BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

YELAHANKA – BANGALORE - 64

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Semester: VII ECE	**Course: Cryptography**	**Subject Code: 17EC744**
Academic Year: 2020-21 Odd Sem	**Course coordinator: Mamatha K R** **Course handled by: MKR,JKB**	**SIE Marks:40** **CIE Marks:60**
	No. of Lecture hours /week: 3	**Total no. of Lecture:40 hours**

COURSE OUTCOMES :

Students will be able to		
CO1	Apply the basic, modern mathematical concepts and pseudorandom number generators required for encryption and decryption of data.	PO1
CO2	Analyse basic cryptographic algorithms to encrypt and decrypt the data	PO2
CO3	Design algorithms related to the concepts of authentication and protection of internet data.	PO3
CO4	Demonstrate the enriched knowledge of cryptographic concepts and web security in a team or individual	PO5,9,10,12

CONTENT:

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

YELAHANKA – BANGALORE – 64

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

STUDY MATERIAL

CRYPTOGRAPHY

17EC744

2020-21

Cryptography: Module 1

Introduction to Number Theory:

Divisibility:

A nonzero b divides a if $a = mb$ (m, a & $b \rightarrow$ integers)

i.e. b divides a if there is no remainder on division.

notation: $b \mid a$ also say <u>b is a divisor of a</u>

The +ve divisors of 24 are 1, 2, 3, 4, 6, 8, 12 & 24.

Properties of divisibility:

1. If $a \mid 1$ then $a = \pm 1$
2. If $a \mid b$ and $b \mid a$ then $a = \pm b$
3. If $b \neq 0$ divides 0
4. If $a \mid b$ and $b \mid c$, then $a \mid c$
5. If $b \mid g$ & $b \mid h$ then $b \mid (mg + nh)$ for arbitrary integers m & n.

Eg:- $2 \mid 6$ & $6 \mid 24$ then $2 \mid 24$.

$2 \mid 6$ & $2 \mid 10$ then $2 \mid (2 \times 6 + 3 \times 10)$ $= 2 \mid 12 + 30$ $= 2 \mid 42$

The Division Algorithm

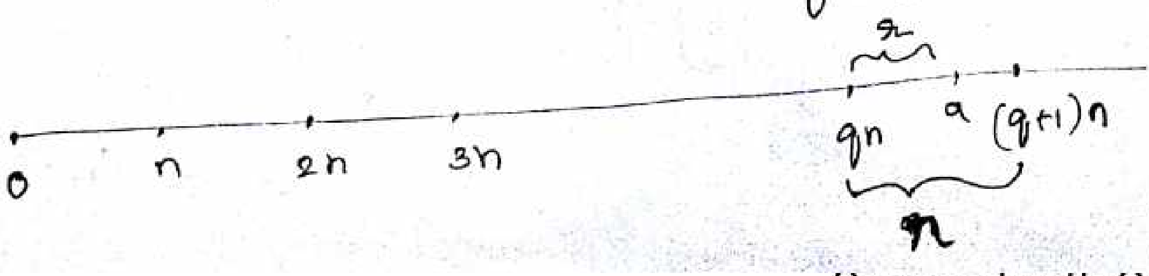
Given any +ve integer n & any nonnegative integer a, if we divide a by n, we get an integer quotient q & an integer remainder r that obey the following relationship:

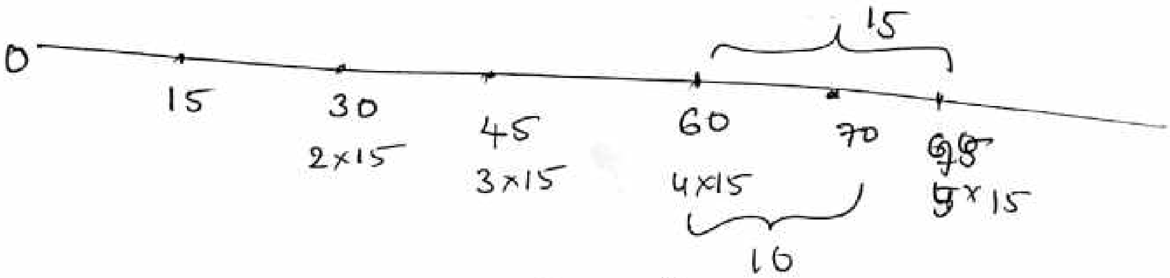
$$a = qn + r \qquad 0 \leq r < n$$

$$q = [a/n]$$

$$\begin{array}{r} 12 \\ n \to 2\overline{)25} \to a \\ \underline{24} \\ 1 \to r \end{array}$$

$25 = 12 \times 2 +$





$70 = (4 \times 15) + 10$

The Euclidean Algorithm

Simple procedure for determining the GCD (Greatest Common Divisor) of 2 +ve integers

$gcd(a,b) \rightarrow$ is the largest integer that divides both a & b. $g(0,0) = 0$. $\boxed{gcd(a,b) = \max[k, \text{ such that } k|a \text{ \& } k|b]}$

Eg:- $gcd(60, 24) = gcd(60, -24) = 12$

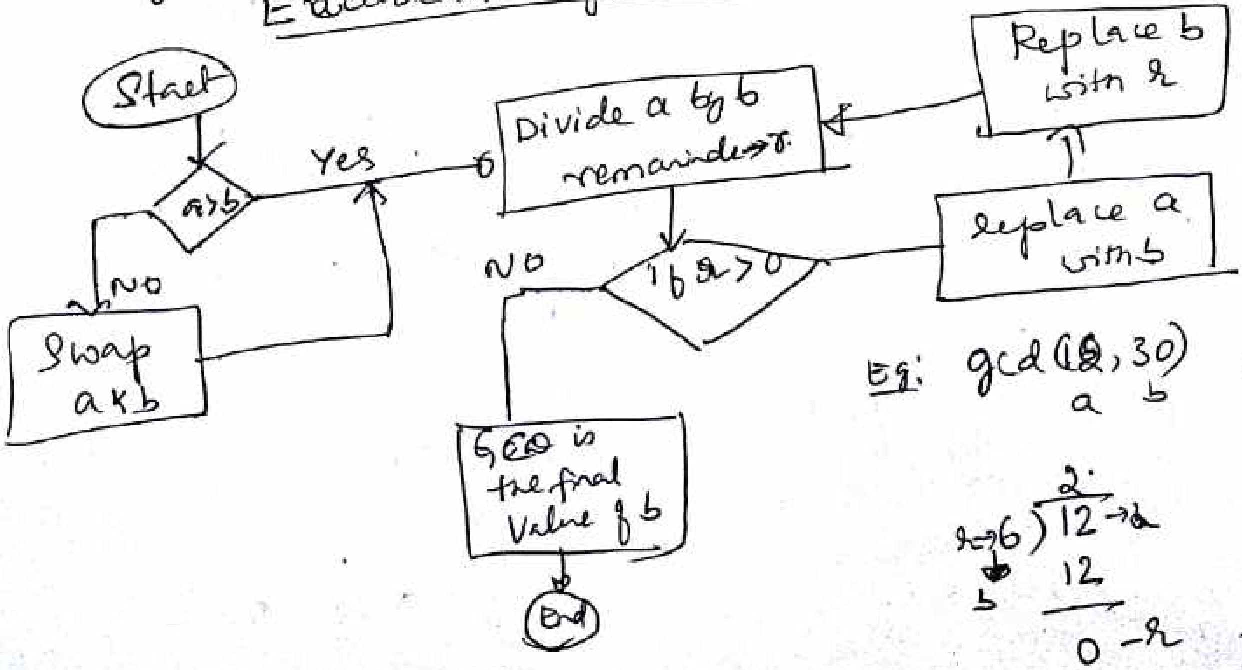
In general $gcd(a,b) = gcd(|a|, |b|)$
$= gcd(-a,-b) = gcd(-a,+b)$
$= gcd(+a,-b) = gcd(a,b)$

$gcd(a, 0) = |a|$

a & b are relatively prime if $gcd(a,b) = 1$

Eg: 8 & 15

Euclidean Algorithm



Eg: $gcd(12, 30)$
a b

$12 \overline{)30}$ a → 2, 24, 6 − r (r>0)

$6 \overline{)12}$ → 2, 12, 0 − r then 6 is gcd.

(2)

Dividing a & b by applying the division algorithm

$a = q_1 b + r_1 \quad 0 \le r_1 < b$

As $b > r_1$, $\quad b = q_2 r_1 + r_2 \quad 0 \le r_2 < r_1$

$r_1 = q_3 r_2 + r_3 \quad 0 < r_3 < r_2$

$\vdots$

$r_{n-1} = q_{n+1} r_n + 0$

$\boxed{d = \gcd(a, b) = r_n.}$

Modular arithmetic

The Modulus:

If a is an integer & n is a +ve integer, we define (a mod n) to be the remainder when a is divided by n. $\boxed{n \rightarrow \text{modulus}}$

Eg:- $11 \bmod 7 = 4$ $\Rightarrow 7\overline{)11}(1$, 7, 4 $\quad 7-4=3$

$-11 \bmod 7 = 3.$

$a = qn + r \quad 0 \le r < n \qquad r = a \bmod n.$

$q = \lfloor a/n \rfloor$

$a = \lfloor a/n \rfloor \times n + (a \bmod n) \rightarrow$ binary operator.

Congruent modulo n: Two integers a & b are said to be congruent modulo n, if $(a \bmod n) = (b \bmod n)$ ie $a \equiv b \pmod n \rightarrow$ congruence relation.

Eg:- $73 \equiv 4 \pmod{23}$ $\qquad 23\overline{)73}(3$, 69, 4

$(a-b)$ is multiple of ~~73~~

ie 69 is multiple of n. $\qquad 23\overline{)40}(0.17$, 23, 17

NOTE: if $a \equiv 0 \pmod n$ then ~~a/n~~ $n | a$

Properties of congruences

1. $a \equiv b \pmod n$ if $n \mid (a-b)$
2. $a \equiv b \pmod n$ implies $b \equiv a \pmod n$
3. $a \equiv b \pmod n$ & $b \equiv c \pmod n$ imply
   $a \equiv c \pmod n$

Eg:- $23 \equiv 8 \pmod 5$ as $23-8 = 15 = 05 \times 03$ = multiple of 5

$-11 \equiv 5 \pmod 8$ as $-11-5 = -16 = 8(-2)$ = multiple of 8.

