VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

COURSE NAME: CRYPTOGRAPHY AND NETWORK SECURITY

COURSE CODE: R16A1051

COURSE INSTRUCTOR: MADHU BABU JANJANAM, ASSOC. PROF, CSE

UNIT: 2

By the end of this session…

- Describe Cryptography and characterization of Cryptographic Algorithms.
- Describe Cryptanalysis and various Cryptanalysis attacks.
- Describe basic functionality of symmetric cryptography.
- Describe various traditional symmetric ciphers

Cryptography

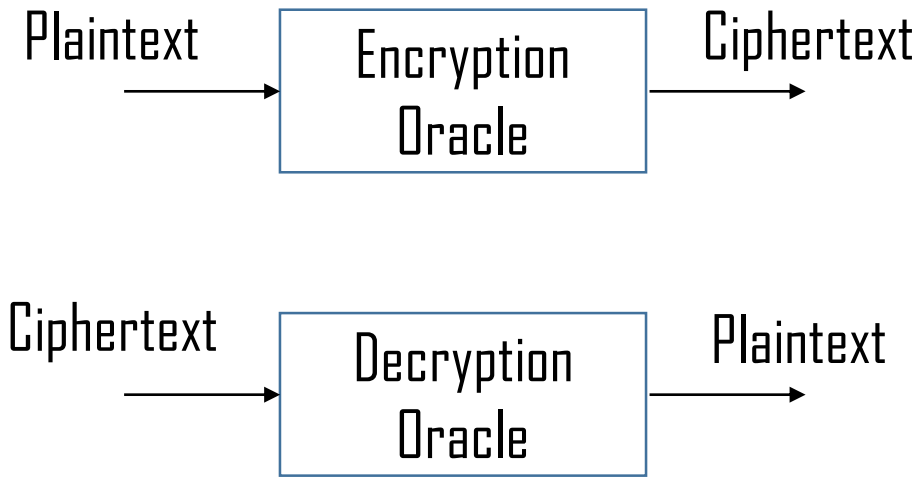
Subtitution Vs Transposition

Symmetric Vs Asymmetric

Stream Vs Block

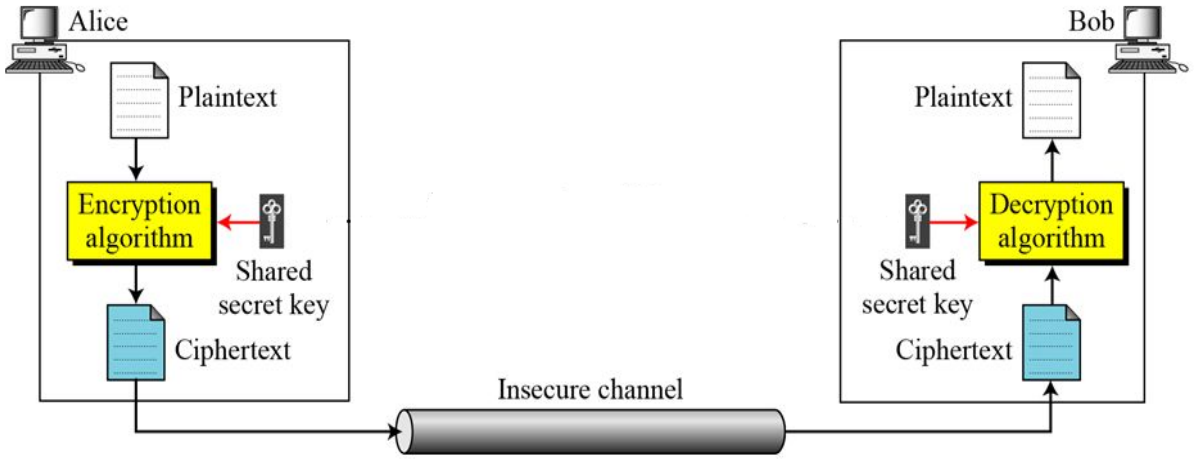
- Study of algorithms called Ciphers, meant for converting plaintext to ciphertext and vice-versa
- Characterized based on three dimensions:
  - Type of operations used for transforming plaintext in to ciphertext.
  - The number of keys used
  - The way in which plaintext is processed

Cryptanalysis



- Meant for procedures used to attack Cryptography.
- 5 types of cryptanalytic attacks
  - Ciphertext only
  - Known Plaintext
  - Chosen Plaintext
  - Chosen Ciphertext
  - Chosen Text
- Even more attacks
  - Brute-force attack
  - Dictionary attack

Symmetric Cryptography

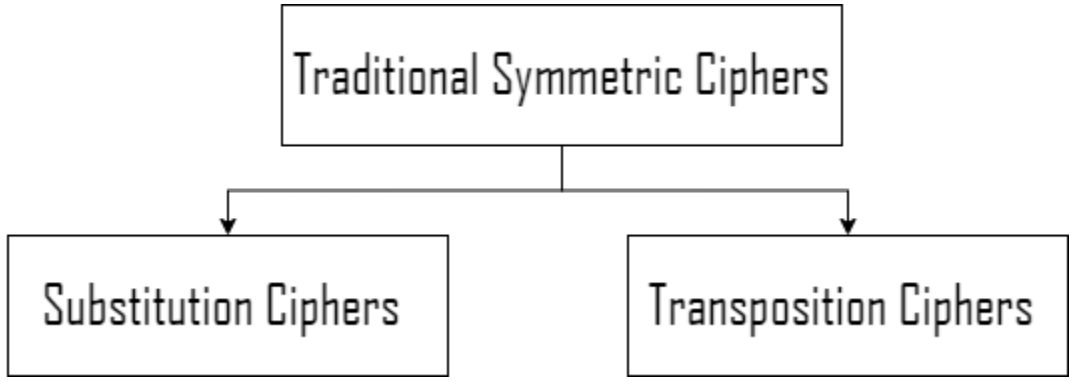


$C = E_k(P)$

$P = D_k(C)$

- Uses single shared secret key for encryption and decryption.
- Sender uses Encryption algorithm and shared secret key to produce ciphertext.
- Receiver uses Decryption algorithm and shared secret key to produce plaintext.
- The data transmission can be done through an insecure channel, where an attacker is one of the participant.
- Combination of Encryption algorithm and Decryption algorithm called cipher.
- Cipher is openly announced and key is kept secret.

Categories of Traditional Symmetric Ciphers



- Categorized into two types
  - Substitution Ciphers
  - Transposition Ciphers
- Substitution Ciphers substitutes another character/symbol in place of original plaintext character.
  - Eg; A is replaced with M, Z is replaced with U.
- Transposition Ciphers changes the position of plaintext characters/symbols to form ciphertext.
  - Eg. Characters in 'help' can be replaced as 'lehp'

Caesar Cipher

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

$$C = E(P, K) = (P + K)\ mod\ 26$$

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

- Earliest known Substitution cipher designed by Julius Caesar.
- Involves replacing each plaintext character with the letter standing 'k' spaces further down,
- For example: With key '3'

M	E	E	T		M	E		A	F	T	E	R		T	H	E		P	A	R	T	Y
P	H	H	W		P	H		D	I	W	H	U		W	K	H		S	D	U	W	B

- Exercise: Find the ciphertext "WELCOME TO VVIT" using key 5.

Cryptanalysis on Caesar Cipher

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

- Brute-force Cryptanalysis is easily possible.
- Total possible keys 25.
- Three important characteristics that made Brute-force attack possible.
  - The encryption and decryption algorithm is known
  - There are only 25 possible keys to try.
  - The language of plaintext is known and easily recognizable.
- Break "GHSW RI FVH".

Monoalphabetic Cipher

- Also one of the substitution cipher,
- Meant for countering Brute-force attacks
- Each character in plaintext is replaced with different character irrespective of numerical key, which make the key length as 26 characters.
- Dramatic increase in key size – 26, so possible number of keys – 26!, > 4×10^{26}

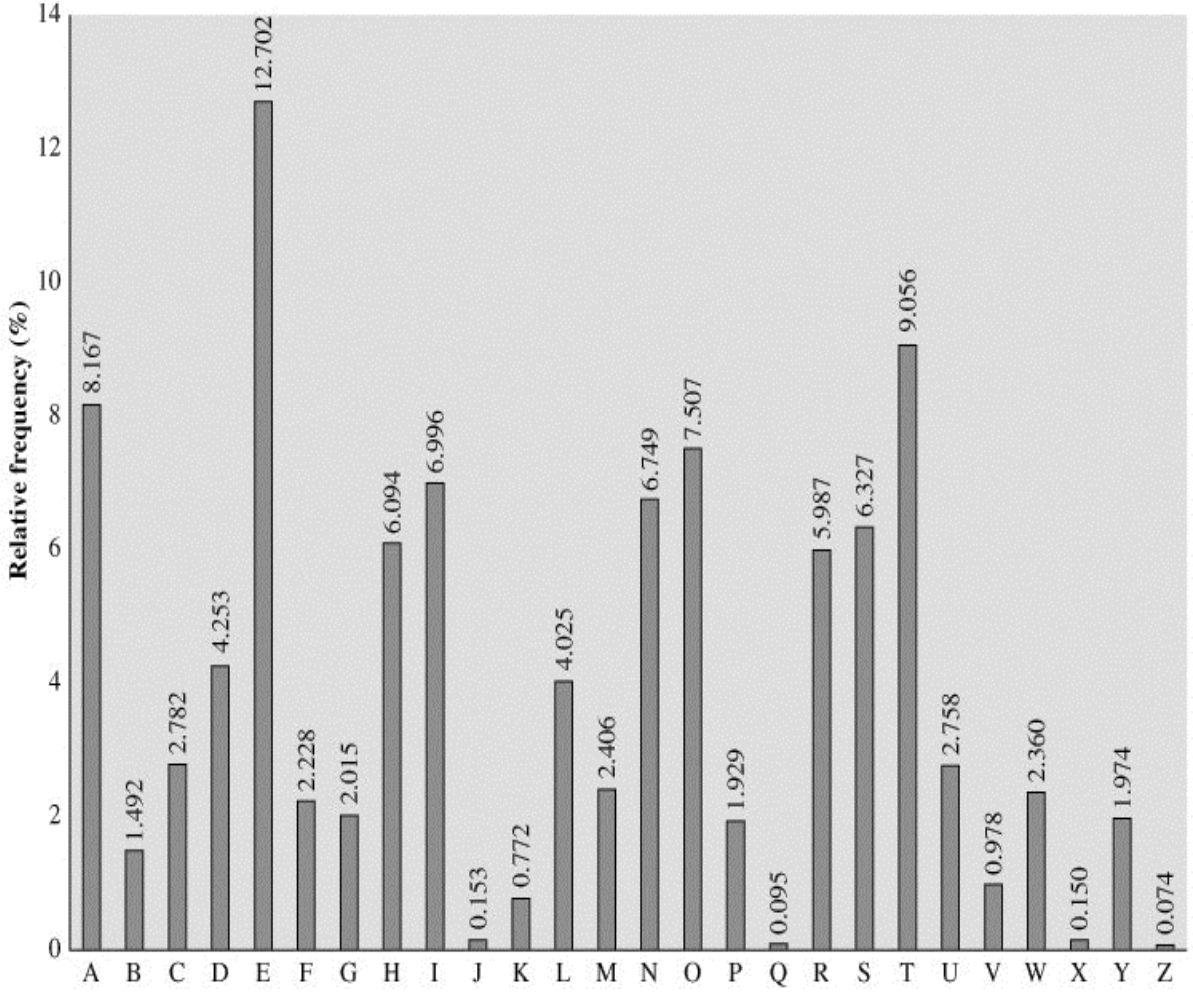
Characters	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
KEY	D	K	V	Q	F	I	B	J	W	P	E	S	C	X	H	T	M	Y	A	U	O	L	R	G	Z	N

Plain Text	I	F	W	E	W	I	S	H	T	O	R	E	P	L	A	C	E	L	E	T	T	E	R	S
Cipher Text	W	I	R	F	R	W	A	J	U	H	Y	F	T	S	D	V	F	S	F	U	U	F	Y	A

Cryptanalysis on Monoalphabetic Cipher

- Now, have a total of 26! Keys = 4×10^{26} keys.
- With so many keys, might think it as secure?
- Problem with language characteristics

