality and authentication of data is called a *security attack*.

What is Replay Attack ?
Replay Attack is a type of security attack to the data sent over a network.
In this attack, the hacker or any person with unauthorized access, captures the traffic and sends communication to its original destination, acting as the original sender. The receiver feels that it is an authenticated message but it is actually the message sent by the attacker. The main feature of the Replay Attack is that the client would receive the message twice, hence the name, **Replay Attack**.



Replay Attack

Note –
Arrows in the above image denote flow of communication.

Prevention from Replay Attack :

1. **Timestamp method –**
   Prevention from such attackers is possible, if timestamp is used along with the data. Supposedly, the timestamp on a data is more than a certain limit, it can be discarded, and sender can be asked to send the data again.
2. **Session key method –**
   Another way of prevention, is by using session key. This key can be used only once (by sender and receiver) per transaction, and cannot be reused.

Article Tags : Computer Networks | Information-Security | Network-security

Recommended Articles

1. Difference between Active Attack and Passive Attack
2. Denial of Service DDoS attack
3. Birthday attack in Cryptography
4. Sybil Attack
5. How to Prevent Man In the Middle Attack?
6. Difference between Threat and Attack
7. What is Zed Attack Proxy?
8. Brute Force Attack
9. What is a Dictionary Attack?
10. What is FTP Spoofing Attack?
11. US Maritime Attack
12. Zero-day Exploit (Cyber Security Attack)

About Flood Attacks

Flood attacks are also known as Denial of Service (DoS) attacks. In a flood attack, attackers send a very high volume of traffic to a system so that it cannot examine and allow permitted network traffic. For example, an ICMP flood attack occurs when a system receives too many ICMP ping commands and must use all its resources to send reply commands.

The Firebox can protect against these types of flood attacks:

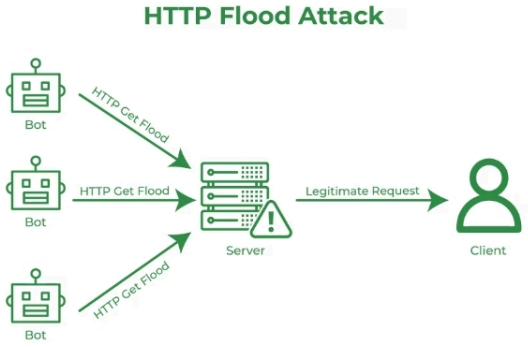
- IPSec
- IKE
- ICMP
- SYN
- UDP

The default configuration of the Firebox is to block flood attacks.

About Flood Attack Thresholds

To prevent flood attacks, in the **Default Packet Handling** page, you can specify thresholds for the allowed number of packets per second for different types of traffic. When the number of packets received on an interface exceeds the specified threshold, the device starts to drop traffic of that type on the interface.

For example, if you set the **Drop UDP Flood Attack** threshold to 1000, the device starts to drop UDP packets from an interface



HTTP Flooding Attack:

- HTTP flooding attack is a layer 7 (Application Layer Attack) which is really dangerous and harmful as it is easy to attack websites using HTTP flooding.
- HTTP flooding is a form of DDoS (Distributed Denial of Service) where an HTTP flood attack makes use of HTTP to get an HTTP post requests to carry out the cyberattack.
- The main purpose of an HTTP flooding Attack is to bring down a desired or needed site/ server by flooding it with a huge number of HTTP requests.
- The huge number of HTTP requests make the site/ server unresponsive and thus bring them down and inaccessible for use.

How to Perform HTTP flooding Attack:

- The flooding of HTTP requests is carried through bots which flood the computer with a huge number of requests. For each request to the server/ computer, thousands of unauthorized requests are flooded along with the user request.
- This huge volume of HTTP requests that flood the browser make use of bots that are supported by Trojan Horses for generating them.
- HTTP GET Requests are easy to create requests and make use of bots for generating a huge flood of requests. While the HTTP POST Requests, on the other hand, involve complex processing and are therefore not preferred over HTTP GET Requests to carry the HTTP flooding process.