$81 \equiv 0 \pmod{27}$ as $(81-0) = 81 = 27 \times 3$ = multiple of 27.

Modular arithmetic operations

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

Eg:- $11 \bmod 8 = 3$ & $15 \bmod 8 = 7$.

① LHS $= [11 \bmod 8 + 15 \bmod 8] \bmod 8 = [3+7] \bmod 8$

$= 10 \bmod 8$

$= 2$

RHS $= (a+b) \bmod n = (11+15) \bmod 8$

$= 26 \bmod 8$

$= 2$.

$$8 \overline{)26}^{\,3},\quad 24,\quad 2$$

② $(11-15) \bmod 8 = -4 \bmod 8 = 4$.

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

③ $[(11 \bmod 8) \times (15 \bmod 8)] \bmod 8 = 21 \bmod 8 = 5$

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

$$8 \overline{)21}^{\,2},\quad 16,\quad 5$$

$$8 \overline{)165}^{\,2},\quad 16,\quad 5$$

Exponentiation is performed by repeated multiplication, as in ordinary arithmetic

Eg:- $11^7 \bmod 13$.

$$\begin{array}{r} 9 \\ 13\,)\,121 \\ 117 \\ \hline 004 \end{array}$$

$11^2 \bmod 13 = 121 \bmod 13 = 4.$

$$\begin{array}{r} 1 \\ 13\,)\,16 \\ 13 \\ \hline 3 \end{array}$$

$11^4 = (11^2)^2 \bmod 13 = 4^2 \bmod 13 = 3.$

$11^7 = 11 \times 11^2 \times 11^4 = 11 \times 4 \times 3 = 132 \bmod 13$

$= 2$

$$\begin{array}{r} 10 \\ 13\,)\,132 \\ 130 \\ \hline 2 \end{array}$$

Properties of modular arithmetic

Commutative laws: $(w + x) \bmod n = (x + w) \bmod n$

$(w \times x) \bmod n = (x \times w) \bmod n$

Associtative Laws: $[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$

$[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n.$

Distributive law : $[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$

Identities : $(0 + w) \bmod n = w \bmod n$

$(1 \times w) \bmod n = w \bmod n$

Additive Inverse $(-w)$: For each $w \in Z_n$ there exists a z such that $(w + z) \equiv 0 \bmod n$

if $Z_n = \{0, 1, \ldots n-1\}$

Arithmetic Modulo 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

Multiplication modulo 8.

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

Additive & multiplicative Inverse modulo 8

w	$-w$	w^{-1}
0	0	–
1	7	1
2	6	–
3	5	3
4	4	–
5	3	5
6	2	–
7	1	7

Peculiarity of modular arithmetic :

1) If $(a+b) \equiv (a+c) \pmod{n}$ then $b \equiv c \pmod{n}$

Eg:- $(5+23) \equiv (5+7) \pmod{8}$ $\qquad 23 \equiv 7 \pmod{8}$

$28 \equiv 12 \bmod 8$ $\qquad (23-7) \equiv \cancel{8 \pmod{8}}$

6

$(28-12) = 16 = 8 \times 2$ multiple of 8.

2) If $(a \times b) \equiv (a \times c) \pmod{n}$ then $b \equiv c \pmod{n}$

If a is relatively prime to n.

Euclidean Algorithm Revisited

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

$$\gcd(55,22) = \gcd(22, 55 \bmod 22)$$
$$= \gcd(22, 11)$$
$$= 11.$$

$$22 \overline{)\,55\,}^{\,2}$$
$$\underline{44}$$
$$11$$

Euclidean algorithm:-

$a = q_1 b + r_1$

$b = q_2 r_1 + r_2$

$r_1 = q_3 r_2 + r_3$

$\vdots$

$r_{n-2} = q_n r_{n-1} + r_n$

$r_{n-1} = q_{n+1} r_n + 0$ then $\gcd(a,b) = r_n$.

Calculate

$r_1 = a \bmod b$

$r_2 = b \bmod r_1$

$r_3 = r_1 \bmod r_2$

$\vdots$

Recursive fn:-

```
Euclid (a,b)
  if (b=0) then return a;
  else return Euclid (b, a mod b);
```

Extended Euclidean Algorithm:

→ Important for later computations in the areas of finite fields & in encryption algorithms, such as RSA.

→ For given integers a & b, the extended Euclidean algorithm not only calculates the GCD d but also additional integers x & y that satisfy the following eqn.

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

Eg:- $\gcd(42,30) = 6$ $\qquad a = 42,\ b = 30.$

$42x + 30y \Rightarrow$ for different values of x & y, (all are divisible by 6)

$y \backslash x$	-2	-1	0	1	2
-2	-144	-102	-60	-18	24
-1	-114	-72	-30	12	54
0	-84	-42	0	42	84
1	-54	-12	30	72	114
2	-24	18	60	102	144.

all entries are divisible by 6.
as 42 & 30 are divisible by 6.

$\therefore$

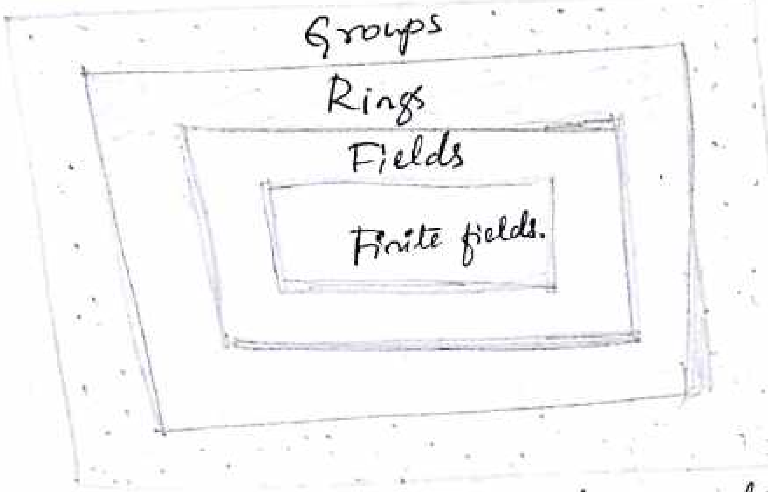
Algorithm:-

$$a = q_1 b + r_1$$
$$b = q_2 r_1 + r_2$$
$$\vdots$$
$$r_{n-1} = q_{n+1} r_n + 0$$

$$r_1 = ax_1 + by_1$$
$$r_2 = ax_2 + by_2$$
$$\vdots$$

GROUPS, RINGS & FIELDS

Groups, rings & fields are the fundamental elements of a branch of mathematics known as abstract algebra / modern algebra.



Fields are a subset of a larger class of algebraic structures called rings, which are in turn a subset of the larger class of groups

Finite fields are a subset of fields — with a finite no. of elements

Groups:— A group G denoted by {G, ·} is a set of elements with a binary operation denoted by · that associates to each ordered pair (a, b) of elements in G such that the following axioms are obeyed:

(A1) Closure: If a & b belong to G, then a·b is also in G

(A2) Associative: $a\cdot(b\cdot c) = (a\cdot b)\cdot c$ for all a, b, c in G.

(A3) Identity element: There is an element e in G such that $(a\cdot e) = e\cdot a = a$ for all a in G.

(A4) Inverse element: For each a in G, there is a' in G such that $a\cdot a' = a'\cdot a = e$.

Group
- Finite Group.
- Infinite Group.

Abelian Group :-

If it satisfies additional condition

(A5) Commutative : $a \cdot b = b \cdot a$ for all a, b in G.

Cyclic group :- $a^3 = a \cdot a \cdot a$

A group G is cyclic if every element of G is a power a^k of a fixed element $a \in G$.

A cyclic group will always be Abelian & may be finite or infinite.

Rings :- R denoted by $\{R, +, \times\}$ → set of elements with 2 binary op^s addition & multiplication.

R satisfies A1 to A5 (Abelian group)

(M1) closure under multiplication: If a & b belongs to R then ab also in R

(M2) Associativity of multiplication: $a(bc) = (ab)c$ for all a, b, c in R

(M3) Distributive laws : $(a+b)c = ac + bc$ for all a, b, c in R.

Ring is a set of elements in which we can do addition, multiplication & subtraction.

Ring is commutative if it satisfies additional condition

(M4) commutative of multiplication: $ab = ba$ for all a, b in R.

Integral Domain :- is a commutative Ring that obeys

(M5) Multiplicative Identity : $a1 = 1a = a$ for all a in R

(M6) No zero divisors : If a, b in R & $ab = 0$ then either $a = 0$ or $b = 0$.

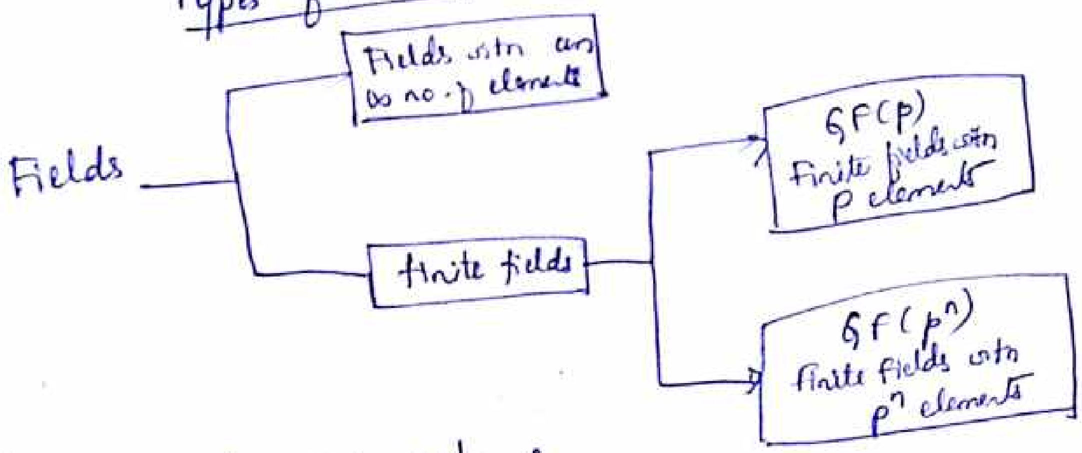
Fields :- denoted by $\{F, +, \times\}$ is a set of elements with 2 binary operations called addition & multiplication such that for all a, b, c in F the following axioms are obeyed. (A1 - M6)

(M7) Multiplicative inverse : there exists an element a^{-1} in F such that $a \cdot a^{-1} = (a^{-1})a = 1$.

