CS434: Cryptography and Security:

Cryptography:
- Art of protecting information by transforming it into an unreadable format.
- Method of protecting info and communication through the use of codes so that only those for whom the info is intended can read and process it.
- Constructing and Analysing protocols that prevent third parties or the public from reading private messages.

Symmetric Cryptography: if keys are same at sender and receiver

Asymmetric Cryptography: if keys are different at sender and receiver.

Encryption: Process of transforming info from readable to unreadable format.

Decryption: Process of transforming info from unreadable to readable format

Key: String of bits used by cryptographic algorithms to transform plain text to cipher text or vice-versa.

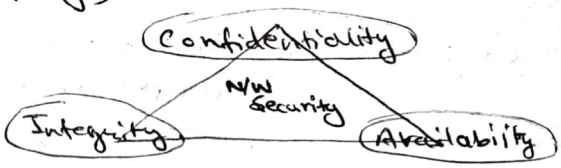
Types of Cryptography:

Symmetric	Asymmetric
• Secret key Cryptography / Private key Cryptography • Simplest kind of encryption technique that involves only 1 key to encrypt and decrypt information. • E.g.: DES (Data Encryption System) AES, RC4, 2DES, 3DES	• Public key cryptography. • Uses two keys (public key and private key) for encryption and decryption. • E.g.: RSA, DSA, Elliptic Curve, Deffie Hellman

NOTE: Msg that is encrypted using a public key can only be decrypted using a private key and vice versa, ie., msg encrypted using a private key can only be decrypted using the public key.

• Performance → Symmetric key algos are faster in execution	• Slower in execution
• less complex & less computation power.	• more complex & more computational power
• Used for transfer of bulk data.	• Secret exchanging the secret key.
• Sharing key between sender and receiver is not safe.	• No problem of key sharing because of private key concept.

* Security Goals: (CIA Triad in Cryptography)
  - Confidentiality
  - Integrity
  - Availability



* Security Services:
  - Data Confidentiality : Protect data from attackers.
  - Data Integrity : Protect data from modification.
  - Authentication : Verification of actual person linked to that data.
  - Non Repudiation : Assurance that someone cannot deny validity of something.
  - Access Control : to whom access should be given can be decided.

* Security Attacks:

1. Passive Attack : Attempts to learn or make use of info from system but does not affect the system resources / modify it.
   a. Release of Message Content
   b. Traffic Analysis.

   Passive attacks are difficult to detect.

2. Active Attack : Attempts to alter system resources / info. can see and modify message.
   a. Masquerade : impersonating; one entity pretends to be another entity.
   b. Modification of msg/data : Delay, reorder, modify, edit.
   c. Replay : involves passive capture of a msg and its subsqent retransmissions to produce an unauthorised effect.
   d. Denial of Services : Prevents normal use of communication facilities; disrupts an entire network

* Security Mechanisms:
  1. Encipherment : Use of mathematical algos to encrypt data.
  2. Digital Signature : Sender can electronically sign the data and receiver can electronically verify data.
  3. Data Integrity : appends data with a short check value.
  4. Authentication exchange : Two entities exchange some msg to prove their identity to each other.
  5. Traffic Padding : Add extra/dummy bits with data while encrypting.
  6. Routing Control : Selecting and Continuously changing different available routes between sender and receiver.
  7. Access Control : proves that user has right access to data
  8. Notarisation : Selecting 3rd Party to control communication blw two entities.

* Traditional Encryption Techniques (classical Encryption Techniques)

1. Substitution : (substitute one letter for another)
   - ↳ Monoalphabetic : ceaser ciphers
   - ↳ Polyalphabetic : playfair, Vigenere, hill cipher, Vernam
2. Transposition : (jumble letters) → ↳ keyed, ↳ keyless
   - ↳ Railfence (keyless)
   - ↳ Row Transposition Cipher. (keyed).
   - ↳ [illegible] Cipher (keyed)
   - ↳ [illegible] Cipher (keyless)

- Railfence Technique: Sequence of diagonals.

  Eg: Plaintext: Hello World!

  Encryption: H e l l o w o r l d !

  Cipher Text: HlOol!elwrd

- Row Transposition Ciphers: write msg in rectangle, row by row, and read msg off, column by column, but permute orders of column.

  Key: int value (unique digit from 0 to 9)

  Eg: Plaintext: Attack postponed until two am

Key:	4	3	1	2	5	6	7
Plaintext ⇒	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z
						dummy bits	

  Cipher: ttna aptm tsuo aodw coix knly petz

- Ceaser Cipher: Shift cipher / additive cipher.

  Key = No. of shifts (int value)

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

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

* Playfair Ciphers: create 5×5 matrix (grid of letters)
  - → Convert text into pairs of alphabet
    - if pair has same letters, separate them and give 'x' as pair to the first one.
    - if last letter is single (cannot form pair), give 'z' as pair.
  - → Place them in matrix. i/j will be one cell combined.
  - → • if both alphabets in same row, replace them with immediate right alphabets
    - if both alphabets in same column, replace them with immediate below alphabets
    - if in diff row/colm, replace with alphabets in same row diff colm and same col diff row

Playfair Cipher example:

Plain text : Hello Johnny boy

Pairing: He lx lo Jo hn ny bo yz

Key : Hidden key

5×5 Matrix:

h	i/j	d	e	n
k	y	a	b	c
f	g	l	m	o
p	q	r	s	t
u	v	w	x	z

he → in lx → mw lo → mf Jo → ng

hn → ih ny → ic bo → cm yz → cv

Cipher text: inmwmfngihiccmcv

* Vigenere Cipher : 26×26 Matrix.