The Appearance of HTTP Flooding Requests:

- HTTP flooding is an extremely dangerous form of cyber attack as the appearance of HTTP flooding requests makes them very difficult to distinguish.
- HTTP flooding requests appear like valid URLs and cannot be questioned or distinguished based on their appearance structure.

Prevention from HTTP Flooding Attacks:

It is very important to know the different ways in which HTTP flooding attacks can be prevented. Below are some of the ways listed to prevent HTTP flooding attacks:-

1. Computational Machines: Computational JavaScript machines can help detect if requests are coming from a bot and help in timely identifying flooding requests.

2. Web Application Firewall: The Web Application firewall will help in protecting from abnormal flooding of calls. Thus, securing the computer system from HTTP flooding attacks.

Article Tags : Ethical Hacking | Ethical Hacking - Network Attacks

Diffie-Hellman Key exchange algorithm

(i) not an encryption algo.
(ii) used to exchange secret keys between 2 users
(iii) we will use asymmetric encryption to exchange the secret key b/w users.
(public & private key concept)

Y this algo?

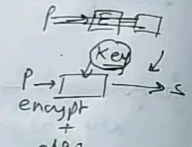
b/c when we are sending a key to receiver, it can be attacked in b/w.

ALGORITHM

(i) consider a prime number 'q'
(ii) select α such that it must be the primitive root of q and $\alpha < q$

a is a primitive root of q if

a mod q
a^2 mod q
a^{q-1} mod q



key concept)

ie when we are sending a key to receiver, it can be attacked in b/w.

ALGORITHM

(i) consider a prime number 'q'

(ii) select α such that it must be the primitive root of q and $\alpha < q$

'a' is a primitive root of q if

$a^1 \bmod q$

$a^2 \bmod q$

$a^3 \bmod q$ . . . . $a^{q-1} \bmod q$

gives results $\{1, 2, 3, \ldots, q-1\}$

ie values shouldnot be repeated & we should have all values in the o/p set from 1 to q-1.

(show example).

key exchange algorithm

primitive root

$\alpha^1 \bmod 7 = 3$
$\alpha^2 \bmod 7 = 2$
$\alpha^3 \bmod 7 = 6$
$\alpha^4 \bmod 7 = 4$
$\alpha^5 \bmod 7 = 5$
$\alpha^6 \bmod 7 = 1$

$q = 7$

$7 \overline{)125}$ 17

Let $q = 7$ (prime)

$\therefore \alpha < q$ i.e It is a primitive root

Let $\boxed{\alpha = 5}$ We can take any of the two primitive root 3 or 5.

α and $q \rightarrow$ global public elements (known to everyone)

Key generation of person1

Assume private key $\boxed{X_A = 3}$ $\because (X_A < q \quad 3 < 7$ yes$)$

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$= 125 \bmod 7$

Both of them dont need any info from anyone. They just use α and q. which are known to all.

key gener

Let P

dating

$(X_B < q)$

2

(show covert scenario diagram)

person2

Secret

$K_2 = (Y_A)^{X_B} \bmod$

$X_B < q$

secret key

secret keys, both

will use public

$\rightarrow$ private key of user

$\rightarrow$ public key of user

$X_A < q$

key exchange algorithm

primitive root

$\alpha^2 \bmod 7 = 3$
$\alpha^3 \bmod 7 = 2$
$\alpha^4 \bmod 7 = 6$
$\alpha^5 \bmod 7 = 4$
$\alpha^6 \bmod 7 = 5$
$\alpha^7 \bmod 7 = 1$

$q = 7$

$7 \overline{)125}$ 17
7
55
49
6

$\frac{625}{7} = 86$ 12
625×5

Let $q = 7$ (prime)

$\therefore \alpha < q$ i.e it is a primitive root

Let $\alpha = 5$ we can take any of the two primitive root 3 or 5.