Cryptanalysis on Monoalphabetic Cipher (Contd...)




```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

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

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

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

By the end of this session…

- Describe the working of Vigenère Cipher, Playfair cipher, Hill cipher.

Vigenère Cipher

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

- As in Caesar Cipher, every character is replaced with another character along same numerical shift key.
- Vigenère cipher consists of several Caesar ciphers in sequence with different shift values.
- To encrypt, a table of alphabets can be used, termed a tabula recta, Vigenère square, or Vigenère table.

Vigenère Cipher (Example)

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

- Plaintext: ATTACKATONCE
- Key: LEMON

Encryption:

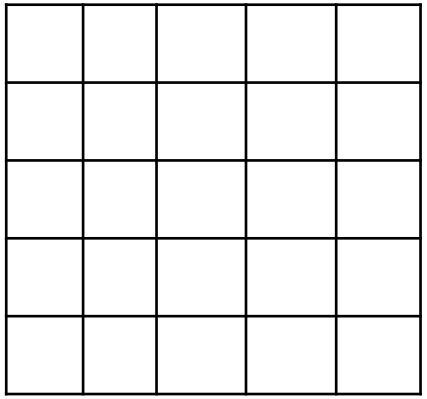
- Plaintext: ATTACKATONCE
- Key: LEMONLEMONLE
- Ciphertext: LXFOPVEFCANI

Exercise: Encrypt "CRYPTOGRAPHY" with key "VVIT"

Vigenère Cipher (Cryptanalysis)

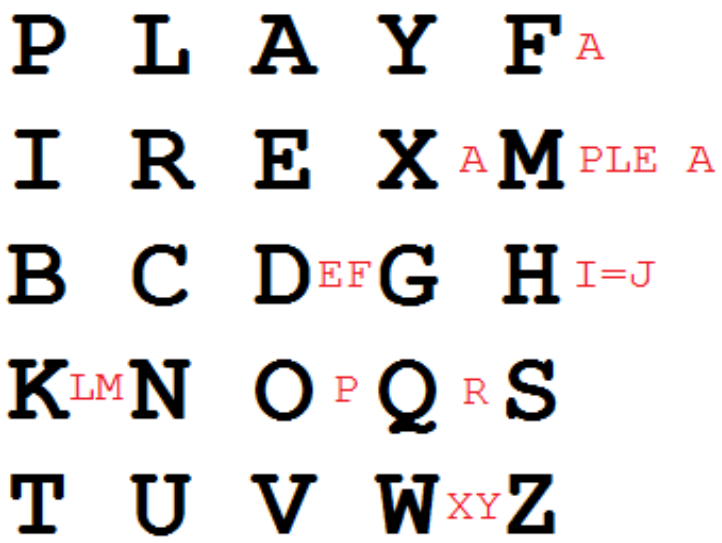
- The idea behind the Vigenère cipher is to disguise plaintext letter frequencies.
- However, using the Vigenère cipher, Plaintext letter can be enciphered as different ciphertext letters at different points in the message, thus defeating simple frequency analysis.
- The primary weakness of the Vigenère cipher is the repeating nature of its key. If a cryptanalyst correctly guesses the key's length, then the ciphertext can be treated as interwoven Caesar ciphers, which individually are easily broken.

Playfair Cipher



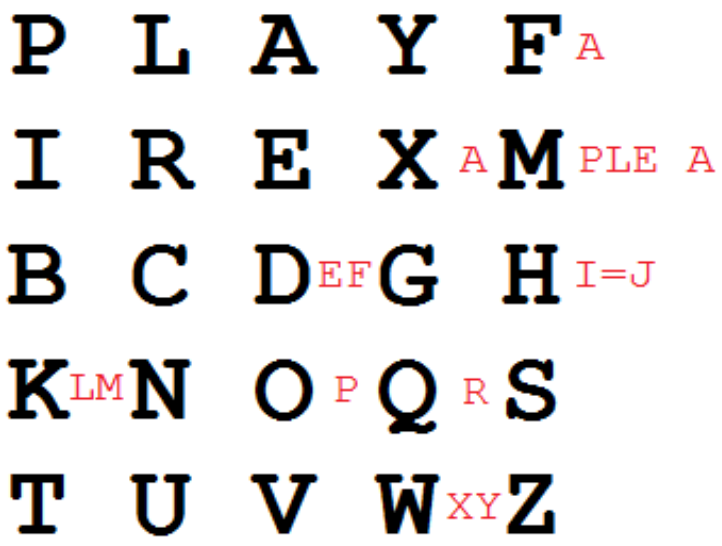
- is a manual symmetric encryption technique and was the first literal digraph substitution cipher.
- invented in 1854 by Charles Wheatstone.
- encrypts pairs of letters (digraphs).
- Playfair is thus significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it

Playfair Cipher Example



- The Playfair cipher uses a 5 × 5 table containing a key word or phrase.
- Key: "playfair example".
- Plaintext: HIDE THE GOLD IN THE TREE STUMP

Playfair Cipher Example (Contd...)



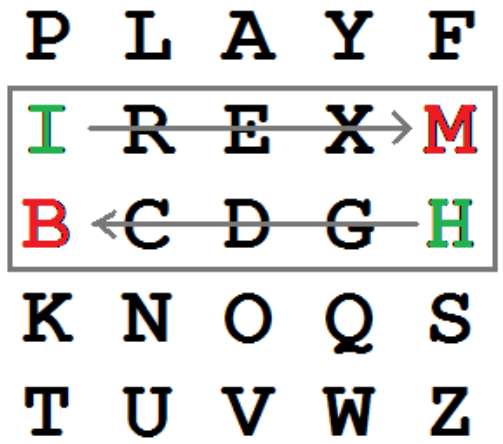
- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- If the letters appear on the same row of your table, replace them with the letters to their immediate right respectively
- If the letters appear on the same column of your table, replace them with the letters immediately below respectively
- If the letters are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair

Playfair Cipher Example (Contd...)



HI

Shape: Rectangle
Rule: Pick Same Rows, Opposite Corners

BM

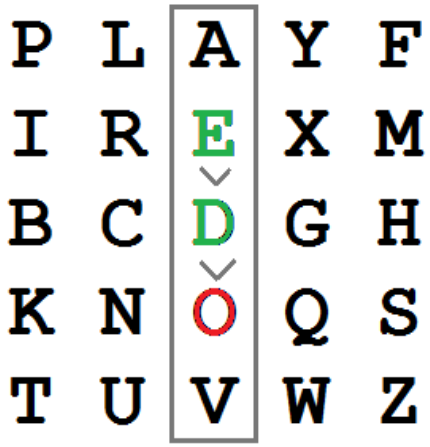
- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- Ciphertext: BM

Playfair Cipher Example (Contd...)



DE

Shape: Column
Rule: Pick Items Below Each Letter, Wrap to Top if Needed

OD

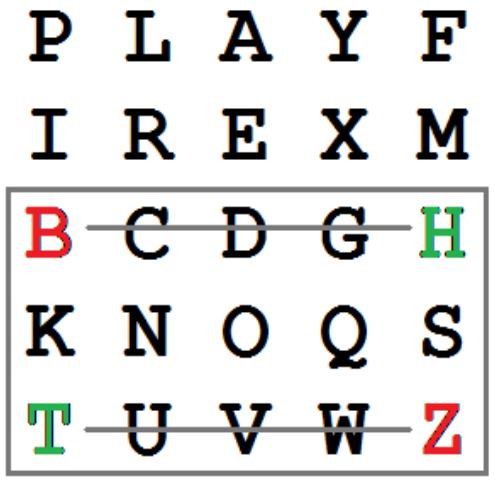
- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- Ciphertext: OD

Playfair Cipher Example (Contd...)



TH

Shape: Rectangle
Rule: Pick Same Rows, Opposite Corners

ZB

- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- Ciphertext: ZB

Playfair Cipher Example (Contd...)

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

EG

Shape: Rectangle
Rule: Pick Same Rows, Opposite Corners

XD

- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- Ciphertext: XD

Playfair Cipher Example (Contd...)

OL

Shape: Rectangle
Rule: Pick Same Rows, Opposite Corners

NA

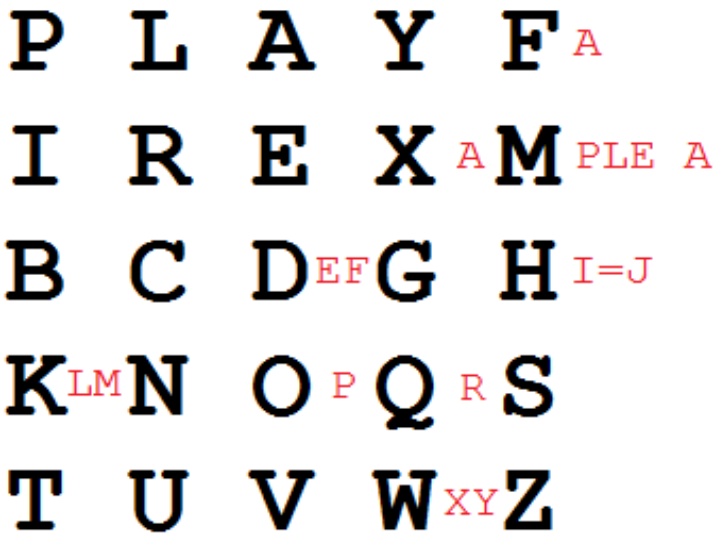
- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

- Ciphertext: NA

Playfair Cipher Example (Contd...)



Exercise: Encrypt "Students of VVIT" with key "Hello World"

- Create pair of letters in the plaintext

Plaintext: HIDE THE GOLD IN THE TREE STUMP

HI DE TH EG OL DI NT HE TR EX ES TU MP

DI	NT	HE	TR	EX	ES	TU	MP
BE	KU	DM	UI	XM	MO	UV	IF

- BM OD ZB XD NA BE KU DM UI XM MO UV IF
- For Decryption, use opposite rules and follow the same process.

Hill Cipher

$$C = (P \times K) \, mod \, 26$$

$$P = (C \times K^{-1}) \, mod \, 26$$

- Polygraphic substitution cipher based on linear algebra
- Invented by Lester S. Hill in 1929.
- To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, again modulus 26
- To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption

Hill Cipher Example (Contd…)

- Plaintext Message: ACT = $\begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix}$
- Key: GYBNQKURP = $\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$
- Encryption:

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

- Which gives "POH"

Hill Cipher Example

- Key: IFKVIVVMI = $\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}^{-1} \equiv \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} mod\ 26$
- Decryption:

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

- Which gives us 'ACT'

Hill Cipher Conditions

1. The Key matrix will have an inverse if and only if it's determinant is not Zero.
2. The determinant of the key matrix must not have any common factors with the modular base i.e., determinant must not be divisible by 2, 13 and 26.

$$a \times b \equiv 1 \, mod \, n$$

By the end of this session...

- Describe the working of Vernam Cipher.
- Describe the working of Railfence Cipher, Route Cipher, and Column Transposition Cipher

Vernam Cipher

$$C = P \oplus K$$

$$P = C \oplus K$$

Encrypt "VVIT" with key "CSE"

- It is a stream cipher.
- Also called as One-time pad cipher.
- Plaintext is exclusive-or with the random key of same length to form the ciphertext.
- Every key must be used only once and a new key must be generated for every message.

Example:

Plaintext: 'IF' = 1001001 1000110

Key = 1010110 0110001

Ciphertext = 0011111 1110110

Rail Fence Cipher

Plaintext: “defend the east wall
Key: 3

d				n				e				t				l
	e		e		d		h		e		s		w		l	
		f				t				a				a		

Ciphertext: dnetleedheswlftaa

Encrypt “we are in cryptography class” with key 4

- It is a transposition cipher that follows a simple rule for mixing up the characters in the plaintext to form the ciphertext.
- Rail-fence cipher is a simpler "write down the columns, read along the rows" cipher
- The railfence cipher offers essentially no communication security, and it will be shown that it can be easily broken even by hand.

Route Cipher

