Network Security & Cyber laws

Unit #1 : Introduction

Information Era : Information is an asset that has a value like any other asset. ∴ Information needs to be secured from attacks.

To secure, Infn need to hidden from unauthorized access - confidentiality; protected from unauthorized change - integrity and available to an authorized entity when it is needed - availability.

In early days, the infn collected & stored on physical files in an organization.

> Restricting the access to a few authorized & trusted people. i.e. Confidentiality

> only a few authorized people allowed to change the contents of the files - Integrity

> Designated person have right to access the files at all times - availability.

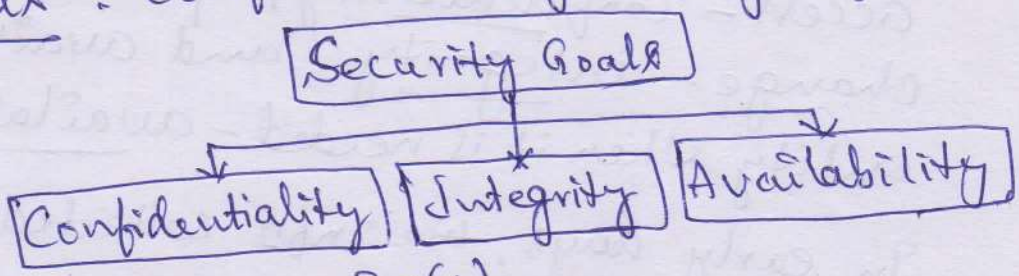
Now a days, information storage became electronic with the help of Computers. However the three security requirements did not change. The files stored in Computers require confidentiality, integrity and availability. Implementation of these requirements is different and more challenging.

* Computer Networks created a revolution in the use of information
  - distributed
  - Send and Receive from/to any distance

But these three requirements have not changed.

* maintain Confidentiality when info transmitted from one Computer to another.
* attacks can threaten these three security goals
* Provide Security Services and
* Implement mechanisms / Techniques

Security Goals : Confidentiality, Integrity & availability.



fig(1)

Confidentiality: is the most common aspect of info Security. To maintain Confidential info, organizations needs to guard against malicious actions

Ex: military: Sensitive info
Industry: Hiding some info from competitors
Banking: Customer's accounts need to be kept secret.

Confidentiality is not only applies to the storage of info, it also applies to the transmission of info. (need to store info in remote Computer, retrieve info from remote Computer)

Integrity: Information needs to be changed constantly.

Ex: Bank: customers deposit, credit or withdraw money, Integrity means that changes need to be done only by authorized entities and through authorized mechanisms. (not only by malicious act, interrupt in the system, but power surge

power surge may also create unwanted changes on some infn >

Availability: The infn created and stored by an organizat -ion needs to be available to authorized entities.
Infn is useless if it not available.
The unavailibity of infn is harmful for an organization as the lack of confidentiality, or integrity.
Ex: Bank: Customers could not access thier accounts for transactions.

Cryptographic Attacks: There are categories of attacks: 1) cryptanalytic 2) Non-cryptanalytic.

Cryptanalytic Attacks: These attacks are combinations of statistical and algebraic techniques and find out the secret key of a cipher. These methods are used to find the mathematical properties of the cryptographic algorithms, finding distinguishers of the output distribution of cryptographic algorithms from uniform distributions.

* Cryptographic algs - for manage distribution and convert it into ciphertext distribution using the key.
* objective of crypto-analysis - to find properties of the cipher which does not emit in a random fn.
i.e. distinguishers, all attacks are fundamentally distinguishers.
The attacker guesses the key and looks for the distinguishing properties.

If the property is detected, the guess is correct, otherwise the next guess is tried.

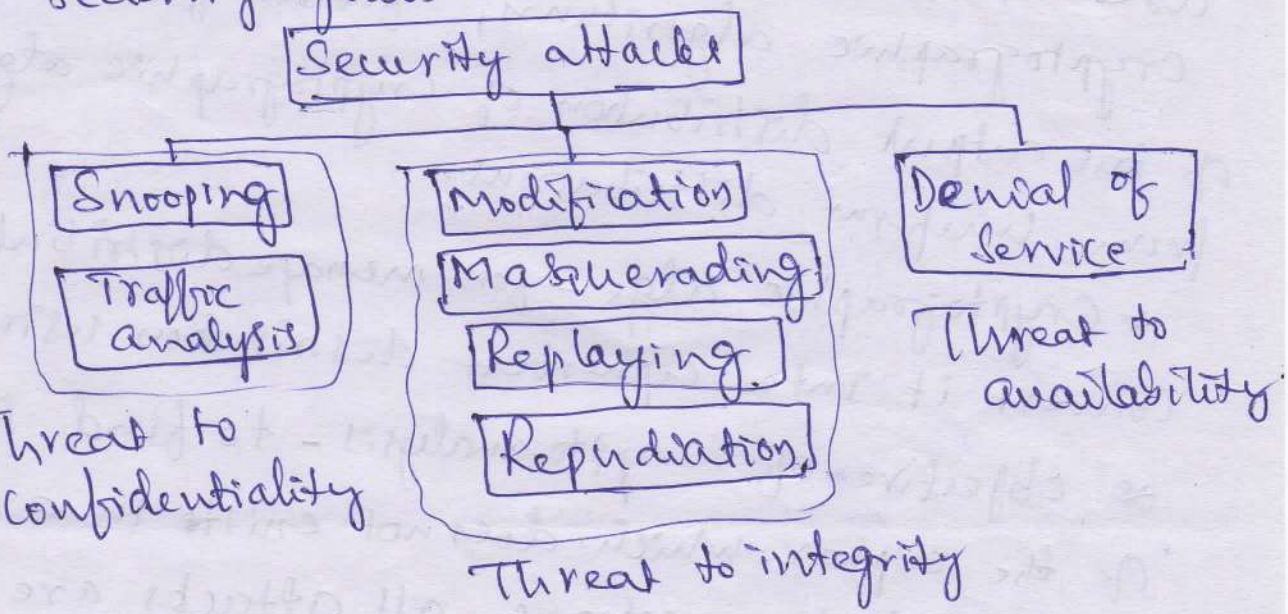
Efficient attacks will try to adopt a "divide and conquer" strategy to reduce the complexity of guessing the key. If attack is theoretically successful, the complexity is totally reduced. attack is practically infeasible.

Non-Cryptanalytic Attacks

* These types of attacks don't exploit the mathematical weakness of the cryptographic algorithms.

* Three goals of Security (confidentiality, Integrity and availability) are very much threatened by this class of attacks.

We can categories attacks into three groups based on the security goals.



fig(2) Taxonomy of attacks with relation to Security goals.

Attacks Threatening Confidentiality

1) Snooping: unauthorized access to or interception of data
Ex: a file transferred through the Internet
[confidential info]
* Data can be made non-intelligible to the interceptor by using encipherment Techniques to prevent Snooping.

2) Traffic Analysis: Attackers might be obtain info by monitoring online traffic
Ex: Find the e-mail address of the sender/Receiver
* Collecting pairs of responses & requests to guess the nature of transaction.

Attacks Threatening Integrity

1) Modification: After intercepting or accessing info, the attacker modifies the info to make it beneficial to themself.
Ex: a customer sends a message to a bank to do some transaction. The attacker intercepts the message & changes the type of transaction.
Some times attackers delete / delays the message to harm

2) Masquerading (Spoofing): happens when the attacker impersonates somebody else.
Ex: an attacker might steal the bank card and PIN of a bank customer & pretend that he is the customer.
* a user tries to contact a bank, but another site pretends that it is the bank & obtain some info from the user.

3) Replaying: The attacker obtain a copy of a message sent by a user and later tries to replay it.

Ex: a person sends a request to bank to ask for payment, the attacker intercepts the message & send it again to receive another payment from the bank.

4) Repudiation: It is performed by one of the two parties in the communication: the sender or the receiver The Sender of the message may be later deny that he has sent the message; the receiver of the message later deny that he has received the message.

Ex: Denial by the sender: bank customer asking bank to send some money to a third party but later denying that he has made such a request

Denial by the receiver may occur when a person buys a product from a manufacturer and pays for it electronically, but the manufacturer later denies having received the payment and asks to be paid.

Attacks Threatening availability

Denial Of Service: is a very common attack. It may slow down or totally interrupt the service of a system. The attacker can use several strategies to achieve this. The attacker send so many bogus requests to server that the server crashes because of the heavy load.

The attacker intercept & delete a server's response to a client, making the client to believe that the server is not responding.

> intercept requests from the client, causing the clients to send requests many times & overload the system.

Passive vs Active attacks

Generally categorize the attacks into two groups: passive and active.

Passive attacks: the attacker's goal is just to obtain information, i.e. does not modify data or harm the system.
- attacks may harms the sender or the receiver
- Attacks that threaten confidentiality – snooping and traffic analysis are passive attacks.
- the system is not affected.
- It is difficult to detect this type of attack until the sender or receiver finds out about the leaking of confidential info.

Active attacks: may change the data or harm the system. Attacks that threaten the integrity & availability are active attacks. Active attacks easier to detect than to prevent, because an attacker can launch them in a variety of ways.

Attacks	passive/active	Threatening
Snooping Traffic analysis	passive	Confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of Service	Active	Availability

Categorization of passive & active attacks

Services and Mechanism

* ITU-T provides some security services to implement. (International Telecomm Union - Telecomm standardization Sector)

Security services: ITU-T (x.800) defined five services

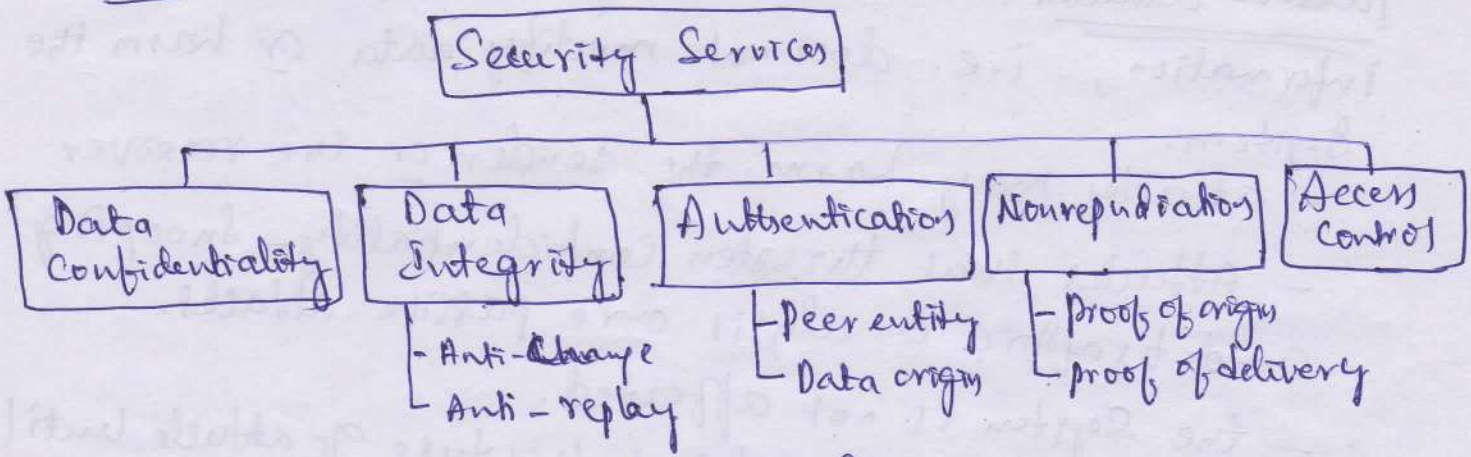


Fig (3) Security services.

These services are related to one or more security goals. & these services have designed to prevent attacks.

* Data Confidentiality is designed to protect data from disclosure attacks., this service is defined broadly and protection against traffic analysis., prevent snooping & traffic analysis attack.

