Computer Security

By

Abdul Hadi M.Alaidi

Chapter 1 1
1.1 Definitions 1
1.2 Security Services 2
1.3 Security Mechanism 4
1.4 Terminology and Background 4
1.5 Basic Cryptographic Algorithms 5
1.5.1 Classical model of encryption 6
Chapter 2 9
2.1 Modular Arithmetic 9
2.2 Greatest Common Divisor(GCD). 10
2.3 Least Common Multiple (LCM). 10
2.4 Multiplicative Inverse 11
2.5 Exercise ..... **Error! Bookmark not defined.**
Chapter 3 13
3.1 The forms of Encryption 13
3.2 Keyless Transposition Ciphers: 13
3.2.1 Keyless Transposition Ciphers: 13
3.2.2 Columnar Transposition Ciphers. 13
3.3 Substitution cipher 14
3.3.1 Monoalphabetic Ciphers. 14
3.3.2 Additive Cipher: 15
3.3.3 Caesar Cipher: - 16
Caesar Cipher 16
Multiplicative Ciphers 18
3.3.4 Affine Ciphers 19
3.4 Polyalphabetic Ciphers 20
3.4.1 Autokey Cipher: - 20
3.4.2 Playfair Key Matrix 21
3.4.3 Hill Cipher 23
3.4.4 One-Time Pad 30
Chapter 4 33
4.1 Introduction 33
4.2 Stream cipher 33

4.3 Block ciphers 34
4.4 Ciphers vs. Block ciphers 34
4.5 Encryption and Decryption with Stream Ciphers 35
4.6 Shift Register-Based Stream Ciphers 35
4.7 Linear Feedback Shift Registers (LFSR) 35
4.8 The Data Encryption Standard (DES) and Alternatives 37
4.9 Introduction to DES 37
4.10 Exercise ..... **Error! Bookmark not defined.**
Chapter 5 39
5.1 Introduction ..... **Error! Bookmark not defined.**
5.2 Exercise ..... **Error! Bookmark not defined.**
Chapter 6 ..... **Error! Bookmark not defined.**
6.1 Principle of mathematical induction **Error! Bookmark not defined.**
6.2 Exercise ..... **Error! Bookmark not defined.**

Chapter 1

Data Security Concepts

1.1 Definitions

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

Information systems security is the ability to provide the services required by the user community while simultaneously preventing unauthorized use of system resources

Network Security - measures to protect data during their transmission

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

Aspects of Security: - 3 aspects of information security:

- security attack
- security service
- security mechanism

Security Attack

any action that compromises the security of information owned by an organization

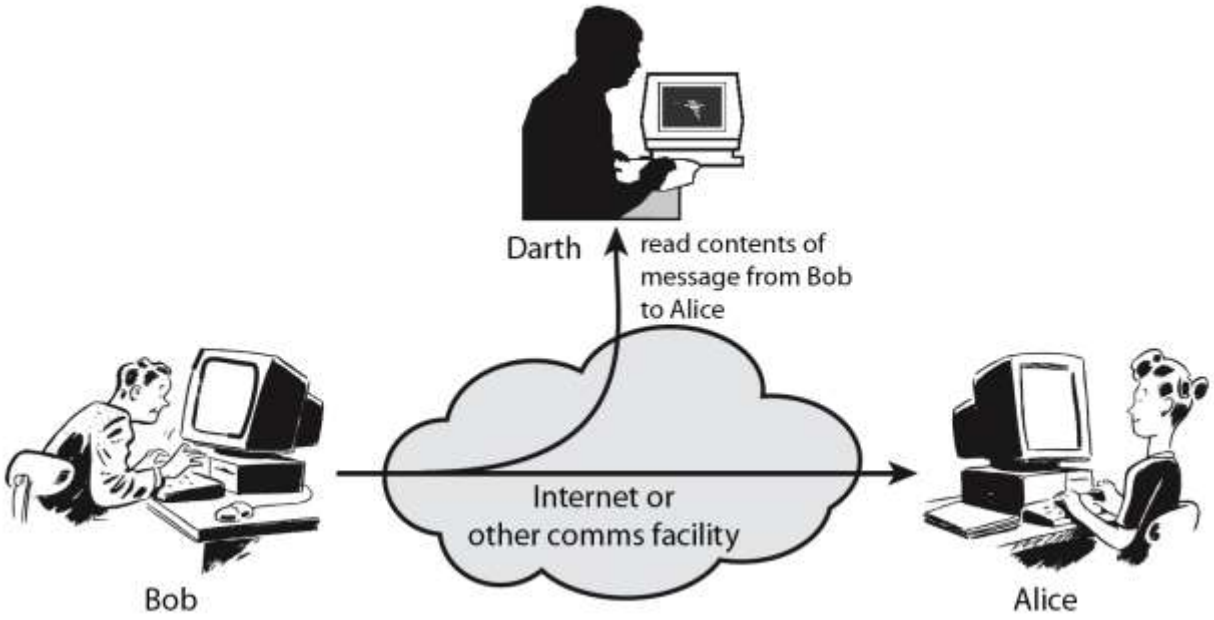
information security is about how to prevent attacks, or failing that, to detect attacks on information-based systems

often threat & attack used to mean same thing

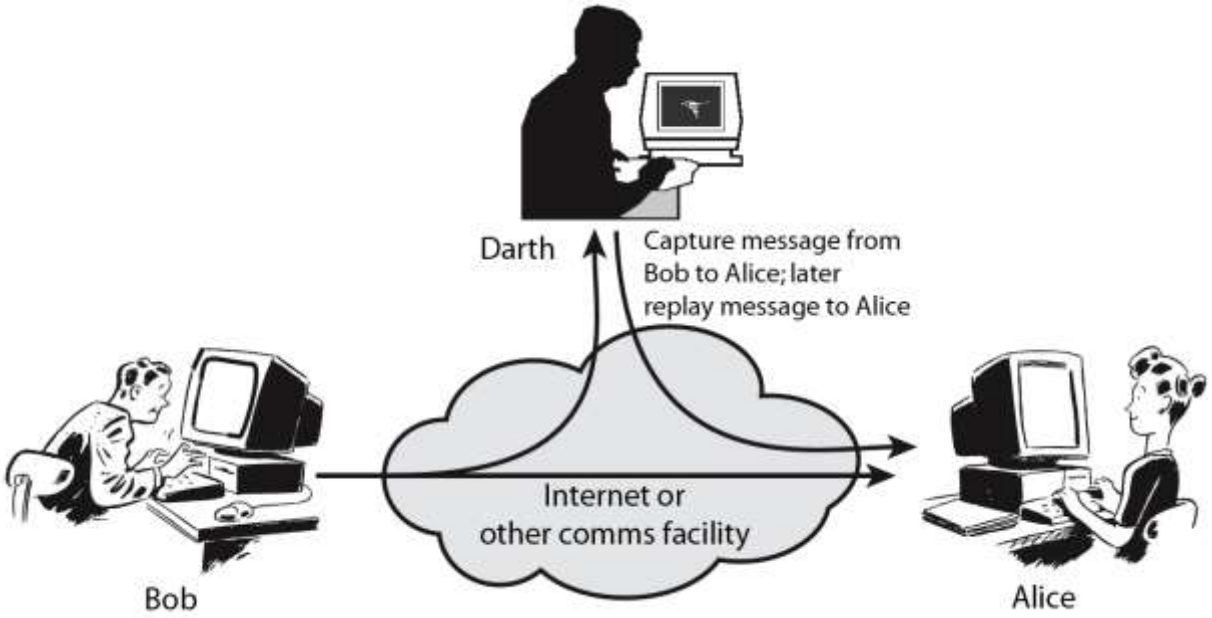
have a wide range of attacks and can focus of generic types of attacks

- passive
- active

Passive Attacks



Active Attacks



1.2 Security Services

1.Confidentiality: - The concept of Confidentiality relate to the protection of information and prevention of unauthorized access or disclosure. The ability to keep data confidential, or secret, is critical to staying competitive in today's business environments.

Examples of Confidentiality

1. Student grade information is an asset whose confidentiality is considered to be very high
    a. The US FERPA Act: grades should only be available to students, their parents, and their employers (when required for the job)
2. Student enrollment information: may have moderate confidentiality rating; less damage if enclosed
3. Directory information: low confidentiality rating; often available publicly

2. Integrity: - Integrity deals with prevention of unauthorized modification of intentional or accidental modification.

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

System integrity: Assures that a system performs its operations in unimpaired manner

Examples of Integrity

- A hospital patient's allergy information (high integrity data): a doctor should be able to trust that the info is correct and current
- If a nurse deliberately falsifies the data, the database should be restored to a trusted basis and the falsified information traced back to the person who did it
- An online newsgroup registration data: moderate level of integrity
- An example of low integrity requirement: anonymous online poll (inaccuracy is well understood)

3. Availability: - assures that the resources that need to be accessed are accessible to authorized parties in the ways they are needed. Availability is a natural result of the other two concepts (confidentiality and integrity).

Examples of Availability

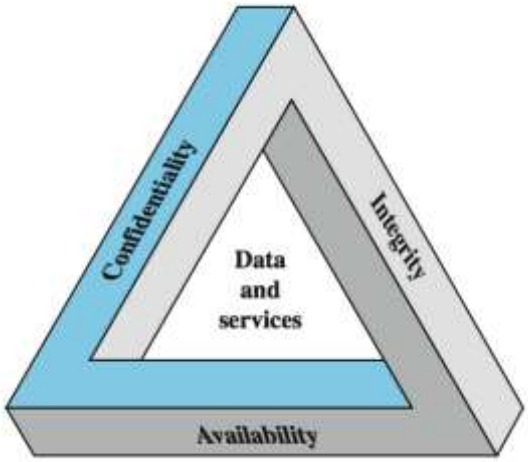
1. A system that provides authentication: high availability requirement
   (a) If customers cannot access resources, the loss of services could result in financial loss
2. A public website for a university: a moderate availably requirement; not critical but causes embarrassment
3. An online telephone directory lookup: a low availability requirement because unavailability is mostly annoyance (there are alternative sources)

4. Authentication is the process by which the information system assures that you are who you say you are; how you prove your identity is authentic. Methods of performing authentication are:

1. User ID and passwords. The system compares the given password with a stored password. If the two passwords match then the user is authentic.