α and $q \rightarrow$ global public elements (known to everyone)

Key generation of person1

Assume ~~private~~ key $X_A = 3$ $\therefore (X_A < q \quad 3 < 7 \text{ yes})$

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 5^3 \bmod 7 = 125 \bmod 7$

$Y_A = 7$

Key generation of person2

Let ~~private~~ key $X_B = 4$ $(X_B < q)$

calculating ~~private~~ public key $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 5^4 \bmod 7 = 2$

$Y_B = 2$

Both of them dont need any info from anyone. They just use α and q which are known to all.

(show covert scenario diagram)

key calculation | by person2

$K_2 = (Y_A)^{X_B} \bmod$

$X_A < q$ and

$X_B < q$

secret key

use publi

Diffie-Hellman key exchange

X → private key of users
Y → public key of users

(3) assume X_A (private key of A) and $X_A < q$

calculate $Y_A = \alpha^{X_A} \bmod q$

Y_A → public key of A

(4) assume X_B (private of B) $X_B < q$

calculate $Y_B = \alpha^{X_B} \bmod q$

Y_B → public key of B

Now we will calculate secret key

To calculate the secret key, both the sender & receiver will use public keys.

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

Let $q = 7$ (prime)

$\alpha < q$ i.e it is a primitive root

let $\alpha = 5$

we can take any of primitive root

α and q → global public elements (known

Key generation of person1

Assume private key $X_A = 3$

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 5^3 \bmod 7$

$Y_A = 7$

Key generation of person2

let private key $X_B = 4$

calculating public key $Y_B =$

Y_A

Secret key calculat

by Alice/person1

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

let $\alpha = 5$

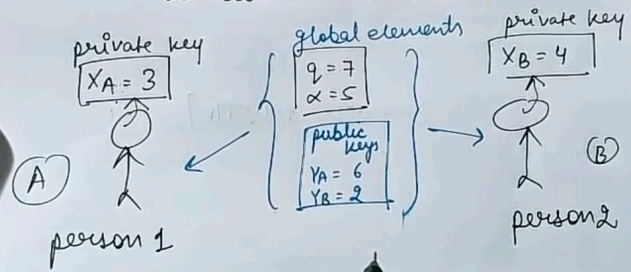
we can take any of the two primitive root 3 or 5.

α and $q \rightarrow$ global public elements (known to everyone)

Key generation of person 1

Assume private key $X_A = 3$ $\quad \because X_A < q$

So current scenario



$X \rightarrow$ public keys

$Y \rightarrow$ private keys

public key of A

(4) assume X_B (private of B)

calculate $Y_B = \alpha^{X_B} \bmod q$

public key of B

Now we will calculate secret

To calculate the secret keys sender & receiver will use

$K_A = (Y_B)^{X_A} \bmod q$ → public keys known to all ← $K_B = (Y_A)^{X_B}$

K ... then ...

is ...

THE ...

public elements (known to everyone)

Key generation of person1

Assume private key $X_A = 3$ ($X_A < q$, 3 < 7 yes)

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 5^3 \bmod 7 = 125 \bmod 7$

$Y_A = 7$

Key generation of person2

Let private key $X_B = 4$ ($X_B < q$)

calculating public key $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 5^4 \bmod 7 = 2$

$Y_B = 2$

Both of them don't need any info from anyone. They just use α and q, which are known to all.

(Show request scenario diagram)

Secret key calculation by Alice (person1)

...

by person2

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

$K = 6^4 \bmod 7$

$K = (36 \times 36) \bmod 7$

$K = 4$

Assume ~~public~~ private key $X_A = 3$ $\quad \because \left(\begin{array}{l} X_A < q \\ 3 < 7 \text{ yes} \end{array} \right)$

(4) ass calculating public key $Y_A = \alpha^{X_A} \bmod q$

ca

$Y_A = 5^3 \bmod 7 = 125 \bmod 7$

$\boxed{Y_A = 6}$

Both of them dont need any info from anyone. They Just use α and q. which are known to all.