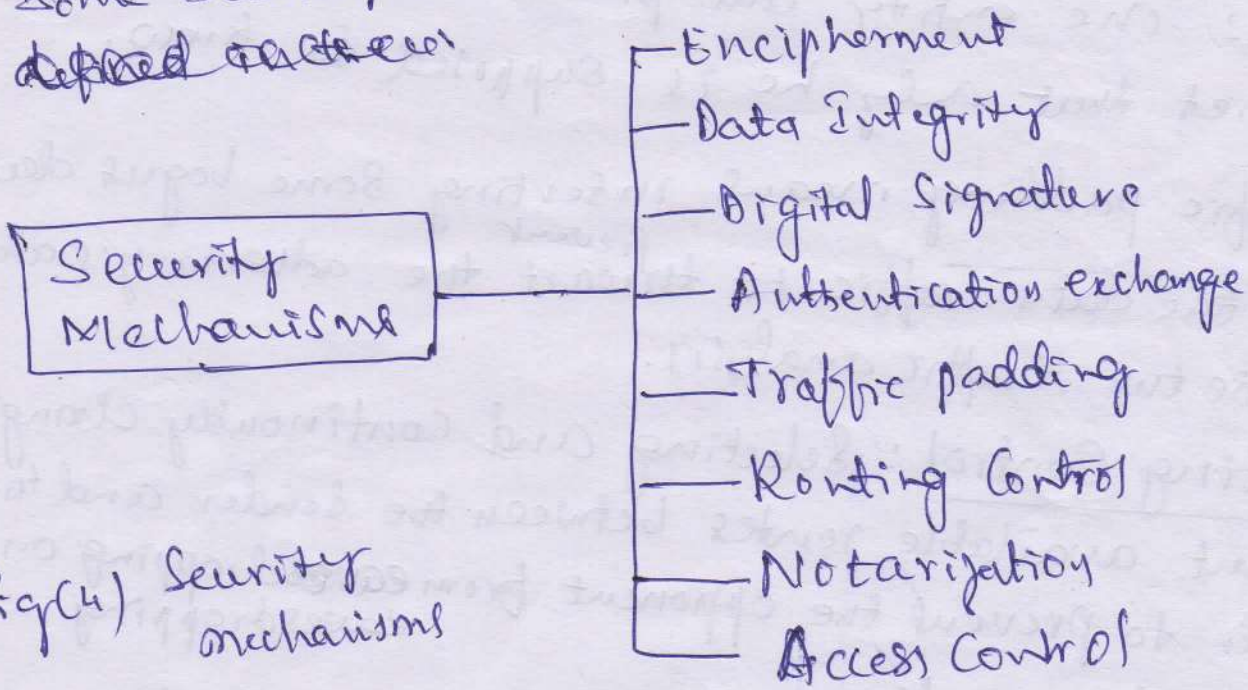
* Data Integrity - to protect data from modification, insertion, deletion and replaying. It may protect whole message or part of the message

* Authentication - provides the authentication of the user at the other end of the line., provides & authen-tication of the sender or receiver during the connection establishment (peer entity authentication). In Connection-oriented service. It authenticates the source of the data (data origin authentication) In Connection-less service.

* Nonrepudiation service protects against repudiation by either the sender or the receiver.
  - receiver can later prove the identity of the sender if the sender denied.

* Access Control provides protection against unauthorized access to data. It is involve reading, writing, modification, executing programs and so on.

Security Mechanism : ITU-T (X.800) also recommends some security mechanisms to provide the security services defined earlier.



fig(4) Security mechanisms

* Encipherment hiding or covering data, provide confidentiality. cryptography & Steganography are used for enciphering.
* Data Integrity mechanism appends to the data a short check value that has been created by a specific process from the data itself. Receiver receives the data with check value, calculate a new checkvalue, compare the new checkvalue with the received data.

If the two check values are the same, the integrity

of data has been preserved.

* Digital Signature – the sender electronically sign the data & receiver electronically verify the signature. Sender has a private key and announced it publicly. The receiver uses this public key to prove that the message is indeed signed by the sender

* Authentication Exchange: Two entity exchange some messages to prove thier identity to each other.
  Ex: one entity can prove that he knows a secret that only he is supposed to know.

* Traffic padding means inserting some bogus data into the data traffic to prevent thwart the adversary's attempt to use the traffic analysis.

* Routing Control: Selecting and continously changing different available routes between the sender and the receiver to prevent the opponent from ~~eaves~~dropping on eavesdropping a particular route.

* Notarigation – Selecting a ~~trusted~~ third ~~trusted~~ party to control the comm2 between two entities.
  Ex: to prevent repudiation. The receiver can involve a trusted party to store the sender request in order to prevent the sender from later denying that he made such a request.

* Access Control uses methods to prove that a user has access right to the data or resources owned by a system. Ex: passwords and PINs.

Relation between Services and mechanism.

Security Service	Security mechanism
Data Confidentiality	Encipherment and routing control
Data Integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital sign. Authentication exchange,
Nonrepudiation	Digital Signature, data integrity, & notarization
Access Control	Access control mechanism

Techniques for Security goals Implementation

* The actual implementation of security goals needs some techniques.

Two Techniques are prevalent : 1) Cryptography - in general
2) Steganography (specific)

1. Cryptography :

* the science and art of transforming messages to make them secure and immune to attacks *

The cryptography is defined as involving three distinct mechanisms : i) Symmetric-key encipherment
ii) ASymmetric-key encipherment
iii) Hashing

i) Symmetric-key encipherment (sometimes called Secret-key encipherment or Secret-key cryptography)

one entity send a message to another entity, say Mr. ABC and Mr XYZ assumes the contents of the message cannot understanding by anyone else, simply eavesdropping over the channel.

Mr ABC sends a message (encrypted message), Mr. XYZ decrypt it using decryption alg.

So Symmetric-key encipherment uses a single secret key for both encryption and decryption
In this enciphering, Mr ABC puts the message in a box and locks the box using the shared secret key; Mr X42 unlocks the box with same key and takes out the message.

ii) Asymmetric-key encipherment: (Sometimes called public-key encipherment or public-key cryptography).
- same as the symmetric-key encipherment, with a few exceptions
Here there are two keys: one public key and one private key. To send a secured message, first encrypts the message using Bob's public key. To decrypt the message, receiver uses his own private key.

iii) Hashing: In this case, a fixed-length message digest is created out of a variable-length message. The digest is normally much smaller than the message.
The both message and the digest are ~~send~~ to be sent to receiver. Hashing is used to provide checkvalues (CS).

Steganography: i.e. Secret writing,
> Cryptography means concealing the contents of a message by enciphering; while
> Steganography means concealing the message itself by covering it with something else.

Historical use: History is full of facts and myths about the use of Steganography.
In China, War messages were written on thin pieces of silk and rolled into a small ball and swallowed by the messenger.

In Rome and Greece, messages were carved on pieces of wood, that were later dipped into wax to cover the writing. Invisible inks (onion juice or ammonia salts) were also used to write a secret message between the lines of covering message or on the back of the paper; the secret message was exposed when the paper was heated or treated with another substance.

In recent times, other methods have been devised.
* Harmless message overwritten in a pencil lead, that is visible only when exposed to light at an angle.
* Hide a secret message inside simple message.
Ex: the first or second letter of each word, composed a secret message. Microdots were also used.

Modern use: (Today).

Text, image, audio or video data can be digitized & insert secret binary info into the data during digitization process. It can also used to protect copyright, prevent tampering, or add extra info.

Text Cover: There are several ways to insert binary data into an innocuous (harmless) text.

Ex: use single space between words to represent the binary digit 0 and double space - binary digit 1.
A - hides the 8-bit binary representation (ASCII)
is 01000001, This book ... is mostly about crypto, not st
0 1 0 0 0 0 1

Another more efficient method is to use a dictionary of words organized according to their grammatical usages.

- dictionary containing 2 articles, 8 verbs, 32 nouns and 4 prepositions.

If we agree to use cover text always use sentences with the pattern article-noun-verb-article-noun.

The Secret binary data divided into 16-bit chunks.

> The first bit represented by an article (Ex 0 for a & 1 for the)

> the next five bits by noun (Subject), the next four bits by verb, next bit by the second article. & last five bits by another noun (object).

Ex: the Secret data "Hi", which is 0100100001001001

can be a sentence like A friend called a doctor.

0 10010 0001 0 01001

Image Cover: Secret data can also be covered under a color image. Digitized images are made of pixels (picture elements), in which normally each pixel uses 24 bits (three bytes). Each byte represents one of the primary colors (red, green or blue). ∴ 2^8 different shades of each color. Least-Significant bit (LSB) of each byte is set to zero. This may make the image a little bit lighter in some areas, but it is not normally noticed.

So we can hide a binary data in the image by keeping or changing the LSB. If binary bit is 0, we keep the bit; if it is 1, we change the bit to 1.

In this way we can hide a character in three pixels.

Ex: Three pixels can represent the letter M.

	R	G	B
pixels P_1	01010001	10111110	01010101
P_2	01011110	10111110	01100101
P_3	01111110	01001010	00010101

Other Covers: The Secret menage can cover under audio (Sound | music), and video.

→ the Secret data can be embedded during or before the Compression.

ASCII Binary

M = 077 = 01001101

Symmetric-key Encipherment

To understand the some (three) techniques (Symmetric Assymetric & Hashing), Mathematical Concepts needed.

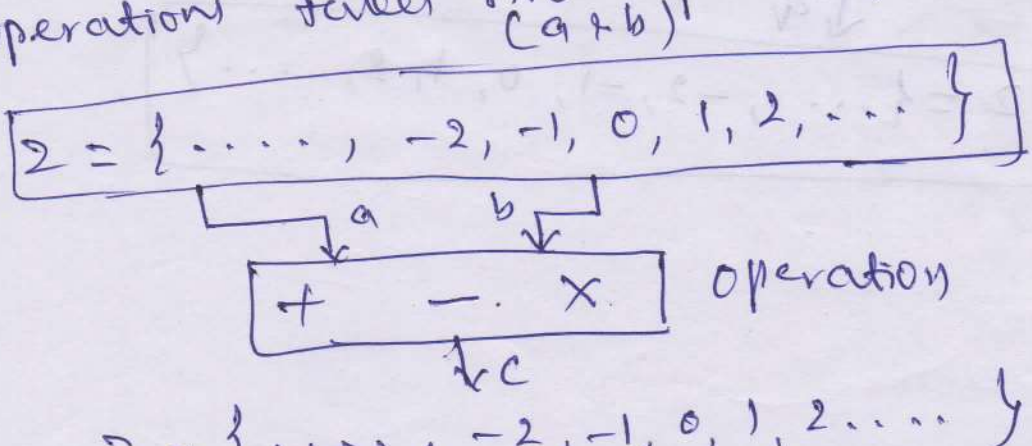
Mathematics of cryptography:

Cryptography is based on some specific areas of mathematics, including number theory, linear algebra and algebraic structures.

Integer Arithmetic: Set of operations, create a background for modular arithmetic.

Set of Integers: (Z), Contains all integral numbers (with no fraction) from negative infinity to positive infinity. $Z = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

Binary operations: In cryptography, three binary opns commonly used to apply to set of integers. Binary opn takes two inputs & produce one output.
Addition, Subtraction, & multiplication, each of these operations takes two inputs & produce one o/p.



fig(2) three binary opns for the set of integers

Ex:

Add: $5+9=14$ $(-5)+9=4$ $5+(-9)=-4$, $(-5)+(-9)=-14$

Subtract: $5-9=-4$ $(-5)-9=-14$ $5-(-9)=14$, $(-5)-(-9)=+4$

Multiply: $5\times9=45$ $(-5)*9=-45$ $5*(-9)=-45$ $(-5)*(-9)=45$

Integer Division: In integer arithmetic, if we divide a by n, we can get q & r.

$a = q \times n + r$. In this relation, a = dividend, q = quotient, n = divisor, r = remainder.

Ex: $a=255$, $n=11$, ∧ they $q=23$, $r=2$.

note: The result of dividing a by n is two integers (q, r).

```
n → 11) 255 (23 → q
    22
    ---
     35
     33
     --
      2 → r
```

$255 = 23 \times 11 + 2$

Two Restrictions: i) divisor is positive integer ($n > 0$)
ii) remainder is non-negative integer ($r \geq 0$).

$Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

$a = q \times n + r$ → r (non negative)

n → positive

q

$Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

Divisibility: It is often encounter in Cryptography.

If a is not zero & let $n=0$, the division relation is $a = q \times n$.

Properties:

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

2: If $a|b$ and $b|a$, then $a = \pm b$

3: If $a|b$ and $b|c$, then $a|c$

4: If $a|b$ and $a|c$, then $a|(m \times b + n \times c)$,

where m & n are arbitrary integers.

Ex: Since $3|15$ and $15|45$, according to the third property, $3|45$.

Since $3|15$ and $3|9$, according to the fourth property $3(15 \times 2 + 9 \times 4)$ which means $3|66$.