E.g.: Plain Text : Give Money.

Key : Lock

G	I	V	E	M	O	N	E	Y
L	O	C	K	L	O	C	K	L

for each column pair, find intersection alphabet from 26×26 matrix

Method 1 using 26×26 table.

26×26 Matrix format:

	a	b	c	d	e	f	g	h	i	j	k	l	m	...	z
A	A	B	C	D	E	F	G	H	I	J	K	L	M		Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N		A
C	C	D	E	F	G	H	I	J	K	L	M	N	O		B
⋮															
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L		Y

Cipher: GI → R IO → W VC → X EK → O ML → X

OO → O NC → P EK → O YL → J

∴ Cipher : RWXOXOPOJ

Method II: Using mod formula:

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

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

Unlike Ceasar Cipher, here K value is different at each case.

Alphabets A → 0, B → 1, C → 2 ... Y → 24, Z → 25.

- Vernam Cipher:

Length of key used for encryption = length of plaintext.

A→0, B→1, C→2 .... Z→25

Convert both Plaintext & key to corresponding Numbers

Add PT + key; if PT+key > 25, subtract 26.

Cipher: Write corresponding letters to that number obtained.

Example: Plaintext: Hello

key : world

Plaintext:	H	E	L	L	O
Num:	7	4	11	11	14
Key:	W	O	R	L	D
	22	14	17	11	3
PT + key	29	18	28	22	17
if >25, subt 26	3	18	2	22	17
Cipher	D	S	C	W	R

∴ Cipher: DSCWR

for decryption, CT − key; if CT-key < 0, add 26.

* Hill Cipher:

To encrypt: $c = KP \bmod 26$

A→0, B→1 ... Z→25

Key: Key matrix must be square matrix.

e.g.: VIEW = $\begin{bmatrix} V & I \\ E & W \end{bmatrix} = \begin{bmatrix} 21 & 8 \\ 4 & 22 \end{bmatrix}$

key: QUICKNESS = $\begin{bmatrix} Q & U & I \\ C & K & N \\ E & S & S \end{bmatrix} = \begin{bmatrix} 16 & 20 & 8 \\ 2 & 10 & 13 \\ 4 & 18 & 18 \end{bmatrix}$

If key is n×n, convert plaintext in vectors of length n.

Example:

Plaintext: ATTACK

Let key = $\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}_{2\times 2}$

$$\begin{bmatrix} A \\ T \end{bmatrix} \quad \begin{bmatrix} T \\ A \end{bmatrix} \quad \begin{bmatrix} C \\ K \end{bmatrix} = \begin{bmatrix} 0 \\ 19 \end{bmatrix} \quad \begin{bmatrix} 19 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 10 \end{bmatrix}$$

Then, $c = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 57 \\ 114 \end{bmatrix} \bmod 26 = \begin{bmatrix} 5 \\ 10 \end{bmatrix} = \begin{bmatrix} F \\ K \end{bmatrix}$

∴ AT → FK

Similarly, $\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 19 \\ 0 \end{bmatrix} \bmod 26 = \begin{bmatrix} 12 \\ 5 \end{bmatrix} = \begin{bmatrix} M \\ F \end{bmatrix}$

$\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 2 \\ 10 \end{bmatrix} \bmod 26 = \begin{bmatrix} 8 \\ 14 \end{bmatrix} = \begin{bmatrix} I \\ O \end{bmatrix}$

∴ ATTACK → FKMFIO

To decrypt, $P = K^{-1} C \bmod 26$.

$$\det^{-1}(K) \equiv \det(K) \times 1 \bmod 26$$

If $K^{-1} = \frac{1}{x}$ [matrix], mod 26

find, $x^{-1} x \equiv 1 \bmod 26 \rightarrow x^{-1} \equiv x \bmod 26$

and $K^{-1} = x^{-1}$ [Matrix] mod 26

✱. Euler's Toitient function ($\phi(n)$):

- $\phi(n)$: Number of Relative Primes / Coprime (Mutually prime).

What is Relative Prime? If $\gcd(a,b) = 1$; a and b are relative primes.

- $\phi(p) = p-1$; where p is prime.
- $\phi(n) = (p-1) * (q-1)$; where p & q are primes
  $= (p-1) * (q-1) * (r-1) \ldots$ where p, q, r... are primes.

✱ Euler's Theorem:

Also called Fermat - Euler Theorem or Euler's Toitient theorem

$$x^{\phi(n)} \equiv 1 \bmod n$$

ie, $x^{\phi(n)} \bmod n = 1 \bmod n$.

✱ Fermat's Theorem:

Special case of Euler's Theorem.

$x^{n-1} \equiv 1 \bmod n$, where n is prime.
x is not divisible by n.

or, $x^n x^{-1} \equiv 1 \bmod n$

or, $\frac{x^n}{x} \equiv 1 \bmod n$

or, $x^n \equiv x \bmod n$, Another form of fermat's theorem

✱ Primality Test (Miller - Robin)

If p is prime, then $p-1$ is even. So, let $a=2$.

$p-1 \equiv a^b \bmod p$; b is $(p-1)/2$ until * odd
i.e., if 28: b = 14, 7

if $p-1 \equiv +1 \bmod p$; composite

if $p-1 \equiv -1 \bmod p$; probable prime.