2. Swipe card, which has a magnetic strip embedded, which would already contain your details, so that no physical data entry takes place or just a PIN is entered.
3. Digital certificate, an encrypted piece of data which contains information about its owner, creator, generation and expiration dates, and other data to uniquely identify a user.
4. key fob, small electronic devices which generate a new random password synchronized to the main computer
5. Biometrics - retinal scanners and fingerprint readers. Parts of the body are considered unique enough to allow authentication to computer systems based on their properties.

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



1.3 Security Mechanism

- feature designed to detect, prevent, or recover from a security attack
- no single mechanism that will support all services required
- however, one particular element underlies many of the security mechanisms in use:
  cryptographic techniques
- hence our focus on this topic

1.4 Terminology and Background

Cryptography is the art or science of keeping messages secret.

Cryptanalysis is the art of breaking ciphers, i.e. retrieving the plaintext without knowing the proper key.

People who do cryptography are cryptographers, and practitioners of cryptanalysis are cryptanalysts.

Cryptography deals with all aspects of secure messaging, authentication, digital signatures, electronic money, and other applications.

Cryptology is the branch of mathematics that studies the mathematical foundations of cryptographic methods.

The various components of a basic cryptosystem are as follows: -

Plaintext. It is the data to be protected during transmission.

Encryption Algorithm. It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

Ciphertext. It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

Decryption Algorithm, It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

Encryption Key. It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

Decryption Key. It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

For a given cryptosystem, a collection of all possible decryption keys is called a **key space**

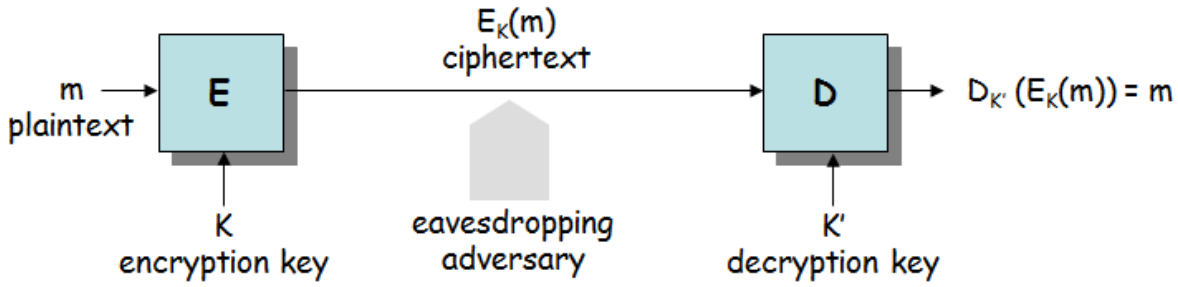
1.5 Basic Cryptographic Algorithms

A cipher is the method of encryption and decryption.

Some cryptographic methods rely on the secrecy of the algorithms. Keyless Cipher is a cipher that does not require the use of a key.

All modern algorithms use a key to control encryption and decryption; a message can be decrypted only if the key matches the encryption key.

The key used for decryption can be different from the encryption key, but for most algorithms they are the same.

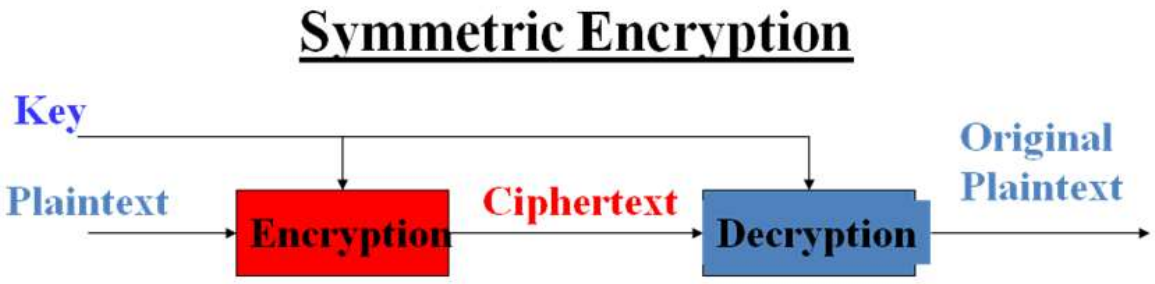


1.5.1 Classical model of encryption

Basic classification of encryption key-based algorithms

1. Symmetric-key or (or secret-key) encryption algorithm.

- Symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key)
- two main types:
  - stream ciphers – operate on individual characters of the plaintext
  - block ciphers – process the plaintext in larger blocks of characters

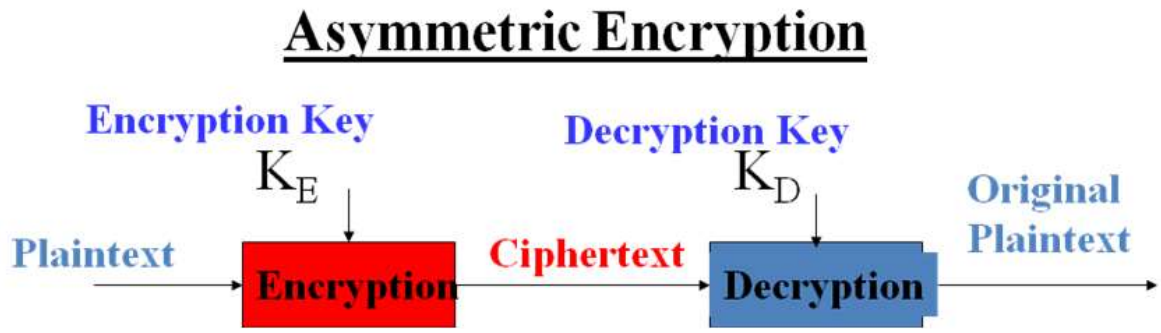


2. Asymmetric (or public-key) encryption algorithms.

algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key.

permit the encryption key to be public (it can even be published in a newspaper), allowing anyone to encrypt with the key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is also called the public key and the decryption key the private key or secret key.

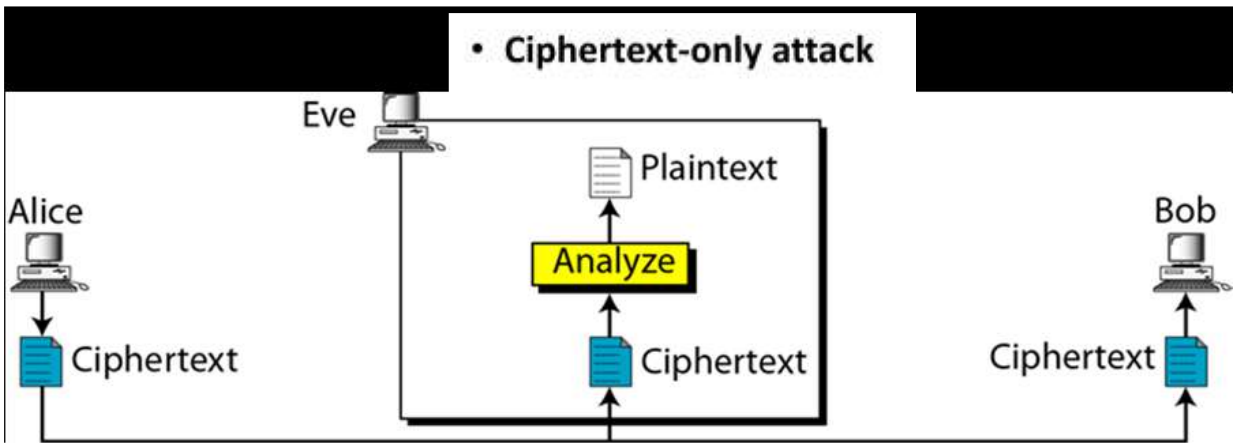
symmetric algorithms are much faster to execute on a computer



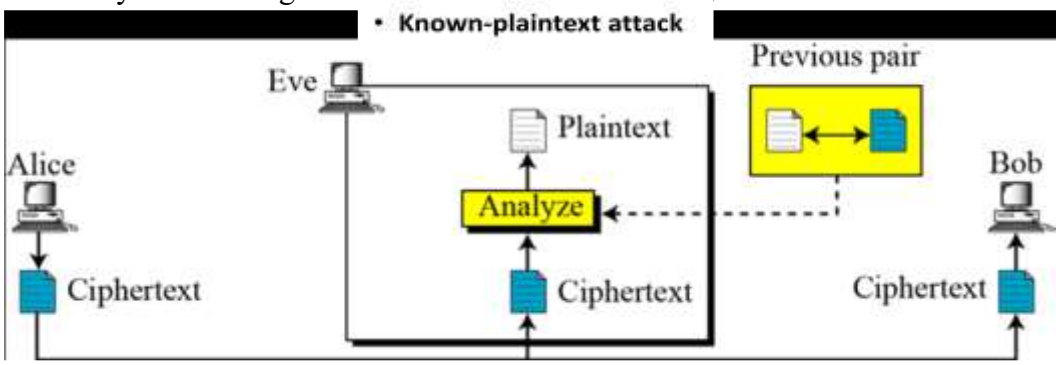
Cryptanalysis and Attacks on Cryptosystems

There are many cryptanalytic techniques. Some of the more important ones for a system implementer are

