Sarvasiddhant Education Society's
SWAMINARAYAN SIDDHANTA INSTITUTE OF TECHNOLOGY
Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University
Nagpur Katol Highway Road, Khapri (Kothe),
Tal.Kalmeshwar, Nagpur, Maharashtra 441501
Department Of Computer Engineering
Session 2023-2024

CNS NOTES

CRYPTOGRAPHY & NETWORK SECURITY

SEMESTER: 7TH SEM (FINAL YEAR)

Subject Incharge: Prof. Ashvini Bais

Asst. Prof. (CE Dept.)

NOTES

CRYPTOGRAPHY & NETWORK SECURITY

Semester: 7th Sem (Final Year)

Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur
FOUR YEAR B. TECH. COURSE
(Revised Curriculum as per AICTE Model Curriculum)
B.Tech VII Semester (Computer Engineering) Scheme & Syllabus

Seventh Semester:-

S. N.	Subject Code	Subject	Teaching Scheme			Evaluation Scheme			Credits	Minimum Passing Marks
			L	T	P	CA	UE	Total		
1	BTCME701T	Cryptography & Network Security	3	1	-	30	70	100	4	45
2	BTCME701P	Cryptography & Network Security-Lab	-	-	2	25	25	50	1	25
3	BTCME702T	Elective – IV	3	-	-	30	70	100	3	45
4	BTCME703T	Elective – V	3	-	-	30	70	100	3	45
5	BTCME704T	Open Elective-II	3	-	-	30	70	100	3	45
6	BTCME705P	Project Work Phase -I	-	-	6	50	50	100	3	50
7	BTCME706P	Report Writing Activity	-	-	2	-	-	-	Audit	Grade
		Total	**12**	**01**	**10**	**195**	**355**	**550**	**17**	

Elective IV: -
1. Deep Learning
2. Block chain Technology
3. Augmented & Virtual Reality
4. Salesforce Technology

Elective V: -
1. Compiler Design
2. Natural Language Processing
3. Introduction to Software Testing

Open Electives:
1. Joy of Computing using Python
2. Data Base Management System
3. Data Visualization

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B.TECH.) DEGREE COURSE
SEMESTER: SEVENTH (CBCS)
BRANCH: Computer Engineering
Subject : Cryptography and Network Security

Subject : Cryptography and Network Security **Subject Code BTCME701T**

Load	Credit	Total Marks	Internal Marks	University Marks	Total
04Hrs (Theory)	03(L)+01(T)	100	30	70	100

Aim :To highlight the features of different technologies involved in Network Security.

Prerequisite(s): Mathematics, Algorithm, Networking

Course Objectives:

1	To develop the student's ability to understand the concept of security goals in various applications and learn classical encryption techniques
2	Apply fundamental knowledge on cryptographic mathematics used in various symmetric and asymmetric key cryptography
3	To develop the student's ability to analyze the cryptographic algorithms.
4	To develop the student's ability to analyze the cryptographic algorithms.

Course Outcomes:
At the end of this course student are able to:

CO1	To understand basics of Cryptography and Network Security and classify the symmetric encryption techniques.
CO2	Understand, analyze and implement the symmetric key algorithm for secure transmission of data.
CO3	Acquire fundamental knowledge about the background of mathematics of asymmetric key cryptography and understand and analyze asymmetric key encryption algorithms and digital signatures.
CO4	Analyze the concept of message integrity and the algorithms for checking the integrity of data.
CO5	To understand various protocols for network security to protect against the threats in the networks.

UNIT-I **[08 Hrs]**

Introduction, Model for network security. Mathematics of cryptography: modular arithmetic, Euclidean and extended Euclidean algorithm. Classical encryption techniques: substitution techniques-Caesar cipher, Vigenere's ciphers, Playfair ciphers and transposition techniques.

UNIT-II **[07 Hrs]**

Symmetric key cryptography: Block Cipher Principles, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), RC4, Key Distribution.

UNIT III **[07 Hrs]**

Asymmetric key cryptography: Euler's Totient Function, Fermat's and Euler's Theorem, Chinese Remainder Theorem, RSA, Diffie Hellman Key Exchange, ECC, Entity authentication: Digital signature.

UNIT IV **[07 Hrs]**

Message Integrity and authentication: Authentication Requirements and Functions, Hash Functions, MD5, Kerberos, Key Management, X.509 Digital Certificate format.

UNIT V **[07 Hrs]**

Network Security: PGP, SSL, Firewalls, IDS, Software Vulnerability: Phishing, Buffer Overflow, SQL Injection, Electronic Payment Types

Text Book:

1. William Stallings, "Cryptography and Network Security: Principles and Standards", Prentice Hall India, 7th Edition, 2017.
2. Bernard Menezes, "Network Security and Cryptography", Cengage Learning, 2010.

Reference Books:

1. Robert Brugge, Mark Rhodes, Heithstruggberg "Network Security, The Complete Reference", Tata McGraw Hill Publication, 2004.
2. Behrouz A. Forouzen, "Cryptography and Network Security", McGraw-Hill publication, 2nd Edition, 2010
3. Bruce Schneier, Applied Cryptography, John Wiley New York, 2nd Edition, 1996.

UNIT I

Security Fundamentals Introduction, Model for Network Security. Mathematics of cryptography: Modular arithmetic, Euclidean and extended Euclidean algorithm. Classical encryption techniques: substitution techniques- Caesar cipher, Vigenere's ciphers, Playfair ciphers and transposition techniques.