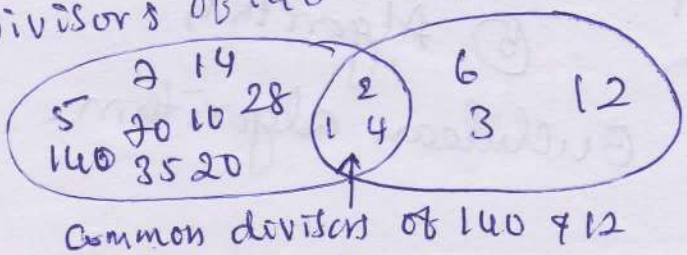
All Divisors: A positive integer can have more than one divisor.

Ex: 32 has six divisors: 1, 2, 4, 8, 16, & 32.

Greatest Common Divisor (GCD): One integer often needed in Cryptography is the GCD of two positive integers. Two positive integers may have many common divisors, but only one greatest common divisor.

Ex: GCD of 12 & 140 are 1, 2, & 4.

4 is the GCD.



"The greatest common divisor of two +ve integers is the largest integer that can divide both integers

Euclidean Algorithm

Finding the GCD of two large +ve integer is not practical. more than 2000 years ago a mathematician, Euclid developed an alg. It is based on the two facts.

1. gcd(a,0) = a
2. gcd(a,b) = gcd(b,r), where r is the remainder of dividing a by b.

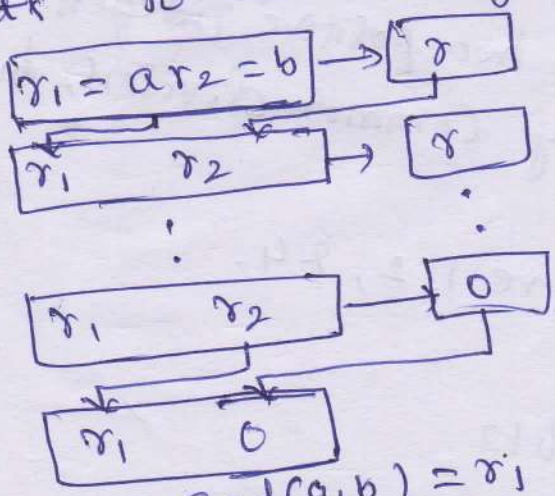
Fact #1: if the second integer is 0, the GCD is the first one.
Fact #2: allows to change the value of a, b until b becomes 0.

Ex: gcd(36,10)

use the second fact several times & the first fact once.

gcd(36,10) = gcd(10,6) = gcd(6,4) = gcd(4,2) = gcd(2,0) = 2.

This means that instead of calculating gcd(36,10), find gcd(2,0). Fig. below shows that above two facts to calculate gcd(a,b).



(a) process

```
r1 ← a; r2 ← b;
{ Initialization }
While (r2 > 0)
{ q = r1/r2;
  r ← r1 − q × r2;
  r1 ← r2;  r2 ← r;
}
gcd(a,b) ← r1
```

(b) Algorithm

Fig (7) Euclidean algorithm.

When $\gcd(a, b) = 1$, a & b are relatively prime.

Ex1: find the greatest common divisor of 2740 & 1760

Initialize $r_1 = 2740$, $r_2 = 1760$

Using Euclidean Algorithm, the following table has been shown the value of q in each step.

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

$\therefore \gcd(2740, 1760) = 20$

Ex2: find the greatest common divisor of 25 & 60

Here first no. is smaller than second number.

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

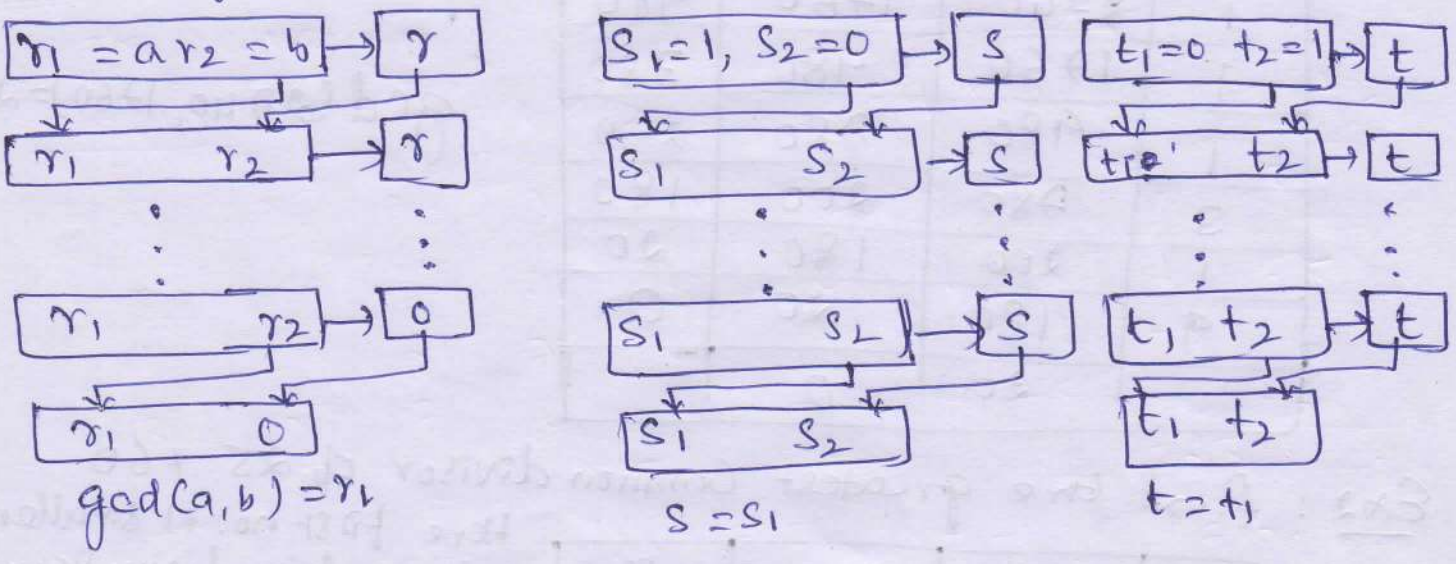
$\therefore \gcd(25, 60) = 5$

The Extended Euclidean Algorithm

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

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

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



fig(a) process.

(b) Algorithm:

{Initialization} $r_1 \leftarrow a$; $r_2 \leftarrow b$; $s_1 \leftarrow 1$, $s_2 \leftarrow 0$;
$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;

```
While(r2 > 0)
{ q ← r1 / r2;
  r ← r1 - q x r2;
  r1 ← r2; r2 ← r;        } updating r's
  s ← s1 - q x s2;
  s1 ← s2; s2 ← s;        } updating s's
  t ← t1 - q x t2;
  t1 ← t2; t2 ← t;        } updating t's
}
gcd(a,b) ← r1;   s ← s1;   t ← t1
```

fig(c) Extended Euclidean algorithm.

The algorithm uses three sets of variables r, s and t's
In each step, r_1, r_2 & r have the same values.
The variables r_1 & r_2 are initialized to a & b, the variables s_1 & s_2 are initialized to 1 & 0, & t_1 & t_2 to 0 & 1.

* r is the remainder of dividing r_1 & r_2
* there is only one quotient q, (r_1/r_2).

Ex-1: Given a = 161 and b = 28, find gcd(a, b) and the values of s & t.

Solution: $r = r_1 - q \times r_2$ $s = s_1 - q \times s_2$ $t = t_1 - q \times t_2$

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

we get gcd(161, 28) = 7, s = -1 & t = 6.
The answer can be tested, (-1) x 161 + 6 x 28 = 7

Ex-2: Given a = 17 and b = 0, find gcd(a, b) and the values of s & t.

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

Note: no calculation for q, r, & s, the first value of r_2 meets termination condition.
So gcd(17, 0) = 17, s = 1 & t = 0, This indicates why we should initialize s_1 to 1 & t_1 to 0, the answer can be tested as shown below

$(1 \times 17) + (0 \times 0) = 17$.

Ex-3: Given $a=0$ and $b=45$, find $\gcd(a,b)$ and the values of s & t.

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

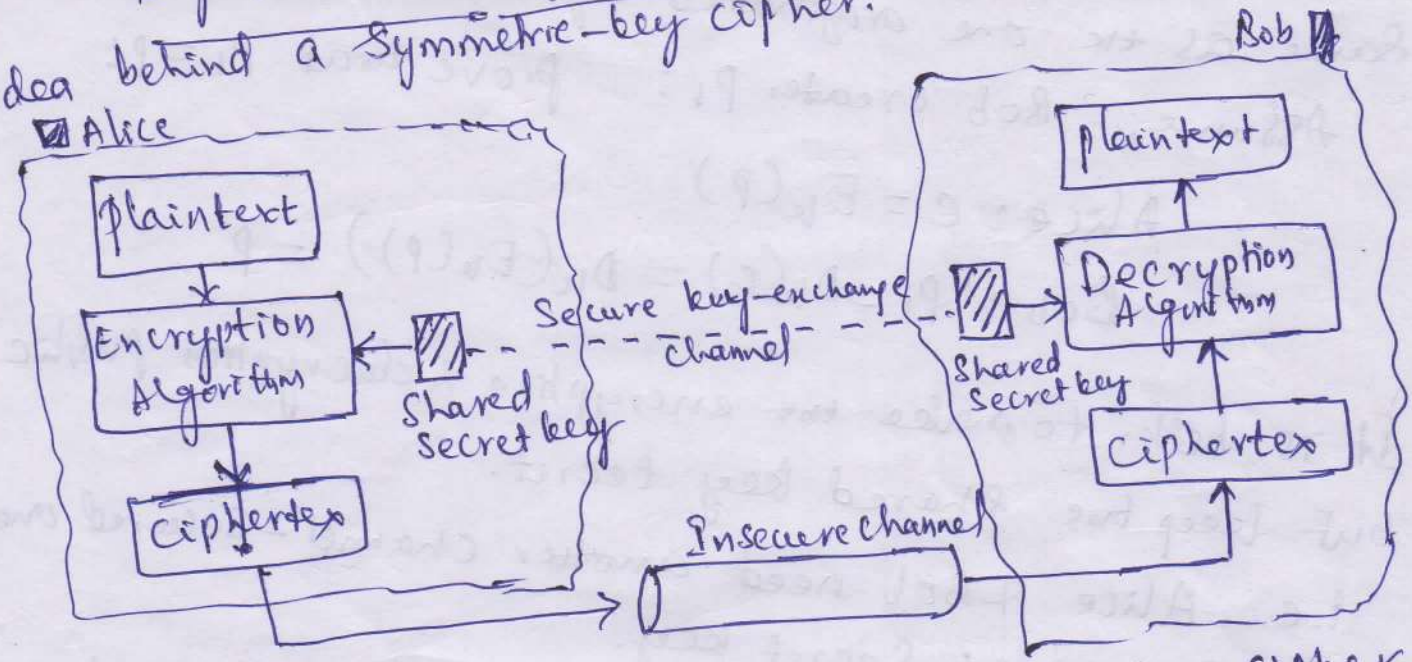
$\therefore \gcd(0,45)=45$, $s=0$ & $t=1$. This indicates why we should initialize s_2 to 0 & t_2 to 1. The answer can be tested: $(0\times 0)+(1\times 45)=45$

Traditional Symmetric-key ciphers

Traditional Symmetric-key ciphers are not used today, but Reasons to study
- they are simpler than modern ciphers and easier to understand.
- they show the basic foundation of Cryptography & one ciphermost
- This helps to understand modern ciphers.

* They provide the rationale for using modern ciphers, because the traditional ciphers can be easily attacked using a computer. These ciphers are no longer secure in this computer age.

Symmetric-key ciphers: fig(1) shows the general idea behind a Symmetric-key cipher.



fig(1) General idea of Symmetric-key cipher

The original message from Alice to Bob is called plaintext. the message that is sent through the channel. is called the ciphertext. To create the ciphertext, using encryption alg. and a shared secret key. (@ sender site)
To create the plaintext, using a decryption alg & the same secret key (@ receiver site)

Symmetric-key encipherment uses a single key for both encryption & decryption.

If P is the plaintext, C is the ciphertext, & k is the key, the encryption Alg $E_k(x)$ — create ciphertext.

decryption Alg $D_k(x)$ — create the plaintext.

$\therefore E_k(x)$ & $D_k(x)$ are inverses of each other.

Encryption: $C = E_k(P)$ Decryption: $P = D_k(C)$.

In which, $D_k(E_k(x)) = E_k(D_k(x)) = x$

We can prove that the plaintext created by Bob is the same as the one originated by Alice.