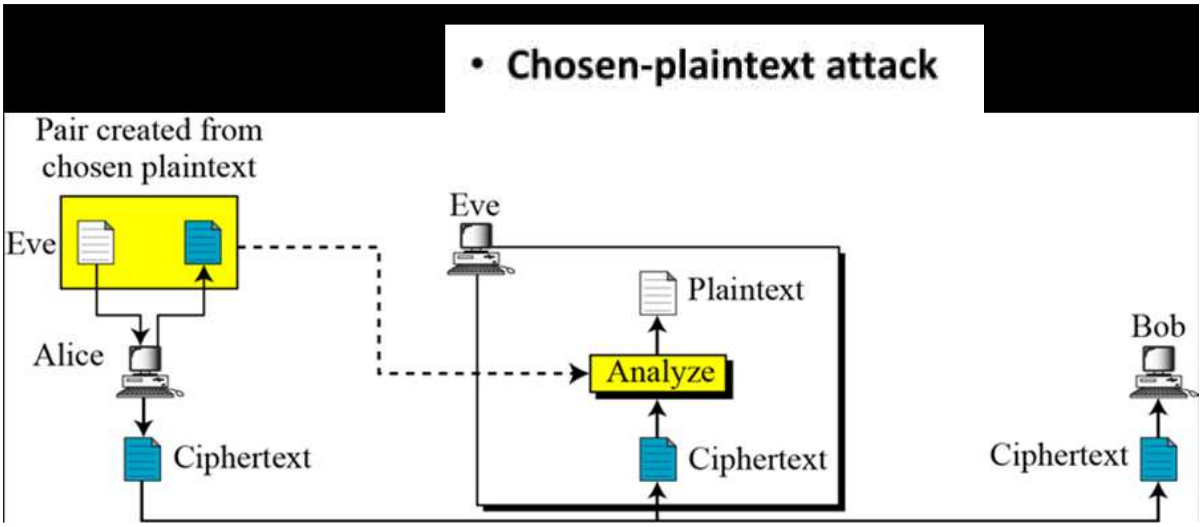
1) **Ciphertext-only attack** (Only know algorithm / ciphertext, statistical, can identify plaintext): This is the situation where the attacker does not know anything about the contents of the message, and must work from ciphertext only. It is very hard.



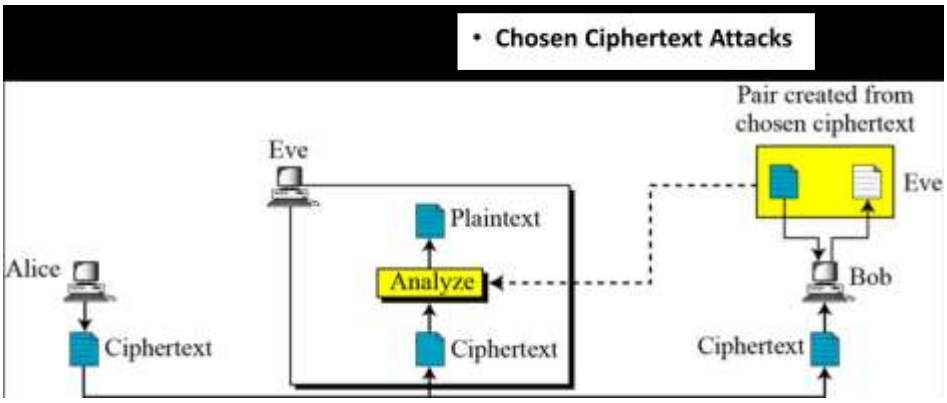
2) **Known-plaintext attack** (know/suspect plaintext & ciphertext to attack cipher): The attacker knows or can guess the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext blocks using this information. This may be done by determining the key used to encrypt the data, or via some shortcut.



3) Chosen-plaintext attack (selects plaintext and obtain ciphertext to attack cipher): The attacker is able to have any text he likes encrypted with the unknown key. The task is to determine the key used for encryption.



4) **Chosen Ciphertext Attacks** (select ciphertext and obtain plaintext to attack cipher): Attacker obtains the decryption of any ciphertext of its choice (under the key being attacked)



Chapter 2

Mathematic

2.1 Modular Arithmetic

several important cryptosystems make use of modular arithmetic. This is when the answer to a calculation is always in the range 0 – m where m is the modulus.

(a mod n) means the remainder when a is divided by n.

$a \; mod \; n \; = \; r$
$a \; div \; n = q$
$a \; = \; qn \; + \; r$
$r \; = \; a - q \; * \; n$
Example :- if a=13 and n=5, find q and r.
$q = 13 \; div \; 5 = 2 \; and \; r = 13 - 2 \; * 5 = 3$ which is equivalent to $(13 \; mod \; 5)$
Example :- find $(-13 \; mod \; 5)$.
This can be found by find the number (b) where $5 * b \; > 13$ then let $b = 3$ and $5 * 3 = 15$ which is less than 13 so

$$-13 \; mod \; 5 = 5 * 3 - 13 = 2$$

Properties of Congruences.

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

$$(a \; mod \; n) = \; (b \; mod \; n) \rightarrow a \; \equiv \; b(mod \; n)$$

The difference between a and b will be a multiple of n So $a - b \; = \; kn$ for some value of k

Examples $4 \; \equiv 9 \; \equiv \; 14 \equiv 19 \; \equiv -1 \; \equiv \; -6 \; mod \; 5$

$$73 \; \equiv \; 4(mod \; 23)$$

Properties of Modular Arithmetic.

1. $[(a \; mod \; n) \; + \; (b \; mod \; n)] \; mod \; n \; = \; (a \; + \; b) \; mod \; n$
2. $[(a \; mod \; n) \; - \; (b \; mod \; n)] \; mod \; n \; = \; (a \; - \; b) \; mod \; n$
3. $[(a \; mod \; n) \times \; (b \; mod \; n)] \; mod \; n \; = \; (a \; \times \; b) \; mod \; n$

Examples

$11\ mod\ 8\ =\ 3;\ 15\ mod\ 8\ =\ 7$

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

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

$[(11\ mod\ 8\ )\ -\ (15\ mod\ 8)]\ mod\ 8\ =\ -4\ mod\ 8\ =\ 4$

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

$[(11\ mod\ 8\ )\ \times\ (15\ mod\ 8)]\ mod\ 8 =\ 21\ mod\ 8\ =\ 5$

$(11\ \times\ 15)\ mod\ 8\ =\ 165\ mod\ 8\ =\ 5$

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

Example

To find $(11^7\ mod\ 13)$do the followings

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

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

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

2.2 Greatest Common Divisor(GCD).

Let a and b be two non-zero integers. The greatest common divisor of a and b, denoted gcd(a,b) is the largest of all common divisors of a and b.

When gcd(a,b) = 1, we say that a and b are relatively prime.

It can be calculated using the following equation: -

GCD(a,b)=***G***CD(b,a mod b)

Example :- find the GCD(72,48).

GCD(89,25)=GCD(25, 89 mod 25)= GCD(25, 14)

GCD(25, 14)=GCD(14, 25 mod 14)= GCD(14,11)

GCD(14,11)=GCD(11, 14 mod 11)= GCD(11,3)

GCD(11,3)=GCD(3, 11 mod 3)=GCD(3, 2)

GCD(3,2)=GCD(2, 3 mod 2)=GCD(2,1)

GCD(2,1)=GCD(1, 2 mod 1)=GCD(1,0) so the GCD(89,25)=1

Example 2: GCD (93, 36)

GCD (93, 36) = GCD (36, 93 mod 36) = GCD (36,21)

GCD (36, 21) = GCD (21, 36 mod 21) = GCD (21,15)

GCD (21, 15) = GCD (15, 21 mod 15) = GCD (15,6)

GCD (15, 6) = GCD (6, 15 mod 6) = GCD (6,3)

GCD (6, 3) = GCD (3, 6 mod 3) = GCD (3,0)

2.3 Least Common Multiple (LCM).

The least common multiple of the positive integers a and b is the smallest positive integer that is divisible by both a and b.

The least common multiple of a and b is denoted by LCM(a, b).

•It can be calculated using the following equation: -

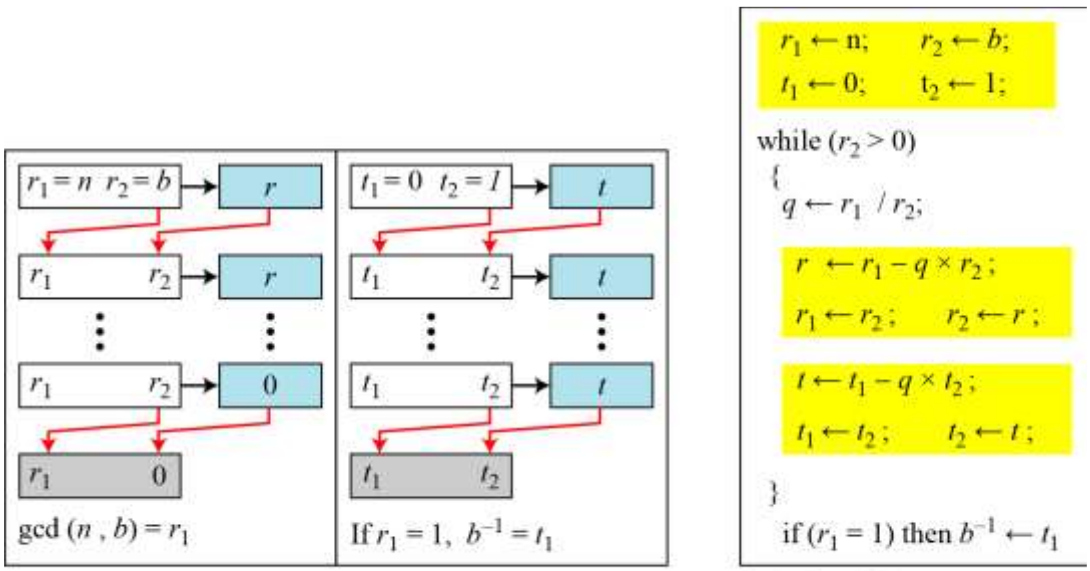
LCM(a, b)=***a*** ∗***b*** / ***G***CD(a, b)

Example :- find the LCM(354,144).

$GCD(354,144) = GCD(144,354\ mod\ 144) = GCD(144,66)$
$GCD(144,66) = GCD(66,144\ mod\ 66) = GCD(66,12)$
$GCD(66,12) = GCD(12,66\ mod\ 12) = GCD(12,6)$
$GCD(12,6) = GCD(6,127\ mod\ 6) = GCD(6,0) = 6$
$LCM(354,143) = (354 * 144)/6 = 8496$

2.4 Multiplicative Inverse

In Zn, two numbers a and b are the multiplicative inverse of each other if The extended Euclidean algorithm finds the multiplicative inverses of b in Zn when n and b are given and gcd (n, b) = 1 as shown in this figure:



a. Process b. Algorithm

Example: - Find the multiplicative inverse of 11 in Z26.