Now

Key generation of person 2

Let ~~public~~ private key $X_B = 4$ $\quad (X_B < q)$

calculating ~~private~~ public key $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 5^4 \bmod 7 = 2$

$\boxed{Y_B = 2}$

$K_A =$

(Show current scenario diagram)

Secret key calculation by Alice/person 1	by person 2
$K_1 = K_A = (Y_B)^{X_A} \bmod q$	$K_B = (Y_A)^{X_B} \bmod q$
$= 2^3 \bmod q$	$K = 6^4 \bmod 7$
$= 2^3 \bmod 7$	$K = (36 \times 36) \bmod 7$
$\boxed{K = 1}$	$\boxed{K = 1}$

calculate $Y_A = \alpha^{X_A} \bmod q$

(Y_A ↓ public key of A)

(4) assume X_B (private of B) $\quad X_B < q$

calculate $Y_B = \alpha^{X_B} \bmod q$

(Y_B ↓ public key of B)

Now we will calculate secret key

To calculate the secret keys, both the sender & receiver will use public keys.

$K_A = (Y_B)^{X_A} \bmod q$ → public keys known to all ← $K_B = (Y_A)^{X_B} \bmod q$

$K_1 = K_2$ then we say exchange is successful.

Description of the algorithm

- Suppose Alice and Bob want to agree upon a key to be used for encryption / decryption message that would be exchanged between them. Then dif fi-hellman key exchange works as follows:

- 1. Alice & bob agree on two large prime numbers: n and g
  (note: these two integer need not be kept secret, insecure channel)
- 2. Alice choose another large random number x and calculate A such that,

$$A = g^x \bmod n$$

- 3. Alice send the number A to Bob.
- 4. Bob independently choose another large random integer y and calculate B such that,

$$B = g^y \bmod n$$

- 5. Bob send the number B to Alice.
- 6. Alice, now compute the secret key K1 as follows:

$$K1 = B^x \bmod n$$

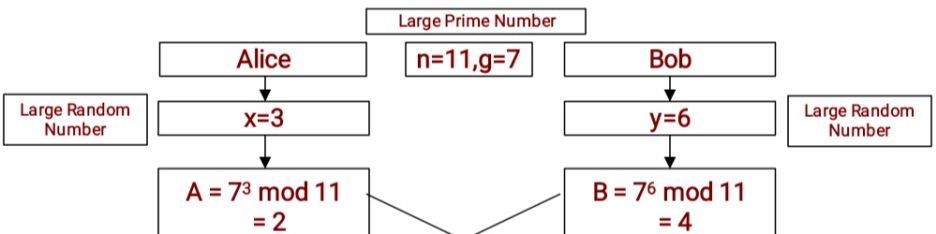
Description of the algorithm

- 7. Bob, now compute the secret key K2 as follows:

$$K2 = A^y \bmod n$$

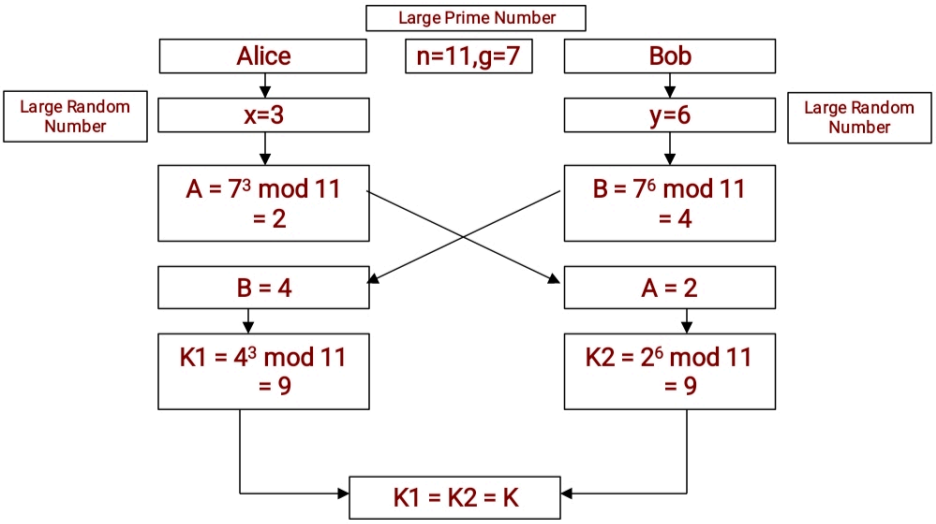
- 8. Surprise, K1=K2=K (which is symmetric key)