Assume, Bob creates P_1; prove that $P_1 = P$:

Alice: $C = E_k(P)$

Bob: $P_1 = D_k(C) = D_k(E_k(P)) = P$.

It is better to make the encryption & decryption public but keep the shared key secret.

i.e. Alice & Bob need another channel - secured one. to exchange the secret key.

* Alice & Bob can meet once & exchange the key personally. So secured channel is the face-to-face exchange of the key.
* They can also trust third party
* ~~No. of keys~~: Both directions using the same key Alice to Bob. & Bob to Alice. This is why the method is called Symmetric.

Another element in Symmetric-key encipherment is the no. of keys.

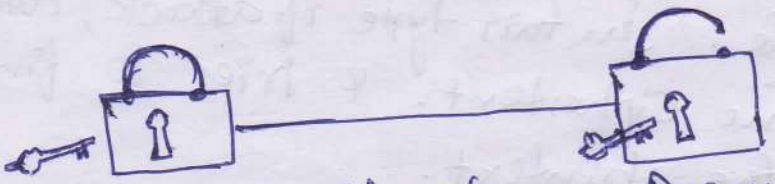
Alice needs another secret key to communicate with another person.

If there are n people in a group who need to communicate with each other, how many keys are needed?

Ans: $(n \times (n-1))/2$

each person needs n-1 keys to communicate with the rest of the group, but the key bet2 A & B can be used in both directions.

In Symmetric-key encipherment, the same key locks and unlocks as shown in fig(2).



Fig(2) Symmetric-key encipherment as locking & unlocking with the same key.

Kerckhoff's principle

Using ciphers to secure & encryption / decryption Algs with secret key keep hide is not recommended.

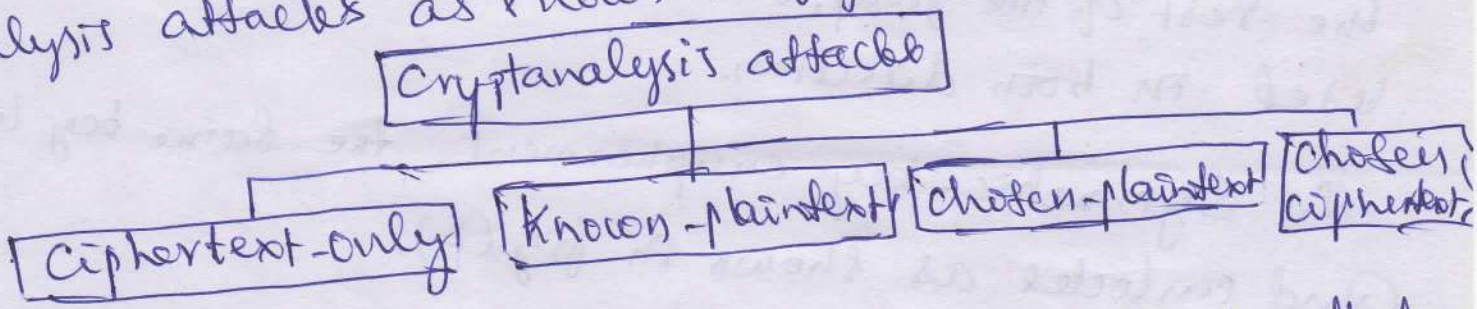
Based on kerckhoff's principle, there is the adversary knows the encryption/decryption Alg. Avoiding the attack based only on the secrecy of the key.

"Guessing the key is so difficult, no need to hide the encryption/decryption Alg."

It is clear that need modern ciphers and key domains for each Alg, large that it makes it difficult for the adversary (opponent) to find the key.

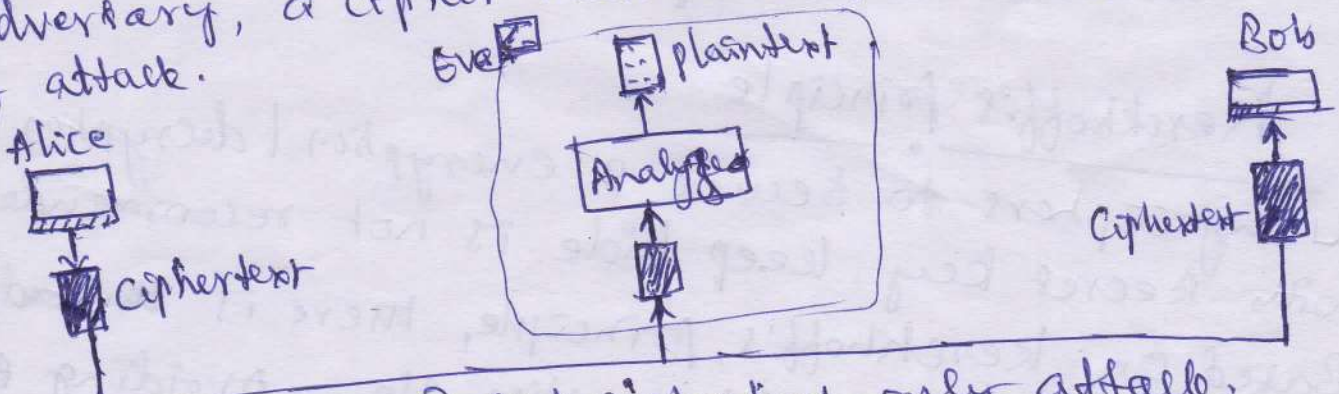
Cryptanalysis : As cryptography is the science & art of creating secret codes, cryptanalysis is the science & art of breaking those codes.

The study of cryptanalysis helps us to create better secret codes. There are four common types of crypta nalysis attacks as shown in fig below.



① Ciphertext-only Attack : In this type of attack, attacker has access to only some ciphertext. & tries to find the corresponding key and the plaintext.

Attacker knows the Alg & can intercept the ciphertext. To thwart (prevent) the decryption of a message by an adversary, a cipher must be very resisting to this type of attack.



Fig(a) ciphertext-only attacks.

Various methods of ciphertext-only attacks

② Brute-force Attack : (Exhaustive-key search method)

In this case, attacker (Eve) tries to use all possible keys. Assume Eve knows the alg & the key domain (list of possible keys)

- Using the intercepted cipher, Eve (attacker) decrypt the ciphertext with every possible key until the plaintext makes sense.

To prevent this type of attack, no. of possible keys must be very large.

(b) Statistical Attack: The Cryptanalyst can benefit from some inherent characteristics of the plaintext language to launch a statistical attack.

Ex: the letter E is the most-frequently used letter in English text.

the cryptanalyst finds the mostly-used character & predicting the corresponding character. After finding a few pairs, the analyst can find the key and use it to decrypt the message.

To prevent this type of attack, the cipher should hide the characteristics of the language.

(c) pattern Attack: Some ciphers hide the characteristics of the language, but may create some pattern in the ciphertext. A cryptanalyst may use a pattern attack to break the cipher. ∴ It is supposed to use random patterns to avoid this attack.

Known-plaintext Attack: In this attack, Eve [attacker has access to some plaintext | ciphertext pairs in addition to the intercepted ciphertext previous pair

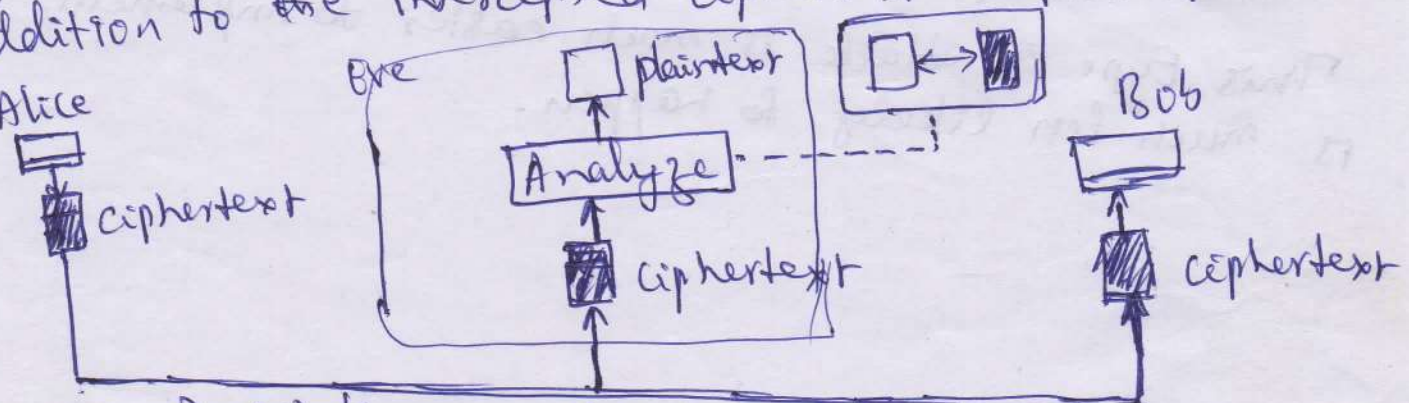


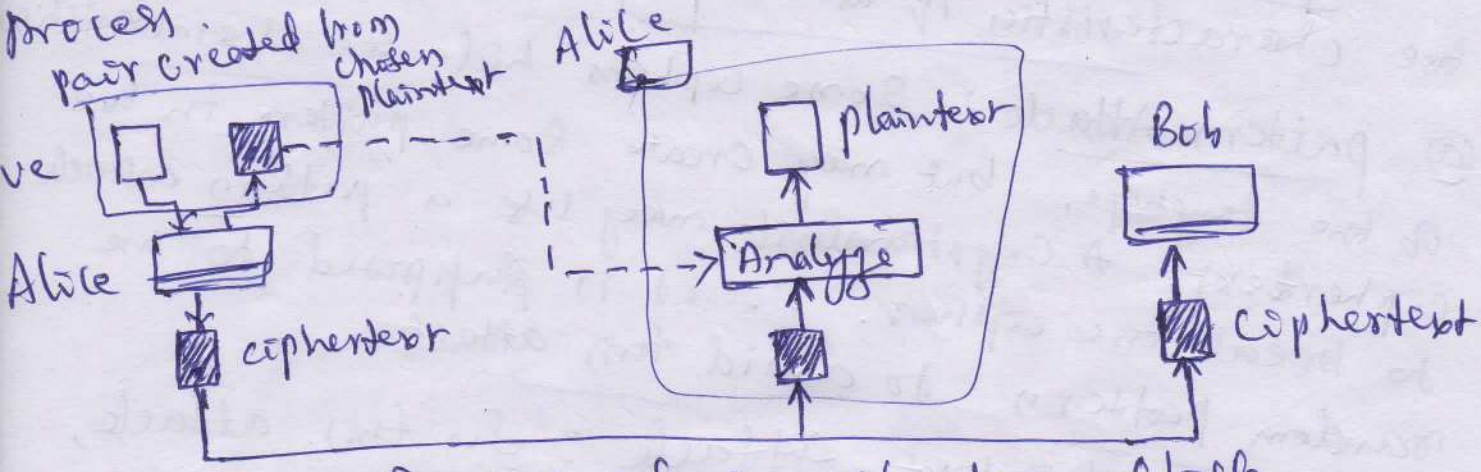
Fig (3) Known-plaintext Attack.

The plaintext/ciphertext pairs have been collected earlier.

Ex: Alice has sent a secret message to Bob, but she has later made the contents of the message public.

Eve/Attacker has kept both the messages to break the next message from Alice to Bob, assuming that Alice has not changed her key. Eve/attacker uses the relationship between the previous pair to analyze the current ciphertext. This attack is easy to implement, however, it is less likely to happen because Alice may have changed her key or may have not disclosed the contents of any previous messages.