Use Fermat's theorem whenever necessary.

* Extended Euclidean Algorithm:

$$d = \gcd(a,b) = ax + by.$$

$r_{-1} = a \qquad r_0 = b$

$y_{-1} = 0 \qquad y_0 = 1$

$x_{-1} = 1 \qquad x_0 = 0$

r is remainder

Onwards, $x_i = x_{i-2} - q_i x_{i-1}$

$y_i = y_{i-2} - q_i y_{i-1}$

q is quotient.

Example: gcd (1759, 550)

i	r_i	q_i	x_i	y_i
-1	1759		1	0
0	550		0	1
1	109	3	1	-3
2	5	5	-5	16
3	4	21	106	-339
4	1	1	-111	355
5	0	4	-	-

$\frac{1759}{550} = 3\frac{109}{550}$

$\frac{550}{109} = 5\frac{5}{109}$

$\frac{109}{5} = 21\frac{4}{5}$

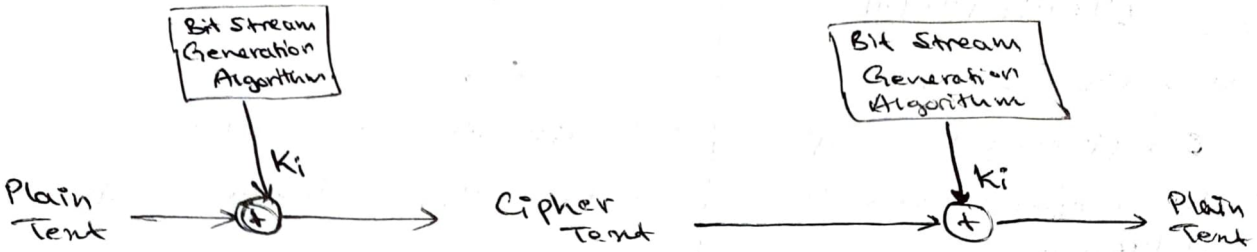
$5/4 = 1\frac{1}{4}$

$x = -111 \quad y = 355 \quad a = 1759 \quad b = 550$

$d = 1759 \times (-111) + 550 \times 355 = 1.$

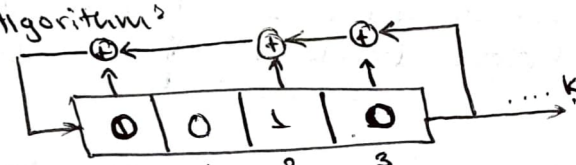
* Stream and Block Cipher:

- **Stream Ciphers:** Encrypts digital data stream one bit/byte at a time. Symmetric key cipher uses 8 bits.



Bit Stream Generation Algorithm:

Example:
LFSR



Right Shift at each step
Leftmost new bit
$= S_0 \oplus S_2 \oplus S_3$

0	1	2	3	Keys
1	1	0	1	0 → K_0
0	1	1	0	1 → K_1
1	0	0	1	0 → K_2
0	1	0	0	1 → K_3
				⋮

and so on.... for K_n required keys

$C_i = P_i \oplus K_i$; $P_i = C_i \oplus K_i$

- **Block Ciphers:** Block of plain text is treated as a whole and used to produce ciphertext of equal length. Symmetric key cipher.
Typically block size of 64 or 128 bit.

[illegible]

Block Cipher	Stream Cipher
• Plain → Cipher text by taking plaintexts block at a time [4] [4] [4] [4]	• Plain → Cipher text by taking 1 bit or 1 byte at a time 1 0 1 1 0 0 0 1
• Uses 64 bits or more.	• Uses 8 bits.
• Complexity is simple	• Complexity is complex.
• Uses confusion as well as diffusion.	• Uses only confusion concept
• Reverse encrypted text is hard	• Reverse encrypted text is easy.
• Eg: ECB (Electronic Code Block) CBC (Cipher Block Chaining)	• E.g: CFB (Cipher Feedback) OFB (Output Feedback)

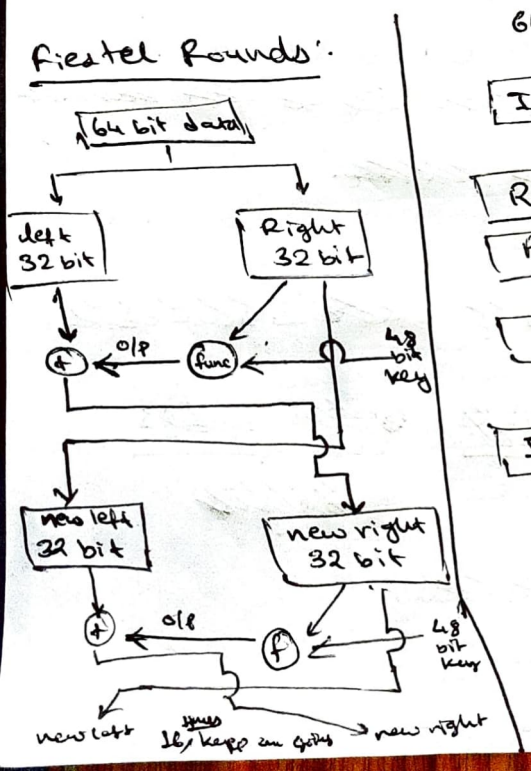
* DES (Data Encryption Standard)