Example of the Diffie -Hellman algorithm



4

Example of the Diffie -Hellman algorithm



5

Note: A,B,K1,K2 are Private to others

Mathematics Theory behind the algorithm

- 1) Alice perform K1 $K1 = B^x \bmod n$ Now what is B? $B = g^y \bmod n$
- If we put B value into K1 then,

$$K1 = (g^y \bmod n)^x \bmod n$$

$$K1 = g^{yx} \bmod n$$

- 2) Bob perform K2 $K2 = A^y \bmod n$ w what is A? $A = g^x \bmod n$
- If we put A value into K2 then,

$$K2 = (g^x \bmod n)^y \bmod n$$

- Now Basic Mathematics says that,

$$K2 = g^{xy} \bmod n$$

$$K^{yx} = K^{xy} \text{ so that } K1=K2=K \text{ (Proof)}$$

6

Problem with the Algorithm

- Can dif fe-hellman solve our problem associated with key exchange? Unfortunately, not quite!
- This algorithm fall into "man in the middle attack"
- This work as follows:

The Chinese Remainder Theorem

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

$X \equiv a_1 \pmod{m_1}$

$X \equiv a_2 \pmod{m_2}$

. . .

$X \equiv a_n \pmod{m_n}$

CRT states that the above equations have a unique solution of the moduli are relatively prime.

The Chinese Remainder Theorem

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

$X \equiv a_1 \pmod{m_1}$

$X \equiv a_2 \pmod{m_2}$

. . .

$X \equiv a_n \pmod{m_n}$

CRT states that the above equations have a unique solution of the moduli are relatively prime.

$X = (a_1M_1\ M_1^{-1} + a_2M_2M_2^{-1} + \ldots + a_nM_nM_n^{-1}) \bmod M$

The Chinese Remainder Theorem

Example 1: Solve the following equations using CRT

$X \equiv 2 \pmod 3$

$X \equiv 3 \pmod 5$

$X \equiv 2 \pmod 7$

Solution:

$$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + a_3 M_3 M_3^{-1}) \bmod M$$

The Chinese Remainder Theorem

$X \equiv a_1 \pmod{m_1}$	$X \equiv 2 \pmod{3}$
$X \equiv a_2 \pmod{m_2}$	$X \equiv 3 \pmod{5}$
$X \equiv a_3 \pmod{m_3}$	$X \equiv 2 \pmod{7}$

Solution:

$X = (a_1M_1 M_1^{-1} + a_2M_2M_2^{-1} + a_3M_3M_3^{-1}) \bmod M$

Given		To Find		
$a_1 = 2$	$m_1 = 3$	M_1	M_1^{-1}	M
$a_2 = 3$	$m_2 = 5$	M_2	M_2^{-1}	
$a_3 = 2$	$m_3 = 7$	M_3	M_3^{-1}	

The Chinese Remainder Theorem

Given		To Find		
$a_1 = 2$	$m_1 = 3$	$M_1 = 35$	M_1^{-1}	$M=105$
$a_2 = 3$	$m_2 = 5$	$M_2 = 21$	M_2^{-1}	
$a_3 = 2$	$m_3 = 7$	$M_3 = 15$	M_3^{-1}	

$$M_1 = \frac{M}{m_1}$$

$$M_1 = \frac{105}{3}$$

$$M_1 = 35$$

$$M_2 = \frac{M}{m_2}$$

$$M_2 = \frac{105}{5}$$

$$M_2 = 21$$

$$M_3 = \frac{M}{m_3}$$

$$M_3 = \frac{105}{7}$$

$$M_3 = 15$$

The Chinese Remainder Theorem

Given		To Find		
$a_1 = 2$	$m_1 = 3$	$M_1 = 35$	$M_1^{-1} = 2$	
$a_2 = 3$	$m_2 = 5$	$M_2 = 21$	$M_2^{-1} = 1$	$M = 105$
$a_3 = 2$	$m_3 = 7$	$M_3 = 15$	$M_3^{-1} = 1$	