Fields, Integral Domain, Commutative Ring, Ring, Abelian Group, Group

(A1) Closure under addition
(A2) Associativity of addition
(A3) Additive identity.
(A4) Additive Inverse
(A5) Commutativity of addition
(M1) Closure under multiplication
(M2) Associativity of multiplication
(M3) Distributive laws.
(M4) Commutativity of multiplication
(M5) Multiplicative Identity
(M6) No zero divisors
(M7) Multiplicative Inverse.

Types of Fields :-



* Infinite fields are not of particular interest in the context of cryptography.
* No. of elements in the field — Order of the field.
* If the order of a finite field is power of a prime p^n where $n \rightarrow$ +ve integer.
* It is generally written as GF(p^n) where
GF $\rightarrow$ Galois Field (Mathematician who studied Finite fields for the first time)

for $n = 1$, the finite field $GF(p)$ will have different structure than that for finite fields with $n > 1$.

$GF(2^n)$ fields are of particular cryptographic interest.

Arithmetic operations of finite field $\rightarrow GF(2)$

+	0	1
0	0	1
1	1	0

Addition (XOR)

×	0	1
0	0	0
1	0	1

Multiplication (AND)

w	$-w$	w^{-1}
0	0	–
1	1	1

Inverse.

Let Z_n is a set of integers $\{0, 1 \ldots n-1\}$

Any integer in Z_n has a multiplicative inverse if & only if that integer is relatively prime to n.

If n is prime then all of the non-zero integers in Z_n are relatively prime to n.

$Z_p \rightarrow$ finite field if it satisfies

Multiplicative inverse (w^{-1}) | for each $w \in Z_p$, $w \neq 0$, there exists a $z \in Z_p$ such that $w \times c \equiv 1 \pmod p$

If $(a \times b) \equiv (a \times c) \pmod p$ then $b \equiv c \pmod p$

multiply by a^{-1} on b.s.

$$(a^{-1}) \times a \times b \equiv ((a^{-1}) \times a \times c) \pmod p$$

$$\boxed{b \equiv c \pmod p}$$

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

addition mod 7

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

multiplicative mod 7.

Additive & multiplicative inverse modulo 7.

w	0	1	2	3	4	5	6
$-w$	0	6	5	4	3	2	1
w^{-1}	–	1	4	5	2	3	6

Finding the Multiplicative Inverse in $GF(p)$

Thro' table ⟶ for smaller values of 'p'.

For large values of p, it is not practical.

If a & b are relatively prime, then b has a multiplicative inverse modulo a. ie $\gcd(a,b) = 1$.

ie

using extended Euclidean algorithm:

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

If $\gcd(a,b) = 1$, then $ax + by = 1$.

From basic equalities of modular arithmetic

$[(ax \bmod a) + (by \bmod a)] \bmod a = 1 \bmod a$

$0 + (by \bmod a) \bmod a = 1 \bmod a$

$\therefore \; by \bmod a = 1.$

then if $by \bmod a = 1$, then $b^{-1} = y$.

Eg:- $a = 1759$, $b = 550$

prime no.

Solu of the eqn $1759x + 550y = d$

$y = 355$ & $b^{-1} = 355$

$550 \times 355 \mod 1759 = 195250 \mod 1759$

$= 1$.

Polynomial Arithmetic

3 classes of polynomial arithmetic are:

1. Ordinary polynomial arithmetic, using the basic rules of algebra.
2. Polynomial arithmetic in which the arithmetic on the coeffts is performed modulo p ie coeffts are in GF(p)
3. Polynomial arithmetic in which the coeffts are in GF(p) & the polynomials are defined modulo a polynomial m(x) whose highest power is some integer n.

Ordinary Poynomial Arithmetic: (+, −, ×)

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

$$= \sum_{i=0}^{n} a_i x^i \qquad a_n \neq 0.$$

if $a_n = 1 \longrightarrow$ constant polynomial. (zero degree)

$f(x) = x^3 + x^2 + 2 \qquad g(x) = x^2 - x + 1.$

then $f(x) + g(x) = x^3 + 2x^2 - x + 3$

$f(x) - g(x) = x^3 + x + 1$

$f(x) \times g(x) = x^5 - \cancel{x^4} + \cancel{x^3} + \cancel{x^4} - x^3 + x^2 + 2x^2 - 2x + 2$

$= x^5 + 3x^2 - 2x + 2$

$f(x)/d(x)$

$x^3+x+1 \overline{) \cancel{x^7} + \cancel{x^5} + \cancel{x^4} + x^3 + x + 1} \; (x^4 + 1 \; q'$

$\cancel{x^7} + \cancel{x^5} + \cancel{x^4}$

$x^3 + x + 1$

$x^3 + x + 1$

0 – remainder.

GCD of polynomial

The polynomial $c(x)$ is said to be the gcd of $a(x)$ & $b(x)$ if the following are true:

1. $c(x)$ divides both $a(x)$ & $b(x)$
2. Any divisor of $a(x)$ & $b(x)$ is a divisor of $c(x)$.

$gcd[a(x), b(x)] \Rightarrow$ is the polynomial of max degree that divides both $a(x)$ & $b(x)$

$$gcd[a(x), b(x)] = gcd[b(x), a(x) \bmod b(x)]$$

Euclidean Algorithm:-

$a(x) = q_1(x)\, b(x) + r_1(x)$ $\qquad r_1(x) = a(x) \bmod b(x)$

$b(x) = r_1(x)\, q_2(x) + r_2(x)$ $\qquad r_2(x) = b(x) \bmod r_1(x)$

$r_1(x) = r_2(x)\, q_3(x) + r_3(x)$

$\vdots$

$r_{n-1}(x) = q_{n+1}(x)\, r_n(x) + 0$ then

$$d(x) = gcd[a(x), b(x)] = r_n(x)$$

Repetitive app^n of division algorithm.

assumes $\rightarrow$ degree of $a(x)$ > degree of $b(x)$

Eg:- Find gcd $[a(x), b(x)]$ for $a(x) = x^6+x^5+x^4+x^3+x^2+x+1$ & $b(x) = x^4+x^2+x+1$.

$$
\begin{array}{r l}
x^4+x^2+x+1 \,\big) & \cancel{x^6}+x^5+\cancel{x^4}+\cancel{x^3}+\cancel{x^2}+x+1 \,\big(\, x^2+x \rightarrow q_1(x) \\
& x^6+x^4+x^3+x^2 \\
\hline
& x^5 + x + 1 \\
& x^5+x^3+x^2+x \\
\hline
& x^3+x^2+1 \rightarrow \text{remainder } r_1(x)
\end{array}
$$

divide $b(x)$ by $r_1(x)$

$$
\begin{array}{r l}
x^3+x^2+1 \,\big) & \cancel{x^4}+x^2+\cancel{x}+1 \,\big(\, x+1 \\
& x^4+x^3+x \\
\hline
& x^3+x^2+1 \\
& x^3+x^2+1 \\
\hline
& 0 \rightarrow
\end{array}
$$

rem $r_1(x) = 0$ then gcd $[a(x), b(x)]$ is

gcd $(a(x), b(x)] = r_1(x) = \underline{x^3+x^2+1}$.

Finite fields of the form $GF(2^n)$

finite fields → order is p^n where p → prime no, n → any +ve integer.

All the axioms for a field are satisfied.

$GF(2^n)$ → set of p^n elements, $n>1$.

Consider an eg: of encryption algorithm that operates on 8 bits then range of integers - represent is 0 to 255. Z_{256} (arithmetic modulo 256) As 256 is not a prime no. But closest prime no. Set of integers is not a field. Set of integers of Z_{251} is a field. 251 is a prime & hence set of integers

Scanned with CamScanner

GF(2)

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

addition

multiplication

		00	01	10	11
0	00	00	00	00	00
1	01	00	01	10	11
x	10	00	10	11	01
$x+1$	11	00	11	01	10
		0	1	x	$x+1$

2
2

Polynomial.

$p(x) = x^2 + x + 1$

00	0	0	0
01	1	1	1
10	2	$0+x+0$	x
11	3	$0+x+1$	$x+1$

$x^2+x+1=0$ $\quad$ $x^2+x=1$

$x^2 = x+1$

$(x+1)(x+1) = x^2+x+x+1$

$= x^2+1$

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Inverse

w	$-w$	w^{-1}
0	0	-
1	1	1
2	2	3
3	3	2

Scanned with CamScanner

Arithmetic in $GF(2^3)$

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
000 0	0	1	2	3	4	5	6	7
001 1	1	0	3	4	5	4	7	6
010 2	2	3	0		6	7	4	5
011 3	3	2	1	0	7	6	5	4
100 4	4	5	6	7	0	1	2	3
101 5	5	4	7	6	1	0	3	2
110 6	6	7	4	5	2	3	0	1
111 7	7	6	5	4	3	2	1	0

Addition

$$\begin{array}{r} 101 \\ 001 \\ \hline 100 \end{array} \qquad \begin{array}{r} 110 \\ 001 \\ \hline 111 \end{array}$$

mod 2

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
000 0	0	0	0	0	0	0	0	0
001 1	0	1	2	3	4	5	6	7
010 2	0	2	4	6				
011 3	0	3	6					
100 4	0	4	3					
101 5	0	5	1					
110 6	0	6	7					
111 7	0	7	5					

Multiplication.

$$\begin{array}{r} 010 \\ 011 \\ \hline 010 \end{array}$$

$$\begin{array}{rl} 100 & -4 \\ 010 & -2 \\ \hline \cancel{000} & \\ 110 & -6 \end{array}$$

AES — Adv. Encryption std. uses arithmetic in the finite field $GF(2^8)$ with irreducible polynomial

$m(x) = x^8 + x^4 + x^3 + x + 1$

$f(x) = x^6 + x^4 + x^2 + x + 1$

$g(x) = x^7 + x + 1$

Module 2

Classical Encryption Techniques

①

Symmetric Encryption or conventional encryption was the only type of encryption in use prior to the development of Public key Encryption in the 1970's.

DES & AES are most widely used Symmetric Ciphers.
Data Encryption Std. Advanced Encryption Std.

Basic Terms:

Plain Text : original message

Cipher Text : coded message

Enciphering / encryption : Process of converting plaintext to cipher text.

Deciphering / Decryption : Restoring the plaintext from the ciphertext.