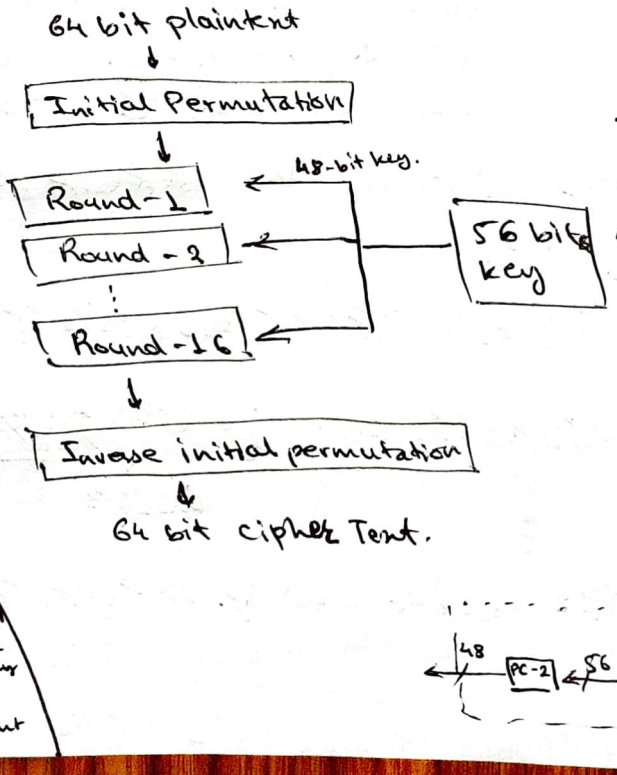
- block cipher
- Symmetric cypher
- Plain Text : 64 bits.
- Non-linear
- Completeness
- Avalanche Effect
- Rounds : 16 rounds.
- Key size: 64/56 bit.

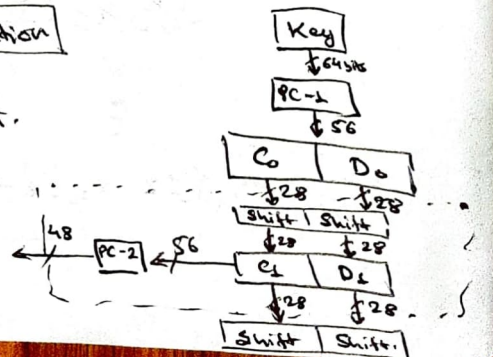
Steps:
1. Initial permutation
2. 16 Feistel Rounds
3. Swapping (Left might swap)
4. Final permutation (Inverse initial permutation.

- Avalanche Effect:- even if just 1 bit is changed in plaintext, in des, the cipher text will differ by 34 bit positions
- If 1 bit is changed in key, 35 bits in cipher changes

Basic Structure:

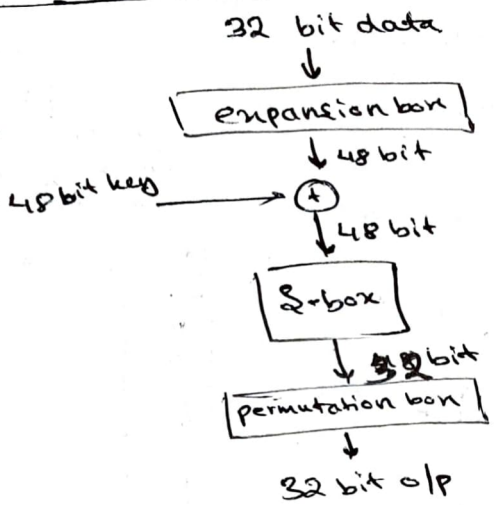






Function definition (func) (f)

DES weakness
- Key size : 56 bit
  easy to cra
- weak keys:
- semi weak keys.
- Possible weak keys
- key Clustering.



Expansion Box Function:

32 bit input. in 1's 0's form.

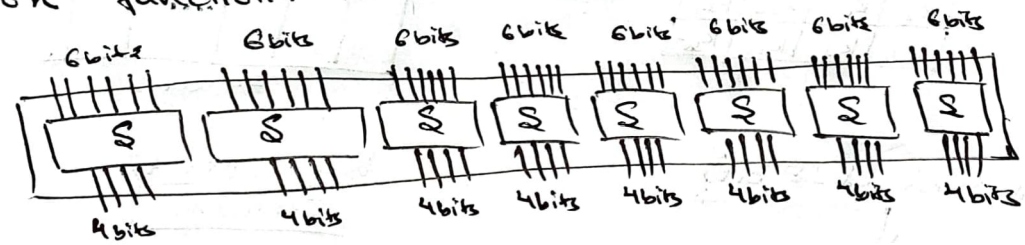
E.g.:

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

Generated 8 bits ← 32 bit input → Generated 8 bits

Generated 48 bits

S-box function: 48 bit distributed as 6×8 bits



How 6 bits converted to 4 bits?

Eg.: 6 bit: 0 0 1 0 1 1

Leftmost - Rightmost bit combined: 01 tells row

Middle 4 bits : 0101 tells column.

S-box table, $(01)_2$ means $(1)_{10}^{th}$ number row.
$(0101)_2$ means $(5)_{10}^{th}$ number column.

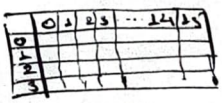


Table starts from (0, 0) .... (3, 15)
Tables have decimal values.
Select that value and convert to 4 bit binary.
It will be output of S-box. Hence, 6 bit → 4 bit.