Encrypt “we are in cryptography class” with key “spiral outwards, anticlockwise”

- The plaintext is first written out in a grid of given dimensions, then read off in a pattern given in the key.
- For example: WE ARE DISCOVERED. FLEE AT ONCE

Ciphertext: EJXCTEDECDAEWRIORFEONALEVSE

- The key: "spiral inwards, clockwise, starting from the top right".

Column Transposition Cipher

- The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen based on key.
- Both the width of the rows and the permutation of the columns are usually defined by a keyword.
- For example, the word ZEBRAS is of length 6 (so the rows are of length 6), and the permutation is defined by the alphabetical order of the letters in the keyword.
- In this case, the order would be "6 3 2 4 1 5".

Column Transposition Cipher - Example

- Plaintext: WE ARE DISCOVERED. FLEE AT ONCE
- Key: ZEBRAS - 6 3 2 4 1 5
- Encryption:

6	3	2	4	1	5
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	Q	K	J	E	U

Cipher text: EVLNE ACDTK ESEAQ ROFOJ DEECU WIREE

Encrypt "we are in cryptography class" with key "THREAT"

Worked Exercise 1 – Caesar Cipher

- Exercise: Find the ciphertext "WELCOME TO VVIT" using key 5

W	E	L	C	O	M	E		T	O		V	V	I	T
5	5	5	5	5	5	5		5	5		5	5	5	5
b	j	q	h	t	r	j		y	t		a	a	q	y

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Worked Exercise 2 – Caesar Cipher

- Exercise: Break GHSW RI FVH.

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

	G	H	S	W		R	I		F	V	H
1	f	g	r	v		q	h		e	u	g
2	e	f	q	u		p	g		d	t	f
3	d	e	p	t		o	f		c	s	e

Worked Exercise 3 – Vigenere Cipher

- Exercise: Encrypt "CRYPTOGRAPHY" with key "VVIT"
- Plaintext: CRYPTOGRAPHY – 12
- Key: VVIT – 4
- Expanded Key: VVITVVITVVIT - 12

C	R	Y	P	T	O	G	R	A	P	H	Y
V	V	I	T	V	V	I	T	V	V	I	T
X	M	G	I	O	J	O	K	V	K	P	R

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Worked Exercise 4 – Playfair Cipher

- Exercise: Exercise: Encrypt "Students of VVIT" with key "Hello World"
- Plaintext: Students of VVIT – studentsofvvit – st ud en ts of vx vi tz
- Key: Hello World

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN							

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR						

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP					

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP	NT				

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

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

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP	NT	HK			

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP	NT	HK	XY		

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP	NT	HK	XY	XG	

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

Worked Exercise 4 – Playfair Cipher (Contd…)

- Plaintext: Students of VVIT – studentsofvvit –
- Key: Hello World

st	ud	en	ts	of	vx	vi	tz
TN	VR	HP	NT	HK	XY	XG	ZW

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

Worked Exercise 5 – Vernam Cipher

- Exercise: Encrypt "VVIT" with key "CSE"
- Plaintext: VVIT – 4
- Key: CSE – 3
- Expanded Key: CSEC - 4

V	0	1	0	1	0	1	1	0
C	0	1	0	0	0	0	1	1
	0	0	0	1	0	1	0	1

V	0	1	0	1	0	1	1	0
S	0	1	0	1	0	0	1	1
	0	0	0	0	0	1	0	1

I	0	1	0	0	1	0	0	1
E	0	1	0	0	0	1	0	1
	0	0	0	0	1	1	0	0

T	0	1	0	1	0	1	0	0
C	0	1	0	0	0	0	1	1
	0	0	0	1	0	1	1	1

Worked Exercise 6 – Railfence Cipher

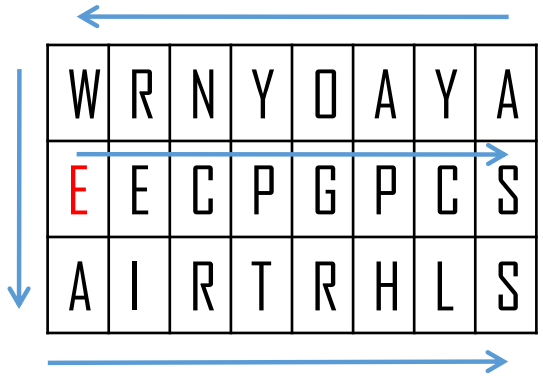
- Exercise: Encrypt "we are in cryptography class" with key 4
- Plaintext: we are in cryptography class
- Key: 4

W						N						O						Y					
	E				I		C				T		G				H		C				S
		A		E				R		P				R		P				L		S	
			R						Y						A						A		

- Ciphertext: WNOYEICTGHCSAERPRPLSRYAA

Worked Exercise 7 – Route Cipher

- Exercise: Encrypt "we are in cryptography class" with key "spiral outwards, anticlockwise"
- Plaintext: we are in cryptography class
- Key: spiral outwards, anticlockwise



Ciphertext: EECPGPCSAYAOYNRWAIRTRHLS

Worked Exercise 8 – Column Transposition Cipher

- Exercise: Encrypt "we are in cryptography class" with key "JUICE"
- Plaintext: we are in cryptography class
- Key: JUICE – 4 5 3 1 2
- Encryption:

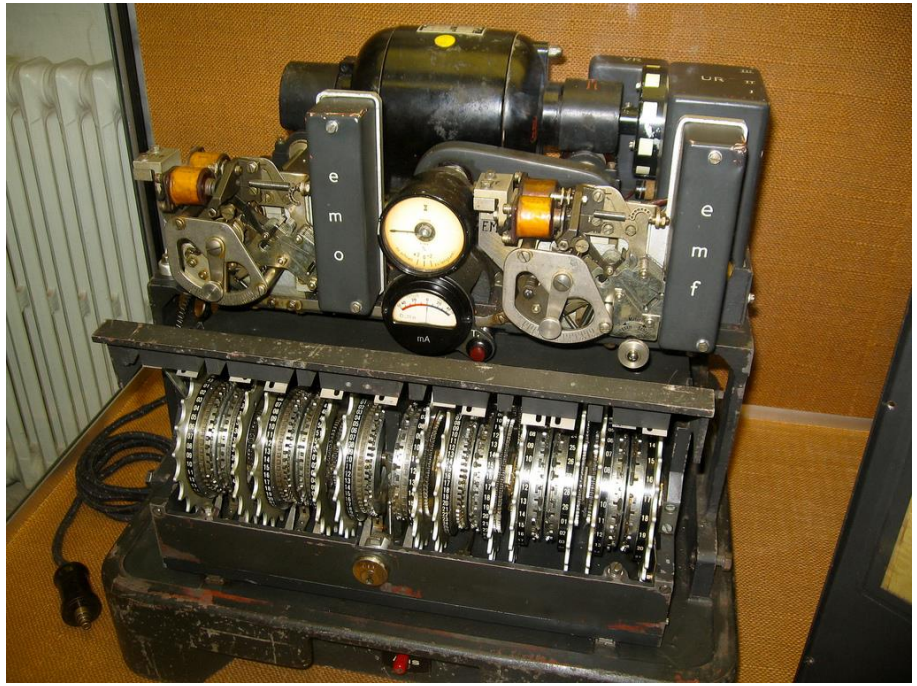
4	5	3	1	2
W	E	A	R	E
I	N	C	R	Y
P	T	O	G	R
A	P	H	Y	C
L	A	S	S	Z

Cipher text: RRGYS EYRCZ ACOHS WIPAL ENTPA

By the end of this session…

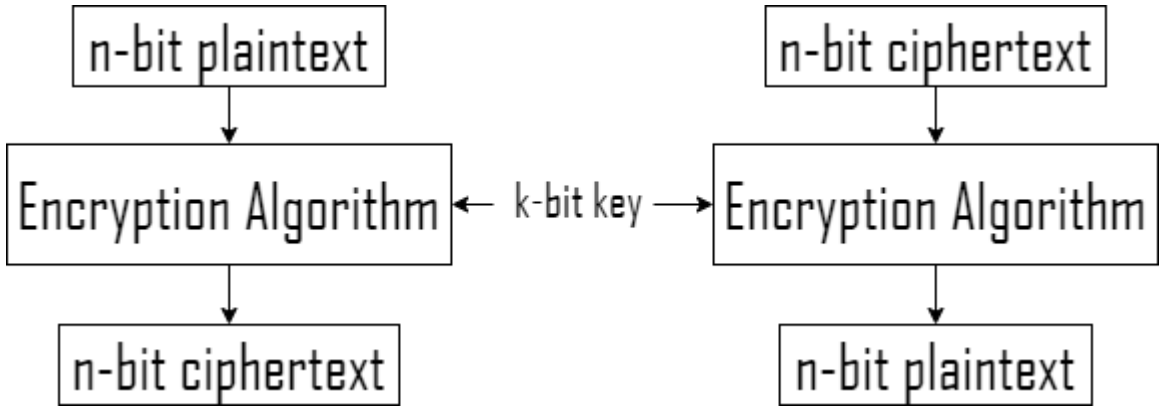
- Explain the concept of modern block ciphers and their characteristics.

Need for Modern Ciphers



1. **World War I**
   (28/07/1914 to 11/11/1918)
   Playfair cipher, Vigenere cipher
2. **World War II**
   (1939 to 1945)
   Rotor Machines, Hill Cipher
3. **ARPANET**
   1966
   First Military Network
4. **Silicon Computer chips**
   1970s
5. **DES**
   1975
   First Symmetric Block Cipher

Modern Block Ciphers



- In Symmetric cryptography, Modern Block Ciphers encrypts n-bit block of plaintext and decrypts n-bit block of ciphertext.
- Encryption and Decryption algorithm uses k-bit key.
- The decryption algorithm must be inverse of encryption algorithm and both the operations must use same secret key.
- If the message block is less than n-bits we need to add padding bits to convert it to n-bit block.

Substitution vs Transposition

- Functionality is same as we see in traditional ciphers but operates on bits instead of characters.
- If the cipher is designed as substitution cipher, 1 bit or 0 bit in plaintext is substituted with 0 bit or 1 bit to form ciphertext. i.e., The plaintext and ciphertext may have different number of 1 bits and 0 bits.
- If the cipher is designed as transposition cipher, only the position of 1 bits and 0 bits are changed. i.e., The number of 1 bits and 0 bits in the plaintext and ciphertext remains same.
- In both cases the number of n-bit possible plaintexts or ciphertexts is 2^n.
- Most of the Modern block ciphers are designed as substitution ciphers.

Full-Size key ciphers

- Full-Size key is long enough to choose every possible plaintext bit is operated with key bit to form ciphertext.
- Full-Size key transposition cipher can be modelled as n-bit permutation with a set of n! ciphertext values. So, we need a key of $\lceil \log_2 n! \rceil$ bits
- Example: 3-bit plaintext will have 3! permutations of ciphertext. To operate on 3-bit plaintext we need the key of size $\lceil \log_2 6 \rceil = 3$ bits.

Plaintext: 101

Permutations: 123, 132, 213, 231, 321, 312

Full-Size key ciphers (Contd…)

- Full-Size key Substitution cipher doesn't transpose bits, but substitutes bits.