Security Fundamentals

Introduction-

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/ data, and telecommunications). This definition introduces three key objectives that are at the heart of computer security:

Confidentiality: This term covers two related concepts:

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

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

Integrity: This term covers two related concepts:

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

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

Availability: Assures that systems work promptly and service is not denied to authorized users. These three concepts form what is often referred to as the CIA triad (Figure 1.1). The three concepts embody the fundamental security objectives for both data and for information and computing services.

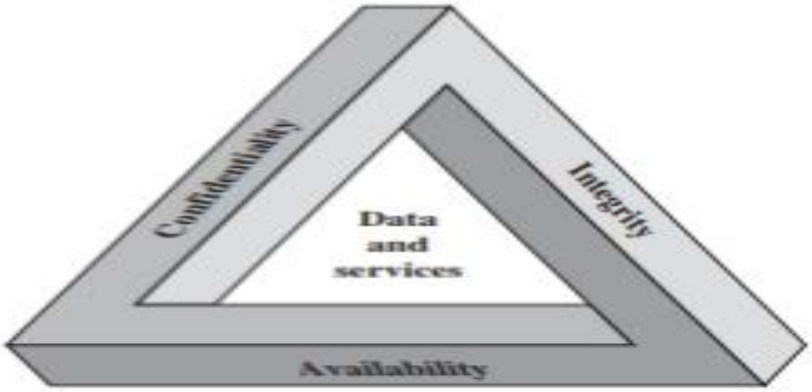


Figure 1.1 The Security Requirements Triad

For example, the NIST standard FIPS 199 (Standards for Security Categorization of Federal Information and Information Systems) lists confidentiality, integrity, and availability as the three security objectives for information and for information systems. FIPS 199 provides a useful characterization of these three objectives in terms of requirements and the definition of a loss of security in each category:

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

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

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

Model for Network Security-

Computer data often travels from one computer to another, leaving the safety of its protected physical surroundings. Once the data is out of hand, people with bad intention could modify or forge your data, either for amusement or for their own benefit. Cryptography can reformat and transform our data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

- Computer Security - generic name for the collection of tools designed to protect data and to thwart hackers

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

Network security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs; conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

A model for much of what we will be discussing is captured, in very general terms, in Figure A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the principals in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals. Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on.

All the techniques for providing security have two components:

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

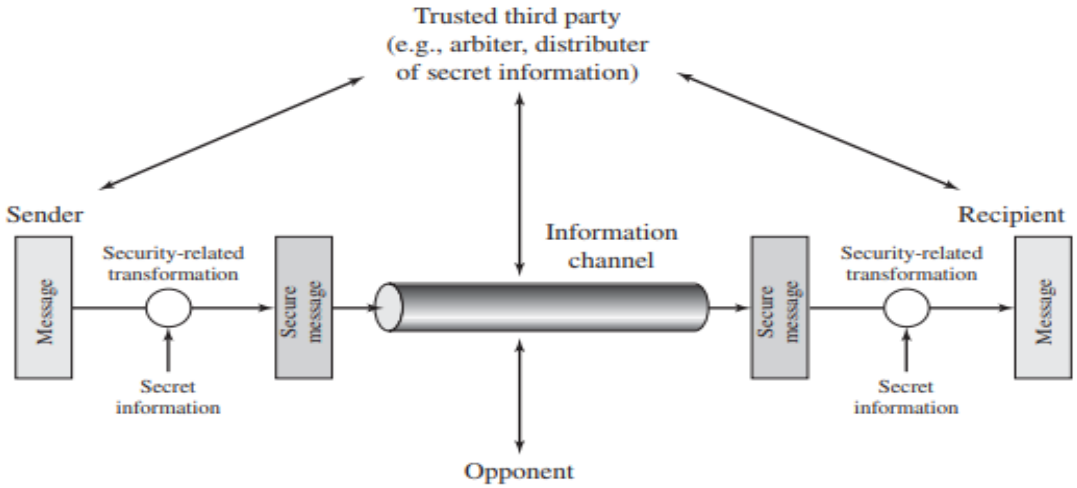


Figure A

- Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception. A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

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

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

Mathematics of Cryptography

Modular Arithmetic-

In modular arithmetic, we select an integer, n, to be our "modulus". Then our system of numbers only includes the numbers 0, 1, 2, 3, ..., n-1. In order to have arithmetic make sense, we have the numbers "wrap around" once they reach n.

Example: If we pick the modulus 5, then our solutions are required to be in the set {0, 1, 2, 3, 4}. We have 2+1=3 and 2+2=4 as usual. Then 2+3=5, which is not in our set, so it wraps around giving 2+3=0. Then 2+4=6, which wraps around to be 1. This may seem strange, but in fact we use it everyday! Consider a clock, we go from 1 o'clock to 2 o'clock, ..., 11 o'clock, 12 o'clock, then back to 1 o'clock, and so on. This is an example of when the modulus is 12 and for clocks we use {1, 2, ..., 12} instead of {0,1, ..., 11}, but these are the same because we consider 0 and 12 to be the same in terms of wrapping around.
How do we write modular arithmetic? Continuing the example above with modulus 5,
we write:
2+1 = 3 (mod 5) = 3
2+2 = 4 (mod 5) = 4
2+3 = 5 (mod 5) = 0
2+4 = 6 (mod 5) = 1

Modular Addition:
Rule for modular addition is:

(a + b) mod m = ((a mod m) + (b mod m)) mod m

Example:
(15 + 17) % 7

= ((15 % 7) + (17 % 7)) % 7

= (1 + 3) % 7

= 4 % 7

= 4

The same rule is to modular subtraction. We don't require much modular subtraction but it can also be done in the same way.