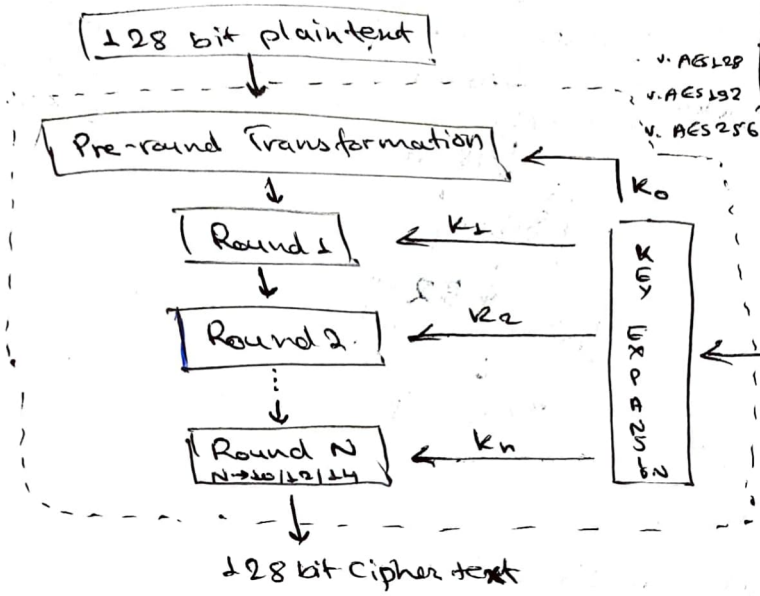
AES (Advanced Encryption Standard).

- block cipher - Symmetric cipher

Block Size : 128bit.

	Rounds	keys
v. AES128	10	128
v. AES192	12	192
v. AES256	14	256

1 word = 32 bits
= 4 bytes



State Matrix (4×4) 16bytes stores o/p from Rounds.

Input Array (4×4) 16bytes → 128 bit → 4 words.

State Array

	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
word 0 Byte 2	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

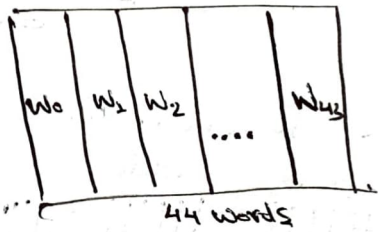
} word 0, Byte 0 : $S_{0,0}$ word 1 Byte 1 $S_{1,1}$ and so on.

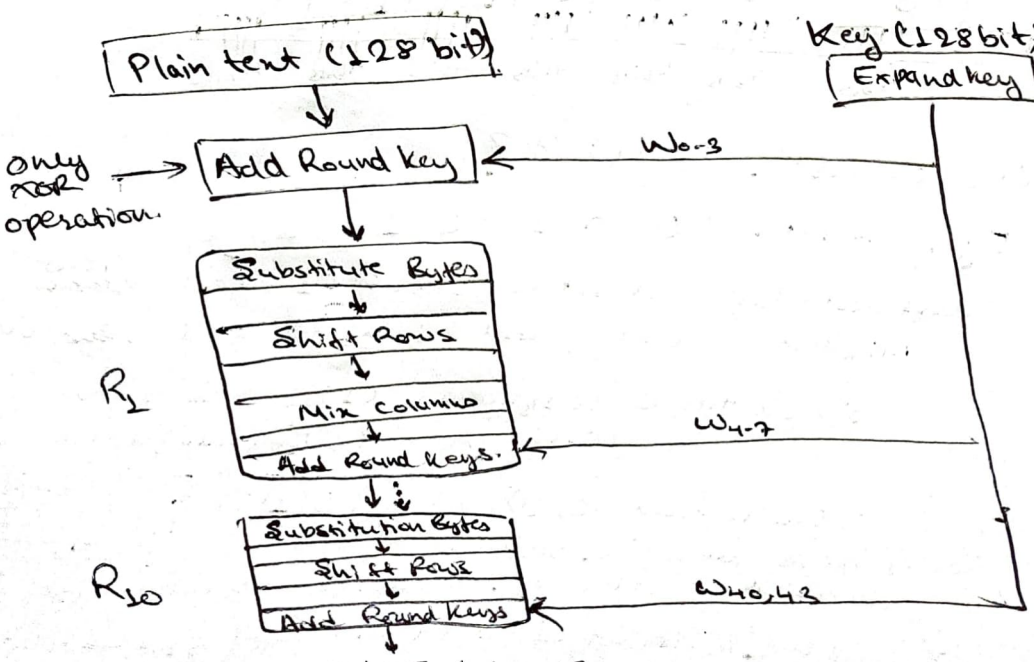
→ [W_0 W_1 W_2 W_3]

Key:

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}

expandkey →

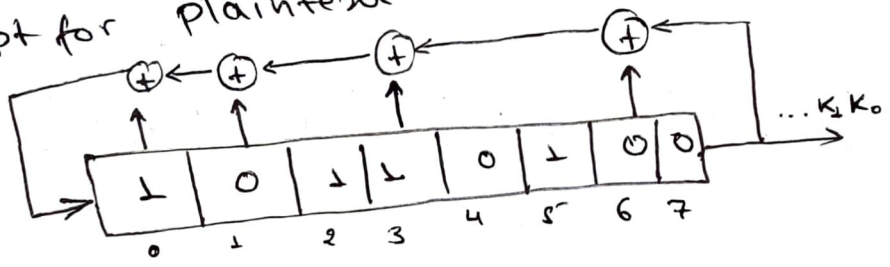




Subhojit Ghimire, 1912160
Computer Science and Engineering

CS434: Linear Feedback Shift Register.