Cryptography : Area of study of different Schemes of Encryption

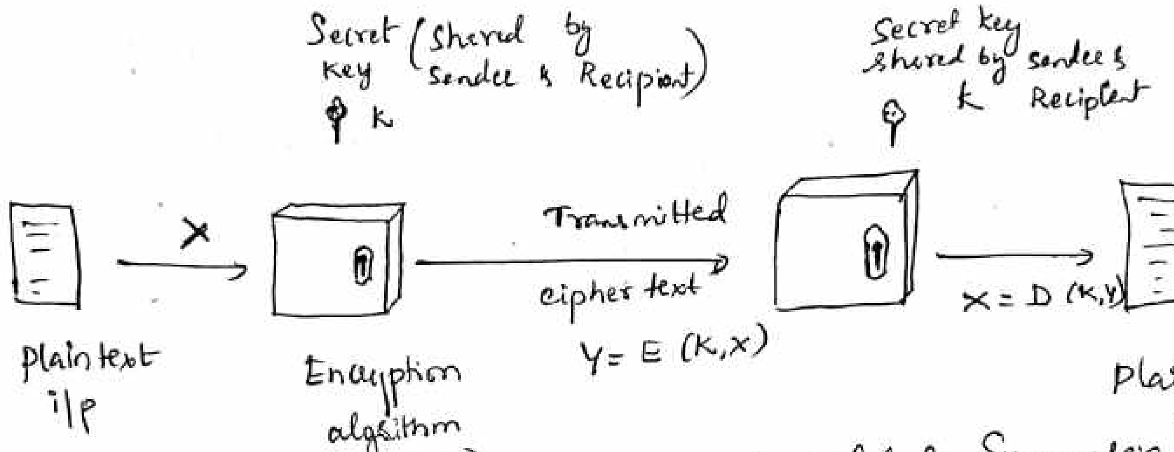
Cipher : cryptographic System.

Cryptanalysis : Techniques used for deciphering a message without any knowledge of the enciphering details.
→ also called as 'breaking the code'

Areas of cryptography & cryptanalysis together are called cryptology

Symmetric Cipher Model

②



Simplified model of Symmetric Encryption

A symmetric encryption scheme has 5 ingredients:

① Plaintext: Original intelligible message or data ie fed into the algorith as i/p.

② Encryption Algorithm: Various substitutions and transformations on the plaintext.

③ Secret Key: is also i/p to the encryption algorithm. The Key is a value independent of the plaintext & of the algorithm. The algorithm will produce a different o/p depending on the specific key being used at the time.

④ Cipher Text: This is the scrambled message produced as o/p. It depends on the plaintext & the secret key.

⑤ Decryption algorithm: Encryption algorithm run in reverse. It takes the ciphertext & the secret key & produces the original plaintext

④ There are 2 requirements for secure use of conventional encryption.

1. We need a strong encryption algorithm. The opponent who knows the algorithm & has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key.

2. Sender & receiver must have obtained copies of the secret key in a secure fashion & must keep the key secure. If someone can discover the key & knows the algorithm, all commn using the key is readable.

Model of Symmetric Cryptosystem

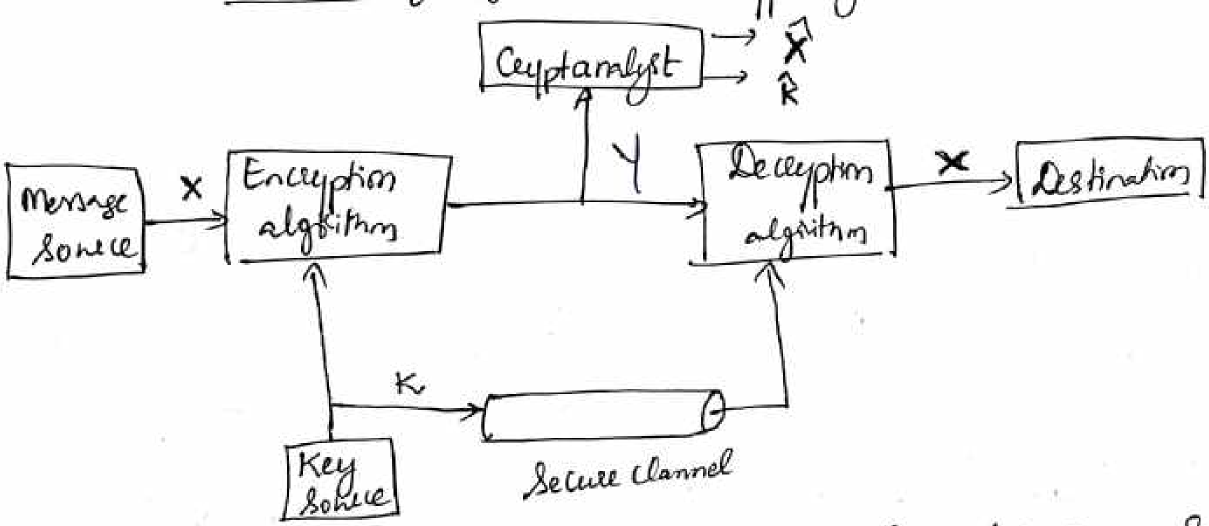


Figure above shows the essential elements of a symmetric encryption scheme. A source produces a message in plaintext, $X = [X_1, X_2 \ldots X_m]$ The m elements of X are letters in some finite alphabet. (Traditionally 26 capital letters) Nowadays binary alphabet {0,1} is typically used.

A key $K = [K_1, K_2 \ldots K_J]$ is generated. If key is generated at the message source, it must also to be provided to the destination by means of some secure channel. With the message X & key K as i/p, the encryption algorithm forms the ciphertext $Y = [Y_1\ Y_2 \ldots Y_N]$

We can write $Y = E(K, X)$

The inteded receiver, in possession of the key, is able to invert the transformation:
$$X = D(K, Y)$$
An opponent, observing Y but not having access to K or X may attempt to recover X or K or both X & K. It is assumed that the opponent knows the Encryption (E) & decryption (D) algorithms.

If the opponent is interested in only the particular message then focus is to recover X by generating plaintext estimate $\hat{X}$. If he is interested in being able to read future messages as well, an attempt is made to recover K by generating an estimate $\hat{K}$.

Cryptography:-

Cryptographic systems are characterized along 3 independent dimensions:

① **The type of operations used for transforming plaintext to cipher text:** Substitutions or Transposition

Substitution: In which each element of the plaintext is mapped into another element (is replaced)

Transposition: the elements in the plain text are rearranged.

② **The no. of keys used:** If both Sender & Receiver use the same key then the system is → Symmetric (Single key) or (Secret key). If the sender & receiver uses different keys, then the system is asymmetric, 2-key & public key encryption.

③ **The way in which the plaintext is processed:**

↳ Block cipher → block of elements at a time is processed.

↳ Stream cipher → processes i/p elements continuously producing one element at a time as o/p.

3

Cryptanalysis and Brute-Force Attack

Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single cipher text. There are two general approaches to attacking a conventional encryption scheme:

• **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–cipher text pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

• **Brute-force attack:** The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

Table 2.1 summarizes the various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst.

Table 2.1 Types of Attacks on Encrypted Messages

Types of Attack	Known to cryptanalyst
Cipher text Only	• Encryption algorithm • Cipher text
Known Plaintext	• Encryption algorithm • Cipher text • One or more plaintext–cipher text pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Cipher text • Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key
Chosen Cipher text	• Encryption algorithm • Cipher text • Cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Cipher text • Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key • Cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Substitution Techniques

In this Techniques the letter of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits then substitution involves replacing plain text bit patterns with cipher text bit patterns.

- Caesar Cipher → [used for short length msg and easy to attack].

↳ Replacing each letter of the alphabet with the letter standing 3 place further down the example:

Plain Text : Meet me after the ~~pargam~~ School.

Cipher : PHHWPHDIWH. . . .

Plain Text : a b c d - - - - z

clipher : D E F G - - - C

Key :- Numerical

$1 \le k \le 26$

$C = (P+K) \bmod 26$

P.T= HELLO

K= 4

~~$C(H) = (8+5+12+12+15) \bmod 26$~~

$C(H) = (8+4) \bmod 26$

$= 12 \bmod 26$

$= 12(L)$.

Cipher of H is L

A→1	Q 17		
B→2	R 18		
C→3	S 19		
D→4	T 20		
E→5	U 21		
F→6	V 22		
G 7	W 23		
H 8	X 24		
I 9	Y 25		
J 10	Z 26		
K 11			
L 12			
M 13			
N 14			
O 15			
P 16			

its p...

(4)

Similarly $C(E) = (5+4) \bmod 26$

$= 9 \bmod 26 = 9 = I$

for E, cipher text character is I.

if Plain Text = ZOO

$C(Z) = (26+4) \bmod 26$

$= 30 \bmod 26 = 4 = D$.

- Playfair cipher → It use 5X5 matrix of letter constructed using a keyword.

P.T = HELLO

key = Network

CT = ?

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

5×5 → 25 Letter (I/J merge).

[IGNORE repeated character Present in key]

→ After entering key into the matrix, Remaining box should be entered which alphabet is not present in key.

→ Divide the P.T. To pair of letters.

→ Differentiate repeated letters in the pair with dummy letter.

→ If pair of plain Text letters are in same Row then replace them with right most letter

→ If the plaintext letters are in same co
replace with beneath letters.

→ if P.T. letters are in diffrent Row and Column then replace : with the character which is collumn corresponding to raw (diagonal position)

HE | LL | O world

wo rl dx

HE | LX | LO,

HE | L - | LO wo rl d

HE → WF

LX → UP

LO → NS

HELXLO → WFUPNS

example.

P-T. = BALLOON

Key = NETWORK

CT = ?