The GCD(26,11)must be 1 in order to find the inverse. By using the extended Euclidean algorithm, we can use this table the inverse of 11 is -7 mod 26=19.

•Or we can find the inverse based on using the equation $n=qn+r$

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	−2
2	11	4	3	1	−2	5
1	4	3	1	−2	5	−7
3	3	1	0	5	−7	26
	1	0		−7	26	

Example: - Find the multiplicative inverse of 11 in Z26.

26=11*2+4
11=4*2+3
4=3*1+1
3=3*1+0

We are now in reverse compensation starting from one as shown

1=4-(3*1)
1=4-(11-(4*2))
1=4-11+4*2
1=3*4-11
1=3*(26-11*2)-11
1=3*26-6*11-11= 3*26-7*11 so the multiplicative inverse of 11 is -7

Example :- Find the multiplicative inverse of 23 in Z100.

100=23*4+8
23=8*2+7
8=7*1+1
7=1*7+0

Now in revers way

1=8-(7*1)
1=8-(23-8*2)
1=8-23+8*2
1=3*8-23
1=3*(100-23*4)-23=3*100-12*23-23=3*100-13*23

So the multiplicative inverse of 23 in Z100 is -23 or 87(-23 mod 100).

Chapter 3

Classical Symmetric Cipher

3.1 The forms of Encryption

Transposition (or permutation) cipher: Transposition cipher keeps the letters the same, but rearranges their order according to a specific algorithm.

Substitution cipher: replacing each element of the plaintext with another element.

Product cipher: using multiple stages of substitutions and transpositions

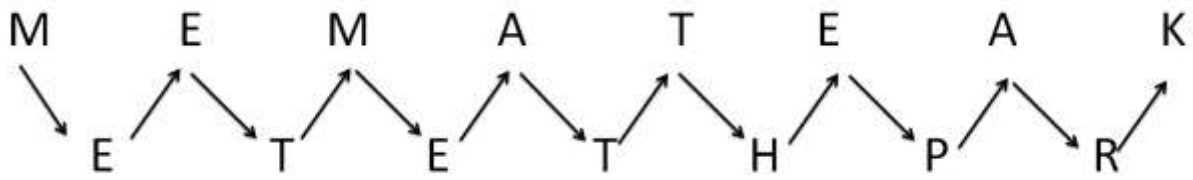
Transposition cipher

3.2 Keyless Transposition Ciphers:

3.2.1 Keyless Transposition Ciphers:

Simple transposition ciphers, which were used in the past, are keyless. A good example of a keyless cipher using the first method is the rail fence cipher. The ciphertext is created reading the pattern row by row. For example, to send the message (Meet me at the park) to Bob, Alice writes

She then creates the ciphertext (MEMATEAKETETHPR).



3.2.2 Columnar Transposition Ciphers.

- Write the message in rows of a fixed length, and then read out again column by column.
- The columns are chosen in some scrambled order.
- Both the length of the rows and the permutation of the columns are usually defined by a key.

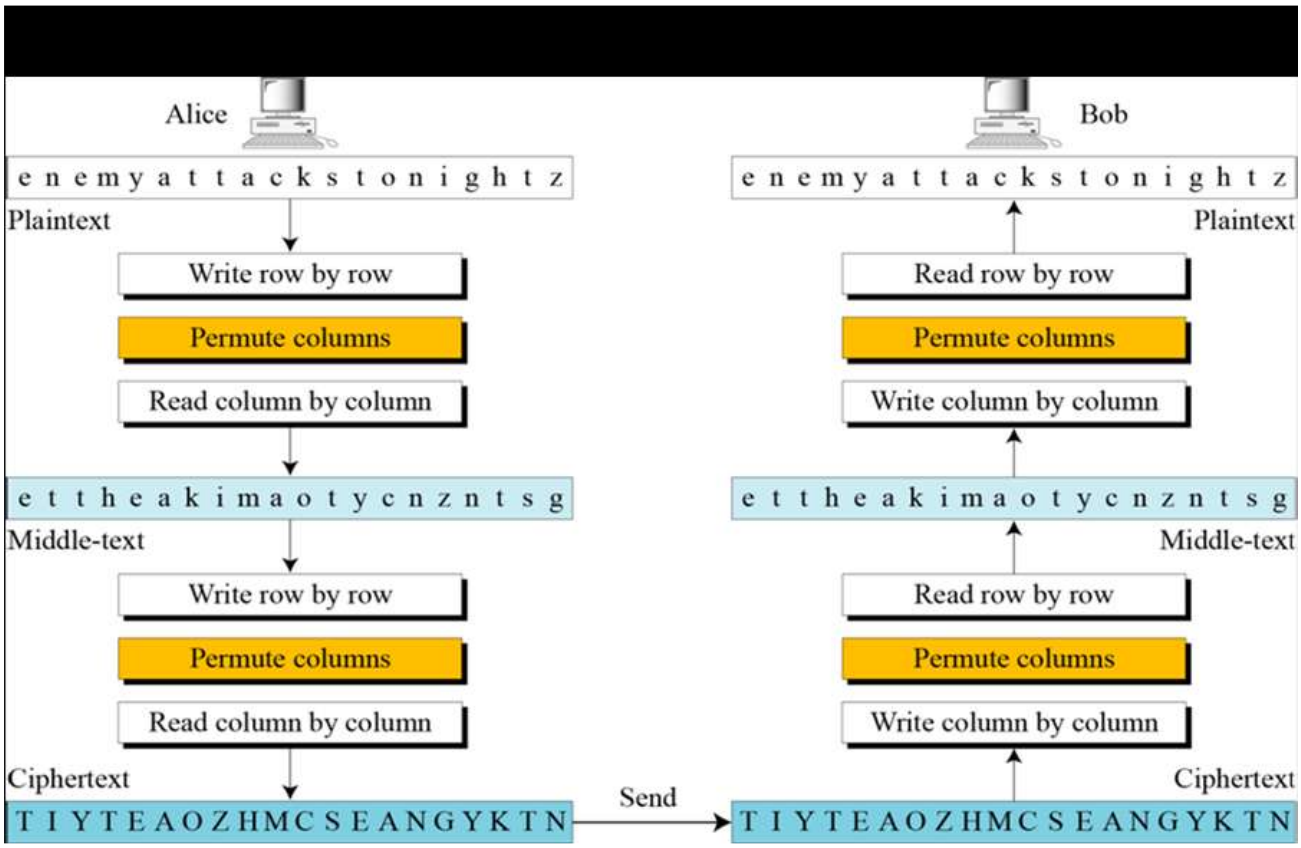
Example: Let the plaintext is (WE ARE DISCOVERED FLEE AT ONCE) the key word be: ZEBRA.

The ciphertext:

Z	E	B	R	A
W	E	A	R	E
D	I	S	C	O
V	E	R	E	D
F	L	E	E	A
T	O	N	C	E

EODAE ASREN EIELO RCEEC WDVFT

Double Columnar Transposition.



3.3 Substitution cipher

3.3.1 Monoalphabetic Ciphers.

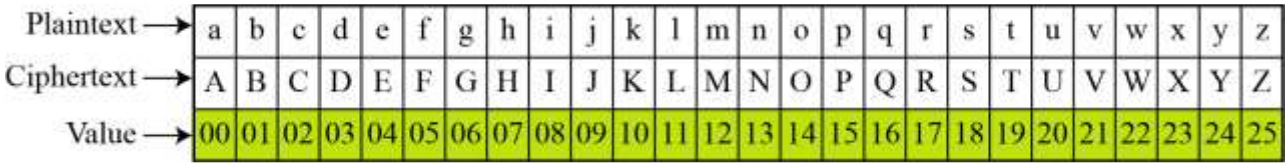
- It is simple substitution
- involves replacing each letter in the message with another letter of the alphabet.

- In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.

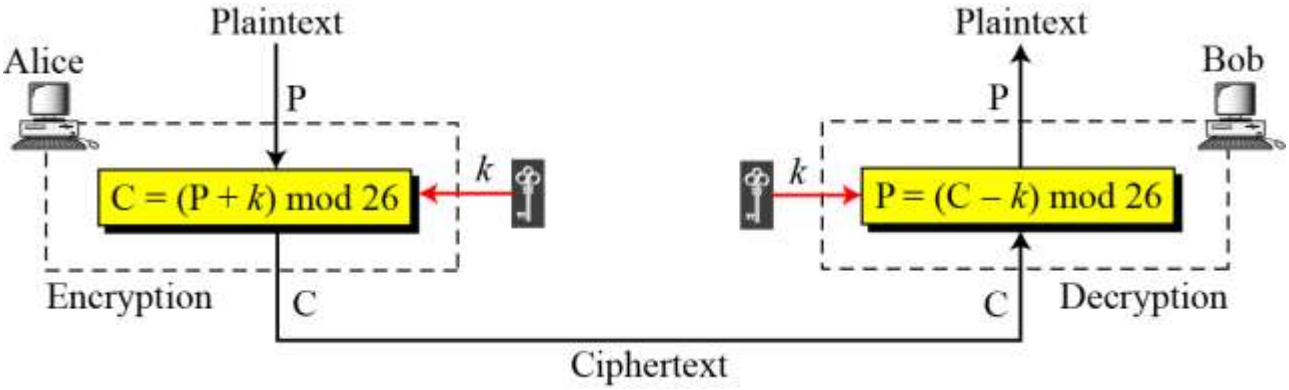
3.3.2 Additive Cipher:

Additive Cipher is the simplest monoalphabetic cipher. It is sometimes called a shift cipher and sometimes a Caesar cipher, but the term additive cipher better reveals its mathematical nature. When the cipher is additive, the plaintext, ciphertext, and key are integers in Z26.