Q. Decrypt for plaintext:



LFSR.

$C_0\ C_1 \ldots\ldots C_{31}$

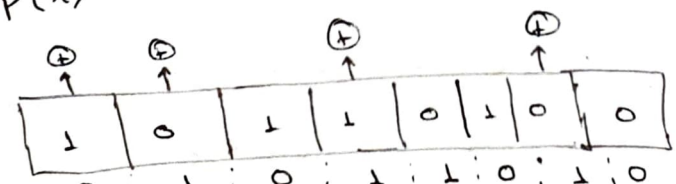
01011111 11110100 00010101 01111010

Find. $P_0, P_1 \ldots P_{31}$ ASCII of. $P_0 \to P_7$; $P_8 \to P_{15}$

$P_{16} \to P_{23}$; $P_{24} \to P_{31}$

Soln:- Primitive Polynomial: $1 + x + x^2 + x^4 + x^7$

$$P(x) = 1 + C_0x + C_1x^2 + C_2x^4 + C_6x^7$$



								Keys
0	1	0	1	1	0	1	0	0
1	0	1	0	1	1	0	1	0
1	1	0	1	0	1	1	0	1
0	1	1	0	1	0	1	1	0
0	0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	0	1
1	1	0	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1
0	0	1	1	0	0	1	1	0
0	0	0	1	1	0	0	1	1
1	0	0	0	1	1	0	0	1
1	1	0	0	0	1	1	0	0
1	1	1	0	0	0	1	1	0
1	1	1	1	0	0	0	1	1
1	1	1	1	1	0	0	0	1
1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	0	0

first set of keys for C_0-C_7

Second sets of keys for C_8-C_{15}

1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	1	1
0	0	0	1	1	0	1	1	1
0	0	0	0	1	1	0	1	1

Third set of keys for $C_{16} - C_{23}$

0	0	0	0	0	1	1	0	1
1	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0
0	1	1	1	0	1	0	0	0
0	0	1	1	1	0	1	0	0

Third set of keys for $C_{24} - C_{31}$

Given, $P_i = C_i \oplus K_i$

So, $C_0 - C_{31}$: 01011111 11110100 00010101 01111010

$\oplus$ $K_0 - K_{31}$: 00101101 01100110 00111100 10110000

$= P_0 - P_{31}$: $\underbrace{01110010}_{2}$ $\underbrace{10010010}_{2}$ $\underbrace{00101010}_{2}$ $\underbrace{11001010}_{2}$

$P_0 - P_7$ $P_8 - P_{15}$ $P_{16} - P_{23}$ $P_{24} - P_{31}$

$(114)_{10}$ $(146)_{10}$ $(42)_{10}$ $(202)_{10}$

Corresponding ASCII:	r	'	*	Ê

∴ Plaintext = r'* Ê

* RSA Algorithm:

- Asymmetric Cryptographic Algo (2 keys, i.e., public and private key concept is used here).
- RSA scheme is a block cipher in which plain text and cipher text are integers between 0 and $n-1$ for some value n.

Algorithm:

1. Key Generation:

i) Select 2 large prime no.s 'p' and 'q'.
ii) Calculate, $n = p * q$.
iii) Calculate, $\phi(n) = (p-1) * (q-1)$
iv) Choose value of **e** such,
$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$
v) Calculate
$d \equiv e^{-1} \bmod \phi(n)$
ie, $ed \equiv 1 \bmod \phi(n)$
vi) Public Key = $\{e, n\}$
vii) Private key = $\{d, n\}$

Example: Given, $p = 3$, $q = 11$; find e and d.

$n = p * q = 3 * 11 = 33$

$\phi(n) = (p-1) * (q-1) = (2 * 10) = 20$

Now, $1 < e < 20$; $\gcd(e, 20) = 1$.

So, let $e = 7$.

Then, $d \equiv e^{-1} \bmod \phi(n)$

$de \equiv 1 \bmod \phi(n)$

ie, $d * 7 \equiv 1 \bmod 20$

So, $(7d \bmod 20 \equiv 1)$ → Do multiplicative inverse. (Method I)

$7 * 1 \bmod 20 \equiv 7$
$7 * 2 \bmod 20 \equiv 6$
$7 * 3 \bmod 20 \equiv 1$

$\therefore d = 3$.

can also be solved by Extended Euclidean Algo (Method II)

Easily finding,
Add 1 to multiples of $\phi(n)$ and find which number is multiple of e. i.e.,
Here, $\phi(n) = 20$.
So, $20*1 = 20 + 1 \rightarrow 21$
$20*2 = 40 + 1 \rightarrow 41$
$20*3 = 60 + 1 \rightarrow 61$
easily, $21/7 \rightarrow 3$ ✓
$41/7 \rightarrow \ldots$
$61/7 \rightarrow \ldots$

(NOTE: d is private key. It need not necessarily be 3 just because it is the first perfect find. 'd' can be anything: 3, 23, 43, 63 ... upto user (sender) to choose)

2. Encryption:

$$C = M^e \bmod n \quad ; \quad M < n$$

3. Decryption

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

NOTE: (e, n) is public key used in encryption
(d, n) is private key used in decryption.