- For a n-bit plaintext, there are 2^n possible ciphertexts. To operated on 2^n ciphertexts we need a key of size $[\log_2(2^n)!]$
- Example: For a 3-bit plaintext can be an integer between 0 to 7. So, it can have any of the value between 0 to 7 as substitution. So there are 8 possible ciphertexts possible. Key size must be 16 bits.

Plaintext: 010 = 2 can be substituted with 0, 1, 2, 3, 4, 5, 6, 7

Ciphertexts possible: 000, 001, 010, 011, 100, 101, 110, 111

Partial-key cipher

- Partial-key cipher is a composition operation and a sub-set of the corresponding full-size key cipher.

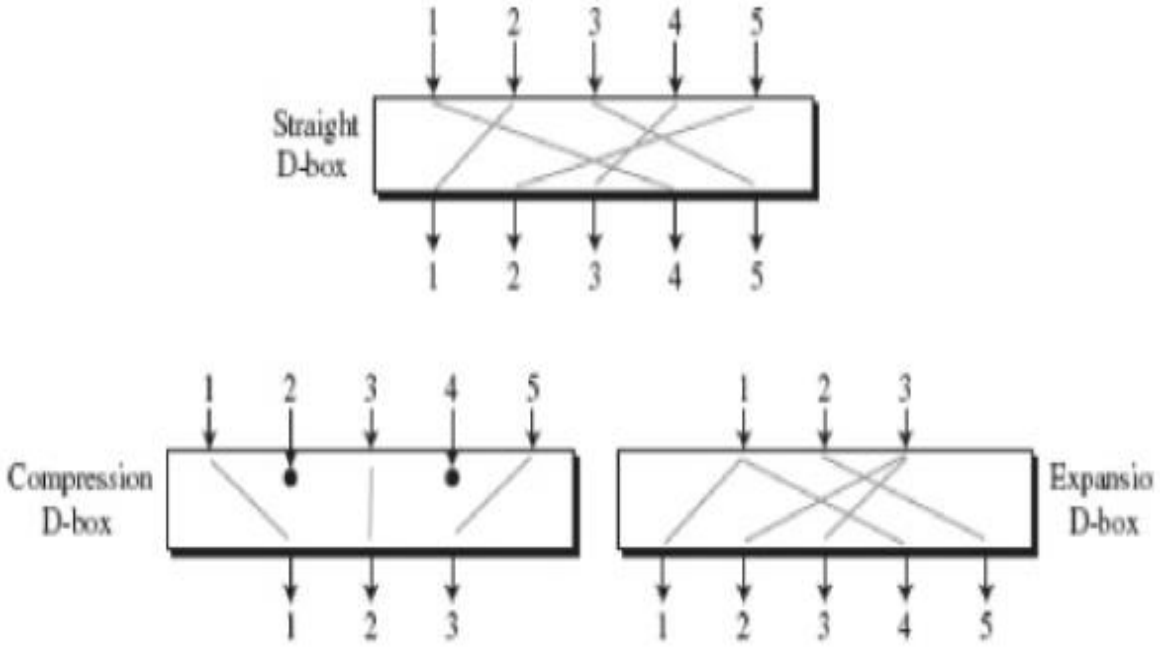
By the end of this session…

- Discuss the components of a modern block cipher.
- Distinguish between the two classes of product ciphers.

Components of a Modern Block Ciphers

- D-boxes
- S-boxes
- Product Ciphers

Diffusion Box (D-box)



- D-box transposes bits, normally keyless.
- Three types of D-boxes
  - Straight D-box (P-box)
  - Expansion D-box
  - Compression D-box
- A Straight D-box with n inputs and n-outputs is a permutation. There are n! possible mappings.
- A Compression D-box with n inputs and m-outputs, where m<n. Some of the inputs are blocked and do not reach the output.
- An Expansion D-box with n inputs and m outputs, where m>n. Some of the inputs are connected more than one output.
- Only Straight D-box is invertible.

Substitution Box (S-box)

Inputs	00	01	10	11
0	00	10	01	11
1	10	00	11	01

- An S-box is an m x n substitution unit, where m and n are not necessarily the same.
- S-box can be keyed or keyless.
- Relationship between input and output is defined by a table or mathematical relation.
- S-box may or may not be invertible, if it is invertible, the number of input bits and output bits must be same.

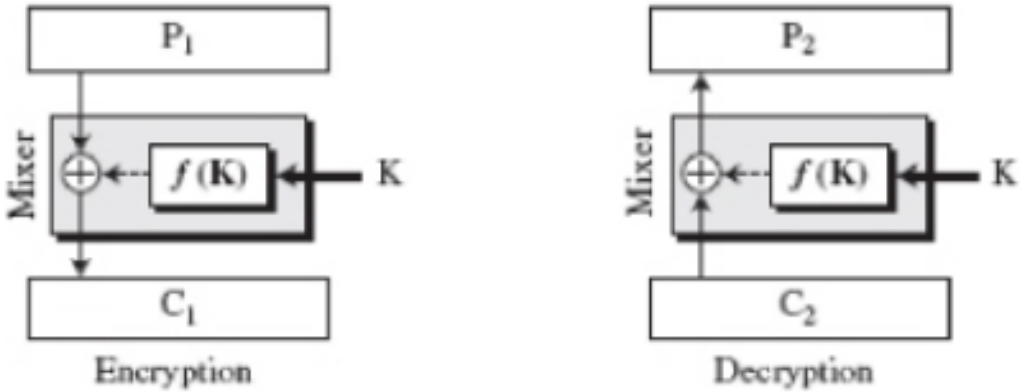
Product Ciphers

- A Product cipher is a combination of Substitutions, Transpositions, Exclusive-or, Circular shifts.
- Introduced by Shannon.
- Works on two important properties:
  - Diffusion
  - Confusion
- Diffusion hides the relationship between the ciphertext and the plaintext.
- Confusion hides the relationship between the ciphertext and the key.
- Diffusion and confusion can be achieved using iterated product ciphers where each iteration (round) is a combination of S-boxes, D-boxes, and other components.
- Modern Block Ciphers are all Product ciphers.

Two Classes of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
  - Feistel Ciphers
  - Non-Feistel Ciphers
- Feistel ciphers use both invertible and non-invertible components. Ex. DES
- Non-Feistel Ciphers use only invertible components. Ex. AES.

Feistel Ciphers (stage – 1)

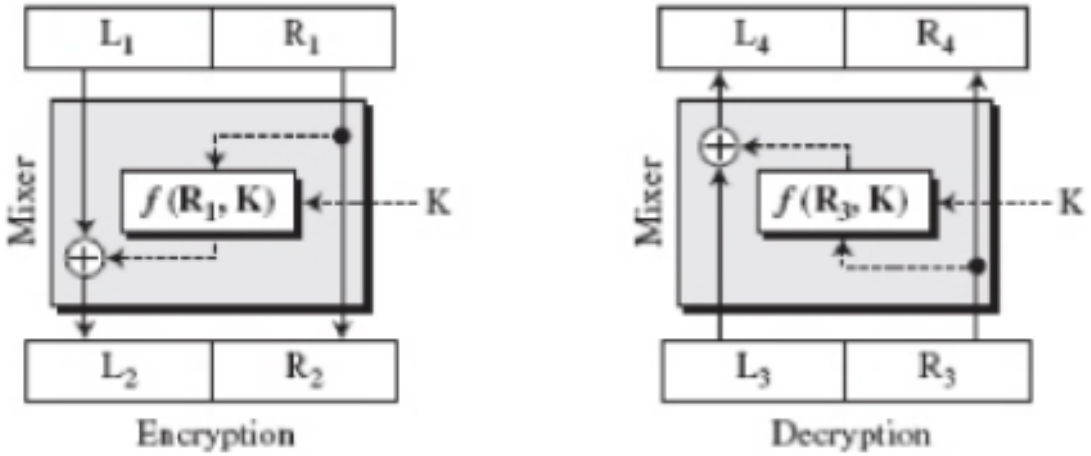


$$C = P \oplus f(k)$$

$$P = C \oplus f(k)$$

- Designed by a researcher named Feistel.
- A Feistel cipher can have three types of components:
  - Self-invertible
  - Invertible
  - Non-invertible.
- Proved and designed that the encryption algorithm and decryption algorithm can be inverses of each other even they have non-invertible units.
- Uses exclusive-or operation to make non-invertible function f(k) to make it invertible.
- The combination of exclusive-or operation and f(k) is called mixer and it is self invertible.

Feistel Cipher (Stage - 2)

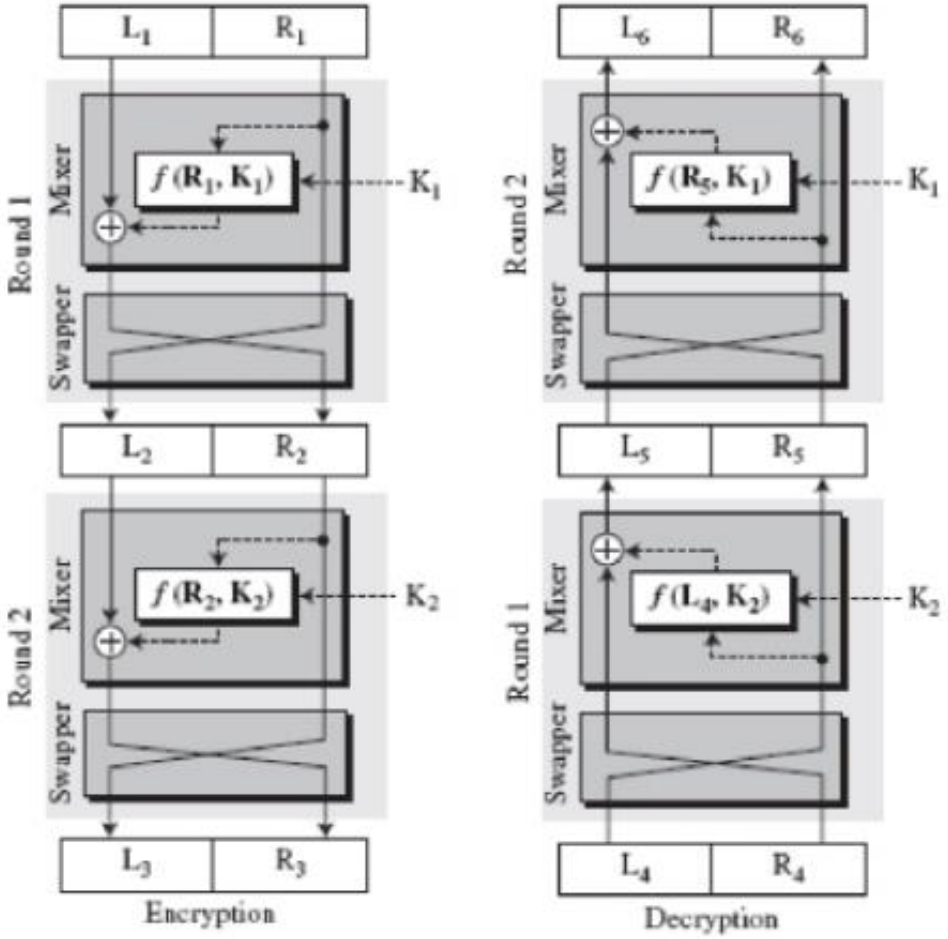


- Part of the plaintext is used as input to encryption algorithm and part of ciphertext is used as input to decryption algorithm along with the key.

$$R_4 = R_3 = R_2 = R_1$$

$$L_n = Ln\_1 \oplus f(R_n\_1, K)$$

Feistel Cipher (Stage - 3)



- The preceding stage has one flaw. The right half of the plaintext never changes.
- To overcome such flaw, a swapper is used to swap left half and right half bits after every round.
- A sub key is generated from the original key for each round to make the cipher more complex.

Non – Feistel Cipher

- Uses only invertible components.
- A component in the plaintext has the corresponding component in the ciphertext.
- For example, S-box must have equal number of input and output bits to make it compatible with non-Feistel cipher.
- No compression, No Expansion are allowed.
- There is no purpose of dividing the plaintext in to two halves.

By the end of this session…

- Summarize the history of Data Encryption Standard (DES)
- Illustrate the structure of DES and Explain its function.

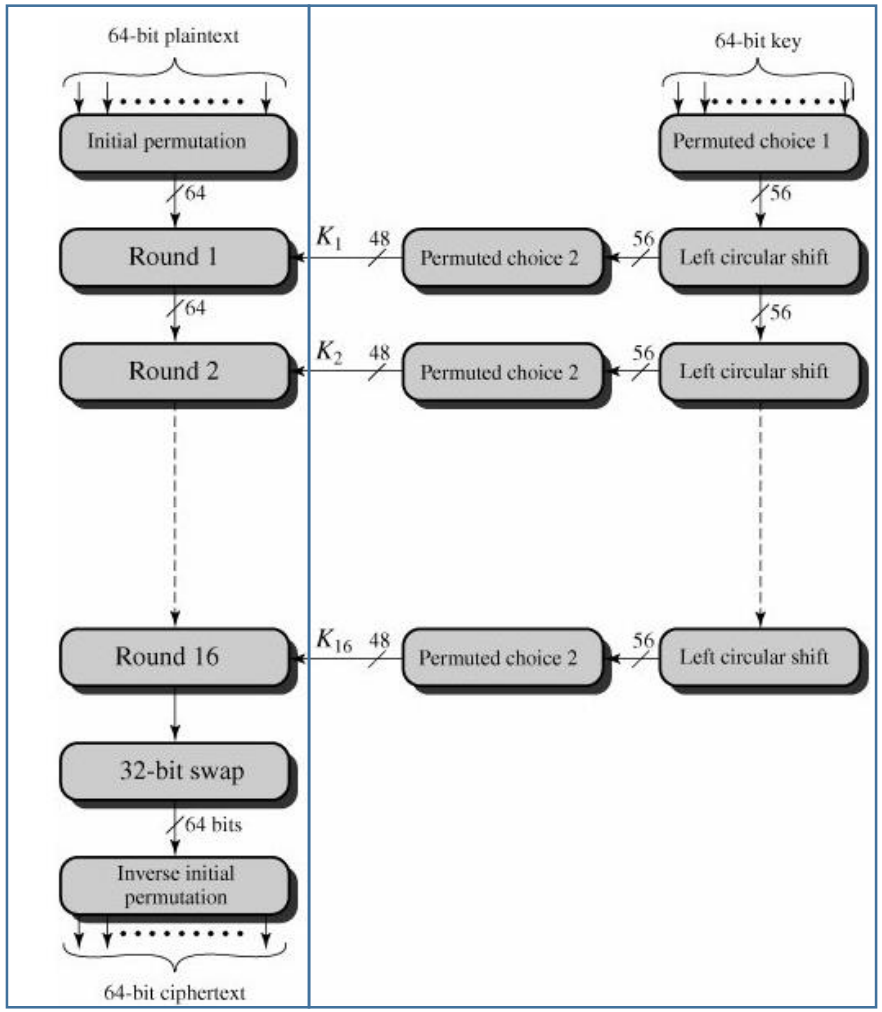
Data Encryption Standard (DES)

- Adopted by National Bureau of Standards in 1977.
- Now adopted by NIST as Federal Information Processing Standard 46 (FIPS PUB 46).
- First known Feistel cipher.
- Encryption and Decryption algorithms works on 64 bit blocks with 56 bit key.

History of DES

- Late 1960s, IBM started a research project led by Horst Feistel.
- Result – Development of an algorithm LUCIFER.
- LUCIFER operates on 64 bit blocks and 128 bits as key.
- LUCIFER sold to Lloyd's of London for use in cash – dispensing system.
- IBM started to develop a marketable commercial encryption products implemented on a single chip, headed by Walter Tuchman and Carl Meyer.
- Result – Refined version of LUCIFER with reduced key size of 56 bits
- In 1973, National Bureau of Standards (NBS) requested for proposals for a national cipher standard, where IBM submitted Tuchman-Meyer project and got adopted in 1977 as DES.
- In 1994, NIST confirmed DES for federal use.
- In 1999, NIST issued a new version of its standard FIPS PUB 46-3 with 3-DES

DES Encryption



- Two inputs to the encryption function
  - Plaintext – 64 bits
  - Key – 56 bits.
- Processing of plaintext in three phases.
  - Initial Permutation (IP)
  - 16 rounds of same function, involves permutation and substitution functions.
  - Inverse initial Permutation (IP^{-1})
- With the exception of initial and inverse initial permutation, DES has the exact structure of Feistel cipher.
- Processing of key
  - Permutation choice – 1
  - A Subkey (K_i) for each round: combination of left circular shift and permutation choice - 2.

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

(a) Initial Permutation (IP)

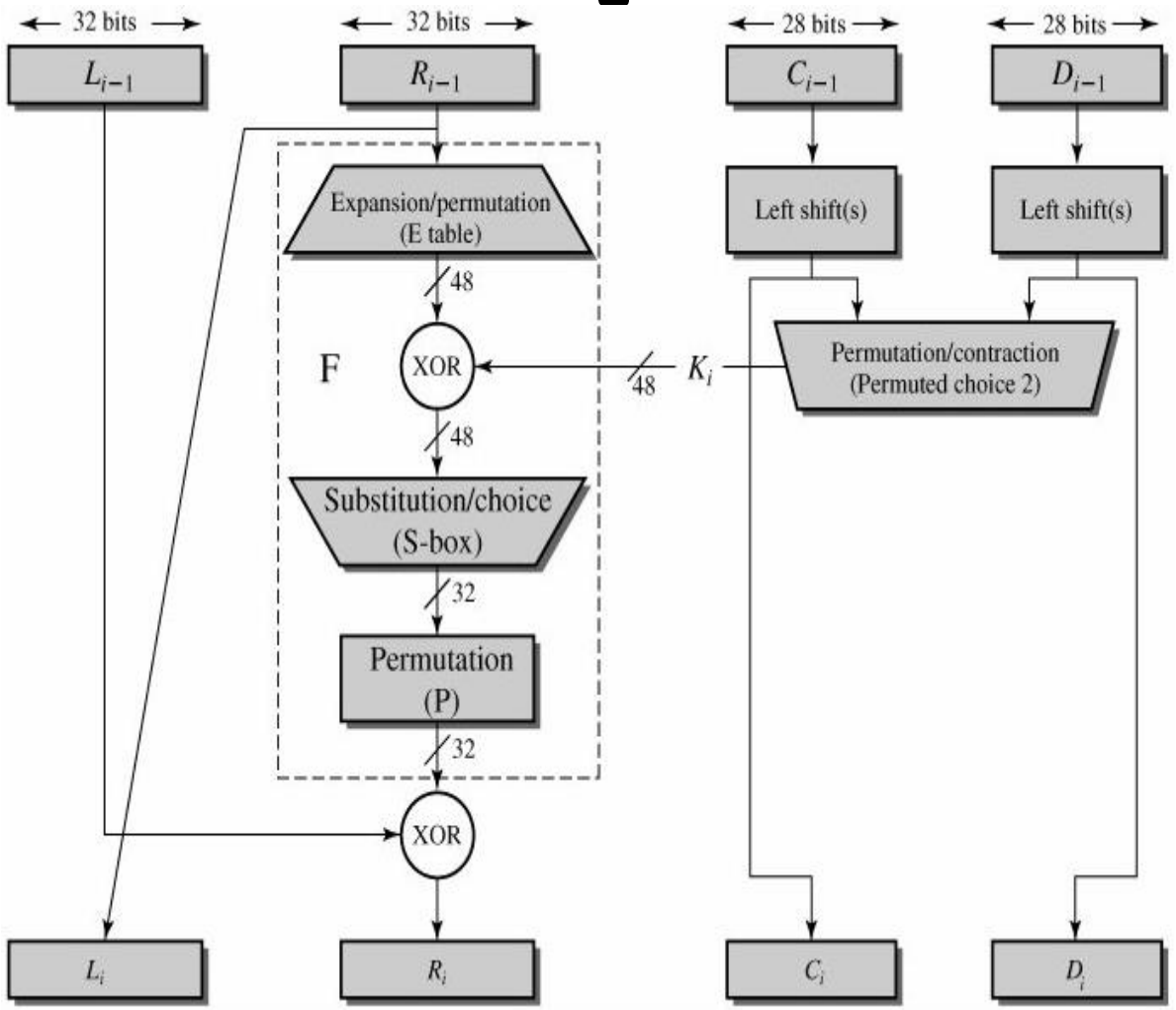
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP^1)

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

- Defined by the tables as shown.
- Input – 64 bits, output – 64 bits.
- Bits at the input are permutated as shown in the table.