Plaintext and ciphertext in Z26



Additive Cipher



Example

Use the additive cipher with key = 15 to encrypt the plain text (hello).

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

Plaintext	h	e	l	l	o
	7	4	11	11	14

Encryption

$(7 + 15) mod\ 26 = 22 \rightarrow W, (4 + 15) mod\ 26 = 19 \rightarrow T, (11 + 15) mod\ 26 = 0 \rightarrow A,$

$(11 + 15)\ mod\ 26 = 0 \rightarrow A, (14 + 15)\ mod\ 26 = 3 \rightarrow D$

Ciphertext **WTAAD**

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

Ciphertext	W	T	A	A	D
	22	19	0	0	3

Decryption

$(22-15) mod\ 26 = 7 \rightarrow h, (19-15) mod\ 26 = 4 \rightarrow e, (0-15) mod\ 26 = 11 \rightarrow l,$

$(0-15) mod\ 26 = 11 \rightarrow l, (3-15) mod\ 26 = 14 \rightarrow 0$

Ciphertext h e l l o

3.3.3 Caesar Cipher: -

Caesar Cipher Named for Julious Caesar. Caesar used a key of 3 for his communications.

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	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

Cryptanalysis of the Caesar cipher: -

Example: - decrypt the following ciphertext:-

wklv phvvdjh lv qrw wrr kdug wr euhdn

By using the above table, replace the characters as show

ciphertext =	wklv	phvvdjh	lv	qrw	wrr	kdug	wr	euhdn
plaintext =	THIS	MESSAGE	IS	NOT	TOO	HARD	TO	BREAK

Example: Eve has intercepted the ciphertext (UVACLYFZLJBYL). Show how she can use a brute-force attack to break the cipher.

Eve tries keys from 1 to 7. With a key of 7, the plaintext is (not very secure),

which makes sense

Ciphertest :uva clyf zljbyl

Key 1 →vwb dmzg amkczm

Key 2 →wxc enah bnldan

Key 3 →xyd fobi comebo

Key 4 →yze gpcj dpnfcp

Key 5 →zaf hqdk eqogdq

Key 6 →abg irel frpher

Key 7 →not very secure

Table of Frequency of characters in English

Letter	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*
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		

Frequency distributions of Plaintext:-

- E
- T
- A, O, R, N , I
- H , C , D , L, M
- .
- .
- X , J ,Z , Q

Example : - Eve has **intercepted** the following ciphertext. Using a statistical attack, find the plaintext.

When Eve tabulates the frequency of letters in this ciphertext, she gets:

h=26, v=17 and so on.

Table 2-2 Frequencies in Example Cipher

Letter	Count	Percent	Letter	Count	Percent
a	0	0.00	n	0	0.00
b	3	1.80	o	4	2.41
c	0	0.00	p	5	2.99
d	11	6.59	q	16	9.58
e	2	1.20	r	9	5.39
f	6	3.61	s	3	1.80
g	4	2.40	t	0	0.00
h	26	15.56	u	8	4.79
i	2	1.20	v	17	10.18
j	5	2.99	w	14	8.38
k	5	2.99	x	5	2.99
l	16	9.58	y	4	2.40
m	0	0.00	z	2	1.20

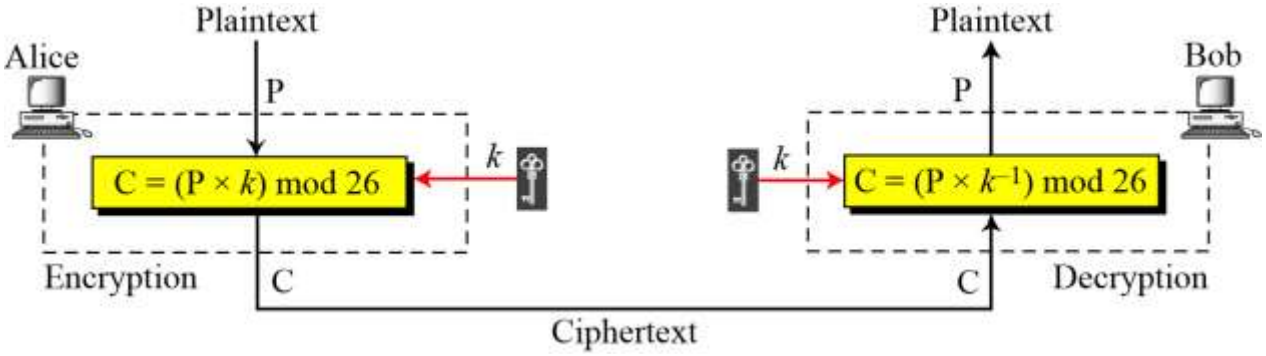
So we will replace each character with the corresponding high frequency in plaintext as shown: -

Plaintext = ENCRYPTION IS A MEANS OF ATTAINING SECURE COMMUNICATION

Which means that the key is =3 ? How?

Multiplicative Ciphers: - In a multiplicative cipher, the plaintext and ciphertext are integers in Z26; the key is an integer in Z26*.

Multiplicative cipher



The key domain for any multiplicative cipher which must be in Z26*, is the set that has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.(why)

Example: - We use a multiplicative cipher to encrypt the message "hello" with a key of 7. The ciphertext is "XCZZU".

Cryptanalyses of the multiplicative cipher based on finding the multiplication inverse of the key (where the multiplication inverse of 7 is 15) as shown

Ciphertext X → 23 Decryption: (23 * 15) mod 26 plaintext= 7→h

Ciphertext C → 2 Decryption: (2 * 15) mod 26 plaintext= 4→e

Ciphertext Z → 25 Decryption: (25 * 15) mod 26 plaintext=11→l

Ciphertext Z → 25 Decryption: (25 * 15) mod 26 plaintext=11→l

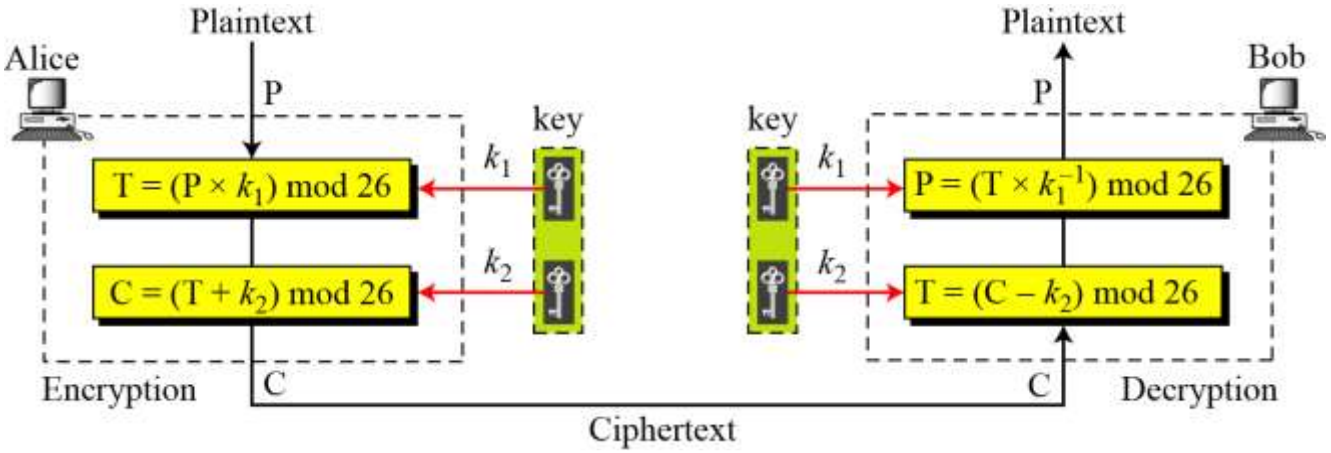
Ciphertext U → 20 Decryption: (20 * 15) mod 26 plaintext=14→o

3.3.4 Affine Ciphers

- The affine cipher uses a pair of keys in which the first key is from Z26* and the second is from Z26. The size of the key domain is 26 × 12 = 312.
- The additive cipher is a special case of an affine cipher in which k1 = 1. The multiplicative cipher is a special case of affine cipher in which k2 = 0.

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

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2



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

P: h → 07	Encryption: (07 × 7 + 2) mod 26	C: 25 → Z
P: e → 04	Encryption: (04 × 7 + 2) mod 26	C: 04 → E
P: l → 11	Encryption: (11 × 7 + 2) mod 26	C: 01 → B
P: l → 11	Encryption: (11 × 7 + 2) mod 26	C: 01 → B
P: o → 14	Encryption: (14 × 7 + 2) mod 26	C: 22 → W

To decrypt the message "ZEBBW" with the key pair (7, 2) in modulus 26. where where the multiplication inverse of **7 is 15**

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

3.4 Polyalphabetic Ciphers

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

The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = (k_1, P_1, P_2, \ldots)$

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

3.4.1 Autokey Cipher: -

Assume that Alice and Bob agreed to use an autokey cipher with initial key value k1 = 12. Now Alice wants to send Bob the message "Attack is today". Enciphering is done character by character as shown :-

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**

3.4.2 Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (minus duplicates)
- fill rest of matrix with other letters in alphabetical order
- eg. using the keyword MONARCHY

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

Encrypting and Decrypting

- plaintext is encrypted two letters at a time
- if a pair is a repeated letter, insert filler like 'X'
    - e.g balloon is treated as ba lx lo on
- if both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
    - e.g ar is encrypted as RM
- if both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom)
    - e.g mu is encrypted as CM
- otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair
    - e.g hs is encrytped as BP, ea is encrypted as IM(or JM)

Example if the key is PROBLEMS use Playfair to encipher the message

SHE WENT TO THE STORE

Solution:

When we pair up the letters they get grouped as follows:

SH EW EN TT OT HE ST OR E

But, we are not allowed to encipher any double letters. So, in this case, we will insert an Q into the plaintext. (If Q is a double letter, then insert another infrequent letter, say X.)