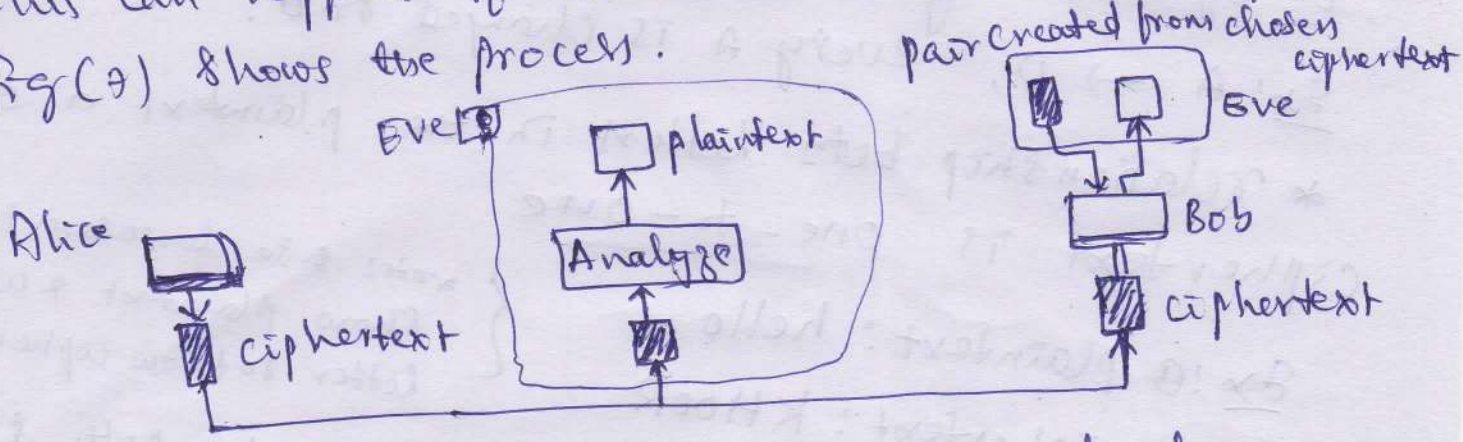
Chosen-plaintext Attack: It is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker herself. Fig(6) shows the process.



Fig(6) Chosen-plaintext attack

This can happen, if Eve has access to Alice's computer. This type of attack is much easier to implement, but it is much less likely to happen.

Chosen-Ciphertext Attack: is similar to the chosen plaintext attack, except that Eve chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair. This can happen if Eve has access to Bob's computer. Fig(7) shows the process.



Fig(7) Chosen-ciphertext attack

Categories of Traditional Ciphers

Two broad categories of Traditional Symmetric-key ciphers
1) Substitution ciphers: replace one symbol with another.
2) Transposition ciphers: reorder the position of symbols.

Substitution ciphers: A substitution cipher replaces one symbol with another.

If the symbols are alphabetic characters, replace one character with another

Ex: replace letter A with D, T with 2.

If the symbols are digits (0 to 9), replace 3 with 7, & 2 with 6.

Substitution ciphers can be categorized into Monoalphabetic ciphers or polyalphabetic ciphers

Monoalphabetic ciphers:

Here, a character/symbol changed to another character or symbol. regardless of the position

Ex: A → D, every A is changed to D.

* relationship betn letters in the plaintext and the ciphertext is one-to-one

Ex:① plaintext: hello
ciphertext: KHOOR

{ note: use lowercase to show plaintext & uppercase letter to show ciphertext.

This cipher is monoalphabetic because both l's (els) are encrypted as O's.

Ex ② The cipher is not monoalphabetic because each l (el) is encrypted by a different character. the first l is encrypted as N; the second as Z.

plaintext: hello ciphertext: ABNZF

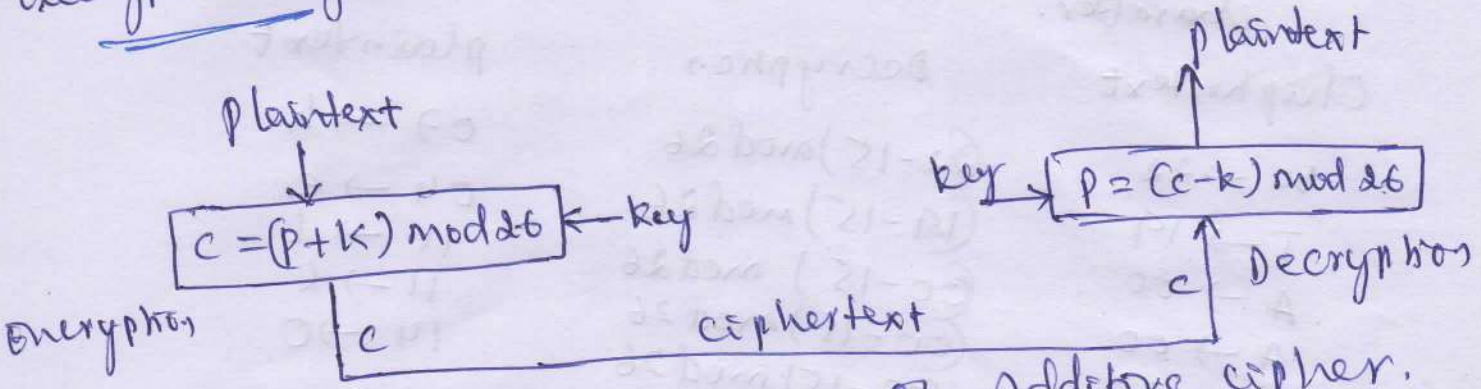
① Additive cipher: The simplest monoalphetic cipher is the Additive cipher.

Assume that the plaintext consists of lowercase letters (a to z), and that the ciphertext consists of uppercase letters (A to Z). To apply mathematical ops, assign numerical values to each letter (lower or uppercase). as shown fig(8).

Plaintext	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
Ciphertext	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
Value	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

fig(8) Representation of plaintext & ciphertext characters in Z_{26}

Each character and secret key assigned (in between integer in Z_{26}. The encryption alg adds the key and decryption alg substracts the key. All opns are done in Z_{26}.



Fig(9) The process of Additive cipher.

We can prove that encryption & decryption are inverse of each other because plaintext created by Bob (P_1) is the same as the one sent by Alice (P).

$P_1 = (C-k) \bmod 26 = (P+k-k) \bmod 26 = P$.

Ex: Use the additive cipher with key = 15 to encrypt the message "hello".

Soln Apply the encryption alg to the plaintext, character by character:

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

∴ Ciphertext is WTAAD.

Note: the cipher is monoalphabetic because two instances of the same plaintext character (l's) are encrypted as the same character (A).

Ex-2: Use the additive cipher with key=15 to decrypt the message "WITAAD".

Soln: Apply the decryption alg to the ~~plaintext~~ ciphertext character by character.

Ciphertext	Decryption	plaintext
W → 22	(22-15) mod 26	07 → h
J → 19	(19-15) mod 26	04 → e
A → 00	(00-15) mod 26	11 → l
A → 00	(00-15) mod 26	11 → l
D → 03	(03-15) mod 26	14 → o

The result is "hello"

Note: operation in modulo 26, a negative result needs to be mapped to Z26 (Ex: -15 becomes 11)

Shift cipher: Historically, additive ciphers are called shift ciphers. The reason is that the encryption & decryption algorithm can be interpreted as "shift key character down" and "shift-key character up" respectively.

Ex: If the key=15, the encryption alg shifts 15 characters down (toward the end of the alphabet). The decryption alg shifts 15 characters up (toward the beginning of the alphabet) When we reach the end or the beginning of the alphabet, we wrap around (manifestation of modulo 26).

{ ... a ..... z a b ... }

a = 00 + 15 = 15 → p

i = 11 + 15 = 26 → A

hello

shift key-down

h = 07 + 15 = 22 W − 15 → h
e = 04 + 15 = 19 T − 15 → e
l = 11 + 15 = 26 A − 15 → l
l = 11 + 15 = 26 A − 15 → l
o = 14 + 15 = 29 D − 15 → o

shift key up

Caesar Cipher: Julius Caesar used an additive cipher to communicate with his officers. For this reason, additive ciphers are sometimes referred to as the Caesar cipher. Caesar used a key of 3 for his communications (key=3)

Note: Additive ciphers are sometimes referred to as shift ciphers or Caesar cipher.

Cryptanalysis: Additive ciphers are vulnerable to ciphertext-only attacks using exhaustive key searches (brute-force attacks). The key domain of the additive cipher is very small; there are only 26 keys. However, one of the keys, zero, is useless (the ciphertext is the same as the plaintext). This leaves only 25 possible keys. Eve/attacker can easily launch a brute-force attack on the ciphertext.

Ex-1: Eve has intercepted the ciphertext "UVACLYPZLJBYL". Show how can use a brute-force attack to break the cipher.

Soln: Eve tries keys from 1 to 7. With a key 7, the plaintext is "not very secure", which makes sense.

Ciphertext: UVACLYPZLJBYL

k=1 → plaintext: tuzbkxeykiaxle
k=2 → plaintext: styajewdxjhzwj
k=3 → : rsxzivcwigyvi
k=4 → : qrwyhubvhfxuh
k=5 → : pqvxgtaugewtg
k=6 → plaintext: opuwfsztfdvsf
k=7 → plaintext: notverysecure

Table-1 : Frequency of characters in English

letter	Freq	let	Freq	let	Freq	let	Freq
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Table-2 Frequency of diagrams & Trigrams

Diagrams : TH HE IN ER AN RE ED ON ES ST EN
AT TO . . .

Trigram: THE ING AND HER ERE ENT THA NTH
WAS ETH FOR OTH

Ex: Cipher text : XLILS4WIMWKSATSVWE . . .
- - - - - . WSVISTJIVW

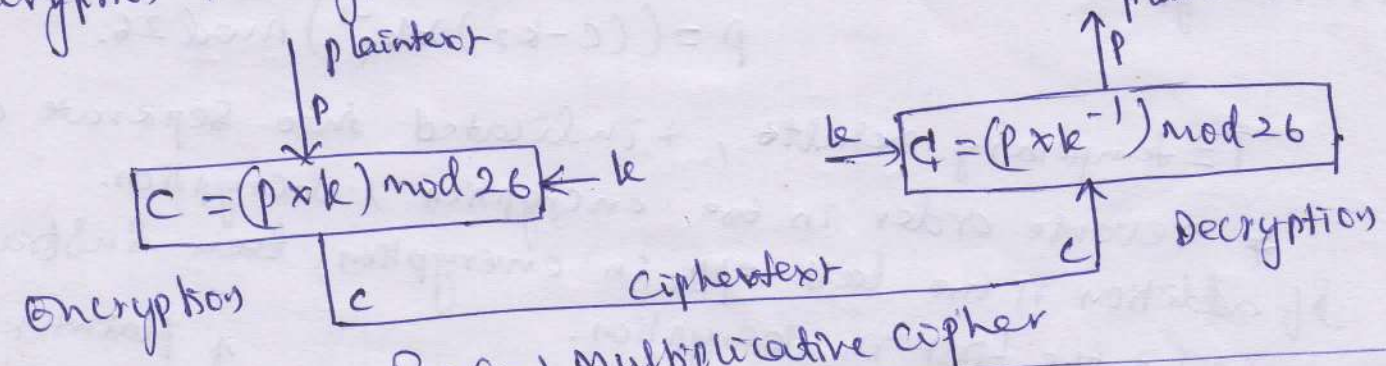
Find the plaintext using Statistical attacks.

Sln : Tabulate the freq. of letters in the ciphertext
I = 14, V = 13, S = 12 & so on.
The most common character is I with 14 occurance
As per the table above the most common character
is E - Frequency is more, so key is 4

X = 23 - 4 = 19 T
L = 11 - 4 = 7 H
S = 08 - 4 = 4 E
L = 11 - 4 = 7 H
S = 18 - 4 = 14 O
4 = 24 - 4 = 20 U
W = 22 - 4 = 18 S
I = 08 - 4 = 4 E

Multiplicative Ciphers: In this ciphers method, the encryption algorithm specifies multiplication of the plaintext by the key and the decryption algorithm specifies division (multiplication), the ciphertext by the key as shown in Fig(10). However, decryption here means multiplying by the multiplicative inverse of the key. (Opns in Z_{26})

Note: key is belong to the set Z_{26}^*, to guarantee that the encryption & decryption are inverses of each other.



Fig(10) Multiplicative cipher

In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z_{26}^*

Ex 1: What is the key domain for any multiplicative cipher?

Soln: The key needs to be in Z_{26}^*. This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.

Since (2, 4, 6, 8, 10, 12, 13, 14, 16, 18, 20, 22 24) mod 26 ⟹ 0

Ex: 2: Using multiplicative cipher, encrypt the message "hello" with a key of 7.

plaintext	Encryption	ciphertext
h → 07	(07 × 07) mod 26	23 → X
e → 04	(04 × 07) mod 26	02 → C
l → 11	(11 × 07) mod 26	25 → Z
l ⇒ 11	(11 × 07) mod 26	25 → Z
D → 14	(14 × 07) mod 26	20 → U

∴ Ciphertext is XCZZU

Affine cipher: The combination of additive and multiplicative ciphers. - a combination of both ciphers with a pair of keys. The first key is used with the multiplicative cipher & second key is used with the additive cipher.

Fig (11) - affine cipher (actually two ciphers), applied one after another.