- Both the tables are inverse of each other as for example, 1st bit in IP is 58th bit and 58th bit in IP^{-1} is 1st bit.
- Both the permutations are keyless and predetermined.
- Reason they are included in DES are not clear.

Details of Single Round



- Plaintext is divided into two halves Left half (L_{i-1}) and Right half (R_{i-1})
- The overall processing at each round is

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \times F(R_{i-1}, K_i)$$

Expansion/Permutation (E Table)

(c) Expansion Permutation (E)

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

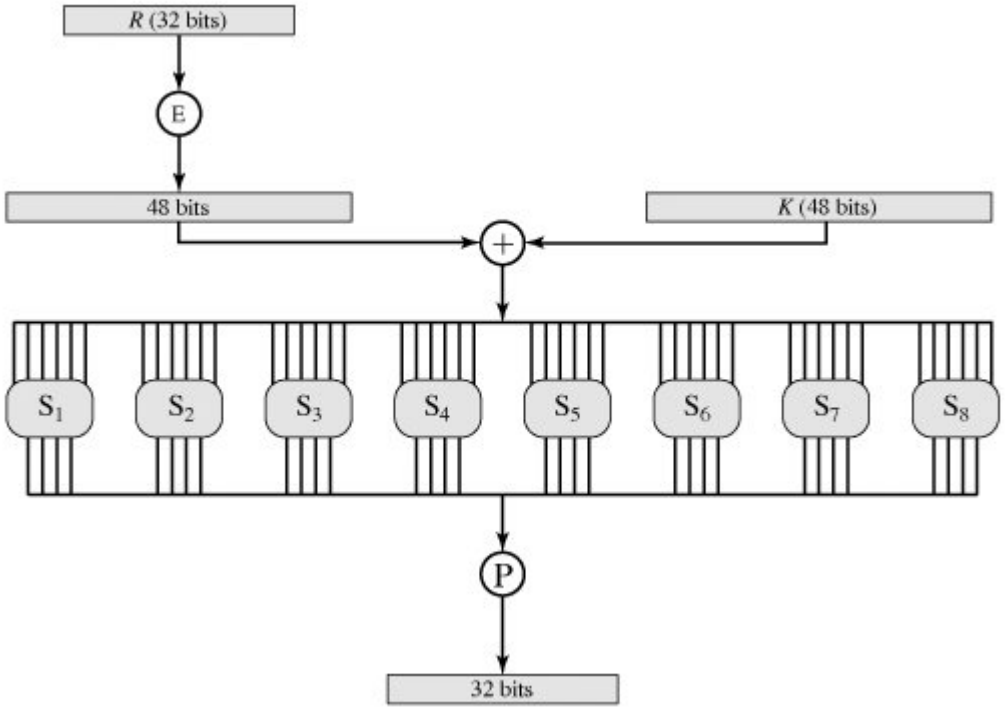
- First operation on Right half of 32 bits.
- The plaintext is expanded to make it 48 bit.
- Involves permutation and Expansion
- Involves duplication of 16 bits.

By the end of this session…

- Illustrate the structure of DES and Explain its function.
- Explain the security developed for DES.

S-boxes

- Consists of set of eight S-boxes
- Each accepts 6 bits as input and produces 4 bits as output.



S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-boxes (Contd...)

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Permutation (P)

(d) Permutation Function (P)

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

- Randomized organization of input 32 bits with the predetermined Permutation table (P).
- Need no key.

Key Generation

(a) Input Key

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

(b) Permuted Choice One (PC-1)

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

- 64-bit key is used input to the algorithm
- Every 8th bit is ignored using Permutation Choice – 1, to produce shaded key of 56 bits.
- The resultant key is divided into two halves labelled as C_0 and D_0.

Key Generation (Contd…)

- At each round, the two halves Ci-1 and Di-1 are subjected to a circular left shit of 1 or 2 bits.

(d) Schedule of Left Shifts

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

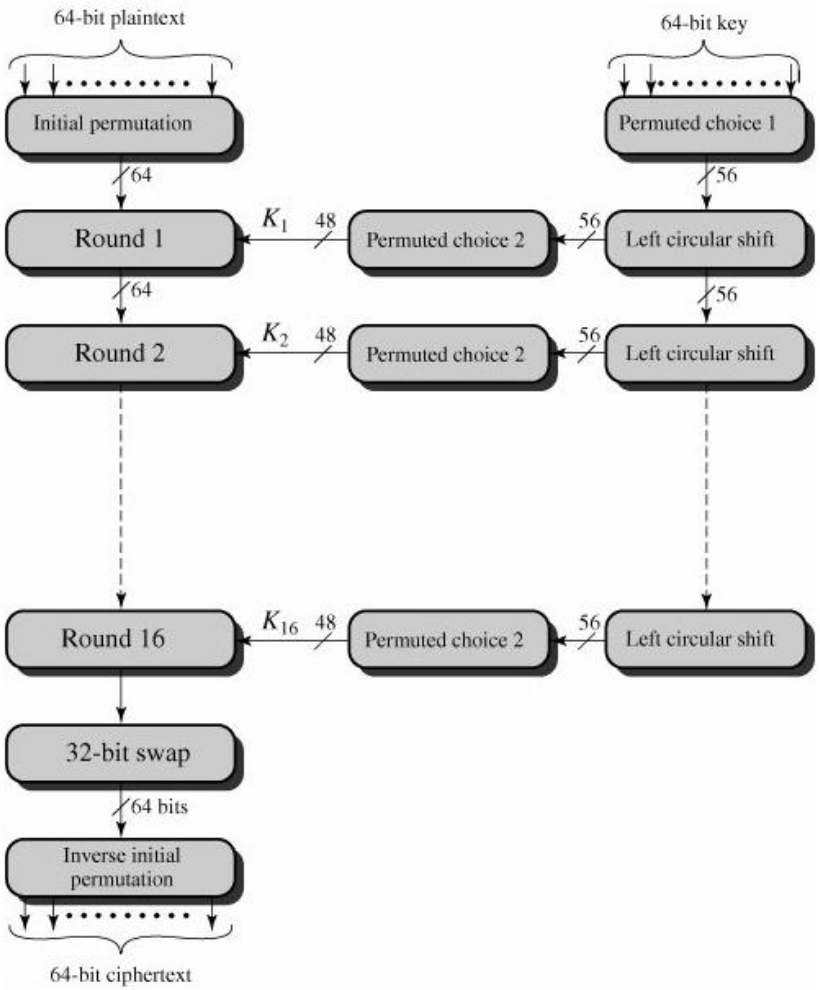
Key Generation (Contd…)

(c) Permuted Choice Two (PC-2)

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

- The output bits from the circular shifts are taken input to Permutated Choice – 2 to make 48 bit round key.

DES Decryption



- As it is a Feistel cipher, DES uses same algorithm for encryption and decryption.
- The only difference is the subkeys are applied in reverse order in decryption algorithm.

DES Demo

https://cryptographyacademy.com/des/

Critics on DES

- Before its adoption as a standard, DES subjected to intense criticism.
- Two areas drew the critics' fire
  - First, the key length was too short
  - The design criteria for internal structure of DES, the S-boxes, P-boxes etc.

Strength of DES – Avalanche Effect

(a) Change in Plaintext

Round	Number of bits that differ
0	1
1	6
2	21
3	35
4	39
5	34
6	32
7	31
8	29
9	42
10	44
11	32
12	30
13	30
14	26
15	29
16	34

- A change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.
- DES exhibits a strong avalanche effect.

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

with the key

0000001 1001011 0100100 1100010 0011100 0011000 0011100 0110010

Strength of DES – Avalanche Effect (Contd…)

(b) Change in Key

Round	Number of bits that differ
0	0
1	2
2	14
3	28
4	32
5	30
6	32
7	35
8	34
9	40
10	38
11	31
12	33
13	28
14	26
15	34
16	35

01101000 10000101 00101111 01111010 00010011 01110110 11101011 10100100

with two keys that differ in only one bit position:

1110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

0110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

Strength of DES – Use of 56 bit key.

- Key length of 56 bits, so 2^{56} possible keys, approximately 7.2×10^{16} keys.
- DES encryption per microsecond would take more than 1000 years to break the cipher, so, brute force attack is impractical.
- In 1977, Diffie and Hellman assumed that the technology existed to build a parallel machine with 1 million encryption devices, each can perform one encryption per microsecond. This makes average break down time to 10 hours. Estimated cost would be $20 million dollars.
- Finally in July 1998, Electronic Frontier Foundation (EFF) had broken DES encryption using special purpose "DES cracker", built for less than $250,000 and took less than 3 days.

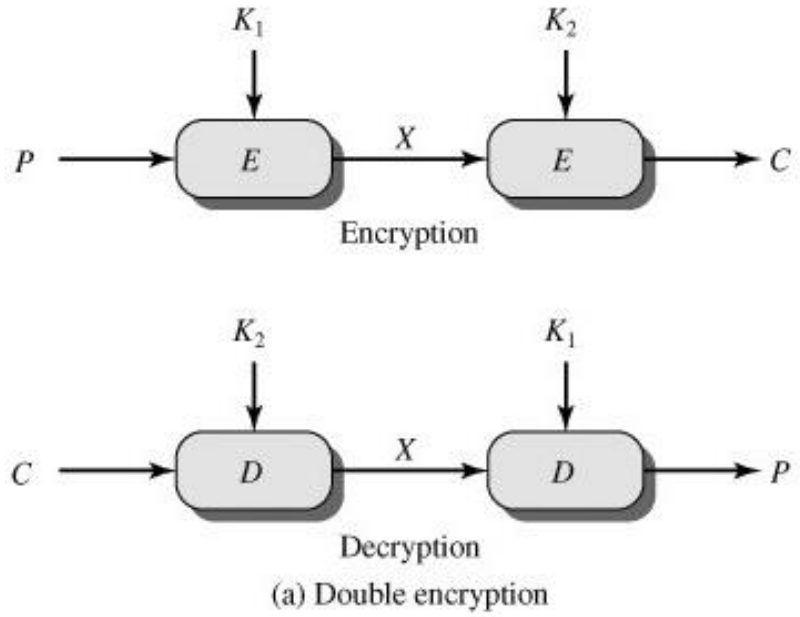
Strength of DES – Nature of the DES Algorithm

- A concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.
- The focus of concern is mainly on eight S-boxes, because, the design criteria for these boxes and indeed entire algorithm were not made public.
- There is a suspicion that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

By the end of this session…

- Explain the concept of multiple DES.
- Discuss in brief the history of Advanced Encryption Standard (AES).
- Describe structure of AES round.

Multiple DES – Double DES



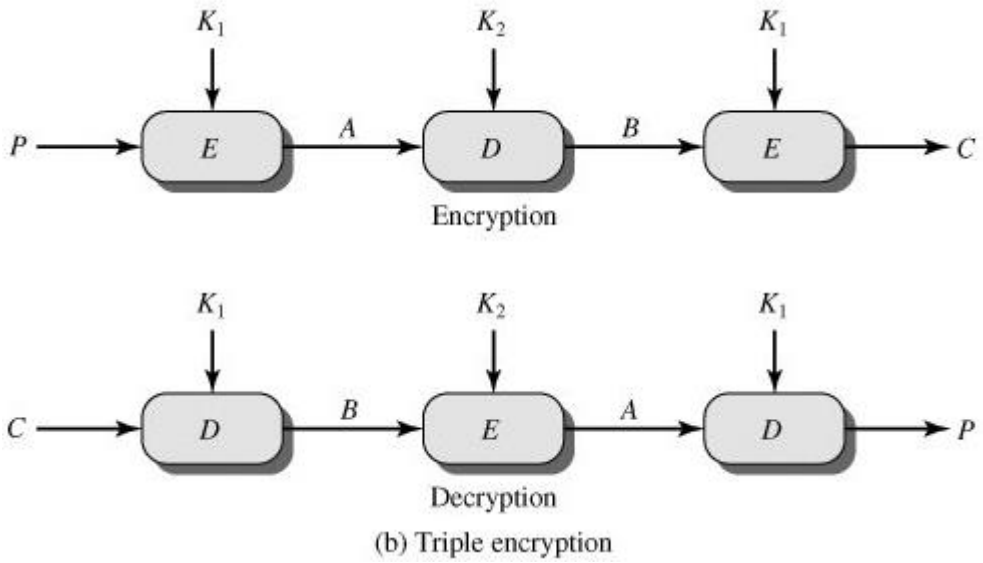
(a) Double encryption

- The simplest form of multiple encryption has two encryption stages and two keys.
- Given a plaintext p and two encryption keys k1 and k2 to generate ciphertext c as

$$C = E(K_2, E(K_1, P))$$
$$P = D(K_1, D(K_2, C))$$

- Dramatic increase in key, i.e., 56 x 2 = 112 bits.
- Possibility of Meet–in–the–Middle attack

Multiple DES – Triple DES



(b) Triple encryption

- Counter the Meet-in-the-middle attack.
- Three stages of encryption with two keys.

$$C = E(K_1, D(K_2, E(K_1, P)))$$
$$P = D(K_1, E(K_2, D(K_1, P)))$$

Advanced Encryption Standard - History

- In 1997, NIST started looking for an alternative for DES and would be called as Advanced Encryption Standard (AES).
- Criteria defined by NIST for selecting AES fall into three areas:
  - Security: minimum of 128 bit key
  - Cost: Computational Efficiency and Storage requirements
  - Implementation: Flexibility and Simplicity.
- August 1998, First AES Candidate Conference, NIST announced 15 out of 21 received algorithms had met the requirements and selected.
- August 1999, Second AES Candidate Conference, NIST announced 5 out of 15 candidates – MARS, RC6, Rijndael, Serpent and Twofish.

Advanced Encryption Standard – History (Contd…)

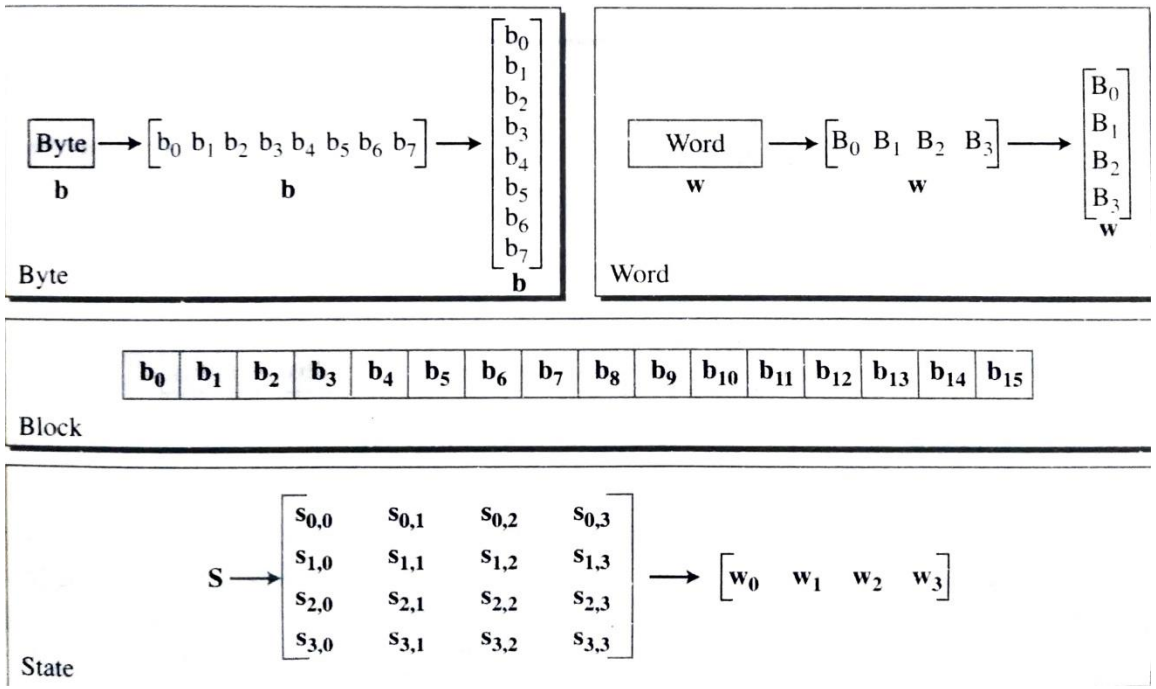
- October 2000, Third AES Candidate Conference, NIST announced Rijndael as Advanced Encryption Standard.
- Rijndael was designed by Belgian researchers Joan Daemen and Vincent Rijment.
- February 2001, NIST announced a draft of Federal Information Processing Standard (FIPS – 197) and made available for review and comments.