SH EW EN TQ TO TH ES TO RE

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

To encipher pairs of letters, adhere to the following rules:

1.If the two letters are on the same row of the chart, like "ES", then replace each letter by the letter to the right. (If necessary, wrap around to the left end of the row. So "ES" encrypts to "MA".

2. If the two letters are on the same column of the chart, like, "TH", then replace each letter by the letter below it. (If necessary, wrap around to the top end of the column.) So "TH" encrypts to "YT".

3. If two letters are on a different row and column, like, "SH", then replace each letter by another letter on its same row, but in the column of the other letter. So "SH" encrypts to "AG".

Using these rules, here is the encryption of the plaintext above:

Plaintext : SH EW EN TQ TO TH ES TO RE

Ciphertext: AG MV MK UT QB YT MA QB PM

To decipher, ignore rule 1. In rules 2 and 3 shift up and left instead of down and right. Rule 4 remains the same. Once you are done, drop any extra Xs that don't make sense in the final message and locate any missing Qs or any Is that should be Js.

Vigenere Cipher

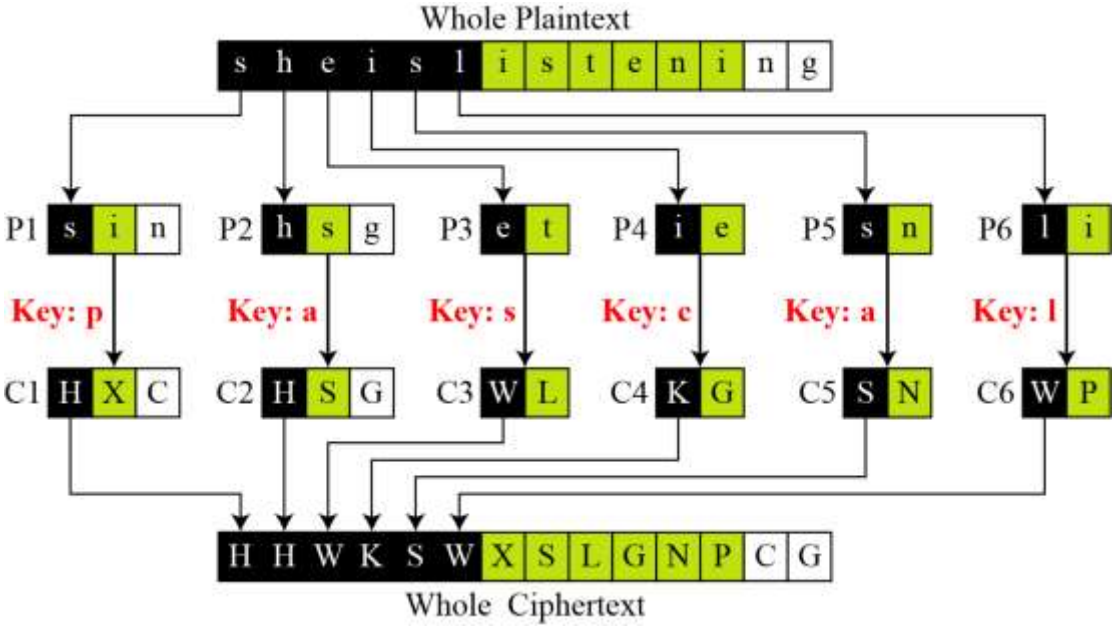
$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad K = [(k_1, k_2, \ldots, k_m), (k_1, k_2, \ldots, k_m), \ldots]$

$\text{Encryption: } C_i = P_i + k_i \qquad \text{Decryption: } P_i = C_i - k_i$

We can encrypt the message “She is listening” using the 6-character keyword “PASCAL” (15, 0, 18, 2, 0, 11).

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	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 values:	**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**

Vigenere cipher can be seen as combinations of m additive ciphers.



3.4.3 Hill Cipher

The Hill Cipher uses matrix multiplication to encrypt a message.

- First, you need to assign two numbers to each letter in the alphabet and also assign numbers to space, . , and ? or !.
- The key space is the set of all **invertible matrices** over Z26. 26 was chosen because there are 26 characters, which solves some problems later on.

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

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

The key matrix in the Hill cipher needs to have a **multiplicative inverse**.

For example, the plaintext "code is ready" can make a 3 × 4 matrix when adding extra bogus character "z" to the last block and removing the spaces. The ciphertext is "OHKNIHGKLISS".

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

a. Encryption

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

b. Decryption

$$M^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{det(M)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Message to encrypt = HELLO WORLD

$$\begin{bmatrix} H \\ E \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} L \\ L \end{bmatrix} = \begin{bmatrix} 11 \\ 11 \end{bmatrix}$$

$$\begin{bmatrix} O \\ W \end{bmatrix} = \begin{bmatrix} 14 \\ 22 \end{bmatrix}$$

$$\begin{bmatrix} O \\ R \end{bmatrix} = \begin{bmatrix} 14 \\ 17 \end{bmatrix}$$

$$\begin{bmatrix} L \\ D \end{bmatrix} = \begin{bmatrix} 11 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \text{x} \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 14+4 \\ 21+16 \end{bmatrix} = \begin{bmatrix} 18 \\ 37 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 18 \\ 11 \end{bmatrix} = \begin{bmatrix} S \\ L \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \text{x} \begin{bmatrix} 11 \\ 11 \end{bmatrix} = \begin{bmatrix} 22+11 \\ 33+44 \end{bmatrix} = \begin{bmatrix} 33 \\ 77 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 7 \\ 25 \end{bmatrix} = \begin{bmatrix} H \\ Z \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \text{x} \begin{bmatrix} 14 \\ 22 \end{bmatrix} = \begin{bmatrix} 28+22 \\ 42+88 \end{bmatrix} = \begin{bmatrix} 50 \\ 130 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 24 \\ 0 \end{bmatrix} = \begin{bmatrix} Y \\ A \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \text{x} \begin{bmatrix} 14 \\ 17 \end{bmatrix} = \begin{bmatrix} 28+17 \\ 42+68 \end{bmatrix} = \begin{bmatrix} 45 \\ 110 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 19 \\ 6 \end{bmatrix} = \begin{bmatrix} T \\ G \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \text{x} \begin{bmatrix} 11 \\ 3 \end{bmatrix} = \begin{bmatrix} 22+3 \\ 33+12 \end{bmatrix} = \begin{bmatrix} 25 \\ 45 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 25 \\ 19 \end{bmatrix} = \begin{bmatrix} Z \\ T \end{bmatrix}$$

HELLO WORLD has been encrypted to **SLHZY ATGZT**

A	1	2	5	7	9	11	15	17	19	21	23	25
A^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Message to encrypt = SLHZYATGZT

$$\begin{bmatrix} S \\ L \end{bmatrix} = \begin{bmatrix} 18 \\ 11 \end{bmatrix} \quad \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \text{x} \begin{bmatrix} 18 \\ 11 \end{bmatrix} = \begin{bmatrix} 108+55 \\ 270+176 \end{bmatrix} = \begin{bmatrix} 163 \\ 446 \end{bmatrix}$$

$$\begin{bmatrix} H \\ Z \end{bmatrix} = \begin{bmatrix} 7 \\ 25 \end{bmatrix} \quad \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \text{x} \begin{bmatrix} 7 \\ 25 \end{bmatrix} = \begin{bmatrix} 42+125 \\ 105+400 \end{bmatrix} = \begin{bmatrix} 167 \\ 505 \end{bmatrix}$$

$$\begin{bmatrix} Y \\ A \end{bmatrix} = \begin{bmatrix} 24 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \text{x} \begin{bmatrix} 24 \\ 0 \end{bmatrix} = \begin{bmatrix} 144+0 \\ 360+0 \end{bmatrix} = \begin{bmatrix} 144 \\ 360 \end{bmatrix}$$

$$\begin{bmatrix} T \\ G \end{bmatrix} = \begin{bmatrix} 19 \\ 6 \end{bmatrix} \quad \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \text{x} \begin{bmatrix} 19 \\ 6 \end{bmatrix} = \begin{bmatrix} 114+30 \\ 285+96 \end{bmatrix} = \begin{bmatrix} 144 \\ 381 \end{bmatrix}$$

$$\begin{bmatrix} Z \\ T \end{bmatrix} = \begin{bmatrix} 25 \\ 19 \end{bmatrix} \quad \begin{bmatrix} 6 & 5 \\ 15 & 16 \end{bmatrix} \text{x} \begin{bmatrix} 25 \\ 19 \end{bmatrix} = \begin{bmatrix} 150+95 \\ 375+304 \end{bmatrix} = \begin{bmatrix} 245 \\ 679 \end{bmatrix}$$

$$\begin{bmatrix} 163 \\ 446 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} H \\ E \end{bmatrix}$$

$$\begin{bmatrix} 167 \\ 505 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 11 \\ 11 \end{bmatrix} = \begin{bmatrix} L \\ L \end{bmatrix}$$

$$\begin{bmatrix} 144 \\ 360 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 14 \\ 22 \end{bmatrix} = \begin{bmatrix} O \\ W \end{bmatrix}$$

$$\begin{bmatrix} 144 \\ 381 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 14 \\ 17 \end{bmatrix} = \begin{bmatrix} O \\ R \end{bmatrix}$$

$$\begin{bmatrix} 245 \\ 679 \end{bmatrix} \text{Mod } 26 = \begin{bmatrix} 11 \\ 3 \end{bmatrix} = \begin{bmatrix} L \\ D \end{bmatrix}$$

SLHZYATGZT has been decrypted to

HELLO WORLD

Encryption: Cipher Tet = (Plain Tet x Key) Mod 26

Decryption: Plain Tet = (Cipher Tet x Key^{-1}) Mod 26

Example: Message ATTACK IS TONIGHT

$$Key = \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$$

Encryption

Message: ATTACK IS TONIGHT

Assign: A-Z 0-25

$$\begin{bmatrix} A & T & T \\ A & C & K \\ I & S & T \\ O & N & I \\ G & H & T \end{bmatrix} = \begin{bmatrix} 0 & 19 & 19 \\ 0 & 2 & 10 \\ 8 & 18 & 19 \\ 14 & 13 & 8 \\ 6 & 7 & 19 \end{bmatrix}$$