Shown only are complex operation for the encryption or decryption such as $C = (P \times k_1 + k_2) \bmod 26$

$P = ((C - k_2) \times k_1^{-1}) \bmod 26.$

T = temporary results, & indicated two separate opns

* reverse order in the encryption & decryption.

If addition is the last opn in encryption, then subtraction should be the first in decryption.

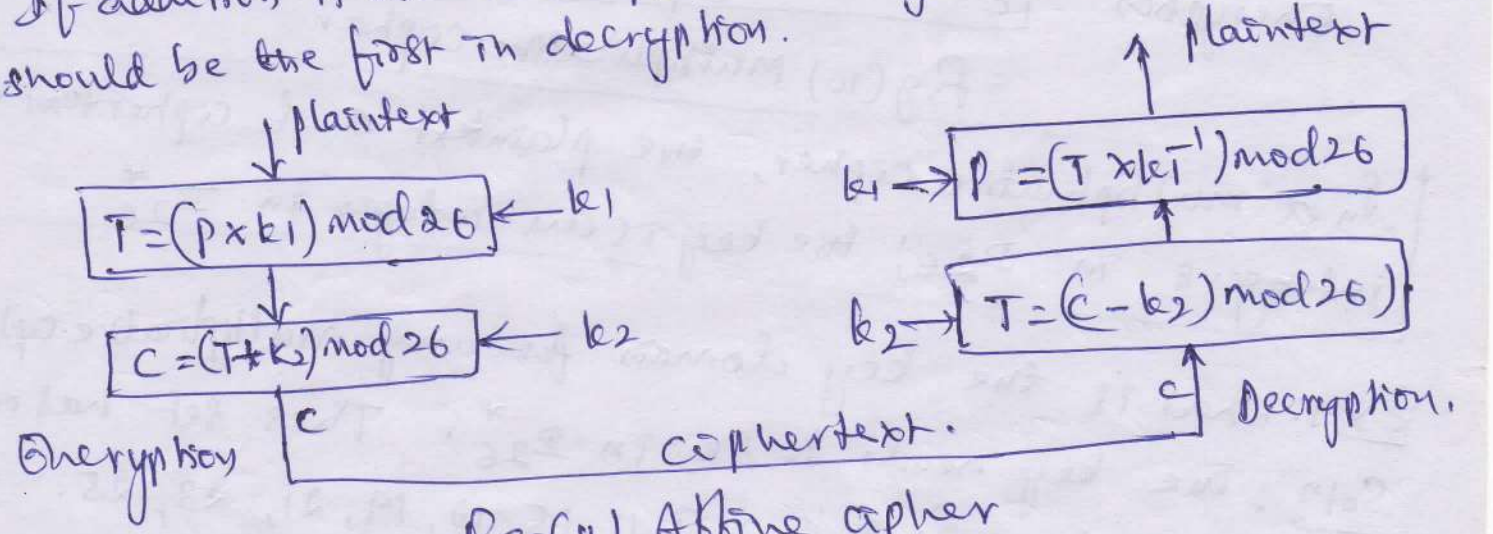


Fig (11) Affine cipher

In the affine cipher, the relationship between the plaintext P and the ciphertext C is

$C = (P \times k_1 + k_2) \bmod 26$ $\quad P = ((C - k_2) \times k_1^{-1}) \bmod 26$

Where k_1^{-1} = multiplicative inverse of k_1

$-k_2$ = additive inverse of k_2

Ex:1) The Affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26}. The size of the key domain is $26 \times 12 = 312$.

12 key numbers

Ex:2) Use an affine cipher to encrypt the message "hello" with the key pair (7, 2).

Solution: Use 7 for the multiplicative key and 2 for the additive key.

h → 07	$(07 \times 7 + 2) \bmod 26$	25 → Z
e → 04	$(04 \times 7 + 2) \bmod 26$	04 → E
l → 11	$(11 \times 7 + 2) \bmod 26$	01 → B
l → 11	$(11 \times 7 + 2) \bmod 26$	01 → B
o → 14	$(14 \times 7 + 2) \bmod 26$	22 → W

∴ ciphertext is ZEBBW.

Ex:3): Use the affine cipher to decrypt the message "ZEBBW" with the key pair (7, 2), in modulus 26.

Soln: Add the additive inverse of $-2 \equiv 24 \pmod{26}$ to the received ciphertext. Then multiply the result by the multiplicative inverse of $7^{-1} \equiv 15 \pmod{26}$ to find the plaintext characters. Because 2 has an additive inverse in Z_{26} and 7 has a multiplicative inverse in Z_{26}^*.

Decryption

Z → 25	$((25-2) \times 7^{-1}) \bmod 26$	07 → h
E → 04	$((04-2) \times 7^{-1}) \bmod 26$	04 → e
B → 01	$((01-2) \times 7^{-1}) \bmod 26$	11 → l
B → 01	$((01-2) \times 7^{-1}) \bmod 26$	11 → l
W → 22	$(22-2) \times 7^{-1} \bmod 26$	14 → o

∴

Monoalphabetic Substitution Cipher

Additive, multiplicative & affine ciphers have small key domains, easy to decrypt, & key is independent from the letters being transferred.

A better solution is to create a mapping between each ~~letter~~ plaintext character & the corresponding ciphertext ~~letter~~ character

Alice & Bob can agree on a table showing the mapping for each character.

Example:

plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	s	t	u	v	...
ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	K	G	Y	L	K	HV	I	J	M	...

An example key for monoalphabetic substitution cipher.

Ex: encrypt the message using the above key

"this message is easy to encrypt but hard to find the key"

The ciphertext is

ICFVQRVVNEFRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Cryptanalysis: The size of the key space is 26! (almost 4×10^{26}).

This makes a brute-force attack extremely difficult for Eve/Attacker even if attacker is using a powerful computer.

"The monoalphabetic ciphers do not change the frequency of characters in the ciphertext, which makes the ciphers vulnerable to statistical attacks."

Polyalphabetic ciphers

In polyalphabetic substitution, each occurrence of a character may have a different substitute.

* a character in the plaintext & ciphertext is one-to-many relationship.

Ex: a ciphered as "D" in the beginning of the text, but as "N" at the middle.

* each ciphertext character dependent on both the Corresponding plaintext character & the position of the plaintext character in the message.

We need to have a key stream $k = (k_1, k_2, k_3 \ldots\ldots)$
k_i used to encipher the ith character in the plaintext to create the ith character in the ciphertext.

① Autokey cipher: In this cipher, the key is a stream of subkeys, each subkey is used to encrypt the corresponding character in the plaintext.
The first subkey is a predetermined value secretly agreed upon by Alice & Bob. The second subkey is the value of the first plaintext character (between 0 & 25). The third subkey is the value of the second plaintext & so on.

$P = P_1 P_2 P_3 \ldots\ldots\ldots$ $C = C_1 C_2 C_3 \ldots$ $k = (k_1, P_1, P_2 \ldots)$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

* Subkeys are automatically created from the plaintext cipher characters during the encryption process.

Ex14: Assume that Alice and Bob agreed to use an autokey cipher with initial key value $k_1 = 12$.

Message is "Attack is today"

Encryption is done character by character, each character is replaced by its integer value. The first subkey is added to create the first ciphertext character.

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y	
P's values:	00	19	19	00	02	10	08	18	19	14	03	00	24	
key stream:	12	00	19	19	00	02	10	08	18	19	14	03	00	
C's values:	12	19	12	19	02	12	18	00	11	7	17	03	24	
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y	

Cryptanalysis: The autokey cipher definitely hides the single-letter frequency statistics of the plaintext. However, it is still as vulnerable to the brute-force attacks as the additive cipher.

The first subkey can be only one of the 25 values (1 to 25). We need polyalphabetic ciphers that not only hide the characteristics of the language but also have large key domains.

⑥ Playfair Ciphers: Used by the British army during World War I. The secret key is made of 25 alphabet letters arranged in a 5 x 5 matrix. (I & J are considered the same when encrypting). Different arrangements of the letters in the matrix can create many different secret keys. One of the possible arrangements is shown in fig(15). We have dropped the letters in the matrix diagonally

starting from the top right-hand corner.

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

= Secret key

Fig (13) An example of a Secret key in the playfair cipher

Before encryption, if the two letters in a pair are the same, a bogus letter is inserted to separate them. After inserting bogus letters, if the no. of characters in the plaintext is odd, one extra bogus character is added at the end to make the no. of characters even.

The cipher uses three rules for encryption.

(a) If the two letters in a pair are located in the same row of the secret key, the corresponding encrypted character for each letter is the next letter to the right in the same row (with wrapping to the beginning of the row if the plaintext letter is the last character in the row).

(b) If the two ..... in the same column ..... is the letter beneath it in the column (with ..... of the column ..... in the column)

(c) If the two letters in a pair are not in the same row or column of the Secret, the corresponding encrypted character for each letter is a letter that is in its own row but in the same column as the other letter

The playfair cipher meets criteria for a polyphabetic cipher. The key is a stream of subkeys in which the subkeys are created two at a time.

In playfair cipher, the key stream and cipher stream are the same.

The encryption alg takes a pair characters from the plaintext & creates a pair of subkeys by following the above rules.

* key stream depends on the position of the character in the plaintext.

Position dependency has a different interpretation here! the subkey for each plaintext character depends on the next or previous neighbor.

* the ciphertext is actually the key stream.

$P = P_1 P_2 P_3 \cdots$ $\quad C = C_1 C_2 C_3 \cdots$ $\quad k = [(k_1, k_2), (k_3, k_4), \ldots]$

Encryption: $C_i = k_i$ $\quad$ Decryption: $P_i = k_i$

Example: Encrypt the plaintext "hello"

Group the letters in two-character pairs, he, ll, o'

we need to insert an x betn two l's,

i.e. he lx lo

he → EC $\quad$ lx → BZ $\quad$ lo → BX

plaintext: hello $\quad$ ciphertext: ECBZBX

Here the cipher is actually a polyalphabetic cipher: two occurrences of the letter 'l' are encrypted as 'B' & 'B'.

Cryptanalysis of a ~~poly~~ playfair cipher:

* a brute-force attack is very difficult.
* the size of the key domain 25! (factorial 25).
* preserve the table arrangement
* cryptanalyst can use only ciphertext-only attacks to break.

Vigenere cipher : very - interesting cipher, designed by Blaise de Vigenere, a 16th century - French mathematician.

A Vigenere cipher uses a different strategy to create the key stream. The key stream is a repetition of an initial secret key stream of length m, where $1 \le m \le 26$.

$(k_1, k_2, \ldots k_m)$ is the initial secret key agreed by Alice & Bob.

$P = p_1 p_2 p_3 \ldots \quad C = C_1 C_2 C_3 \ldots \quad k = [(k_1, k_2, \ldots k_m), (k_1, k_2 \ldots k_m) \ldots]$.

Encryption: $C_i = p_i + k_i$ Decryption: $p_i = C_i - k_i$

* vigenere key stream does not depend on the plaintext characters; it depends only on the position of the character in the plaintext. i.e.
* the key stream can be created without knowing what the plaintext is.

Ex: Encrypt the message "She is listening" using the 6-char - acter keyword "PASCAL". The initial key stream is (15, 0, 18, 2, 00, 11). The key stream is the repetition of the initial key stream

plaintext :	s h e i s l	i s t e n i	n g
p's value :	18 07 04 08 18 11	08 18 19 04 13 08	13 06
key stream :	15 00 18 02 00 11	15 00 18 02 00 11	15 00
C's value :	07 07 22 10 18 22	23 18 11 6 13 19	02 06
Ciphertext :	H H W K S W	X S L G N T	C G

Hill Cipher : is invented by Lester S. Hill.

Here, plaintext is divided into equal size blocks. The blocks are encrypted one at a time in such a way that each character in the block contributes to the encryption of other characters in the block. That is why this cipher belongs to block ciphers (others are belongs to stream ciphers).

In a Hill cipher, the key stream is a square matrix of size $m \times m$, m is the size of the block.

Key matrix K can be written as

$$K = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & & & \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

each element of the matrix is k_{ij}.

key in the Hill cipher.

If m characters in the plaintext block $P_1, P_2 \dots P_m$, the corresponding characters in the ciphertext block are $C_1, C_2, \dots C_m$.

then
$$C_1 = P_1 k_{11} + P_2 k_{21} + \dots + P_m k_{m1}$$
$$C_2 = P_1 k_{12} + P_2 k_{22} + \dots + P_m k_{m2}$$
$$\dots$$
$$C_m = P_1 k_{1m} + P_2 k_{2m} + \dots + P_m k_{mm}$$

- Each ciphertext character, such as C_1 depends on all plaintext characters in the block ($P_1, P_2 \dots P_m$).
- Aware that not all square matrices have multiplicative inverse in $\mathbb{Z}_{26}$. So Alice & Bob should be careful in selecting the key. Bob will not be able to decrypt if the matrix does not have a multiplicative inverse.