* Chinese Remainder Theorem:

- It states that there always exists an 'x' that satisfies the given congruence.

$x \equiv rem[m] \pmod{num[m]}$

$x \equiv rem[0] \pmod{num[0]}$

$x \equiv rem[1] \pmod{num[1]}$

$\vdots$

$x \equiv rem[n-1] \pmod{num[n-1]}$

Here, num[0], num[1] ... num[n-1], all must be coprime to one another.

i.e., $gcd(0,1) = gcd(1,n-1) = gcd(n-1,0) = 1$

~~Example~~ if, $x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

$x \equiv a_3 \pmod{m_3}$

(i) $gcd(m_1, m_2) = gcd(m_2, m_3) = gcd(m_3, m_1) = 1$

(ii) $x = (M_1 X_1 a_1 + M_2 X_2 a_2 + M_3 X_3 a_3 + \dots + M_n X_n a_n) \bmod M$

$\rightarrow M = m_1 * m_2 * m_3 \dots m_n$

~~Skip~~ $M_i = \frac{M}{m_i}$ i.e, $M_1 = \frac{M}{m_1} = \frac{m_1 m_2 m_3}{m_1} = m_2 m_3$

$\rightarrow$ i.e., $M_1 = m_2 m_3$; $M_2 = m_1 m_3$; $M_3 = m_1 m_2$

$M_i X_i \equiv 1 \bmod m_i$

Example:

$x \equiv 1 \pmod 5$

$x \equiv 1 \pmod 7$

$x \equiv 3 \pmod{11}$

Here, $a_1 = 1$, $a_2 = 1$, $a_3 = 3$

$m_1 = 5$, $m_2 = 7$, $m_3 = 11$

$gcd(5,7) = gcd(7,11) = gcd(5,11) = 1$.

$M = m_1 * m_2 * m_3 = 5 * 7 * 11 = 385$

$M_1 = m_2 * m_3 = 7 * 11 = 77$

$M_2 = m_1 * m_3 = 5 * 11 = 55$

$M_3 = m_1 * m_2 = 5 * 7 = 35$

$M_1 X_1 \equiv 1 \bmod m_1 \Rightarrow 77 X_1 \equiv 1 \bmod 5 \Rightarrow 2X_1 \equiv 1 \bmod 5 \Rightarrow X_1 = 3$

$M_2 X_2 \equiv 1 \bmod m_2 \Rightarrow 55 X_2 \equiv 1 \bmod 7 \Rightarrow 6X_2 \equiv 1 \bmod 7 \Rightarrow X_2 = 6$

$M_3 X_3 \equiv 1 \bmod m_3 \Rightarrow 35 X_3 \equiv 1 \bmod 11 \Rightarrow 2X_3 \equiv 1 \bmod 11 \Rightarrow X_3 = 6$

$\therefore X_1 = 3$; $X_2 = 6$; $X_3 = 6$

So now, $x = (M_1 X_1 a_1 + M_2 X_2 a_2 + M_3 X_3 a_3) \bmod 385$

$= (77 * 3 * 1 + 55 * 6 * 1 + 35 * 6 * 3) \bmod 385$

$= (231 + 330 + 630) \bmod 385$

$\therefore x \equiv 1191 \bmod 385 \equiv 36$ $\therefore x = 36$

* Deffie-Hellman Key Exchange Algorithm: (Elgamal Cryptosystem)

- NOT an 'encryption' algo.
- Used to exchange 'secret' keys between 2 users.
- Uses asymmetric encryption to exchange the secret key.

ALGORITHM

i) Consider a prime number 'q'

ii) Select primitive root, α of q such that $\alpha < q$

$\phi(n)$: primitive 'prime' count

$\phi(\phi(n))$: primitive 'root' count

'a' is primitive root of q, if

$a \bmod q$
$a^2 \bmod q$
$a^3 \bmod q$
$\vdots$
$a^{q-1} \bmod q$

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

for example

$3^1 \bmod 7 \equiv 3$
$3^2 \bmod 7 \equiv 2$
$3^3 \bmod 7 \equiv 6$
$3^4 \bmod 7 \equiv 4$
$3^5 \bmod 7 \equiv 5$
$3^6 \bmod 7 \equiv 1$

all 1 - 6 obtained without repetition.

$5^1 \bmod 7 \equiv 5$
$5^2 \bmod 7 \equiv 4$
$5^3 \bmod 7 \equiv 6$
$5^4 \bmod 7 \equiv 2$
$5^5 \bmod 7 \equiv 3$
$5^6 \bmod 7 \equiv 1$

Here, $q = 7$
Let's take $\alpha = 5$

So, 3 and 5 are primitive roots of 7.

$\phi(7) = 6$ primitive primes

$\phi(\phi(7)) = \phi(6) = (3-1) * (2-1) = 2$ primitive roots, i.e., 3 and 5.

Key generation of Person A:

iii) Assume X_A (private key of A) and $X_A < q$

Calculate, $Y_A = \alpha^{X_A} \bmod q$

(Y_A: public key of A.)