$M_1 \times M_1^{-1} = 1 \bmod m_1$

$35 \times M_1^{-1} = 1 \bmod 3$

$35 \times 2 = 1 \bmod 3$

$M_1^{-1} = 2$

$M_2 \times M_2^{-1} = 1 \bmod m_2$

$21 \times M_2^{-1} = 1 \bmod 5$

$21 \times 1 = 1 \bmod 5$

$M_2^{-1} = 1$

$M_3 \times M_3^{-1} = 1 \bmod m_3$

$15 \times M_3^{-1} = 1 \bmod 7$

$15 \times 1 = 1 \bmod 7$

$M_3^{-1} = 1$

The Chinese Remainder Theorem

Example 1: Solve the following equations using CRT

$X \equiv 2 \pmod 3$

$X \equiv 3 \pmod 5$

$X \equiv 2 \pmod 7$

$a_1 = 2$	$m_1 = 3$	$M_1 = 35$	$M_1^{-1} = 2$	M=105
$a_2 = 3$	$m_2 = 5$	$M_2 = 21$	$M_2^{-1} = 1$	
$a_3 = 2$	$m_3 = 7$	$M_3 = 15$	$M_3^{-1} = 1$	

Solution:

$X = (a_1M_1 M_1^{-1} + a_2M_2M_2^{-1} + a_3M_3M_3^{-1}) \bmod M$

$= (2x35x2 + 3x21x1 + 2x15x1) \bmod 105$

$= 233 \bmod 105$

$X = 23$

Introduction to Chinese Remainder Theorem

We are given two arrays num[0..k-1] and rem[0..k-1]. In num[0..k-1], every pair is coprime (gcd for every pair is 1). We need to find minimum positive number x such that:

```
x % num[0]    =  rem[0],
x % num[1]    =  rem[1],
.....................
x % num[k-1]  =  rem[k-1]
```

Basically, we are given k numbers which are pairwise coprime, and given remainders of these numbers when an unknown number x is divided by them. We need to find the minimum possible value of x that produces given remainders.

Examples :

```
Input:  num[] = {5, 7}, rem[] = {1, 3}
Output: 31
Explanation:
31 is the smallest number such that:
  (1) When we divide it by 5, we get remainder 1.
  (2) When we divide it by 7, we get remainder 3.

Input:  num[] = {3, 4, 5}, rem[] = {2, 3, 1}
Output: 11
Explanation:
11 is the smallest number such that:
  (1) When we divide it by 3, we get remainder 2.
  (2) When we divide it by 4, we get remainder 3.
  (3) When we divide it by 5, we get remainder 1.
```

Chinese Remainder Theorem states that there always exists an x that satisfies given congruences.

Below is theorem statement adapted from wikipedia.

Let num[0], num[1], ...num[k-1] be positive integers that are pairwise coprime. Then, for any given sequence of integers rem[0], rem[1], ... rem[k-1], there exists an integer x solving the following system of simultaneous congruences.

$$\begin{cases} x \equiv rem[0] & (\text{mod } num[0]) \\ \quad \cdots \\ x \equiv rem[k-1] & (\text{mod } num[k-1]) \end{cases}$$

Furthermore, all solutions x of this system are congruent modulo the product, prod = num[0] * num[1] * ... * nun[k-1]. Hence

$$x \equiv y \ (\text{mod } num[i]), \quad 0 \leq i \leq k-1 \quad \Longleftrightarrow \quad x \equiv y \ (\text{mod } prod).$$

The first part is clear that there exists an x. The second part basically states that all solutions

Basically, we are given k numbers which are pairwise coprime, and given remainders of these numbers when an unknown number x is divided by them. We need to find the minimum possible value of x that produces given remainders.