BA | LL | OO | N

BA | LX | LO | ON

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

BA → CB

LX → UP

LO → NS

ON → NE

BALXLOON → CBUPNSNE

HILL CIPHER ⑤

This algorithm developed by the mathematician Laster Hill in 1929

[cipher text = (Plaintext × Key) mod 26]

encryption

Key = HILL

Plain Text = CIPHER

A − 0	L → 11
B → 1	M → 12
C → 2	N → 13
D → 3	O → 14
E → 4	P → 15
F → 5	Q → 16
G → 6	R → 17
H → 7	S → 18
I → 8	T → 19
J → 9	U → 20
K → 10	V → 21
	W → 22
	X → 23
	Y → 24
	Z → 25

$$\begin{pmatrix} H & I \\ L & L \end{pmatrix}\begin{pmatrix} C \\ I \end{pmatrix} = \left[\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}\begin{pmatrix} 2 \\ 8 \end{pmatrix}\right] \bmod 26$$

$$= \begin{pmatrix} 78 \\ 110 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 0 \\ 6 \end{pmatrix} = \begin{pmatrix} A \\ G \end{pmatrix}$$

$$\left[\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}\begin{pmatrix} 15 \\ 7 \end{pmatrix}\right] \bmod 26 = \begin{pmatrix} 105 + 56 \\ 165 + 77 \end{pmatrix} = \begin{pmatrix} 161 \\ 242 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 5 \\ 8 \end{pmatrix} = \begin{pmatrix} F \\ I \end{pmatrix}$$

Ans:- Cipher text = A G F I I X

Decription.

[Plaintext = (Cipher text × Key^{-1}) mod 26]

$$\text{Key}(K) = \begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix}$$

$$\text{Key}^{-1}(K) = \frac{\text{adj}(K)}{}$$

Find −ve no. mod.

$(-51) \bmod = 10 = 9$

$n = qm + R$

$\Rightarrow -51 = -6.10 + R$

ting feature ... perty offers ...

J)

$$K^{-1} = \frac{\begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix}}{77 - 88}$$

$$= \frac{\begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix}}{-11} = \frac{1}{-11}\begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix}$$

how to find out $\frac{1}{-11}$ mod 26 :

$-11 \bmod 26 = 15$

$15 \times x = 1 \bmod 26$

$\Rightarrow x = 1 \bmod 26 / 15 = 7$

	add 1	divide by 15
26	27	1.8 X
52	53	3.53 X
78	79	5.26 X
104	105	7 ✓

$$\therefore K^{-1} = 7\begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \bmod 26$$

$$= 7\begin{bmatrix} 11 & 18 \\ 15 & 7 \end{bmatrix} = \begin{bmatrix} 77 & 126 \\ 105 & 49 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 25 & 22 \\ 1 & 23 \end{bmatrix}$$

$$P.T = [CT \times K^{-1}] \bmod 26$$

$$= \begin{bmatrix} 0 \\ 6 \end{bmatrix}\begin{bmatrix} 25 & 22 \\ 1 & 23 \end{bmatrix} \bmod 26$$

$$= \begin{pmatrix} 132 \\ 138 \end{pmatrix} \bmod 26 = \begin{bmatrix} 2 \\ 8 \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix}$$

Eg :- Hill cipher $K = \begin{pmatrix} 3 & 7 \\ 15 & 12 \end{pmatrix}$ $P = (H\ I)$

$C = ?$

$C = PK \bmod 26$.

<u>Encryption</u>:- $P = (H\ I) = (7, 8)$

$\therefore \quad C = [7, 8] \begin{bmatrix} 3 & 7 \\ 15 & 12 \end{bmatrix} = [11, 15]$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad L \quad P.$

$C = (L, P)$

<u>Decryption</u>; $P = CK^{-1} \bmod 26$

$K^{-1} = \det K^{-1} (-1)^{i+j} D_{ij}$

$3 \times 12 - 7 \times 15 = 36 - 105 = -69. \bmod 26$

$= -17 \bmod 26$

$\det K = 9 \bmod 26.$

$\boxed{\det K^{-1} = 3}$

$3 \begin{bmatrix} 12 & -7 \\ -15 & 3 \end{bmatrix} = \begin{bmatrix} 36 & -21 \\ -45 & 9 \end{bmatrix} = \begin{bmatrix} 10 & 5 \\ 7 & 9 \end{bmatrix}$

$P = [11, 15] \begin{bmatrix} 10 & 5 \\ 7 & 9 \end{bmatrix} = \begin{pmatrix} 110 + 105 & 55 + 135 \end{pmatrix}$

$= (215, 190) = (7, 8)$

$= (H, I).$

Scanned with CamScanner 30

Eg:- ATTACK IS TONIGHT

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

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

$$C = PK \bmod 26 = \begin{pmatrix} 0 & 19 & 19 \\ 0 & 2 & 10 \\ 8 & 18 & 19 \\ 14 & 13 & 8 \\ 6 & 7 & 19 \end{pmatrix} \begin{pmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{pmatrix} \bmod 26$$

$C_1 =$

$= 0 \times 3 + 19 \times 20 + 19 \times 9 = 551 \bmod 26$
$= 05$ F

$= 0 \times 10 + 19 \times 9 + 19 \times 4 = 13$ N

$= 0 \times 20 + 19 \times 17 + 19 \times 17 = 22$ W.

$\boxed{\text{ATT} \longrightarrow \text{FNW.}}$

Decryption:- $P = CK^{-1} \bmod 26$.

Det K^{-1} ? $\quad \det K = 3(9 \times 17 - 4 \times 17) - 10(20 \times 17 - 9 \times$

$+ 20(20 \times 4 - 9 \times 9$

$= (-1635) \bmod 26 = (-23) \bmod 26$

$= 3$

$(\det K)^{-1} = \frac{1}{3} \bmod 26 = 09.$

$$K^T = \begin{bmatrix} 3 & 20 & 9 \\ 10 & 9 & 4 \\ 20 & 17 & 17 \end{bmatrix}$$

$$\text{minor} = \begin{bmatrix} 9 \times 17 - 4 \times 17 & +90 & -10 \\ +187 & -129 & 349 \\ -1 & -178 & -173 \end{bmatrix}$$

$+85$

$\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$

1

2

$$= \begin{bmatrix} 85 & -90 & -10 \\ -187 & -129 & 369 \\ -1 & 78 & -173 \end{bmatrix}$$

$$K^{-1} = (\text{Det } K^{-1}) \times \text{adj}(K)$$

$$= 9 \times \begin{bmatrix} \; \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 765 & -810 & -90 \\ -1683 & -1161 & 3141 \\ -9 & 702 & -1557 \end{bmatrix} \bmod 26$$

$$K^{-1} = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$$

$$P = C K^{-1} \bmod 26$$

$$= \begin{bmatrix} 5 & 13 & 22 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} K^{-1} \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 0 & 19 & 19 \end{bmatrix} = ATT$$

Polyalphabetic Ciphers:

These features are common:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

Vigenere Cipher:

$C = E(K, P) = (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26 \ldots\ldots$

Key length should be same as plaintext.

Eg:- " deceptive "

Key: d e c e p t i v e d e c e p t i v e d e c e p t i v e

plain text: w e a r e d i s c o v e r e d s a v e y o u r s e l f

Cipher text. Z I C $\boxed{\text{V T W}}$ Q N G R Z G $\boxed{\text{V T W}}$ A V Z H C Q Y G L M G J

Key	3	4	2	4	15	19	08	21	4	3	4	2	4	15	19	8	21	4	. . .	4
plain text	22	4	0	17	4	3	8	18	2	14	21	4	17	4	3	18	0	21		5
cipher text	25	8	2	21	19	22	16	13	6	17	25	6	21	19	22	0	21	25	. . .	9

$C_i = p_i + (k_{i \bmod m}) \bmod 26$

$p_i = (C_i - k_{i \bmod m}) \bmod 26$

Vernam Cipher:- AXT Eggn Gilbert Vernam (1918)

Data bits rather than letters. $C_i = p_i \oplus k_i$

XOR opn.

$p_i = C_i \oplus k_i$

One-time Pad — perfect secrecy — Cryptosystem.

An army Signal Corp Officer, Joseph Mauborgne proposed an improvement to the Vernam → Security.

One key for one message & discard. → Random key which is as long as the message with no repetition.

Limitations:- (difficulties)

① There is the practical problem of making large quantities of random keys

② Key distribution & protection. (plain text is large then key is also)

"XMCKL"

H e l l o
7 4 11 11 14
Key 23 12 2 10 11

30 16 13 21 25
4 16 13 21 25
E Q N V Z. — cipher text.

Transposition Techniques

a sort of Permutation on the plaintext letters.

Simplest — rail fence technique.

meet me after the party

m e m a t r h p r y → meeting.
e t e f e t e a t

cipher: MEMATRHPRYETEFETEAT

Encryption:-

Complex one:-

Key: 4 3 1 2 5 6 7.

	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

cipher:- TTNAAPTMTSUOAODWCOIXKNLYPETZ

Double Transposition

Key: 4 3 1 2 5 6 7

	4	3	1	2	5	6	7
I/p:	t	t	n	a	a	p	t
	m	t	s	u	o	a	o
	d	w	c	o	i	x	k
	n	l	y	p	e	t	z

o/p.: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ → Difficult to cryptanalysis.

Steganography :-

Not encryption → plaintext message is hidden

Time consuming to construct, arrangement of words or letters which hides the real message.

Eg:- every first letter of each word

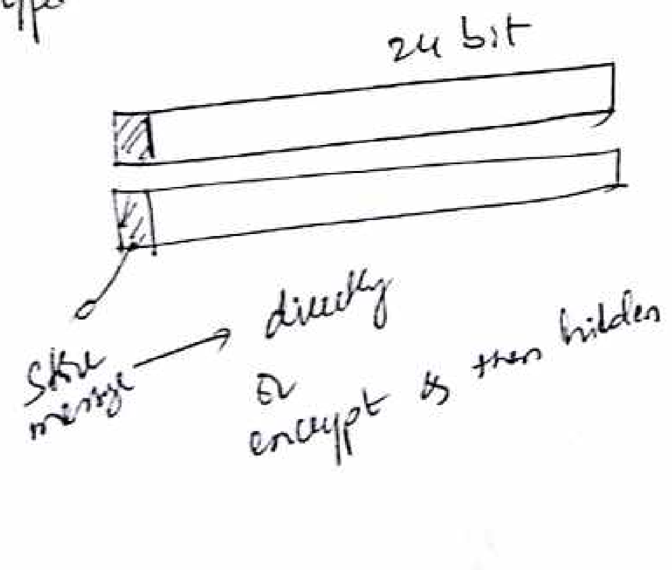
Various techniques :-

① Character Marking : Selected letters of printed or typewritten text are overwritten in pencil. (marks are not visible unless the paper is ~~hidden~~ held at an angle to bright light)

② Invisible ink :

③ Pin punctures :

④ Typewriter correction ribbon :



Traditional Block Cipher Structure

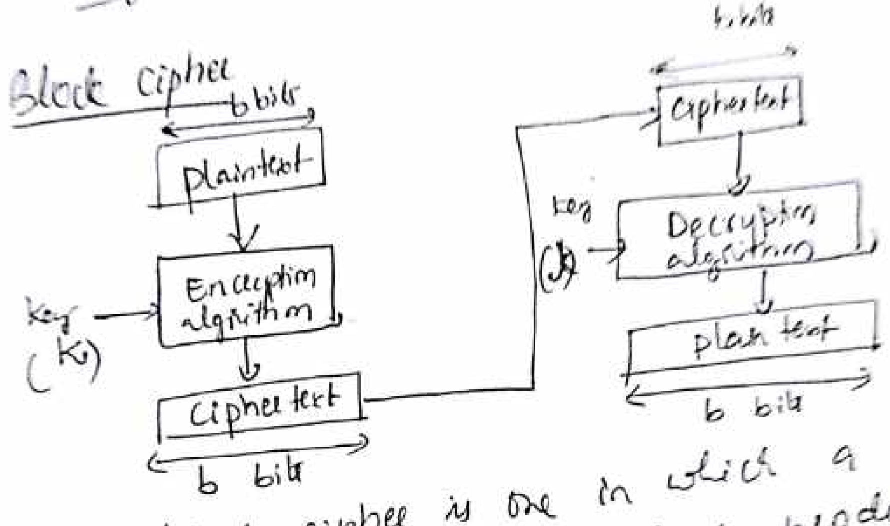
Stream & Block ciphers

Stream cipher using algorithmic bit stream generator

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.