Modular Multiplication:
The Rule for modular multiplication is:

(a x b) mod m = ((a mod m) x (b mod m)) mod m

Example:
(12 x 13) % 5

= ((12 % 5) x (13 % 5)) % 5

= (2 x 3) % 5

= 6 % 5

= 1

Modular Division:
The modular division is totally different from modular addition, subtraction and multiplication. It also does not exist always.

(a / b) mod m is not equal to ((a mod m) / (b mod m)) mod m.

This is calculated using the following formula:

(a / b) mod m = (a x (inverse of b if exists)) mod m

Modular Inverse:
The modular inverse of a mod m exists only if a and m are relatively prime i.e. gcd(a, m) = 1. Hence, for finding the inverse of an under modulo m, if (a x b) mod m = 1 then b is the modular inverse of a.

Example: a = 5, m = 7 (5 x 3) % 7 = 1 hence, 3 is modulo inverse of 5 under 7.

Modular Exponentiation:
Finding a^b mod m is the modular exponentiation. There are two approaches for this – recursive and iterative. Example:
a = 5, b = 2, m = 7

(5 ^ 2) % 7 = 25 % 7 = 4

Associativity:

(a+b)+c = a+(b+c) mod n

Commutativity:

a+b = b+a mod n

Distributivity:

(a+b).c = (a.c)+(b.c) mod n

- also can chose whether to do an operation and then reduce modulo n, or reduce then do the operation, since reduction is a homomorphism from the ring of integers to the ring of integers modulo n
- a+/-b mod n = [a mod n +/- b mod n] mod n
- (the above laws also hold for multiplication)

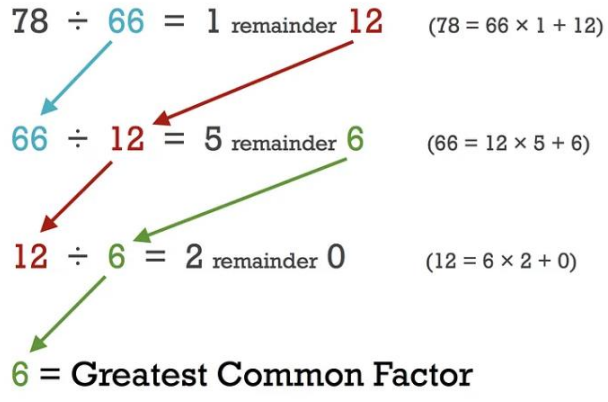
- if n is constrained to be a prime number p then this forms a Galois Field modulo p denoted GF(p) and all the normal laws associated with integer arithmetic work.

Euclidean Algorithm-

The Euclidean algorithm is a way to find the greatest common divisor of two positive integers. GCD of two numbers is the largest number that divides both of them. A simple way to find GCD is to factorize both numbers and multiply common prime factors. Everyone knows about the greatest common divisor(gcd). It is the greatest number (say *N*) that divides both numbers *a* and *b* without leaving a remainder. Numbers *a* and *b* are called co-prime when they satisfy gcd(a,b)=1. The below image will give you an exact idea of how this Euclid's gcd algorithm (This algorithm assumes a > 0 and b > 0) is working.

Basic Euclidean Algorithm for GCD:
The algorithm is based on the below facts.

- If we subtract a smaller number from a larger one (we reduce a larger number), GCD doesn't change. So if we keep subtracting repeatedly the larger of two, we end up with GCD.
- Now instead of subtraction, if we divide the smaller number, the algorithm stops when we find the remainder 0.



A much more efficient method is the Euclidean algorithm, which uses a division algorithm such as long division in combination with the observation that the gcd of two numbers also divides their difference. To compute gcd(48,18), divide 48 by 18 to get a quotient of 2 and a remainder of 12. Then divide 18 by 12 to get a quotient of 1 and a remainder of 6. Then divide 12 by 6 to get a remainder of 0, which means that 6 is the gcd. Note that we ignored the quotient in each step except to notice when the remainder reached 0, signalling that we had arrived at the answer. Formally the algorithm can be described as:

$$\gcd(a,0) = a$$
$$\gcd(a,b) = \gcd(b, a \bmod b),$$

where

$$a \bmod b = a - b\left\lfloor \frac{a}{b} \right\rfloor.$$

If the arguments are both greater than zero then the algorithm can be written in more elementary terms as follows:

$$\gcd(a,a) = a,$$
$$\gcd(a,b) = \gcd(a-b,b) \quad \text{, if } a > b$$
$$\gcd(a,b) = \gcd(a,b-a) \quad \text{, if } b > a$$

Euclid's Algorithm is used to find the Greatest Common Divisor (GCD) of two numbers a and n, a<n, use fact if a and b have divisor d so does a-b, a-2b

GCD (a,n) is given by:

let g0=n
g1=a
gi+1 = gi-1 mod gi
whengi=0 then (a,n) = gi-1
eg find (56,98)
g0=98
g1=56
g2 = 98 mod 56 = 42
g3 = 56 mod 42 = 14
g4 = 42 mod 14 = 0
hence (56,98)=14

Extended Euclidean Algorithm-

The Extended Euclidean Algorithm follows the same steps as Euclid's algorithm but with some extra information to be tracked. Extended Euclidean algorithms are widely used in Cryptography, especially in calculating the Modulo Inverse Multiplicative (when integers *a* and *b* are co-prime) for deriving the key pairs of RSA public-key encryption.