Let, $X_A = 3$
So, $Y_A \equiv 5^3 \bmod 7$
$\equiv 6$

Key generation of Person B:

iv) Assume X_B (private key of B) and $X_B < q$

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

Let $X_B = 4$
So, $Y_B \equiv 5^4 \bmod 7$
$\equiv 2$

v) Calculate secret key:

$K_1 = (Y_B)^{X_A} \bmod q$

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

$K_1 \equiv 2^3 \bmod 7 \equiv 1$
$K_2 \equiv 6^4 \bmod 7 \equiv 1$

Y_B and Y_A are public key known to all.

if, $K_1 = K_2$, then key exchange is successful.

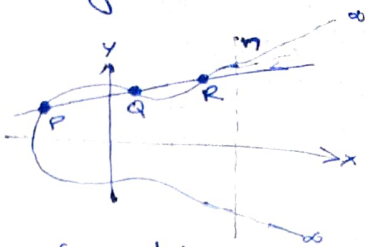
So, $K_1 = K_2 = K$

When A sends Msg to B,	When B sends Msg M to A
$C_1 = KM \bmod q$	$C_2 = KM \bmod q$
when B decrypts A's msg,	When A decrypts B's Msg M,
$M = C_1 K^{-1} \bmod q$	$M = C_2 K^{-1} \bmod q$

* Elliptic Curve Cryptography: (ECC).

- Asymmetric (public) key cryptosystem.
- It provides equal security with smaller key, i.e, Small key size and high security.
- Makes use of Elliptic curves:

$$y^2 = x^3 + ax + b$$



- symmetric to x-axis
- if we draw a line, it will touch max of 3 points

- Trapdoor function: function that is easy to compute in one direction, but difficult to compute in opposite direction (finding inverse) without special information.

ALGORITHM:

- Let $E_q(a,b)$ be elliptic curve.
  Consider eqn $Q = kP$ (see fig ↑)
  where, Q, P are points on curve and $k < n$
  if P and k given, easy to find Q
  if Q and P given, extremely difficult to find k).
  q is prime number or an integer of form 2^m.
- Let G, be a point on curve whose order is large value of n.
  (generator point)
- User A key generation,
  Select private key n_A ; $n_A < n$
  Calculate public key P_A ; $P_A = n_A * G$
- User B key generation,
  Select private key n_B; $n_B < n$
  Calculate public key P_B; $P_B = n_B * G$.
- Secret key, $K = K_A = K_B$
  $\therefore K = n_A P_B = n_B P_A$
- ENCRYPTION: (A sends to B)
  - Let Msg be M.
  - Let point P_m be the point on ellipticcurve where msg M is encoded.
  - choose random positive integer K.
  - Cipher point, $C_m = \{kG, P_m + kP_B\}$
- Decryption: (B decrypts msg)
  - Multiply 1st point in pair with receiver's secret key.
    $kG * n_B$
  - Subtract it from 2nd point in pair.
    $P_m + kP_B - (kG * n_B)$

  we know, $P_B = n_B * G$.
  So, $P_m + kP_B - kP_B = P_m$
  So, P_m is the original point.

NOTE: * Modulus of -ve numbers:

Eg.: $x \equiv -51 \bmod 10$. $\quad x \equiv -N \bmod m$

$\cong -51 \bmod (-60)$

i.e., mod m; make it in -ve such that multiple of m in -ve is less than -N then subtract -N-m

i.e, $-51 - (-60) = -51 + 60 = 9$

$\therefore -51 \bmod 10 \equiv 9$.

Eg.: $-15 \bmod 7 \equiv 6$

$\cong -15 \bmod -21$; $-15 - (-21) = 21 - 15 = 6$ ✓

* Multiplicative Inverse:

$x \equiv y^{-1} \bmod m$

$\Rightarrow yx \equiv 1 \bmod m$.

Then find, (multiples of m) + 1

Then, (multiples of m+1) / y.

Example: $x \equiv 7^{-1} \bmod 20$

$7x \equiv 1 \bmod 20$

(multiples of 20)+1 → $20*1+1 = 21$
$20*2+1 = 41$
$20*3+1 \ldots = 61$.

(multiples of 20+1) / y → $21/7 \Rightarrow 3$. ✓ perfect
$41/7 \Rightarrow 5.xxx$
$61/7 \rightarrow 8.xxx$

$\therefore x \equiv 7^{-1} \bmod 20 \equiv 3$.

* modulus of huge numbers where fermat's theorem cannot be used.

Example: $31^8 \bmod 11$

$\Rightarrow 31^3 \bmod 11 * 31^3 \bmod 11 * 31^2 \bmod 11$

$\equiv 3 \bmod 11 * 3 \bmod 11 * 4 \bmod 11$

$\equiv (3*3*4) \bmod 11$

$\equiv 36 \bmod 11$

$\equiv 3$

Break down into smaller chunks. Use calculator. find mod value of smaller chunks multiply mod vals, and mod them all again.

* Digital Signature:

- Used for msg authentication, non repudiation and msg integrity.
- Not used for confidentiality.
- When a document is signed digitally, the signature is sent as a separate document, i.e. sender sends two documents → msg and signature

* Authentication Functions:

1. Message Encryption : (cipher text acts as authenticator)
2. MAC (Msg Authentication Code).
3. Hash Functions (H)

- **Hash functions:**
  - does not use a key.
  - Takes in variable size msg and generates fixed length o/p
  - Only i/p is the msg.

$H(M) = h$ → fixed length code 'h' (hash code)

M → variable length msg

Hash function AKA compression fn.

- **SHA**

	SHA-1	SHA224	256	384	512
Msg Digest size	160	224	256	384	512
Msg Size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
No. of Rounds	80	64	64	80	80