Eg:- Vigenere, Vernam & one time pad using stream

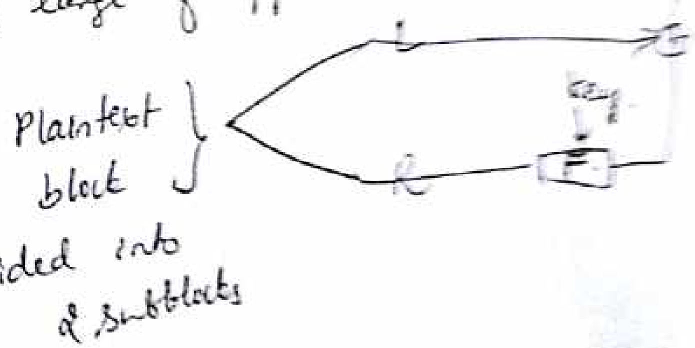
Block cipher

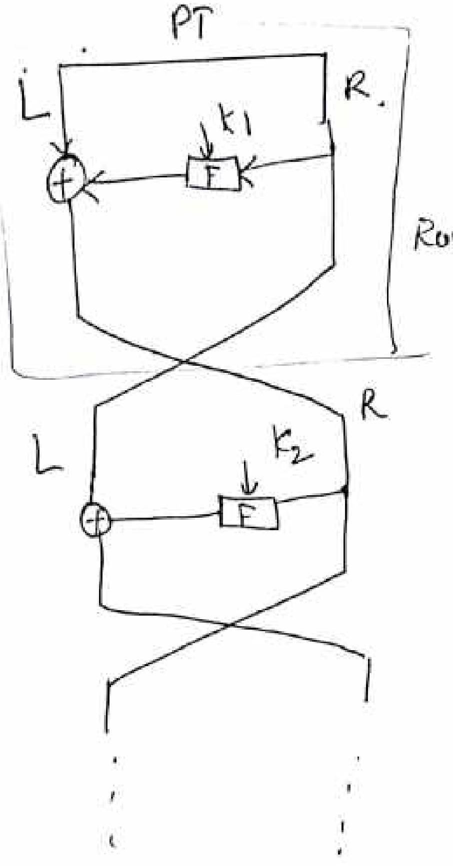


A block cipher is one in which a block of plaintext is treated as a whole & used to produce a ciphertext block of equal length.

Block ⟶ has broader range of appl^ns than stream cipher.

Feistel Cipher :-





all f^ns are applied on right hand blocks & result is EXORed with left hand block
DES → follows the Feistel Structure.

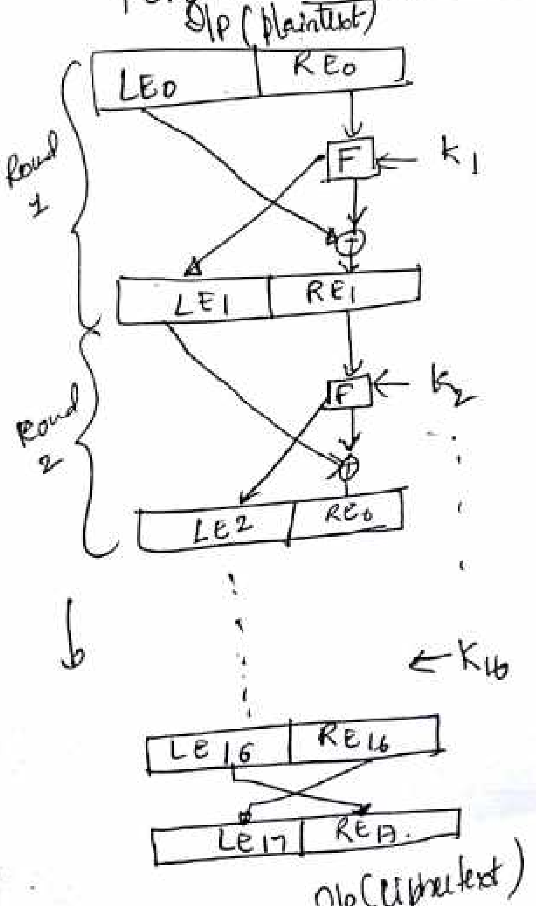
<u>Design principles</u>

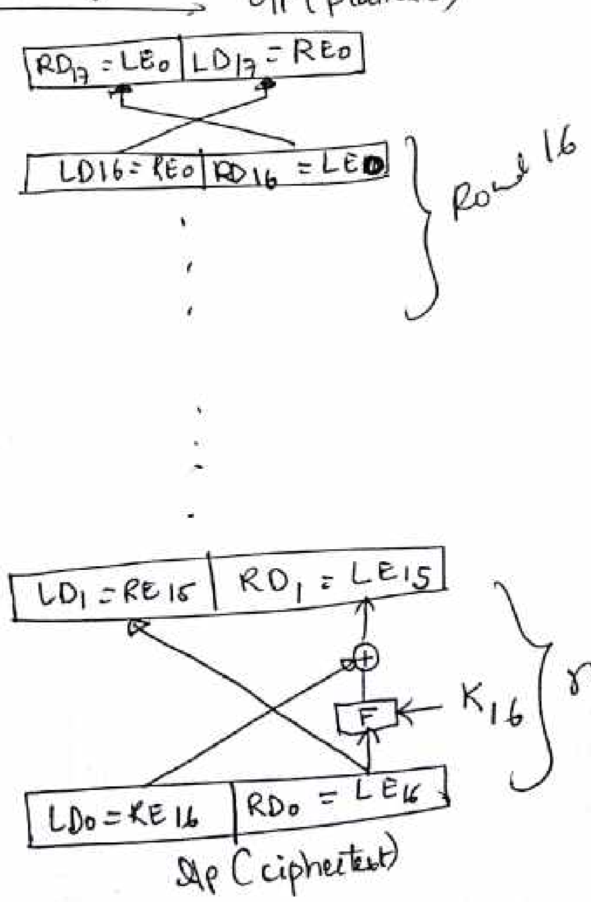
1. Block size ↑ → security ↑. but speed ↓.
2. Key size — " —
3. <u>No. of rounds</u> } → 16 rounds.
4. <u>Subkeys const</u> } → algorithm
5. <u>Round function</u> } → ↑ complexity ↑
6. ~~plain text in 2 equal halves~~

2 other considerations

1. Fast Encryption / decryption.
2. Ease of analysis.

~~Feistel~~ Feistel Decryption Algorithm (16 rounds).





Algorithm:- $LE_{16} = RE_{15}$

$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$

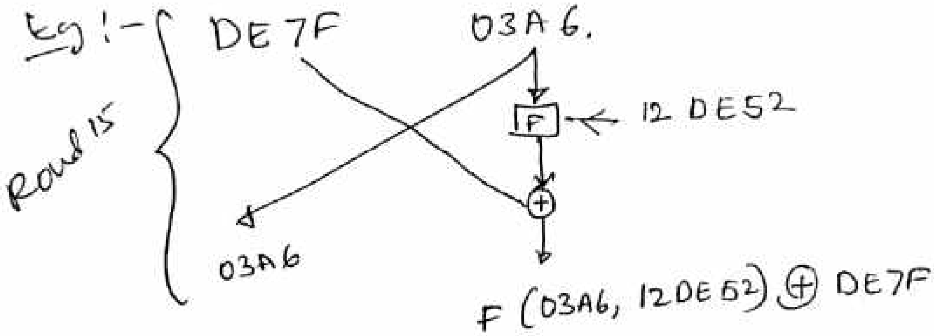
on decryption side:

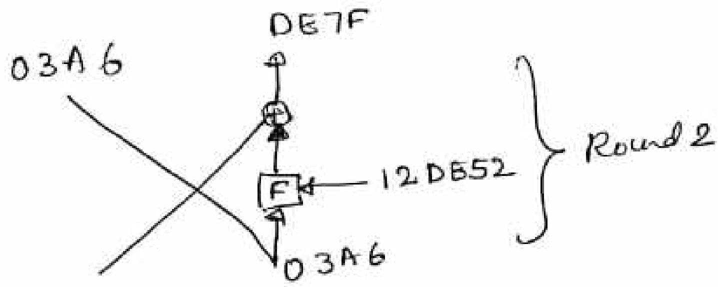
$LD_1 = RD_0 = LE_{16} = RE_{15}$

$RD_1 = LD_0 \oplus F(RD_0, K_{16})$

$= RE_{16} \oplus F(RE_{15}, K_{16})$

Eg:-





F (03A6, 12DE52) ⊕ DE7F

DES: Data Encryption Standard.

Until the introduction of AES in 2001, DES was the most widely used encryption scheme

DEA (Data Encryption Algorithm) ⟶ DES (1977) by National Bureau of Stds.

DES ⟶ dominant Symmetric encryption algorithm, especially in financial appⁿ.

DES Encryption

The overall scheme for DES encryption is illustrated in Figure 4.5. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.[8]

Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*.