- December 2001, Added FIST – 197 in to Federal Register and published.

AES - Introduction

Nr	Key Size
10	128
12	192
14	256

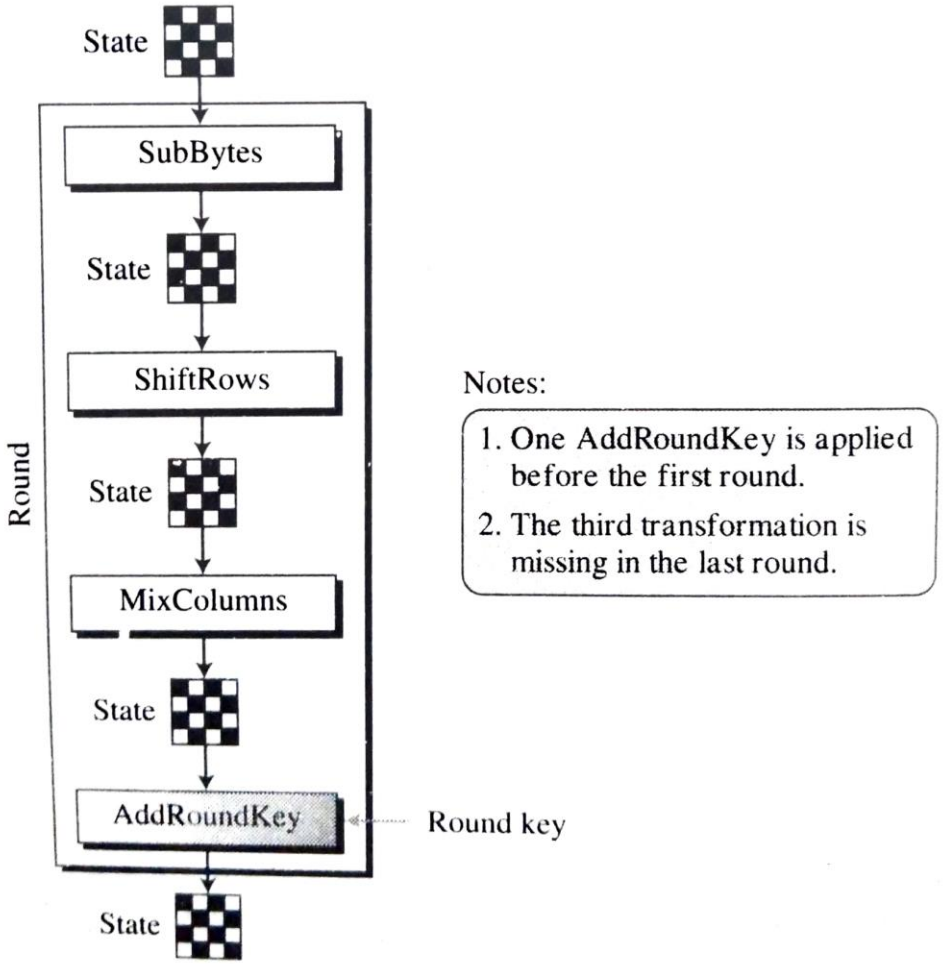
- AES is a non – Feistel cipher block cipher.
- Encrypts a plaintext block of size 128 bits.
- Uses three variants of key size – 128, 192, or 256 bits.
- Number of rounds depends on key size selected and vice versa.
- Encryption algorithm can also be used as Decryption algorithm, but the round keys are applied in the reverse order.
- The number of round keys = number of rounds + 1.

AES – Data Units



- bit (b): a bit is a binary digit with a value 0 or 1.
- Byte (B): A group of eight bits is treated as Byte. Generally represented as a row matrix of 8 bits or column matrix of 8 bits
- Word (w): A word is a group of 32 bits or 4 Bytes. Generally represented as a row matrix of 4 Bytes or column matrix of 4 Bytes.
- Block: A block in AES is a group of 128 bits or 16 Bytes or 4 words. Generally represented as a row matrix of 16 bytes.
- State: AES operates plaintext block in different rounds in different stages. At the beginning and end of every stage the data block is referred as State. Generally represented as 4 x 4 matrix.

AES - Structure of each round



- Each round, expect the last, uses four stages that are invertible.
  - Substitute Bytes (SubBytes)
  - Shift Rows (ShiftRows)
  - Mix Columns (MixColumns)
  - Add Round Key (AddRoundKey)
- Last Round doesn't have MixColumns stage.
- Each stage takes a state and creates another state to be used for the next stage or next round.
- At decryption, InvSubBytes, InvShiftRows, InvMixColums, and AddRoundKey is used.

By the end of this session...

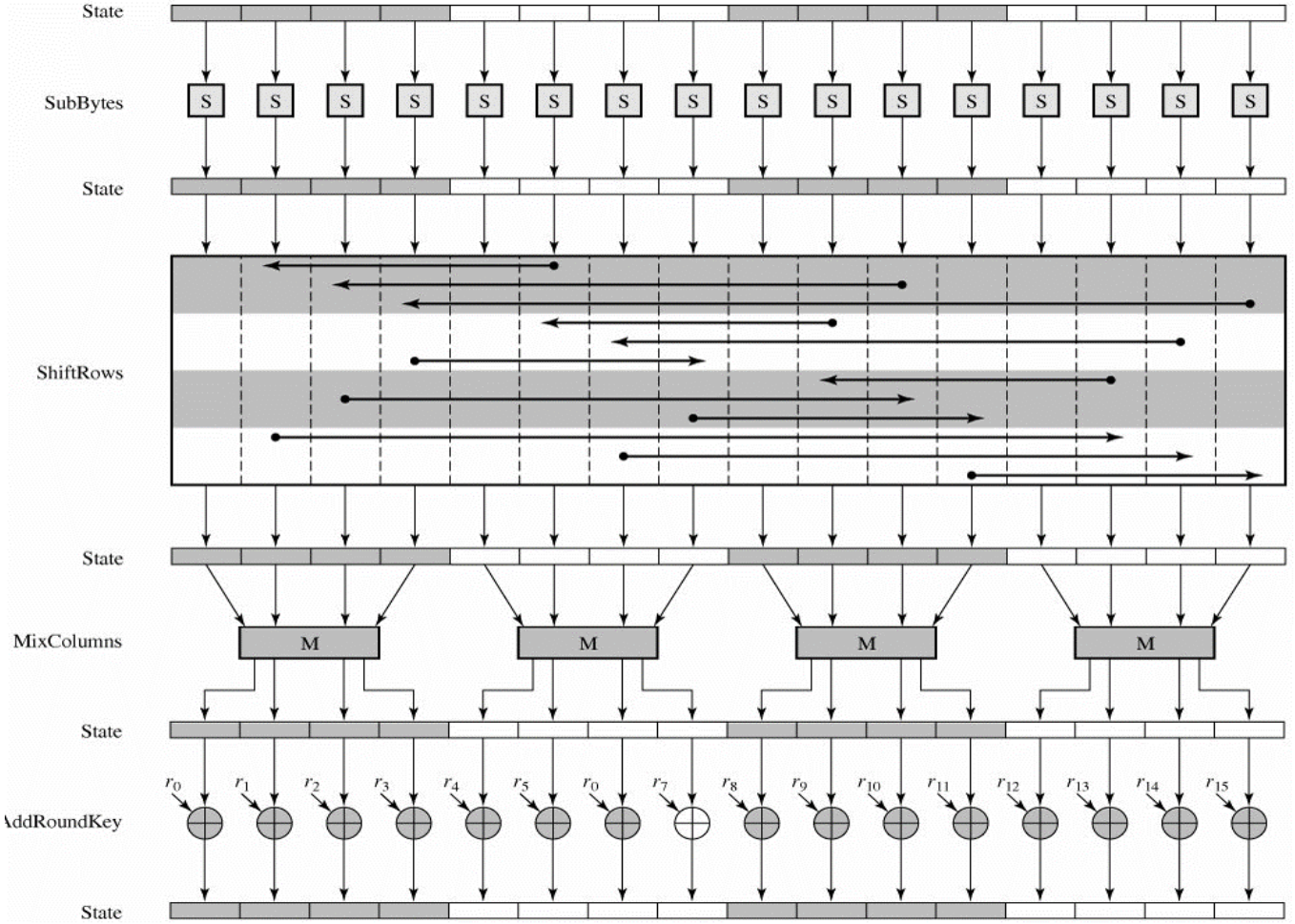
- Explain the stages in each round of AES.
- Illustrate the process of Key expansion in AES.

AES – Processing of Plaintext into state

A	E	S	U	S	E	S	A	M	A	T	R	I	X	Z	Z
0	4	18	20	18	4	18	0	12	0	19	17	8	23	25	25
00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$

AES – Stages in each round



- Four Stages
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey
- All the four operations are invertible.

Substitution Bytes - SubBytes

		y														
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
x	**0**	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

- First stage in each round.
- Purpose of this round is to involve substitutions in AES.
- Based on the predefined Substitution table
- Among the two hexadecimal digits, left digit defines the row and the right digit defines the column.
- Transformation is done one byte at a time

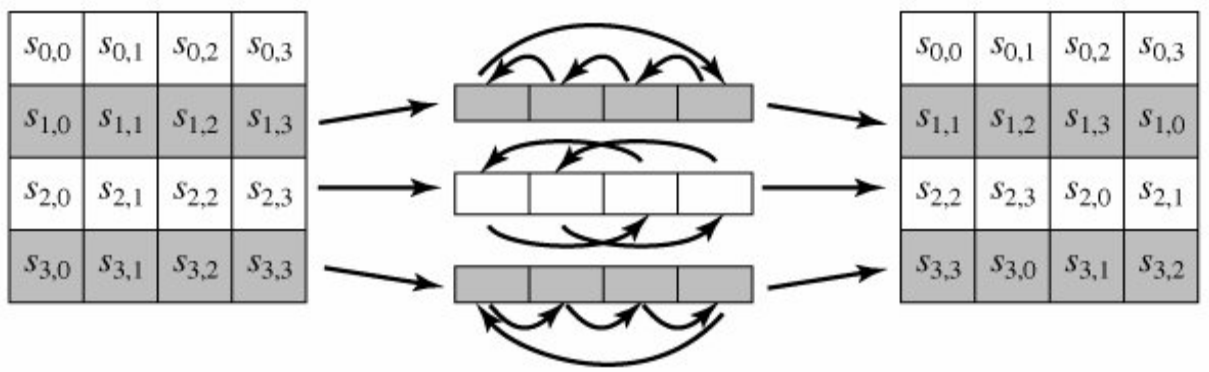
SubBytes - Example

		y														
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
x	**0**	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$

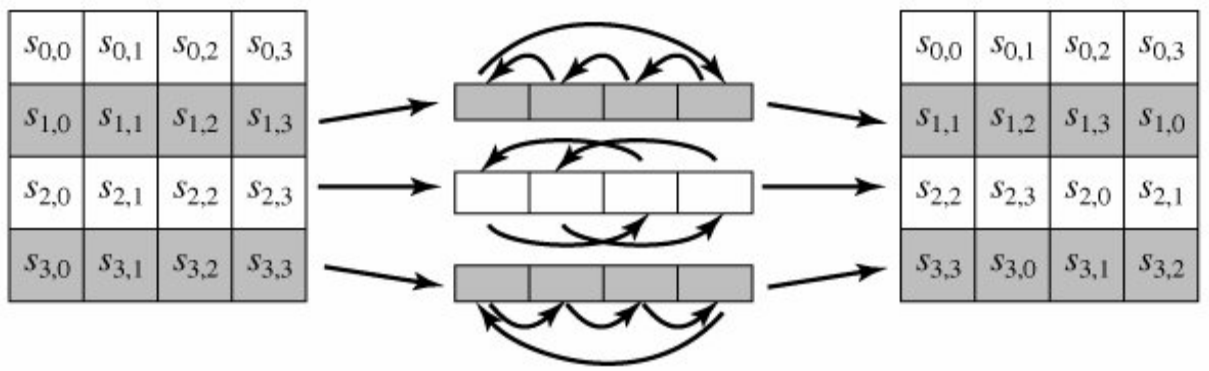
$$\begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & F2 & 63 & 26 \\ C9 & C9 & 7D & D4 \\ FA & 63 & 82 & D4 \end{bmatrix}$$

ShiftRows



- Second Stage in each AES round.
- Purpose of this round is make transpositions in AES.
- The ShiftRows stage follows the transpositions as:
  - 0th row – 0 bytes shifts (no shift)
  - 1st row – 1 byte left shift
  - 2nd row – 2 bytes left shift
  - 3rd row – 3 bytes left shift

ShiftRows - Example

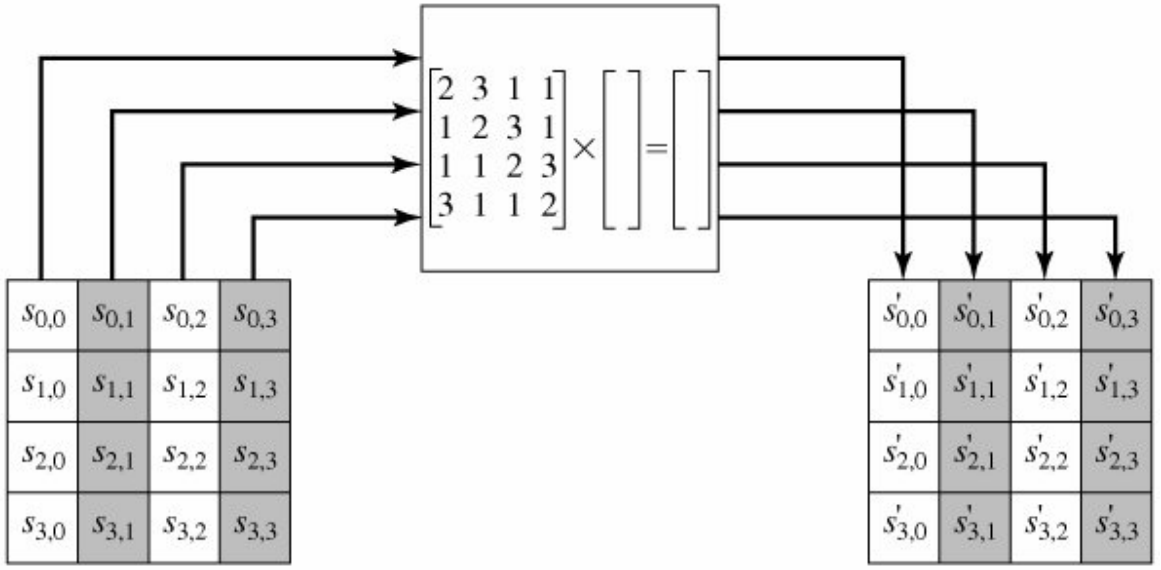


- 0th row – 0 bytes shifts (no shift)
- 1st row – 1 byte left shift
- 2nd row – 2 bytes left shift
- 3rd row – 3 bytes left shift

$$\begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & F2 & 63 & 26 \\ C9 & C9 & 7D & D4 \\ FA & 63 & 82 & D4 \end{bmatrix}$$

$$\begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & 63 & 26 & F2 \\ 7D & D4 & C9 & C9 \\ D4 & FA & 63 & 82 \end{bmatrix}$$

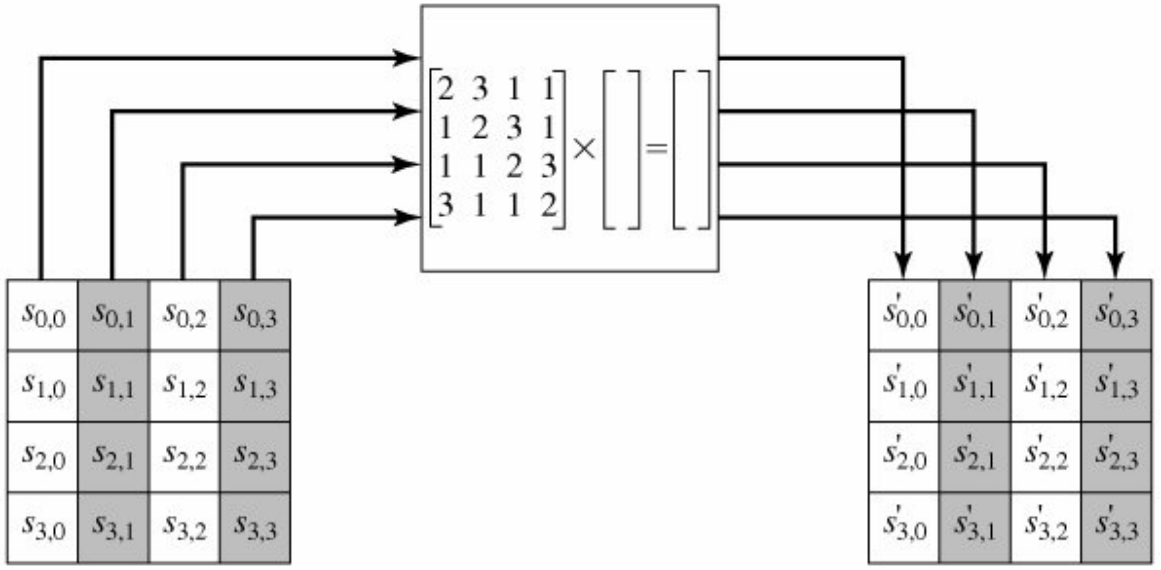
MixColumns



$$
\begin{aligned}
s'_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\
s'_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\
s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\
s'_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})
\end{aligned}
$$

- This stage operates at the column level, where it transforms each column of the state to a new column.
- Actually a matrix multiplication of a state column by a constant square matrix to form a new column.

MixColumns - Example



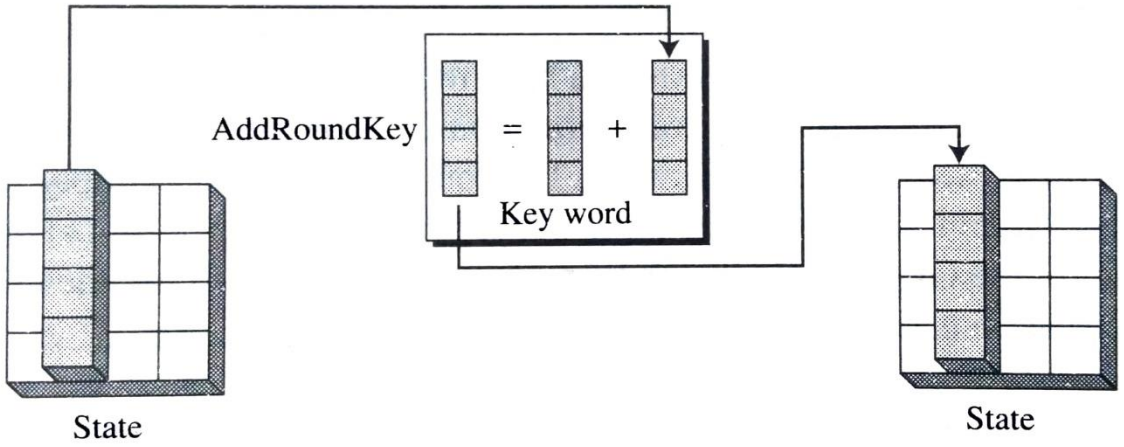
$$s'_{0,j} = (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j})$$
$$s'_{3,j} = (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})$$

$$\begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & 63 & 26 & F2 \\ 7D & D4 & C9 & C9 \\ D4 & FA & 63 & 82 \end{bmatrix}$$

$$\begin{bmatrix} 63 \\ F2 \\ 7D \\ D4 \end{bmatrix} \times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 62 \\ CF \\ 0C \\ 99 \end{bmatrix}$$

$$\begin{bmatrix} 62 & 02 & 27 & 26 \\ CF & 92 & 91 & 0D \\ 0C & 0C & F4 & D6 \\ 99 & 18 & 30 & 74 \end{bmatrix}$$

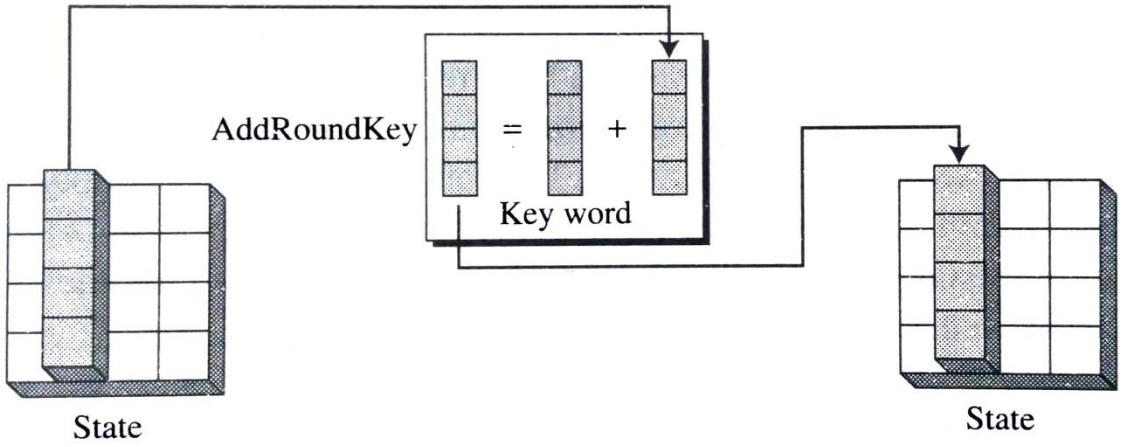
AddRoundKey



- Also processes one column at a time.
- Adds the round key word with each state column matrix.
- Bit-wise addition is nothing but bit-wise xor operation.

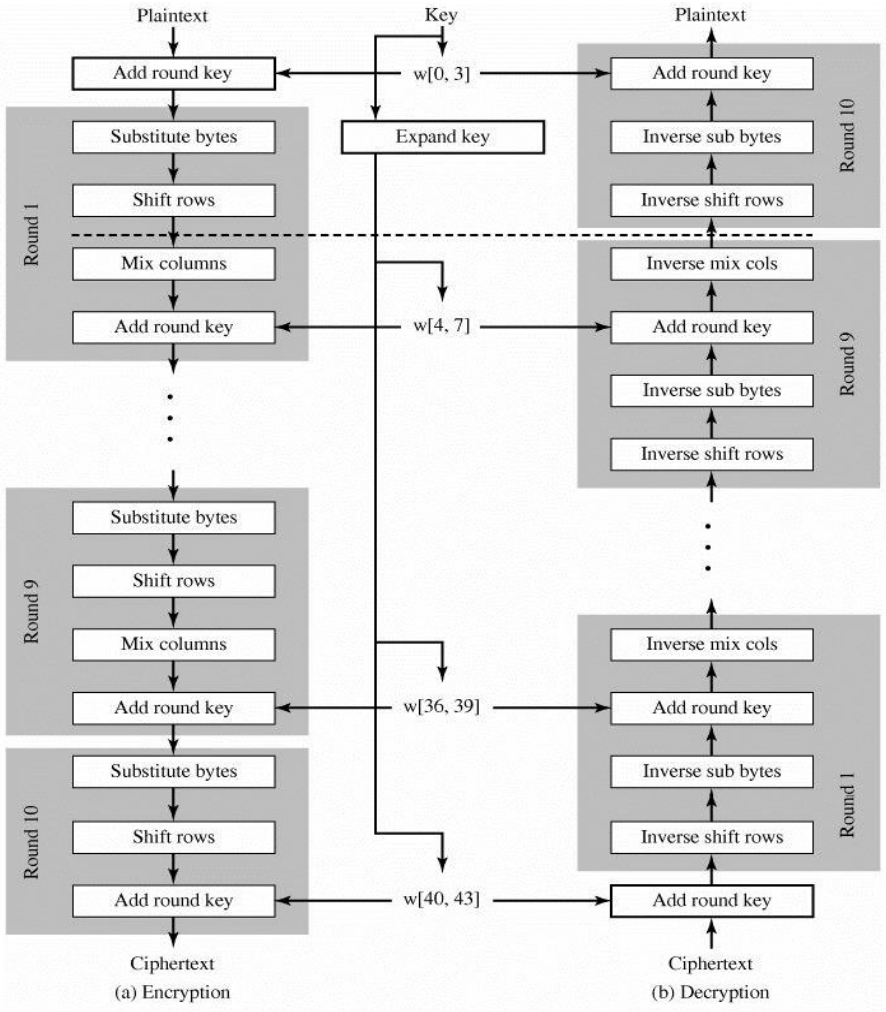
AddRoundKey

$$\begin{bmatrix} 62 & 02 & 27 & 26 \\ CF & 92 & 91 & 0D \\ 0C & 0C & F4 & D6 \\ 99 & 18 & 30 & 74 \end{bmatrix} + \begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix}$$



$$\begin{bmatrix} CE & 1B & 0F & 71 \\ B8 & 68 & 40 & 51 \\ 6A & D0 & DD & D6 \\ 6A & 39 & 71 & 1E \end{bmatrix}$$

AES Structure



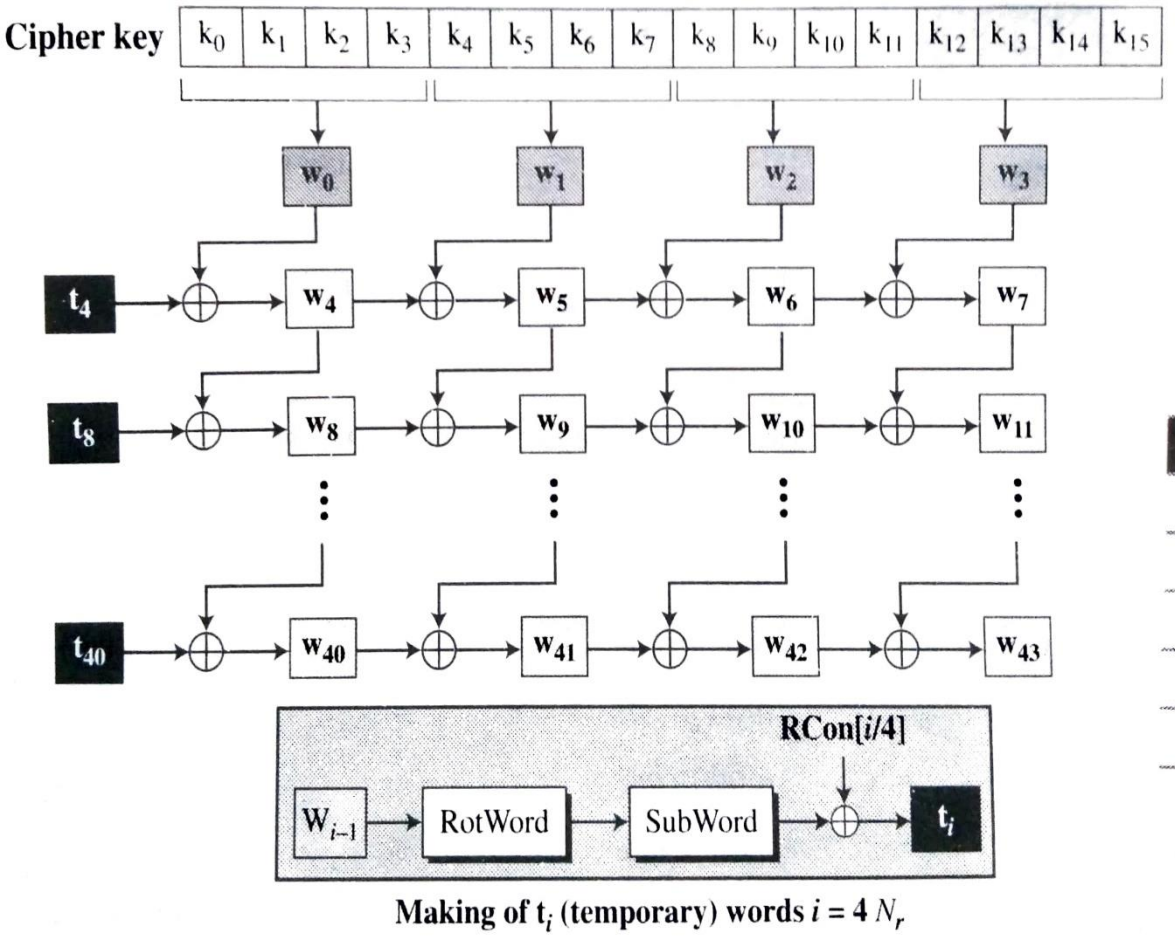
- Each round has four stages, except the last round.
- Key has three variants **128**, 192, 256.
- AES with key 128 bits has 10 rounds
- Each round takes 128 bits = 16 Bytes = 4 words = 1 state as a key to operate with plaintext.
- So, to operate the plaintext in 10 rounds, we need 40 + 4 words = 44 words as subkey.
- As the given key of 128 bits form only 4 words, we need a key expansion algorithm which makes 128 bit original key to 44 words

AES Key Expansion

w_0	w_1	w_2	w_3
k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

- The initial key of 128 bits can form 4 words $k_0=[w0, w_1, w_2, w_3]$ which can be used as subkeys in round 0.
- The key expansion algorithm needs to generate remaining subkeys k_1 to k_{10} = 40 words for the AddRoundKey operation in 10 rounds.
- The remaining 40 words in keys k_1 to k_{10} are generated as a function of first 4 words in k_0.