Examples :

```
Input:  num[] = {5, 7}, rem[] = {1, 3}
Output: 31
Explanation:
31 is the smallest number such that:
  (1) When we divide it by 5, we get remainder 1.
  (2) When we divide it by 7, we get remainder 3.

Input:  num[] = {3, 4, 5}, rem[] = {2, 3, 1}
Output: 11
Explanation:
11 is the smallest number such that:
  (1) When we divide it by 3, we get remainder 2.
  (2) When we divide it by 4, we get remainder 3.
  (3) When we divide it by 5, we get remainder 1.
```

Chinese Remainder Theorem states that there always exists an x that satisfies given congruences.

Below is theorem statement adapted from wikipedia.

Let num[0], num[1], ...num[k-1] be positive integers that are pairwise coprime. Then, for any given sequence of integers rem[0], rem[1], ... rem[k-1], there exists an integer x solving the following system of simultaneous congruences.

$$\begin{cases} x \equiv rem[0] & (\text{mod } num[0]) \\ \cdots \\ x \equiv rem[k-1] & (\text{mod } num[k-1]) \end{cases}$$

Furthermore, all solutions x of this system are congruent modulo the product, prod = num[0] * num[1] * ... * nun[k-1]. Hence

$$x \equiv y \ (\text{mod } num[i]), \quad 0 \leq i \leq k-1 \quad \Longleftrightarrow \quad x \equiv y \ (\text{mod } prod).$$

The first part is clear that there exists an x. The second part basically states that all solutions (including the minimum one) produce the same remainder when divided by-product of num[0], num[1], .. num[k-1]. In the above example, the product is 3*4*5 = 60. And 11 is one solution, other solutions are 71, 131, .. etc. All these solutions produce the same remainder when divided by 60, i.e., they are of form 11 + m*60 where m >= 0.

A **Naive Approach to find x** is to start with 1 and one by one increment it and check if dividing it with given elements in num[] produces corresponding remainders in rem[]. Once we find such an x, we return it.

Below is the implementation of Naive Approach.

C++ Java Python3 C# PHP Javascript

Fermat's Little Theorem

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

Fermat's Little Theorem

Example 1: Does Fermat's theorem hold true for p=5 and a=2?

Solution:

Given: p=5 and a=2.

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

$2^{5-1} \equiv 1 \pmod{5}$

$2^4 \equiv 1 \pmod{5}$

$16 \equiv 1 \pmod{5}$

Euler's Theorem

For every positive integer 'a' & 'n', which are said to be relatively prime, then $a^{\Phi(n)} \equiv 1 \bmod n$.

Euler's Totient Function

- Denoted as $\Phi(n)$.
- $\Phi(n)$ = Number of positive integers less than 'n' that are relatively prime to n.

Euler's Totient Function

Example 1: Find $\Phi(5)$.

Solution:

Here n=5.

Numbers less than 5 are 1, 2, 3 and 4.

GCD	Relatively Prime?
GCD (1, 5) = 1	✓
GCD (2, 5) = 1	✓
GCD (3, 5) = 1	✓
GCD (4, 5) = 1	✓

$\therefore \Phi(5) = 4$.

Euler's Totient Function

	Criteria of 'n'	Formula
$\Phi(n)$	'n' is prime.	$\Phi(n) = (n-1)$
	n = p x q. 'p' and 'q' are primes.	$\Phi(n) = (p-1) \times (q-1)$
	n = a x b. Either 'a' or 'b' is composite. Both 'a' and 'b' are composite.	$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \ldots .$ where $p_1, p_2, \ldots$ are distinct primes.

Euler's Totient Function

Example 3: Find $\Phi(35)$.

Solution:

Here n=35.

'n' is a product of two prime numbers 5 and 7.

Let us assign p=5 and q=7.