The extended Euclidean algorithm updates the results of gcd(a, b) using the results calculated by the recursive call gcd(b%a, a). Let values of x and y calculated by the recursive call be x_1 and y_1. x and y are updated using the below expressions.

Input: $a = 35, b = 15$

Output: $gcd = 5, x = 1, y = -2$

$ax + by = gcd(a, b)$

$gcd(a, b) = gcd(b\%a, a)$

$gcd(b\%a, a) = (b\%a)x_1 + ay_1$

$ax + by = (b\%a)x_1 + ay_1$

$ax + by = (b - [b/a] * a)x_1 + ay_1$

$ax + by = a(y_1 - [b/a] * x_1) + bx_1$

Comparing LHS and RHS,

$x = y_1 - [b/a] * x_1$

$y = x_1$

Output :

```
gcd(35, 15) = 5
```

How is Extended Algorithm Useful?

The extended Euclidean algorithm is particularly useful when a and b are coprime (or gcd is 1). Since x is the modular multiplicative inverse of "a modulo b", and y is the modular multiplicative inverse of "b modulo a". In particular, the computation of the modular multiplicative inverse is an essential step in RSA public-key encryption method.

Unlike normal integer arithmetic, sometimes a number in modular arithmetic has a unique inverse

a -1 is inverse of a mod n if a.a -1 = 1 mod n

wherea,x in {0,n-1} o eg 3.7 = 1 mod 10

if (a,n)=1 then the inverse always exists can extend Euclid's Algorithm to find Inverse by keeping track of gi = ui .n + vi .a ·

Extended Euclid's (or Binary GCD) Algorithm to find Inverse of a number a mod n (where (a,n)=1) is:

Inverse(a,n) is given by:

g0=n u0=1 v0=0
g1=a u1=0 v1=1
Let
t y = gi-1 div gi
gi+1 = gi-1 - y.gi = gi-1 mod gi

ui+1 = ui-1 - y.ui
vi+1 = vi-1 - y.vi

when gi=0 then Inverse(a,n) = vi-1

Example
eg: want to find Inverse(3,460):
i y g u v
0 - 460 1 0
1– 3 0 1
2 153 1 1 -153
3 3 0 -3 460
hence Inverse(3,460) = -153 = 307 mod 460

Classical Encryption Techniques

Substitution Techniques

Substitution ciphers-

In cryptography, a substitution cipher is a method of encrypting by which units of plaintext are replaced with ciphertext, according to a fixed system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing the inverse substitution.

Substitution ciphers can be compared with transposition ciphers. In a transposition cipher, the units of the plaintext are rearranged in a different and usually quite complex order, but the units themselves are left unchanged. By contrast, in a substitution cipher, the units of the plaintext are retained in the same sequence in the ciphertext, but the units themselves are altered. A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with cipher text bit patterns.

There are a number of different types of substitution cipher.

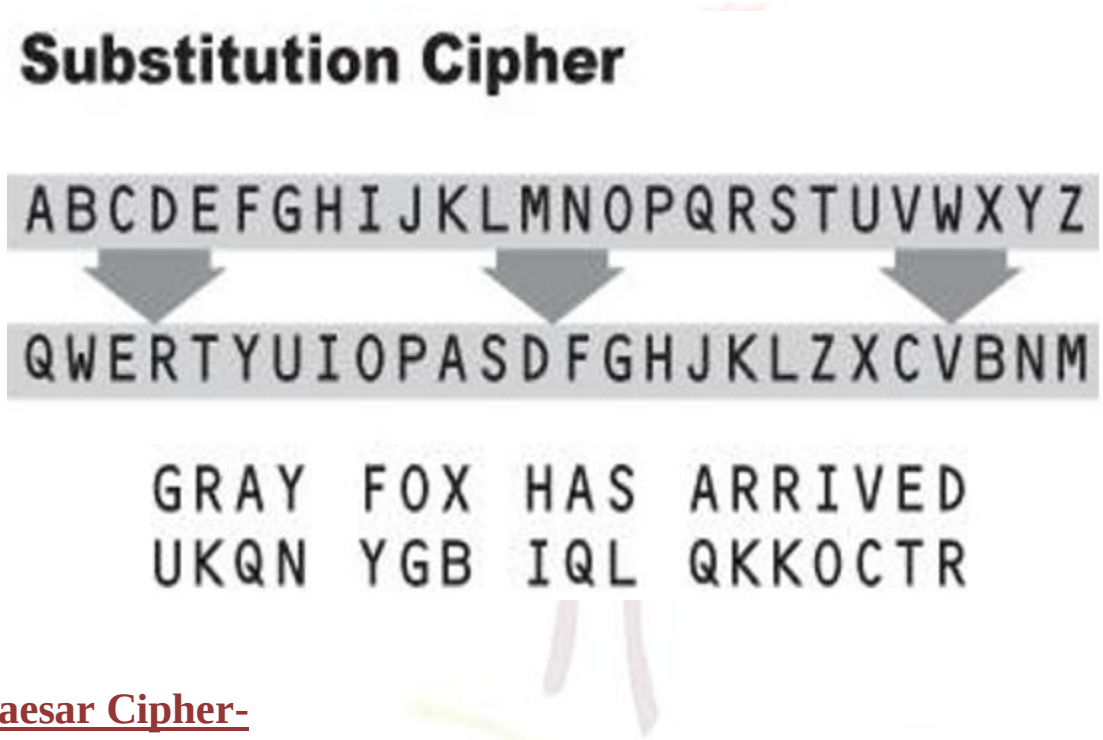
There are several types of substitution cryptosystems:

- Simple substitution cipher; a cipher that operates on larger groups of letters is termed poly-graphic.
- Mono-alphabetic substitution involves replacing each letter in the message with another letter of the alphabet uses fixed substitution over the entire message.
- Poly-alphabetic substitution involves using a series of mono-alphabetic ciphers that are periodically reused cipher uses a number of substitutions at different positions in the message,

where a unit from the plaintext is mapped to one of several possibilities in the ciphertext and vice versa.

- Homophonic substitution makes it possible to have each letter of the plaintext message correspond to a possible group of other characters.
- Poly-graphic substitution involves replacing a group of characters in the message with another group of characters.

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



Caesar Cipher-

Caesar cipher (shift cipher) is the earliest known use of a substitution cipher and the simplest was by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

e.g., plain text : pay more money

Cipher text: SDB PRUH PRQHB

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

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

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

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

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

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

The decryption algorithm is simply

P = D(C) = (C-k) mod 26

- The Caesar cipher is a simple encryption technique that was used by Julius Caesar to send secret messages to his allies. It works by shifting the letters in the plaintext message by a certain number of positions, known as the "shift" or "key".

- 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.

- For example, if the shift is 3, then the letter A would be replaced by the letter D, B would become E, C would become F, and so on. The alphabet is wrapped around so that after Z, it starts back at A.

- Here is an example of how to use the Caesar cipher to encrypt the message "HELLO" with a shift of 3:

1. Write down the plaintext message: HELLO
2. Choose a shift value. In this case, we will use a shift of 3.
3. Replace each letter in the plaintext message with the letter that is three positions to the right in the alphabet.

   H becomes K (shift 3 from H)

   E becomes H (shift 3 from E)

   L becomes O (shift 3 from L)

   L becomes O (shift 3 from L)

   O becomes R (shift 3 from O)

4. The encrypted message is now "KHOOR".

- To decrypt the message, you simply need to shift each letter back by the same number of positions. In this case, you would shift each letter in "KHOOR" back by 3 positions to get the original message, "HELLO".

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:

- It is not secure against modern decryption methods.
- Vulnerable to known-plaintext attacks, where an attacker has access to both the encrypted and unencrypted versions of the same messages.
- The small number of possible keys means that an attacker can easily try all possible keys until the correct one is found, making it vulnerable to a brute force attack.
- It is not suitable for long text encryption as it would be easy to crack.
- It is not suitable for secure communication as it is easily broken.
- Does not provide confidentiality, integrity, and authenticity in a message.

Features of Caesar cipher:

1. Substitution cipher: The Caesar cipher is a type of substitution cipher, where each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

2. Fixed key: The Caesar cipher uses a fixed key, which is the number of positions by which the letters are shifted. This key is known to both the sender and the receiver.

3. Symmetric encryption: The Caesar cipher is a symmetric encryption technique, meaning that the same key is used for both encryption and decryption.

4. Limited keyspace: The Caesar cipher has a very limited keyspace of only 26 possible keys, as there are only 26 letters in the English alphabet.

5. Vulnerable to brute force attacks: The Caesar cipher is vulnerable to brute force attacks, as there are only 26 possible keys to try.

6. Easy to implement: The Caesar cipher is very easy to implement and requires only simple arithmetic operations, making it a popular choice for simple encryption tasks.

Rules for the Caesar Cipher:

1. Choose a number between 1 and 25. This will be your “shift” value.
2. Write down the letters of the alphabet in order, from A to Z.
3. Shift each letter of the alphabet by the “shift” value. For example, if the shift value is 3, A would become D, B would become E, C would become F, and so on.
4. Encrypt your message by replacing each letter with the corresponding shifted letter. For example, if the shift value is 3, the word “hello” would become “khoor”.

5. To decrypt the message, simply reverse the process by shifting each letter back by the same amount. For example, if the shift value is 3, the encrypted message “khoor” would become “hello”.

Algorithm for Caesar Cipher:
Input:
1. Choose a shift value between 1 and 25.
2. Write down the alphabet in order from A to Z.
3. Create a new alphabet by shifting each letter of the original alphabet by the shift value. For example, if the shift value is 3, the new alphabet would be:
4. 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
   D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
5. Replace each letter of the message with the corresponding letter from the new alphabet. For example, if the shift value is 3, the word “hello” would become “khoor”.
6. To decrypt the message, shift each letter back by the same amount. For example, if the shift value is 3, the encrypted message “khoor” would become “hello”.

Procedure:
- Traverse the given text one character at a time .
- For each character, transform the given character as per the rule, depending on whether we’re encrypting or decrypting the text.
- Return the new string generated.