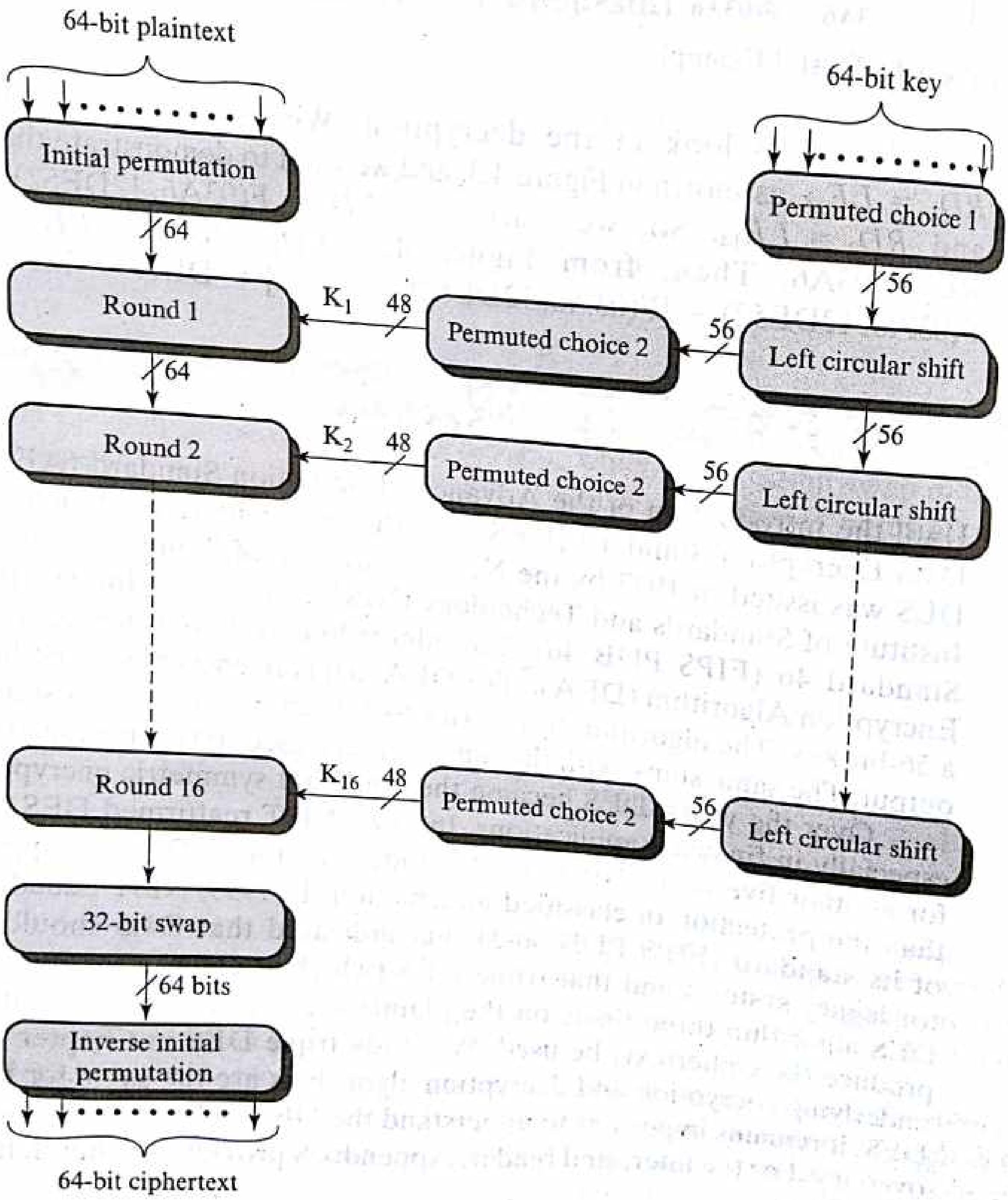


Figure 4.5 General Depiction of DES Encryption Algorithm

- **Key size:** Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
- **Number of rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.
- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function F:** Again, greater complexity generally means greater resistance to cryptanalysis.

There are two other considerations in the design of a Feistel cipher:

- **Fast software encryption/decryption:** In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.
- **Ease of analysis:** Although we would like to make our algorithm as difficult as possible to cryptanalyze, there is great benefit in making the algorithm easy to analyze. That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength. DES, for example, does not have an easily analyzed functionality.

FEISTEL DECRYPTION ALGORITHM The process of decryption with a Feistel cipher

Now we would like to show that the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})$$

On the decryption side,

$$\begin{aligned} LD_1 &= RD_0 = LE_{16} = RE_{15} \\ RD_1 &= LD_0 \oplus \mathrm{F}(RD_0, K_{16}) \\ &= RE_{16} \oplus \mathrm{F}(RE_{15}, K_{16}) \\ &= [LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})] \oplus \mathrm{F}(RE_{15}, K_{16}) \end{aligned}$$

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $RE_{15} \| LE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the ith iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

Rearranging terms:

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i) = RE_i \oplus \mathrm{F}(LE_i, K_i)$$

Thus, we have described the inputs to the ith iteration as a function of the

Avalanche property of DES Module 3

Change in 1 bit of i/p ⟶ many bits change in o/p.

AES :-

2001 → NIST, Symmetric block cipher
replaced DES → many appl^ns
It is complex than the RSA (public key cipher)

AES Structure :-

plaintext block size → 128 bits / 16 bytes
Key length — 16, 24 & 32 bytes (128, 192 & 256 bits)
we have AES-128, AES-192 & AES-256

128 bit block plaintext is depicted as 4x4 sq. matrix of bytes.

Cipher — N rounds. → depends on Key length.
16 bytes → 10 rounds
24 bytes → 12 rounds & 32 bytes → 14 rounds

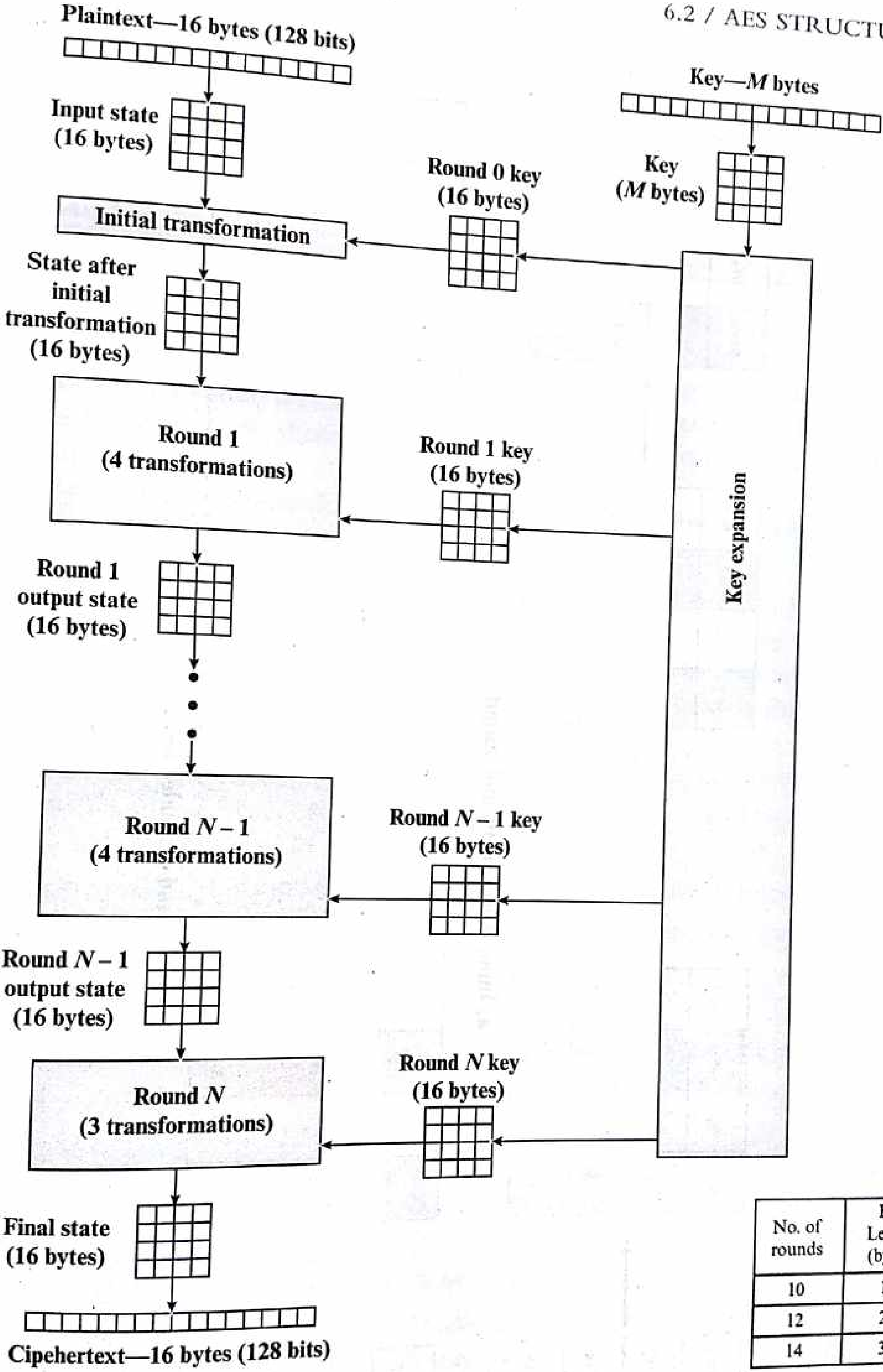
First N-1 rounds consists of $\boxed{\text{4 distinct transformations.}}$
→ SubBytes, ShiftRows, MixColumns & AddRoundKey.
Final Nth round has only 3 transformation f^n.

→ Each transformation takes one or more 4x4 matrices as i/p & produces 4x4 matrix as o/p.

o/p of final round → Cipher text.

4x4 → block is copied to 'State' array after the final stage,
State is copied to an o/p matrix

Key → Sq. matrix of bytes → then it is expanded.