Encryption

Cipher Tet = (Plain Tet x Key) Mod 26

$$\begin{bmatrix} 0 & 19 & 19 \\ 0 & 2 & 10 \\ 8 & 18 & 19 \\ 14 & 13 & 8 \\ 6 & 7 & 19 \end{bmatrix} \times \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix} mod\ 26$$

$$\begin{bmatrix} 551 & 247 & 646 \\ 130 & 58 & 204 \\ 555 & 318 & 789 \\ 374 & 289 & 638 \\ 329 & 199 & 562 \end{bmatrix} mod\ 26 = \begin{bmatrix} 5 & 13 & 22 \\ 0 & 6 & 22 \\ 9 & 6 & 9 \\ 10 & 3 & 13 \\ 17 & 17 & 16 \end{bmatrix}$$

$$\begin{bmatrix} A & T & T \\ A & C & K \\ I & S & T \\ O & N & I \\ G & H & T \end{bmatrix} ==> \begin{bmatrix} F & N & W \\ A & G & W \\ J & G & J \\ K & D & N \\ R & R & Q \end{bmatrix}$$

Decryption Plain Tet = (Cipher Tet x Key^{-1}) Mod 26

You need to find: key^{-1}

key^{-1} =[Det (Key)]$^{-1}$ X Adj(Key)

Step 1: Find Determinant of Key

Adj (key)

Step 2: Transpose Key Matrix

Step 3: Find Minor

Step 4: Find Co-Factor

Decryption

$$Key = \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$$

Step 1: Find Determinant of Key

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} \square & \square & \square \\ \square & e & f \\ \square & h & i \end{vmatrix} - b\begin{vmatrix} \square & \square & \square \\ d & \square & f \\ g & \square & i \end{vmatrix} + c\begin{vmatrix} \square & \square & \square \\ d & e & \square \\ g & h & \square \end{vmatrix}$$

$$= a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

$$= aei + bfg + cdh - ceg - bdi - afh.$$

- Det (Key) = $\begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$ Mod 26 = 03

  = 3*(9*17 – 17*4) – 10*(20*17 – 17*9) + 20*(20*4 – 9*9)

  = (-1635) Mod 26 = (-23) Mod 26 = 03

  $[Det\ (Key)]^{-1}$ = 03^{-1} Mod 26 = 09

Step 2: Transpose Key Matrix

$$\text{Key} = \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$$

$$\textbf{Trans(Key)} = \begin{bmatrix} 3 & 20 & 9 \\ 10 & 9 & 4 \\ 20 & 17 & 17 \end{bmatrix}$$

Step 3: Find Minor

- Trans(Key) = $\begin{bmatrix} 3 & 20 & 9 \\ 10 & 9 & 4 \\ 20 & 17 & 17 \end{bmatrix}$ $\begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$ *To find Minor of*

$a11 = a22*a33 - a32*a23 = 85$

$a11 = 85$ $a12 = 90$ $a13 = (-10)$
$a21 = 187$ $a22 = (-129)$ $a23 = (-349)$
$a31 = (-1)$ $a32 = (-78)$ $a33 = (-173)$

$$\textbf{Minor} = \begin{bmatrix} 85 & 90 & -10 \\ 187 & -129 & -349 \\ -1 & -78 & -173 \end{bmatrix}$$

Step 4: Find Co-Factor

$$\textbf{Minor} = \begin{bmatrix} 85 & 90 & -10 \\ 187 & -129 & -349 \\ -1 & -78 & -173 \end{bmatrix}$$

Put Sign According to (-1) $^{i+j}$

$$\begin{bmatrix} + & - & + \\ - & + & - \\ + & - & + \end{bmatrix} \rightarrow \begin{bmatrix} 85 & -90 & -10 \\ -187 & -129 & 349 \\ -1 & 78 & -173 \end{bmatrix}$$

$$Key^{-1} = [Det\ (Key)]^{-1} \times Adj\ (Key)$$

$$= 09 * \begin{bmatrix} 85 & -90 & -10 \\ -187 & -129 & 349 \\ -1 & 78 & -173 \end{bmatrix} Mod\ 26$$

$$= \begin{bmatrix} 765 & -810 & -90 \\ -1683 & -1161 & 3141 \\ -9 & 702 & -1557 \end{bmatrix} Mod\ 26 = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$$

Finally Plain Text = (Cipher Text x Key^{1}) Mod 26

$$= \begin{bmatrix} 5 & 13 & 22 \\ . & . & . \\ . & . & . \\ . & . & . \\ . & . & . \end{bmatrix} \times \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 520 & 227 & 409 \\ . & . & . \\ . & . & . \\ . & . & . \\ . & . & . \end{bmatrix} Mod\ 26 = \begin{bmatrix} 0 & 19 & 19 \\ . & . & . \\ . & . & . \\ . & . & . \\ . & . & . \end{bmatrix} = A\ T\ T$$

Exercise

3.4.4 One-Time Pad

The one-time pad, which is a provably secure cryptosystem, was developed by **Gilbert Vernam** in 1918.The message is represented as a binary string (a sequence of 0's and 1's using a coding mechanism such as ASCII coding.

The key is a truly random sequence of 0's and 1's of the same length as the message.

The encryption is done by adding the key to the message modulo 2, bit by bit. This process is often called *exclusive or*, and is denoted by XOR. The symbol ⊕ is used

a	b	$c = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

message =‘IF’

then its ASCII code =(1001001 1000110)

key = (1010110 0110001)

Encryption:

1001001 1000110 plaintext

1010110 0110001 key

0011111 1110110 ciphertext

Decryption:

0011111 1110110 ciphertext

1010110 0110001 key

1001001 1000110 plaintext

Why OTP is provably secure?

- The security depends on the randomness of the key.
- It is hard to define randomness.
- In cryptographic context, we seek two fundamental properties in a binary random key sequence:

1. Unpredictability:
2. Balanced (Equal Distribution):

Unpredictability:

Independent of the number of the bits of a sequence observed, the probability of guessing the next bit is not better than ½. Therefore, the probability of a certain bit being 1 or 0 is exactly equal to ½.

Balanced (Equal Distribution):

The number of 1's and 0's should be equal.

Chapter 4

Modern Symmetric Ciphers

(Stream Cipher and Block Cipher)

4.1 Introduction

Symmetric cryptography is split into block ciphers and stream ciphers, which are easy to distinguish.

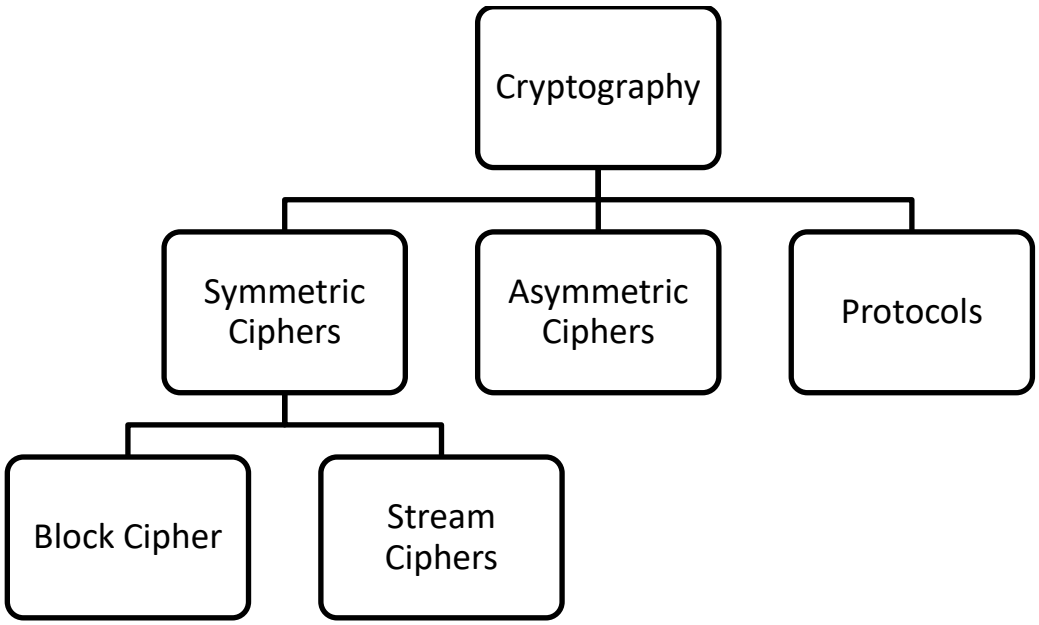


Fig. 4.1 Main areas within cryptography

4.2 Stream cipher

A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along. Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate.

Stream ciphers encrypt bits individually. This is achieved by adding a bit from a key stream to a plaintext bit. There are synchronous stream ciphers where the key stream depends only on the key, and asynchronous ones where the key stream also depends on the ciphertext. If the dotted line in Fig. 4.2 is present, the stream cipher is an

asynchronous one. Most practical stream ciphers are synchronous ones. An example of an asynchronous stream cipher is the cipher feedback (CFB) mode introduced in

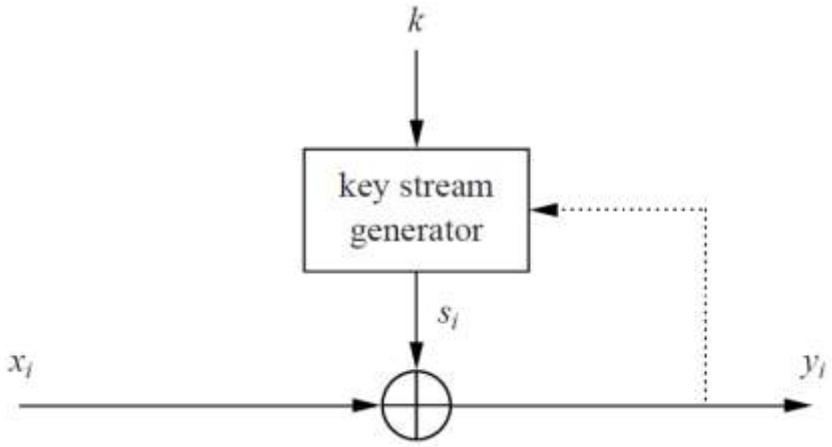


Fig. 4.2 Synchronous and asynchronous stream ciphers

4.3 Block ciphers

Block ciphers encrypt an entire block of plaintext bits at a time with the same key. This means that the encryption of any plaintext bit in a given block depends on every other plaintext bit in the same block. In practice, the vast majority of block ciphers either have a block length of 128 bits (16 bytes) such as the advanced encryption standard (AES), or a block length of 64 bits (8 bytes) such as the data encryption standard (DES) or triple DES (3DES) algorithm. All of these ciphers are introduced in later chapters.