AES Key Expansion (Contd...)



Making of t_i (temporary) words $i = 4\ N_r$

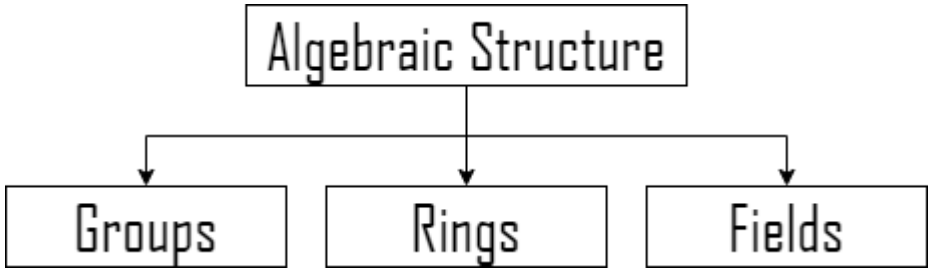
- If $i\ mod\ 4 = 0, w_i = t \oplus w_{i-4}$, where $t = SubWord(RotWord(w_{i-1})) \oplus RCon_{i/4}$.
- If $i\ mod\ 4 \neq 0, w_i = w_{i-1} \oplus w_{i-4}$.

Round	Constant (RCon)	Round	Constant (RCon)
1	$(\underline{01}\ 00\ 00\ 00)_{16}$	6	$(\underline{20}\ 00\ 00\ 00)_{16}$
2	$(\underline{02}\ 00\ 00\ 00)_{16}$	7	$(\underline{40}\ 00\ 00\ 00)_{16}$
3	$(\underline{04}\ 00\ 00\ 00)_{16}$	8	$(\underline{80}\ 00\ 00\ 00)_{16}$
4	$(\underline{08}\ 00\ 00\ 00)_{16}$	9	$(\underline{1B}\ 00\ 00\ 00)_{16}$
5	$(\underline{10}\ 00\ 00\ 00)_{16}$	10	$(\underline{36}\ 00\ 00\ 00)_{16}$

By the end of this session…

- Summarize the concept of common algebraic structures.
- Define Groups and their properties with examples..

Mathematics of Symmetric Cryptography: Algebraic Structures



- Discussed about five different sets of numbers, Z, Z_n, Z_n^*, Z_p, Z_p^*.
- Cryptography requires a set of integers and specific operations that are defined for the sets.
- The combination of set of integers and the operations that are applied on the elements of the set is called an Algebraic Structure.
- There are three common algebraic structures
  - Groups
  - Rings
  - Fields

Properties (Axioms) of Algebraic Structures

- Closure: If a and b are two elements of set, then $c = a \blacksquare b$ is also an element of the same set.
- Associativity: If a, b, and c are elements of set, then $(a \blacksquare b) \blacksquare c = a \blacksquare (b \blacksquare c)$.
- Commutativity: For all a and b in the given set, we have $a \blacksquare b = b \blacksquare a$.
- Existence of Identity: For all a in the set, there exists an element e, called the identity element, such that $e \blacksquare a = a \blacksquare e = a$.
- Existence of Inverse: For each a in the set, there exists an element a^{-1}, called the inverse of a, such that $a \blacksquare a^{-1} = a^{-1} \blacksquare a = e$.

$\blacksquare$ - Operates with

Group (G)

- A Group (G) is a combination of set of elements with a binary operator "▪", Represented as, G = <{}, ▪>, that satisfies four properties:
  - Closure
  - Associativity
  - Existence of Identity
  - Existence of Inverse.
- The sets are Z, Z_n, Z_n^*, Z_p, Z_p^* and the operators are addition/subtraction or multiplication
- A Commutativity Group, also called an Abelian Group, satisfies the property commutativity along with the above four properties.

Group (G) – Example - 1

- $G = \langle Z_{10}, + \rangle$

1. Satisfies Closure: 3+2 mod 10 = 5, 5 is also an element in the same set.
2. Satisfies Associativity: 4+(3+2) mod 10 = (4+3)+2 mod 10.
3. Satisfies Commutativity: for all elements in Z_{10}, satisfies 4+3 mod 10 = 3+4 mod 10
4. Satisfies Identity: As identity element is 0, then 3+0 mod 10 = 0+3 mod 10 = 3.
5. Satisfies Inverse: As the operator is +, we need to have additive inverse, 3+(-3) = 0 or (-3)+3 = 0.

So, the Group $G = \langle Z_{10}, + \rangle$ is a commutative group.

Group (G) – Example - 2

- $G = \langle Z_{10}, * \rangle$

1. Satisfies Closure: 3*2 mod 10 = 6, 6 is also an element in the same set.
2. Satisfies Associativity: 4*(3*2) mod 10 = (4*3)*2 mod 10.
3. Satisfies Commutativity: for all elements in Z_{10}, satisfies 4*3 mod 10 = 3*4 mod 10
4. Satisfies Identity: As identity element is 1, then 3*1 mod 10 = 1*3 mod 10 = 3.
5. Satisfies Inverse: As the operator is *, we need to have multiplicative inverse, not all elements in Z_{10} will have multiplicative inverse.

So, the Group $G = \langle Z_{10}, * \rangle$ is not a commutative group.

Group (G) – Example - 3

- $G = \langle Z_{11}, * \rangle$

1. Satisfies Closure: 3*2 mod 11 = 6, 6 is also an element in the same set.
2. Satisfies Associativity: 4*(3*2) mod 11 = (4*3)*2 mod 11.
3. Satisfies Commutativity: for all elements in Z_{11}, satisfies 4*3 mod 11 = 3*4 mod 11
4. Satisfies Identity: As identity element is 1, then 3*1 mod 11 = 1*3 mod 11 = 3.
5. Satisfies Inverse: As the operator is *, we need to have multiplicative inverse, all the elements in Z_{11} will have a multiplicative inverse.

So, the Group $G = \langle Z_{11}, * \rangle$ is a commutative group.

Interesting Points

- Residue set under modulo n with operator "+" is always an abelian group. i.e., G = $\langle Z_n, + \rangle$ is an abelian group.
- Residue set under modulo n with operator "*" is not always an abelian group.
- Residue set under modulo p with both operators "+" and "*" is always an abelian group.
- Residue and Coprime Set under modulo n with both operators "+" and "*" is always an abelian group.

By the end of this session

- Explain different concepts of Groups and their properties.
- Write the definition of Ring with examples.

More on Groups (G)

- Finite Group: A Group with finite number of elements in the set. Otherwise, Infinite Group.

Eg. G = $<Z_{11}, +>$ is a finite group, where G = <Z, +>

- Order of a Group: The number of elements in the set is the order of group, represented as |G|. If the Group is finite, then the order is finite, Otherwise, Order is infinite.
- Subgroups: A Subset H of a group G is a subgroup of G if H itself is a group with respect to operations on G.
- Cyclic Subgroups: If a subgroup of a group can be generated using the power of an element, the subgroup is called the cyclic subgroup. Ex: Four subgroups can be made from the group G = $<Z_6, +>$, they are $H_1 = <\{0\}, +>$, $H_2 = <\{0,2,4\}, +>$, $H_3 = <\{0,3\}, +>$, H4= G

Exercise on Cyclic Subgroups

- G = $<Z_{10}, +> = <\{0,1,2,3,4,5,6,7,8,9\},+>$

H0 = <{0},+>

H1 = <{0,1,2,3,4,5,6,7,8,9},+>

H2 = <{0,2,4,6,8}, +> i.e., 0 mod 10 = 0, 2 mod 10 = 2, 4 mod 10 = 4, 6 mod 10 = 6, 8 mod 10 = 8, 10 mod 10 = 0, 12 mod 10 = 2 .....

H3 = <{0,3,6,9},+>, H4 = <{0,4,8},+> are not cyclic subgroup, because 12 mod 10 = 2

H5 = <{0,5},+>

H6 = <{0,6},+>, H7=<{0,7},+>, H8=<{0,8},+>, H9 = <{0,9},+> are not cyclic subgroup.

More on Groups (G) (Contd...)

- Cyclic Groups: A Cyclic Group is a group that is its own cyclic subgroup.

Eg: G = $<Z_{10}, +>$ has cyclic subgroups as H0, H1, H2, H5, among them H1 = G

- This means G is a cyclic subgroup of itself, So, G is called as Cyclic Group.
- The element that generates the cyclic subgroup as the group itself is called "Generator".