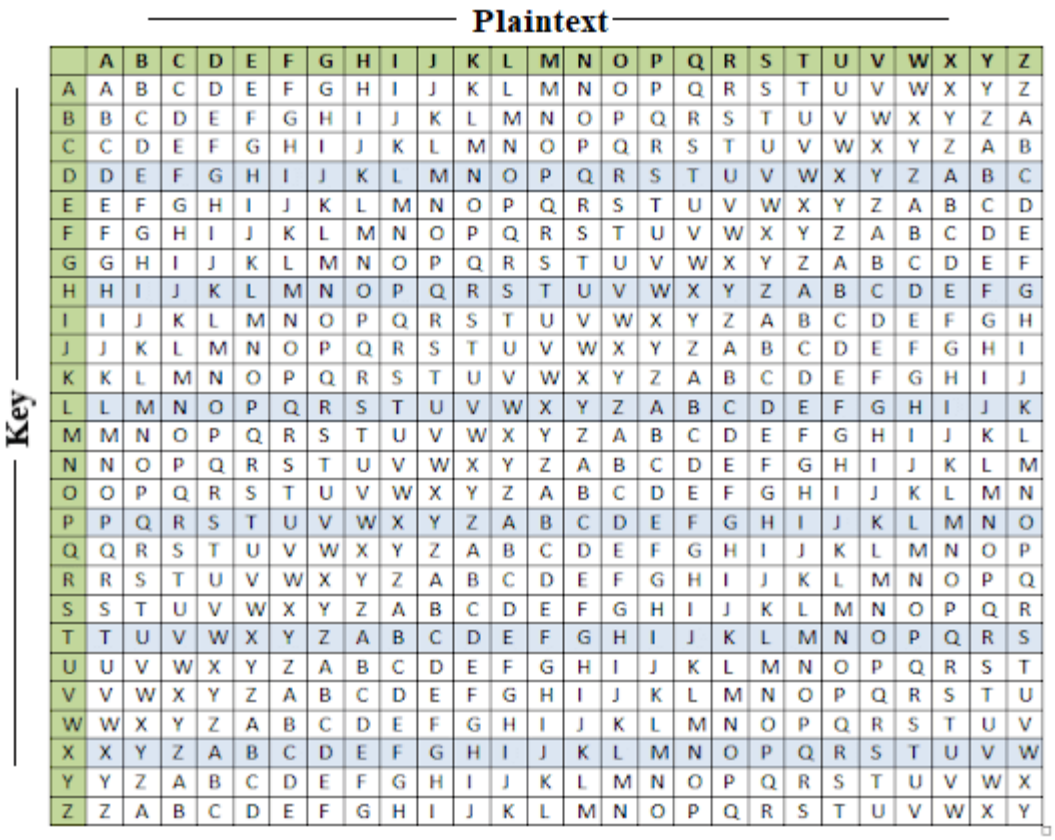
Vigenere’s Cipher-

The vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement. This algorithm was first described in 1553 by Giovan Battista Bellaso**.** It uses a Vigenere table or Vigenere square for encryption and decryption of the text. The vigenere table is also called the tabula recta.

Two methods to perform the vigenere cipher.

Method 1-

When the vigenere table is given, the encryption and decryption are done using the vigenere table (26 * 26 matrix) in this method.



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

Example: The plaintext is "JAVATPOINT", and the key is "BEST".

To generate a new key, the given key is repeated in a circular manner, as long as the length of the plain text does not equal to the new key.

J	A	V	A	T	P	O	I	N	T
B	E	S	T	B	E	S	T	B	E

Encryption

The first letter of the plaintext is combined with the first letter of the key. The column of plain text "J" and row of key "B" intersects the alphabet of "K" in the vigenere table, so the first letter of ciphertext is "K".

Similarly, the second letter of the plaintext is combined with the second letter of the key. The column of plain text "A" and row of key "E" intersects the alphabet of "E" in the vigenere table, so the second letter of ciphertext is "E".

This process continues continuously until the plaintext is finished.

Ciphertext = KENTUTGBOX

Decryption

Decryption is done by the row of keys in the vigenere table. First, select the row of the key letter, find the ciphertext letter's position in that row, and then select the column label of the corresponding ciphertext as the plaintext.

K	E	N	T	U	T	G	B	O	X
B	E	S	T	B	E	S	T	B	E

For example, in the row of the key is "B" and the ciphertext is "K" and this ciphertext letter appears in the column "J", that means the first plaintext letter is "J".

Next, in the row of the key is "E" and the ciphertext is "E" and this ciphertext letter appears in the column "A", that means the second plaintext letter is "A".

This process continues continuously until the ciphertext is finished.

Plaintext = JAVATPOINT

Method 2-

When the vigenere table is not given, the encryption and decryption are done by Vigenar algebraically formula in this method (convert the letters (A-Z) into the numbers (0-25)).

Formula of encryption is,

$$E_i = (P_i + K_i) \bmod 26$$

Formula of decryption is,

$$D_i = (E_i - K_i) \bmod 26$$

If any case (D_i) value becomes negative (-ve), in this case, we will add 26 in the negative value.

Where,

E denotes the encryption.

D denotes the decryption.

P denotes the plaintext.

K denotes the key.

Example: The plaintext is "JAVATPOINT", and the key is "BEST".