Ex: Using matrices allows Alice to encrypt the whole plaintext. (plaintext is an $1 \times m$ matrix, l = no. of blocks).

Sol: Plaintext: "code is ready"

we can make a 3×4 matrix when adding extra bogus character "z" to the last block & removing the spaces.

$$P = \begin{bmatrix} c & o & d & e \\ i & s & r & e \\ a & d & y & z \end{bmatrix} = \begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}$$

Encryption

$$\underset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} = \underset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} \underset{k}{\begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}}$$

Decryption

$$\underset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} = \underset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} \underset{k^{-1}}{\begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}}$$

∴ The ciphertext is "OHKNIHGKLISS"

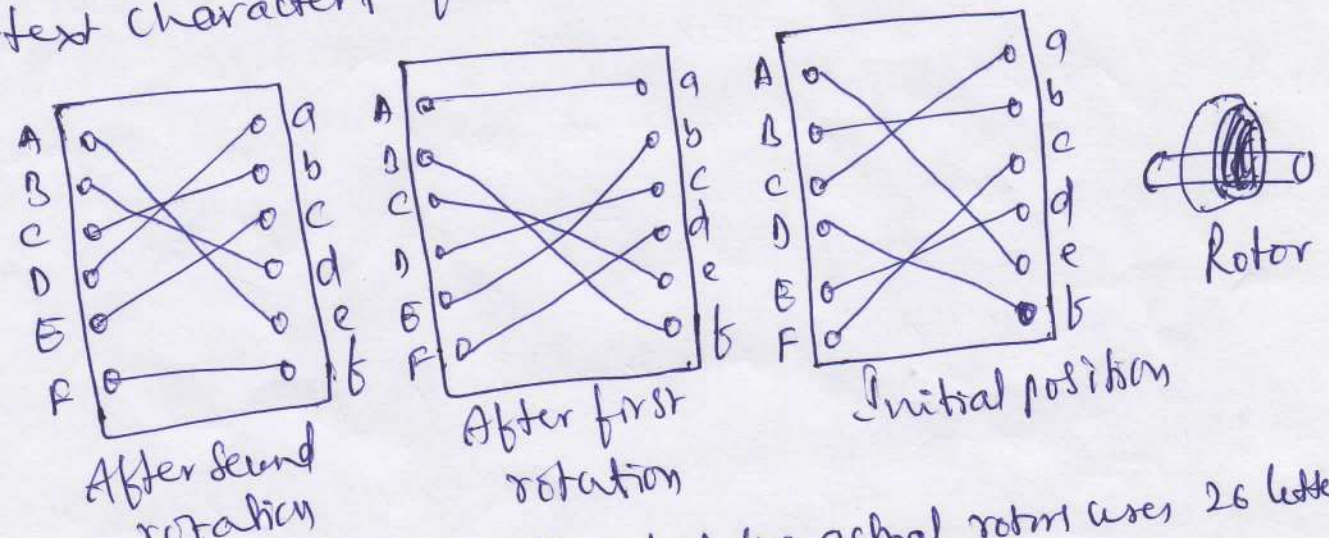
Bob can decrypt the message using the inverse of the key matrix.

One-Time pad: one of the goals of cryptography is perfect secrecy. the perfect secrecy can be achieved if each plaintext is encrypted with a key randomly chosen from a key domain.

Sender changes the key each time, using random sequence of integer, This called one-time pad, invented by Vernam. In this cipher, the key has the same length as the plaintext and is chosen completely in random.

A one-time pad is a perfect cipher, but it is almost impossible to implement commercially.

Rotor cipher: Although one-time pad ciphers are not practical, one step toward more secured encipherment is the rotor cipher. It uses the idea behind monoalphabetic substitution but changes the mapping between the plaintext & the ciphertext characters for each plaintext character.



Here the rotor uses only 6 letters, but the actual rotors uses 26 letters.

The initial setting (position) of rotor is the secret key between Alice & Bob. The first plaintext character is encrypted using the initial setting; the second character is encrypted after the first rotation and so on. Ex: bee is encrypted as "BAA" if the rotor is stationary, (the monoalphabetic), but it will be encrypted as "BCA"

Transposition Ciphers : It changes the location of the symbols. A symbol in the first position of the plaintext may appear in the tenth position of the ciphertext. etc.

* A transposition cipher reorders (transposes) the symbols. *

Keyless Transposition Ciphers : Simple & used in the past.

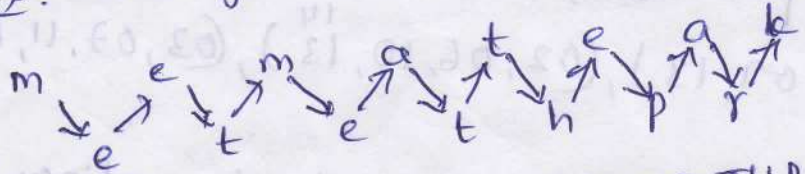
There are two methods for permutation of characters.

i) the text is written into a table column by column & then transmitted row by row.

ii) the text is written into the table row by row and they transmitted column by column.

Rail fence cipher : In this cipher, the plaintext is arranged in two lines as a zigzag pattern (column by column): the ciphertext is created reading the pattern row by row.

Ex: Message: Meet me at the park.



cipher text: MEMATEAKETETHPR

Receiver (Bob) : divide the received ciphertext in half The first half forms the first row; the second half, the second row. Bob reads the result in zigzag.

Ex. Alice & Bob can agree on the no. of columns and use the second method. Write the same plaintext, row by row, in a table of four columns.

m e e t
m e a t
T h e p
a r k

cipher text : MMTAEEHREAEKTTP

Ex-3: The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
01	05	09	13	02	06	10	14	03	07	11	15	04	08	12

The second character in the plaintext has moved to the fifth position in the ciphertext; third to ninth position & so on. the characters are permuted, there is a pattern in the permutation : (01, 05, 09, 13), (02, 06, 10, 14), (03, 07, 11, 15) and (04, 08, 12).

In each section, the difference between the two adjacent number is 4.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Meet me at the park

MMTAEEHREAEKTTP

Keyed Transposition ciphers:

In this type of ciphers, divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.

Ex: message: Enemy attacks tonight

Alice & Bob have agreed to divide the text into groups of five characters and then permute the characters in each group.

e·nemy attac kston ightz ← bogus character.

The key used for encryption and decryption is a permutation key, which shows how the characters are permuted. For this message, assume that Alice & Bob used the following key:

Encryption ↓	3	1	4	5	2
	1	2	3	4	5

↑ decryption.

The third character in the plaintext block becomes the first character
the first —"— second character
& so on (in ciphertext)

The permutation yields:

EEMYN TAACT TKONS HITZG

Alice sends the ciphertext "EEMYNTAACTTKONSHITZG" to Bob. Bob divides the ciphertext into 5-character groups and using the key in the reverse order, finds the plaintext

Combining Two Approaches

* It achieve better Scrambling. Encryption & decryption is done in three steps.

① the text is written into a table row by row,

② the permutation is done by reordering the columns.

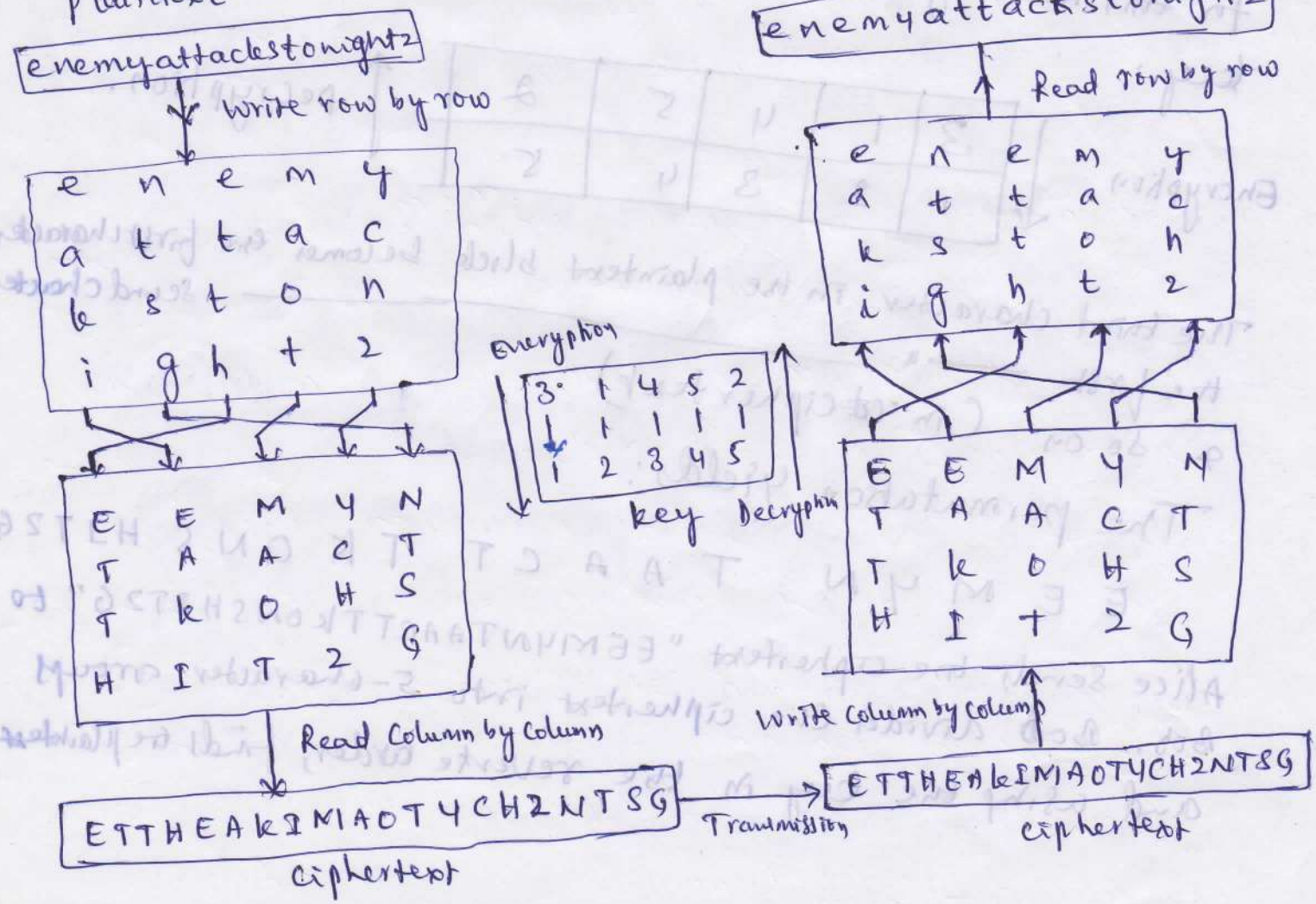
③ the new table is read column by column.

the first & third steps provide a keyless global reordering; the second step provides a blockwise keyed reordering.

This type of cipher is also known as Columnar transposition ciphers.

Ex: message: Enemy attacks tonight

Alice: (Encryption)

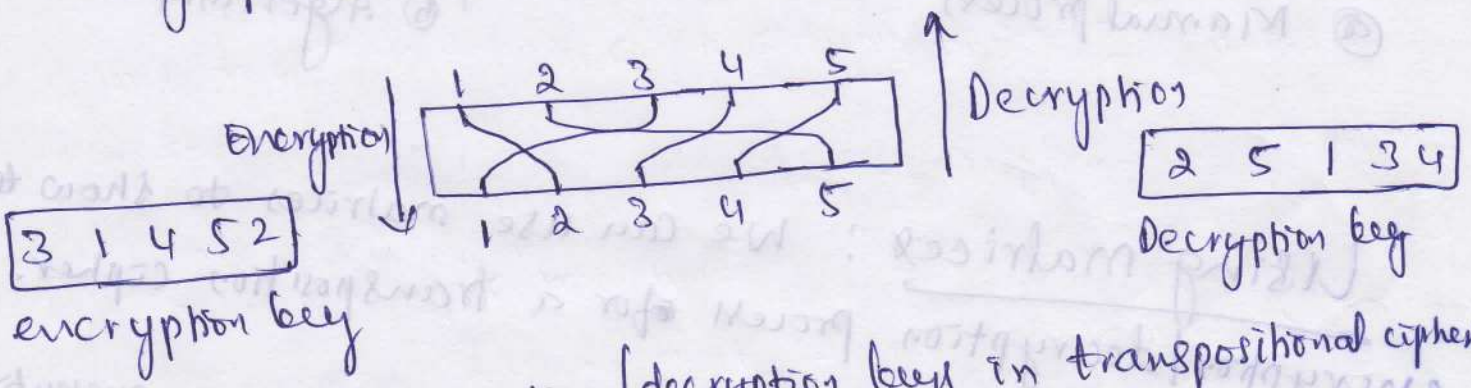


Keys: a single key is used in two directions: downward for encryption, upward for decryption.