4.4 Ciphers vs. Block ciphers

1- In practice, in particular for encrypting computer communication on the Internet, block ciphers are used more often than stream ciphers.

2- Because stream ciphers tend to be small and fast, they are particularly relevant for applications with little computational resources, e.g., for cell phones or other small embedded devices. A prominent example for a stream cipher is the A5/1 cipher, which is part of the GSM mobile phone standard and is used for voice encryption. However, stream ciphers are sometimes also used for encrypting Internet traffic, especially the stream cipher RC4.

3- Traditionally, it was assumed that stream ciphers tended to encrypt more efficiently than block ciphers. Efficient for software-optimized stream ciphers means that they need fewer processor instructions (or processor cycles) to encrypt one bit of plaintext. For hardware-optimized stream ciphers, efficient

means they need fewer gates (or smaller chip area) than a block cipher for encrypting at the same data rate. However, modern block ciphers such as AES are also very efficient in software. Moreover, for hardware, there are also highly efficient block ciphers, such as PRESENT, which are as efficient as very compact stream ciphers.

4.5 Encryption and Decryption with Stream Ciphers

As mentioned above, stream ciphers encrypt plaintext bits individually. The question now is: How does encryption of an individual bit work? The answer is surprisingly simple: Each bit xi is encrypted by adding a secret key stream bit s_i modulo 2.

The plaintext, the ciphertext and the key stream consist of individual bits, $i.e., x_i, y_i, s_i \in \{0,1\}$.

$Encryption: y_i = e_{s_i}(x_i) \equiv x_i + s_i \; mod \; 2.$

$Decryption: x_i = d_{s_i}(y_i) \equiv y_i + s_i \; mod \; 2.$

Note **Modulo 2** addition is equivalent to the **XOR operation**

4.6 Shift Register-Based Stream Ciphers

As we have learned so far, practical stream ciphers use a stream of key bits $s_1, s_2, \ldots$ that are generated by the key stream generator, which should have certain properties. An elegant way of realizing long pseudorandom sequences is to use linear feedback shift registers (LFSRs). LFSRs are easily implemented in hardware and many, but certainly not all, stream ciphers make use of LFSRs. A prominent example is the A5/1 cipher, which is standardized for voice encryption in GSM. As we will see, even though a plain LFSR produces a sequence with good statistical properties, it is cryptographically weak. However, combinations of LFSRs, such as A5/1 or the cipher **Trivium**, can make secure stream ciphers. It should be stressed that there are many ways for constructing stream ciphers. This section only introduces one of several popular approaches.

4.7 Linear Feedback Shift Registers (LFSR)

An LFSR consists of clocked storage elements (*flip-flops*) and a feedback path. The number of storage elements gives us the degree of the LFSR. In other words, an LFSR with $m \; flip - flops$ is said to be of degree m. The feedback network computes the input for the last flip-flop as XOR-sum of certain flip-flops in the shift register. Example 1. Simple LFSR We consider an LFSR of degree m = 3 with flip-flops FF_2, FF_1, FF_0, and a feedback path as shown in Fig. 4.3. The internal state bits are denoted by s_i and are shifted by one to the right with each clock tick. The rightmost state bit is also the current output bit. The leftmost state bit is computed in the feedback path, which is the XOR sum of some of the flip-flop values in the previous clock period. Since the XOR

is a linear operation, such circuits are called linear feedback shift registers. If we assume an initial state of $(s_2 = 1, s_1 = 0, s_0 = 0)$,

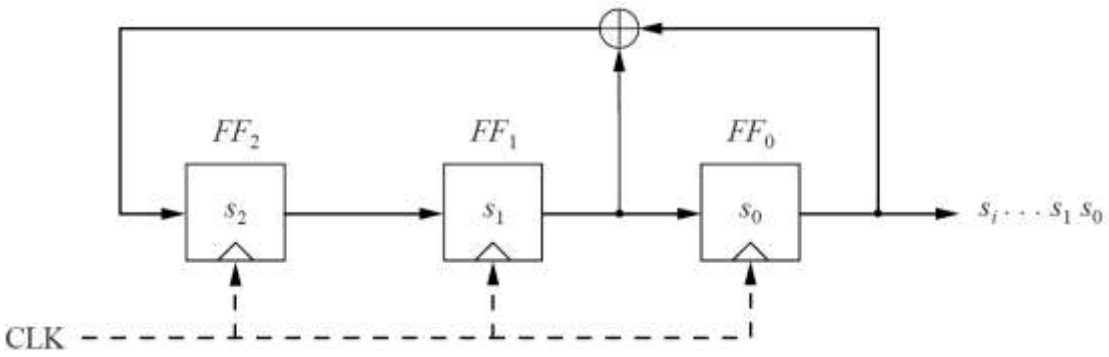


Fig. 4.3 Linear feedback shift register of degree 3 with initial values s_2, s_1, s_0

Table 4.1 gives the complete sequence of states of the LFSR.

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Note that the rightmost column is the output of the LFSR. One can see from this example that the LFSR starts to repeat after clock cycle 6. This means the LFSR output has period of length 7 and has the form:

0010111 0010111 0010111…

There is a simple formula which determines the functioning of this LFSR. Let's

look at how the output bits s_i are computed, assuming the initial state bits s0, s1, s2:

$$s_3 \equiv s_1 + s_0 mod\ 2$$

$$s_4 \equiv s_2 + s_1 mod\ 2$$

$$s_5 \equiv s_3 + s_2 mod\ 2$$

$$\vdots$$

In general, the output bit is computed as:

$$s_i + 3 \equiv s_i + 1 + s_i mod\ 2$$

where i = 0,1,2, . . .

We will now look at general LFSRs

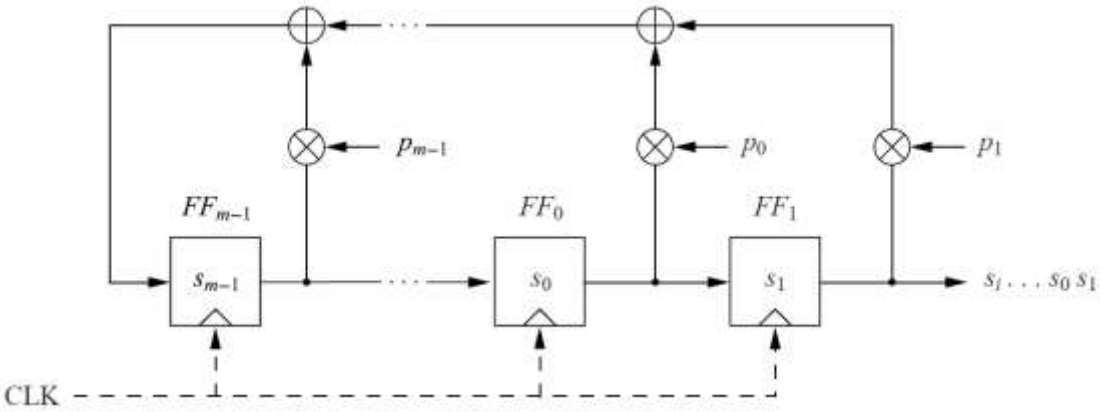


Fig. 4.4 General LFSR with feedback

$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \bmod 2; \quad s_i, p_j \in \{0,1\};\ i = 0,1,2,\ldots$$

The maximum sequence length generated by an LFSR of degree m is $2^m - 1$.

4.8 The Data Encryption Standard (DES) and Alternatives

The Data Encryption Standard (DES) has been by far the most popular block cipher for most of the last 30 years. Even though it is nowadays not considered secure against a determined attacker because the DES key space is too small, it is still used in legacy applications. Furthermore, encrypting data three times in a row with DES — a process referred to as 3DES or triple DES — yields a very secure cipher which is still widely used today (Section 3.5 deals with 3DES.) Perhaps what is more important, since DES is by far the best-studied symmetric algorithm, its design principles have inspired many current ciphers. Hence, studying DES helps us to understand many other symmetric algorithms.

4.9 Introduction to DES

In 1972 a mildly revolutionary act was performed by the US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST): the NBS initiated a request for proposals for a standardized cipher in the USA. The idea was to find a single secure cryptographic algorithm which could be used for a variety of applications. Up to this point in time governments had always considered cryptography, and in particular cryptanalysis, so crucial for national security that it had to be kept secret. However, by the early 1970s the demand for encryption for commercial applications such as banking had become so pressing that it could not be

ignored without economic consequences. The NBS received the most promising candidate in 1974 from a team of cryptographers working at IBM. The algorithm IBM submitted was based on the cipher Lucifer. Lucifer was a family of ciphers developed by Horst Feistel in the late 1960s, and was one of the first instances of block ciphers operating on digital data. Lucifer is a Feistel cipher which encrypts blocks of 64 bits using a key size of 128 bits.

In order to investigate the security of the submitted ciphers, the NBS requested the help of the National Security Agency (NSA), which did not even admit its existence at that point in time. It seems certain that the NSA influenced changes to the cipher, which was rechristened DES. One of the changes that occurred was that DES is specifically designed to withstand differential cryptanalysis, an attack not known to the public until 1990. It is not clear whether the IBM team developed the knowledge about differential cryptanalysis by themselves or whether they were guided by the NSA. Allegedly, the NSA also convinced IBM to reduce the Lucifer key length of 128 bit to 56 bit, which made the cipher much more vulnerable to brute-force attacks.