Eg: 1, 3, 7 and 9 generates the same group so, these are called generators of G = $<Z_{10}, +>$

Lagrange's Theorem

- Given a group G of order |G|, the orders of the subgroups can be easily determined if the divisors of |G| can be found.

Eg: The order of G = $<Z_{10}, +>$ is 10. the divisors of 10 are 1, 2, 5 and 10. This means that the group can have only 4 subgroups H0, H1, H2, H5.

Order of an Element in Group

- The order of an element 'a' in a group, ord(a), is the order of the cyclic group it generates.

Eg. In G = $\langle Z_6, + \rangle$, the orders of the elements are ord(0) = 1, ord(1) = 6, ord(2) = 3, ord(3)=2, ord(4) = 3, ord(5) = 6.

Exercise: find the order of all elements in G = $\langle Z_{10}, + \rangle$

Ring (R)

- A Ring, denoted as R=<{....}, ▪, Δ>, is an algebraic structure with two operations.
  - The first operation must satisfy all five properties. i.e., Closure, Associativity, Commutativity, Identity, Inverse
  - The second operation must satisfy only the first two. i.e., Closure and Associativity
- Eg; R = <Z, +, x> is a Ring

By the end of this session…

- Define Fields and their properties with examples.

Field (F)

- A Field, denoted by F=<{....}, ▪, Δ> is an algebraic structure with two binary operations of addition and multiplication, both operators must satisfy all the five properties.
- The identity element for addition is 0 and the identity element for multiplication is 1.
- The additive inverse of element x in the set is –x and the multiplicative inverse is x^{-1} which satisfies $x.x^{-1} = 1 \bmod n$.
- Furthermore, multiplication must distribute over addition

i.e., $a \times (b + c) = (a \times b) + (a \times c)$ or $(a + b) \times c = (a \times c) + (b \times c)$

Eg. F=<Z_p, +, ×>

Finite Fields ($GF(p^n)$)

- A field with finite number of elements is a Finite field.
- Galois proved that for a field to be finite, the number of elements should be p^n.
- So, Finite fields are usually called as Galois Fields and denoted as $GF(p^n)$.
- If n = 1, we have GF(p), where GF(2) is the most common field with two elements in the set {0,1}.

Addition		
+	0	1
0	0	1
1	1	0

Multiplication		
×	0	1
0	0	0
1	0	1

Additive inverse		
a	0	1
-a	0	-1

Multiplicative inverse		
a	0	1
a^{-1}	-	1

Exercise - 1

- GF(5) = <{0,1,2,3,4}, +, x>

Addition					
+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Multiplication					
+	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Additive inverse					
a	0	1	2	3	4
-a	0	4	3	2	1

Multiplicative inverse					
a	0	1	2	3	4
a^{-1}	-	1	3	2	4

GF(2^n) Fields

- In computers, these positive integers are stored in the computer as n-bit words, where n=1,2,3,4,….., which represents integers as 0 to 2^n-1, So, the modulus is 2^n.

Eg: for n=2, we have $2^n = 2^2$ integers can be represented and the integer values are 0 to 2^2-1 and the values are 0,1,2,3

- To work on fields in computers, we must use GF(2^n) fields, where the set contains 2^n elements for n bit words.

For example, n = 3, the set must contain 2^3 elements, where F=<{000,001,010,011,100,101,110,111},+,x>

- Now, 2^n is not a prime, so it doesn't satisfies the properties with the regular operations to make it field. So, to make it a field we need to define new operations.

GF(2^n) Fields

Addition				
+	00	01	10	11
00	**00**	01	10	11
01	01	**00**	11	10
10	10	11	**00**	01
11	11	10	01	**00**

Multiplication				
+	00	01	10	11
00	00	00	00	00
01	00	**01**	10	11
10	00	10	11	**01**
11	00	11	**01**	10

- 2-bit binary Addition can be computed as bit-wise xor

  Eg. 00 + 01 = 0 + 0, 0 + 1 = 01

- 2-bit binary Multiplication can be computed as:

10	01
ab	cd

=

1	0
ad + bc + ac	ac + bd

By the end of this session…

- Emphasize on the finite fields of type GF(2^n) using polynomials.

Polynomials

1	0	0	1	1	0	0	1
$1x^7$	$0x^6$	$0x^5$	$1x^4$	$1x^3$	$0x^2$	$0x^1$	$1x^0$

$$x^7 + x^4 + x^3 + 1$$

$$x^5 + x^2 + x$$

$0x^7$	$0x^6$	$1x^5$	$0x^4$	$0x^3$	$1x^2$	$1x^1$	$0x^0$
0	0	1	0	0	1	1	0

- It is easier to work with a representation of n-bit words as a polynomial of degree n-1 for the values in GF(2^n).

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0x^0$$

- Where x^i is the i^{th} term and a_i is called coefficient of the i^{th} term.
- The power of x defines the position of the bit in n-bit word.
- The coefficient defines the value of the bits.

Operations on Polynomials - Addition

- Add two polynomials $x^5 + x^2 + x$ and $x^3 + x^2 + 1$ in GF(2^8)

P1	$0x^7$	$0x^6$	$1x^5$	$0x^4$	$0x^3$	$1x^2$	$1x^1$	$0x^0$
P2	$0x^7$	$0x^6$	$0x^5$	$0x^4$	$1x^3$	$1x^2$	$0x^1$	$1x^0$
Add	$0x^7$	$0x^6$	$1x^5$	$0x^4$	$1x^3$	$0x^2$	$1x^1$	$1x^0$

$$x^5 + x^3 + x + 1$$

- Addition operation on polynomials can be performed as an xor operation on the coefficients.
- The additive identity for a polynomial is zero polynomial. i.e., P + E = P
- The additive inverse of a polynomial is the polynomial itself. i.e., P+P = E

Multiplication

- Multiply two polynomials $x^5 + x^2 + x$ and $x^3 + x^2 + 1$ in GF(2^8)

$$x^5(x^3 + x^2 + 1) + x^2(x^3 + x^2 + 1) + x(x^3 + x^2 + 1)$$

$$x^8 + x^7 + x^5 + x^5 + x^4 + x^2 + x^4 + x^3 + x$$

$$x^8 + x^7 + x^3 + x^2 + x$$

- Is the sum of the multiplication of each term of the first polynomial with each term of the second polynomial.
- We need to remember three points
  - Coefficient multiplication is done in binary., GF(2).
  - Multiplying x^i and x^j results in x^{i+j}.
  - Multiplication may create terms with degree more than n-1.
- The multiplicative identity is 1. For example, in GF(2^8), the multiplicative identity is the bit pattern 00000001.
- For finding the multiplicative inverse, we need to use Extended Euclidean algorithm in the modulus and the polynomial.

Modulus

- Find $x^8 + x^7 + x^3 + x^2 + x$ mod $x^8 + x^4 + x^3 + x + 1$

$$
\begin{array}{r|l}
 & 1 \\
\hline
x^8 + x^4 + x^3 + x + 1 & x^8 + x^7 + x^3 + x^2 + x \\
 & x^8 + x^4 + x^3 + x + 1 \\
\hline
 & x^7 + x^4 + x^2 + 1
\end{array}
$$

- Addition of two polynomials never creates a polynomial out of the n-1.
- Multiplication of two polynomials may create a polynomial with degree more than n-1.
- So, We need to apply modulus operation by dividing the polynomial with prime polynomial or Irreducible polynomial.
- A Prime polynomial cannot be factored and cannot be divisible by other polynomials with degree less than n.
- The number of terms in an prime polynomial is odd.
- Eg: $x^8 + x^4 + x^3 + x + 1$