Encryption: $E_i = (P_i + K_i) \bmod 26$

Plaintext	J	A	V	A	T	P	O	I	N	T
Plaintext value (P)	09	00	21	00	19	15	14	08	13	19
Key	B	E	S	T	B	E	S	T	B	E
Key value (K)	01	04	18	19	01	04	18	19	01	04
Ciphertext value (E)	10	04	13	19	20	19	06	01	14	23
Ciphertext	K	E	N	T	U	T	G	B	O	X

Decryption: $D_i = (E_i - K_i) \bmod 26$

If any case (Di) value becomes negative (-ve), in this case, we will add 26 in the negative value. Like, the third letter of the ciphertext;

N = 13 and S = 18

$D_i = (E_i - K_i) \bmod 26$

$D_i = (13 - 18) \bmod 26$

$D_i = -5 \bmod 26$

$D_i = (-5 + 26) \bmod 26$

$D_i = 21$

Ciphertext	K	E	N	T	U	T	G	B	O	X
Ciphertext value (E)	10	04	13	19	20	19	06	01	14	23
Key	B	E	S	T	B	E	S	T	B	E
Key value (K)	01	04	18	19	01	04	18	19	01	04
Plaintext value (P)	09	00	21	00	19	15	14	08	13	19
Plaintext	J	A	V	A	T	P	O	I	N	T

Playfair Cipher-

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.
It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

The Playfair Cipher Encryption Algorithm:
The Algorithm consists of 2 steps:

1. Generate the key Square(5×5):
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.
   - The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
2. Algorithm to encrypt the plain text: The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

For Example-

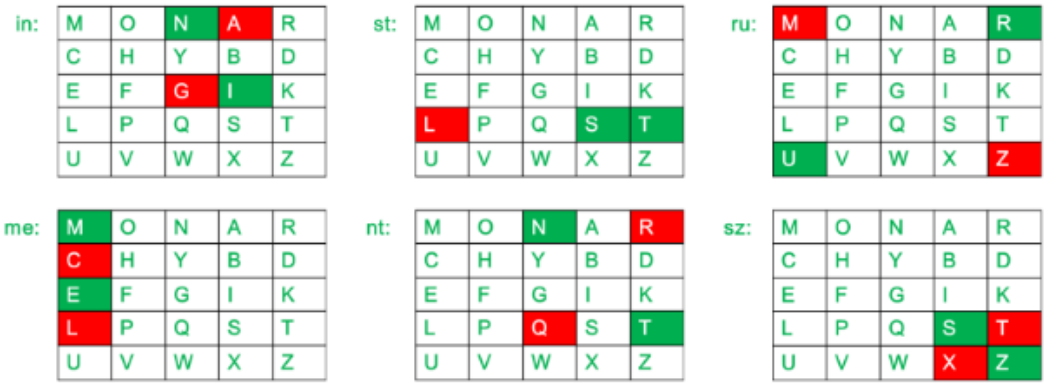
PlainText: "instruments"
After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'

1. Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.
Plain Text: “hello”
After Split: ‘he’ ‘lx’ ‘lo’
Here ‘x’ is the bogus letter.

2. If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter
Plain Text: “helloe”
AfterSplit: ‘he’ ‘lx’ ‘lo’ ‘ez’
Here ‘z’ is the bogus letter.

Rules for Encryption:

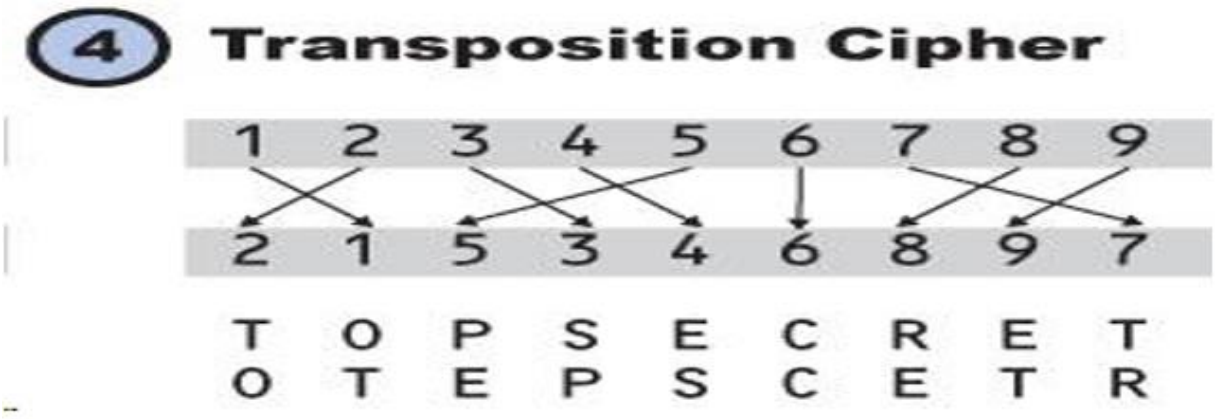
- If both the letters are in the same column: Take the letter below each one (going back to the top if at the bottom).
- If both the letters are in the same row: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
- If neither of the above rules is true: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.



Transposition Techniques-

In cryptography, a transposition cipher is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed (the plaintext is reordered). Mathematically a bijective function is used on the characters' positions to encrypt and an inverse function to decrypt.

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



Rail fence-

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

Plaintext = meet at the school house

This technique is a type of Transposition technique and does is write the plain text as a sequence of diagonals and changing the order according to each row.

It uses a simple algorithm,

1. Writing down the plaintext message into a sequence of diagonals.
2. Row-wise writing the plain-text written from above step.