$\Phi(n) = (p-1) \times (q-1)$

$\Phi(35) = (5-1) \times (7-1)$

$\Phi(35) = 4 \times 6$

$\Phi(35) = 24$

Euler's Totient Function

Example 4: Find $\Phi(1000)$.

Solution:

Here $n = 1000 = 2^3 \times 5^3$.

Distinct prime factors are 2 and 5.

$$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \ldots .$$

$$\Phi(1000) = 1000 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right)$$

$$\Phi(1000) = 1000 \times \left(\frac{1}{2}\right)\left(\frac{4}{5}\right)$$

$$\Phi(1000) = 400$$

Euler's Totient Function

Example 5: Find $\Phi(7000)$.

Solution:

Here $n = 7000 = 2^3 \times 5^3 \times 7^1$

Distinct prime factors are 2, 5 and 7.

$$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\left(1 - \frac{1}{p_3}\right)\ldots.$$

$$\Phi(7000) = 7000 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right)\left(1 - \frac{1}{7}\right)$$

$$\Phi(7000) = 7000 \times \left(\frac{1}{2}\right)\left(\frac{4}{5}\right)\left(\frac{6}{7}\right)$$

$\Phi(7000) = 2400$

Euler's Theorem

Example 1: Prove Euler's theorem hold true for a=3 and n=10.

Solution:

Given: a=3 and n=10.

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

$3^{\Phi(10)} \equiv 1 \pmod{10}$

$\Phi(10) = 4$

$3^4 \equiv 1 \pmod{10}$

$81 \equiv 1 \pmod{10}$

Euler's Theorem

Example 2: Does Euler's theorem hold true for a=2 and n=10?

Solution:

Given: a=2 and n=10.

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

$2^{\Phi(10)} \equiv 1 \pmod{10}$

$\Phi(10) = 4$

$2^4 \equiv 1 \pmod{10}$

$16 \equiv 1 \pmod{10}$

Therefore, Euler's theorem does not hold for a=2 and n=10.

Euler's Theorem

Example 3: Does Euler's theorem hold true for a=10 and n=11?

Solution:

Given: a=10 and n=11.

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

$10^{\Phi(11)} \equiv 1 \pmod{11}$

$\Phi(11) = 10$

$10^{10} \equiv 1 \pmod{11}$

$-1^{10} \equiv 1 \pmod{11}$

$1 \equiv 1 \pmod{11}$

Therefore, Euler's theorem holds for a=10 and n=11.

Modular Arithmetic

- ★ System of arithmetic for integers.
- ★ Wrap around after reaching a certain value called modulus.
- ★ Central mathematical concept in cryptography.

Properties of Modular Arithmetic

Property	Expression
Commutative Laws	$(a + b) \bmod n = (b + a) \bmod n$ $(a \times b) \bmod n = (b \times a) \bmod n$
Associative Laws	$[(a + b) + c] \bmod n = [a + (b + c)] \bmod n$ $[(a \times b) \times c] \bmod n = [a \times (b \times c)] \bmod n$
Distributive Laws	$[a \times (b + c)] \bmod n = [(a \times b) + (a \times c)] \bmod n$
Identities	$(0 + a) \bmod n = a \bmod n$ $(1 \times a) \bmod n = a \bmod n$
Additive Inverse	For each $a \in Z_n$, there exists a '-a' such that $a + (-a) \equiv 0 \bmod n$

Understanding GCD – Example 2

	25	150
Divisors	1, 5, 25	1, 2, 3, 5, 6, 10, 15, 25, 30, 50, 75, 150
Common Divisors	1, 5, 25	
Greatest Common Divisor (GCD)	25	

$\therefore GCD(25, 150) = 25$

Euclid's Algorithm for finding GCD

Find the GCD(12, 33).

Q	A	B	R
2	33	12	9
1	12	9	3
3	9	3	0
X	3	0	X

Relatively Prime Numbers

Two numbers are said to be relatively prime, if they have no prime factors in common, and their only common factor is 1.