* Creating two keys from graphical representation (one for encryption & one for decryption).

The keys are stored in tables with one entry for each column. The entry shows the source/destination column number, from the position of the entry.

The graphical representation of the key as shown below



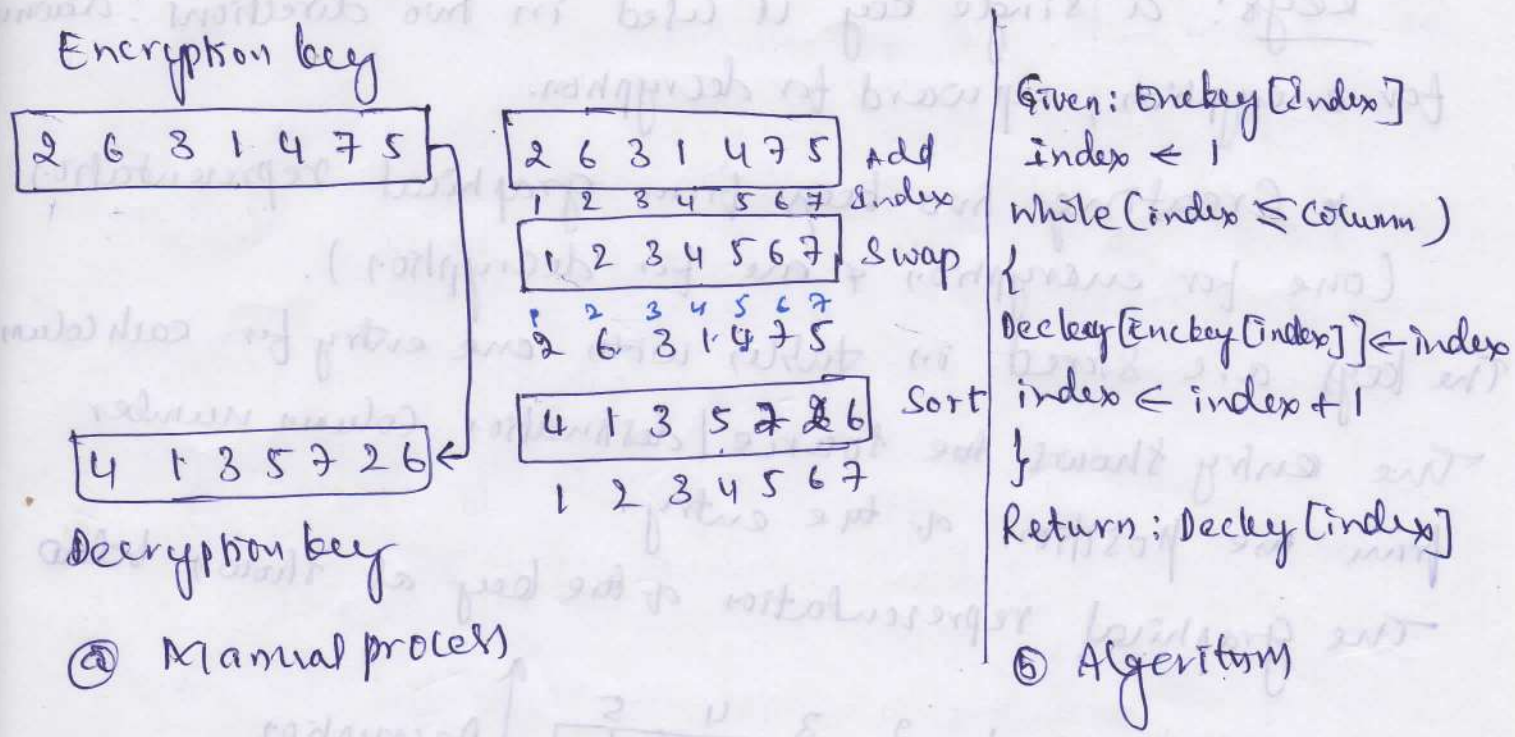
Fig(22) Encryption/decryption keys in transpositional ciphers.

The encryption key is (3 1 4 5 2). the first entry shows that column 3 in the source becomes column 1 in the destination.

The decryption key is (2 5 1 3 4). The first entry shows that column 2 in the source becomes column 1 in the dest.

How can the decryption key be created if the encryption key is given or vice versa?

The process can be done manually in a few steps. as shown in fig(23). first add indices to the key table, then swap the contents and indices, finally sort the pairs according to the index.



ⓐ Manual process

ⓑ Algorithm

Using matrices: We can use matrices to show the encryption/decryption process for a transposition cipher. The plaintext & ciphertext are 1xm matrices representing the numerical values of the characters; keys are square matrices of size m x m.

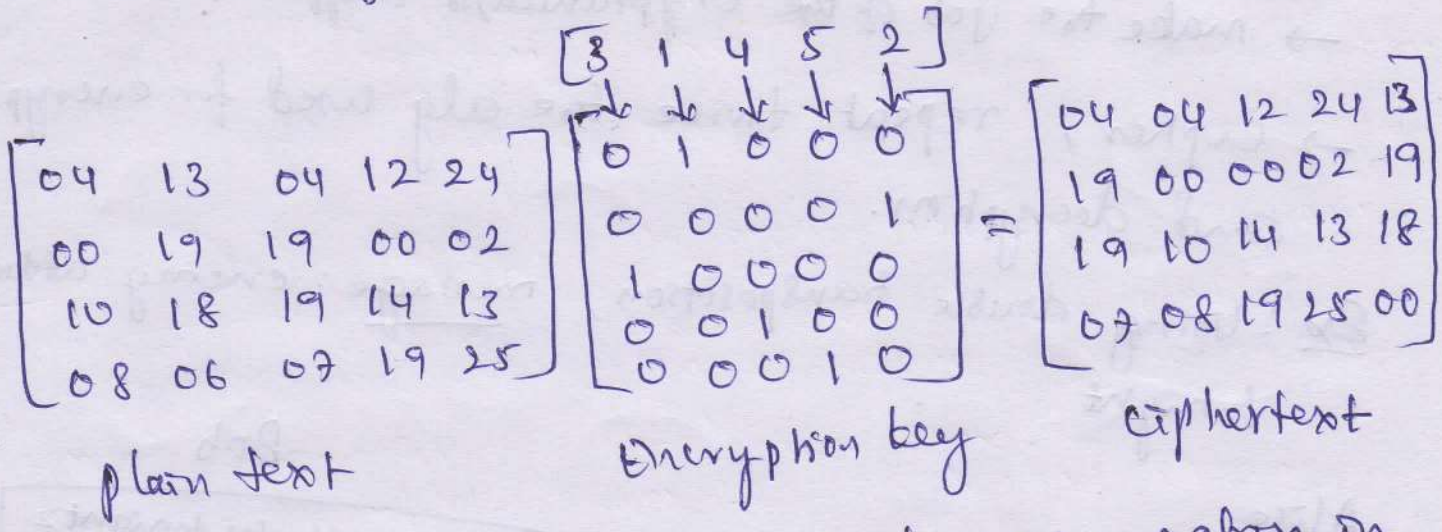
In permutation matrix, every row or column has exactly one 1 and the rest of the values are 0's.

Encryption is performed by multiplying the plaintext matrix by the key matrix to get the ciphertext matrix.

Decryption: multiplying the ciphertext by the inverse key matrix to get the plaintext matrix.

* decryption matrix is the inverse of the encryption matrix, however there is no need to invert the matrix. the encryption matrix can simply be transposed (swapping the rows & columns) to get the decrypt. matrix

Ex: Shows (fig(24)) the encryption process.



fig(24) Representation of the key as a matrix in the transposition cipher.

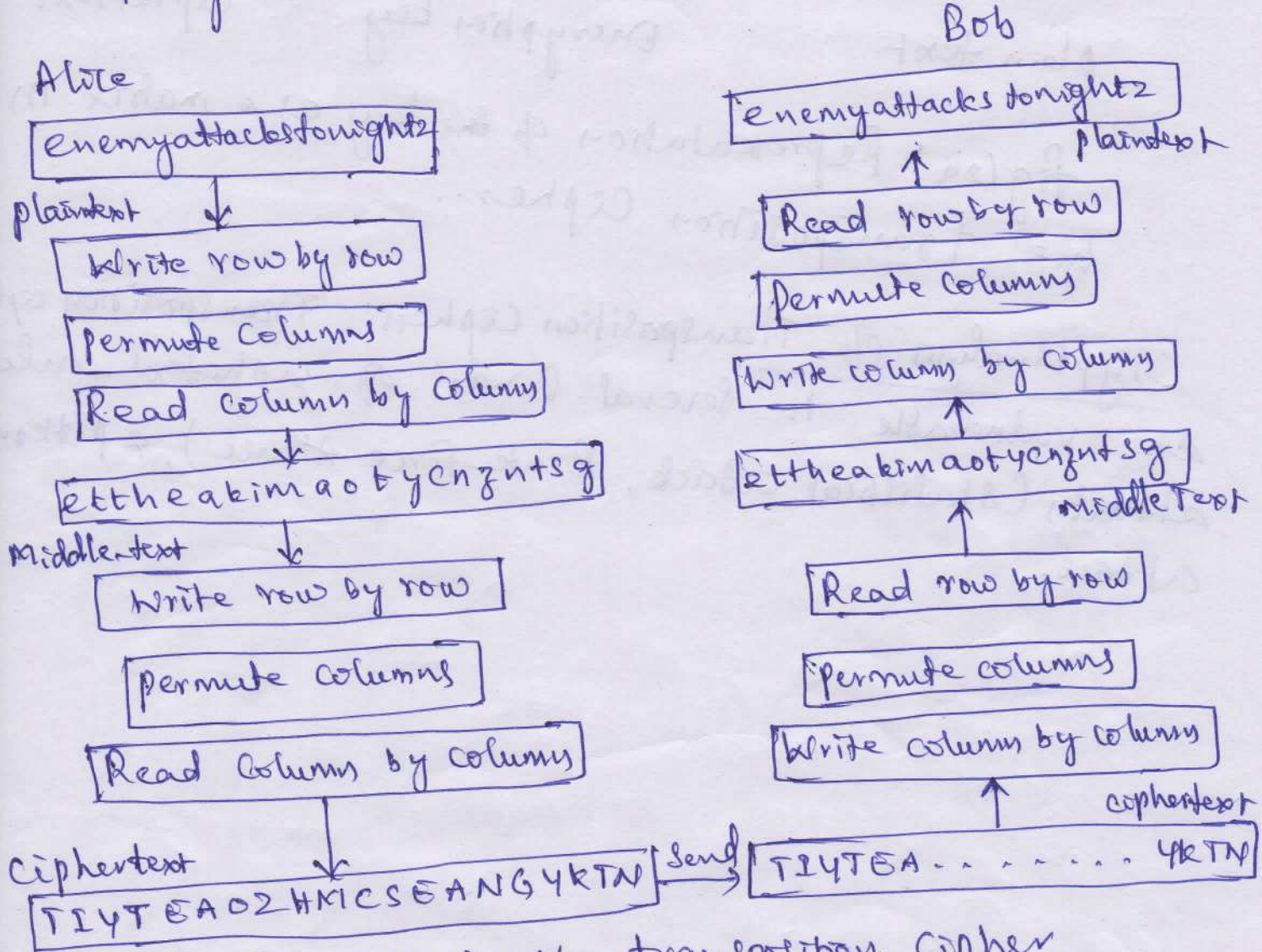
Cryptanalysis of Transposition ciphers: Transposition ciphers are vulnerable to several kinds of ciphertext-only attacks (Statistical attack, Brute-force Attack), & pattern attack.

Double Transposition ciphers

→ make the job of the cryptanalyst difficult.

→ cipher, repeat twice the alg used for encryption and decryption.

Ex: Using double transposition, message: enemy attacks tonight



Fig(25) Double transposition cipher