Rail Fence cipher

- eg. write message out as: (depth=3)

  m e m a t r h t g p r y

  e t e f e t e o a a t

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

- giving ciphertext

  mtaehoptemfregayeetttar

To encipher this message with a rail fence of depth 3, we write the message as follows:

m e a t e c o l o s

e t t h s h o h u e

The encrypted message is

MEATECOLOSETTHSHOHUE

Row Transposition Ciphers-

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

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

Key = 4 3 1 2 5 6 7

PT = m e e t a t t

h e s c h o o

l h o u s e

CT = ESOTCUEEHMHLAHSTOETO

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

Row Transposition Ciphers

- Plaintext is written row by row in a rectangle.
- Ciphertext: write out the columns in an order specified by a key.

Key: 3 4 2 1 5 6 7

a	t	t	a	c	k	p
o	s	t	p	o	n	e
d	u	n	t	i	l	t
w	o	a	m	x	y	z

Plaintext:

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

Vernam Cipher (One Time Pad) -

One Time Pad Cipher is an unbreakable cryptosystem. It represents the message as a sequence of 0s and 1s. this can be accomplished by writing all numbers in binary, for example, or by using ASCII. The key is a random sequence of 0‟s and 1‟s of same length as the message. Once a key is used, it is discarded and never used again. The Vernam Cipher has a specific subset one-time pad, which uses input ciphertext as a random set of non-repeating character. The thing to notice here is that, once an input cipher text gets used it will never be used again hence one-time pad and length of cipher-text is the size that of message text. One time pad should be discarded after every single use and this technique is proved highly secure and suitable for small messages but illogical if used for long messages.

Algorithm:

1. Plain text character will be represented by the numbers as A=0, B=1, C=2,... Z=25.
2. Add each corresponding number of a plain text message to the input cipher text alphabet numbers.
3. If the sum is greater than or equal to 26, subtract 26 from it.
4. Translate each number back to corresponding letters and we got our cipher text.

The system can be expressed as Follows:

Ci = Pi+ Ki Ci - ith binary digit of cipher text Pi - ith binary digit of plaintext Ki - ith binary digit of key

Exclusive OR operation

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

Pi = CiKi +

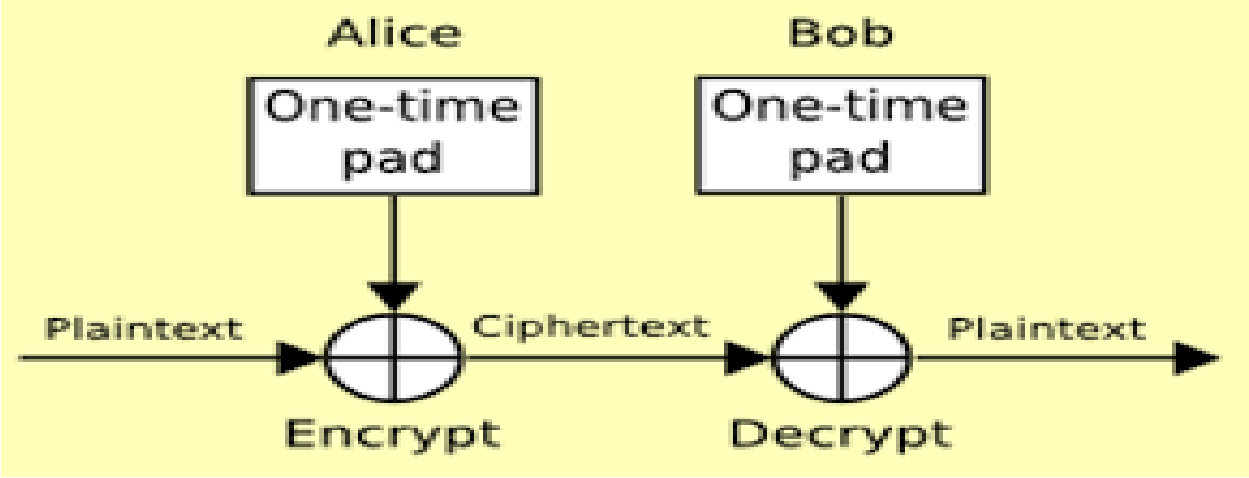
e.g., plaintext = 0 0 1 0 1 0 0 1

Key = 1 0 1 0 1 1 0 0 23

---------- ciphertext = 1 0 0 0 0 1 0 1

- Advantage: Encryption method is completely unbreakable for a ciphertext only attack.
- Disadvantages It requires a very long key which is expensive to produce and expensive to transmit.

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



Example:

Our message is **"INCLUDEHELP"** and input cipher text is **"ATQXRZWOBYV"**

	I	N	C	L	U	D	E	H	E	L	P
Plain text:	8	13	2	11	20	3	4	7	4	11	15
One-time pad:	0	19	16	23	17	25	22	14	1	24	21
	A	T	Q	X	R	Z	W	O	B	Y	V
Initial Total:	8	32	18	34	37	28	26	21	5	35	36
Subtract 26, if >25:	8	6	18	8	11	2	0	21	5	9	10
Cipher Text:	I	G	S	I	L	C	A	V	F	J	K

Example of Vernam Cipher

Book/Running-Key Cipher-

This technique also (incorrectly) known as running key cipher. This technique very simple and similar to our previous Vernam Cipher. For getting a cipher, some portion of text from a book is used as a one-time pad, rest it works in same way as Vernam cipher does.