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 6.1 AES Encryption Process

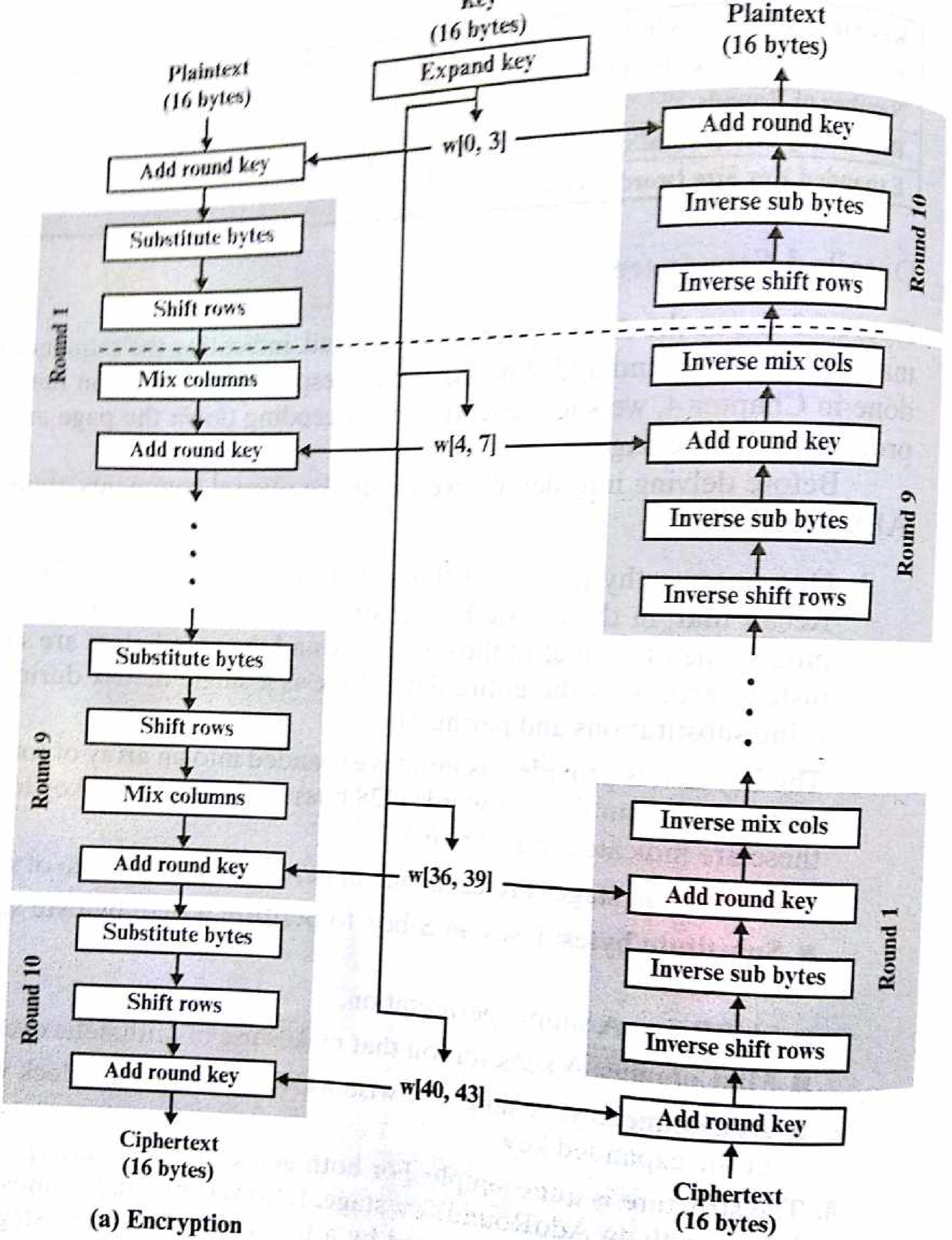
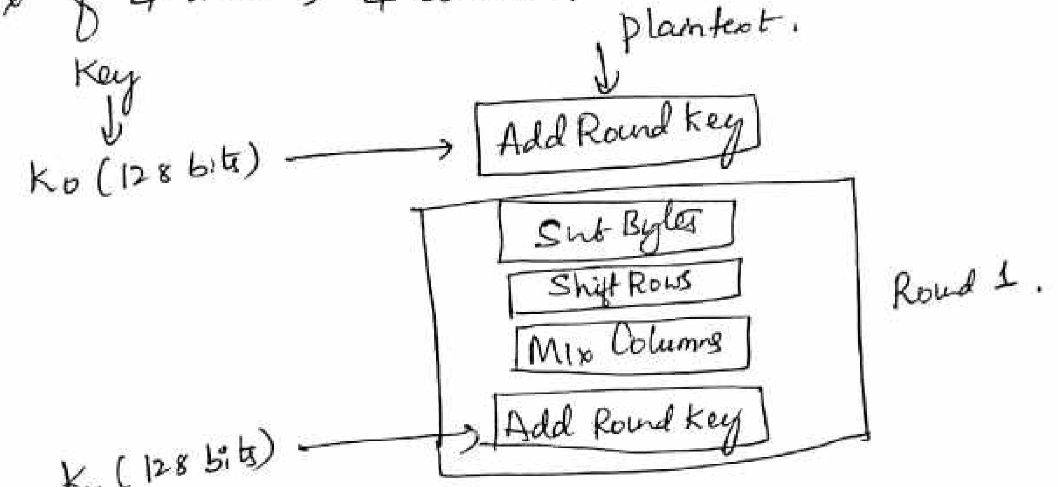


Figure 6.3 AES Encryption and Decryption

AES Transformation functions

① Substitute Bytes Transformation (SubBytes)

16 i/p bytes are substituted by looking up a fixed table (S-Box) given in design. The result is in a matrix of 4 rows & 4 columns.



AES S-Box implements inverse multiplication in $GF(2^8)$ & uses 16x16 matrix of bytevalues → S Box → Encryption & Inverse S-Box for decryption

[DES uses 8 different S-boxes, but AES always uses only one S-Box & IS Box]

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix} \underset{\text{InvSubByte.}}{\overset{\text{SubByte}}{\rightleftarrows}} \begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & F2 & 63 & 26 \\ C9 & C9 & 7D & D4 \\ FA & 63 & 82 & D4 \end{bmatrix}$$

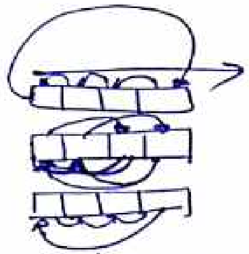
SubBytes & InvSubBytes transformations are inverses of eachother.

The SubBytes transformation repeats a routine called **Subbyte** 16 times, each iteration transforms one byte. In the Subbyte routine, the multiplicative inverse of the byte is found in $GF(2^8)$ with the irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$ as the modulus.

Shift Rows Transformation

First row of state is not altered.
Second —"— 1 byte circular left shift
Third —"— 2 bytes of —"—
Fourth —"— 3 bytes —"—

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Inverse Shift Row transformation → performs the circular shifts in the opp. direction for each of the last 3 rows.
circular rt. shift.

Mix Column Transformation

Multiplication of bytes is done in $GF(2^8)$ with modulus (1000 1101) or $x^8+x^4+x^3+x+1$

→ operates on each column individually.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & & & \\ S_{30} & & & \end{bmatrix} = \begin{bmatrix} S'_{00} & S'_{01} & & \\ S'_{10} & & & \\ & & & \\ & & & S'_{33} \end{bmatrix}$$

$$S'_{0,j} = (2 \cdot S_{0,j}) \oplus (3 \cdot S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$
$$S'_{1,j} = S_{0,j} \oplus (2 \cdot S_{1,j}) \oplus (3 \cdot S_{2,j}) \oplus S_{3,j}$$
$$S'_{2,1} = S_{0,j} \oplus S_{1,j} \oplus (2 \cdot S_{2,j}) \oplus (3 \cdot S_{3,j})$$
$$S'_{3,j} = (3 \cdot S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 \cdot S_{3,j})$$

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
46	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$([02]\cdot[87]) \oplus ([03]\cdot[6E]) \oplus [46] \oplus [A6] = [47]$

$[87] \oplus [02]\cdot[6E] \oplus ([03]\cdot[46]) \oplus [A6] = [37]$

$= [94]$

$= [ED]$

$[02]\cdot[87] = 0001\ 0101.$

$[0000\ 0010]\cdot[1000\ 0111] = x[x^7 + x^2 + x + 1]$

$= \cancel{x^8} + \cancel{x^3} + x^2 + \cancel{x}$

$\cancel{x^8} + x^4 + \cancel{x^3} + \cancel{x} + 1$

$x^4 + x^2 + 1$

$\boxed{0001\ 0101}$

$1 \quad 5$

$[03]\cdot[6E] = \boxed{1011\ 0010}$

$[0000\ 0011]\cdot[0110\ 1110] = (x+1)(x^6 + x^5 + x^3 + x^2 + x)$

$= x^7 + \cancel{x^6} + x^4 + \cancel{x^3} + \cancel{x^2} + \cancel{x^6} + x^5 + \cancel{x^3} + \cancel{x^2} + x$

$= x^7 + x^5 + x^4 + x.$

$= \boxed{1011\ 0010}$

$[02][87]$	0001	0101
$[03]\cdot[6E]$	1011	0010
$[46]$	0100	0110
$[A6]$	1010	0110
	0100	0011
	(4	7)

AddRoundKey Transformation

Forward and Inverse Transformations In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. As shown in Figure 6.5b, the operation is viewed as a columnwise operation between the 4 bytes of a **State** column and one word of the round key; it can also be viewed as a byte-level operation. The following is an example of AddRoundKey:

$$\begin{array}{|c|c|c|c|}\hline 47 & 40 & A3 & 4C \\ \hline 37 & D4 & 70 & 9F \\ \hline 94 & E4 & 3A & 42 \\ \hline ED & A5 & A6 & BC \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline AC & 19 & 28 & 57 \\ \hline 77 & FA & D1 & 5C \\ \hline 66 & DC & 29 & 00 \\ \hline F3 & 21 & 41 & 6A \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline EB & 59 & 8B & 1B \\ \hline 40 & 2E & A1 & C3 \\ \hline F2 & 38 & 13 & 42 \\ \hline 1E & 84 & E7 & D6 \\ \hline \end{array}$$

The first matrix is **State**, and the second matrix is the round key.

The **inverse add round key transformation** is identical to the forward add round key transformation, because the XOR operation is its own inverse.

Rationale The add round key transformation is as simple as possible and affects every bit of **State**. The complexity of the round key expansion, plus the complexity of the other stages of AES, ensure security.

Figure 6.8 is another view of a single round of AES, emphasizing the mechanisms and inputs of each transformation.

Module 7.

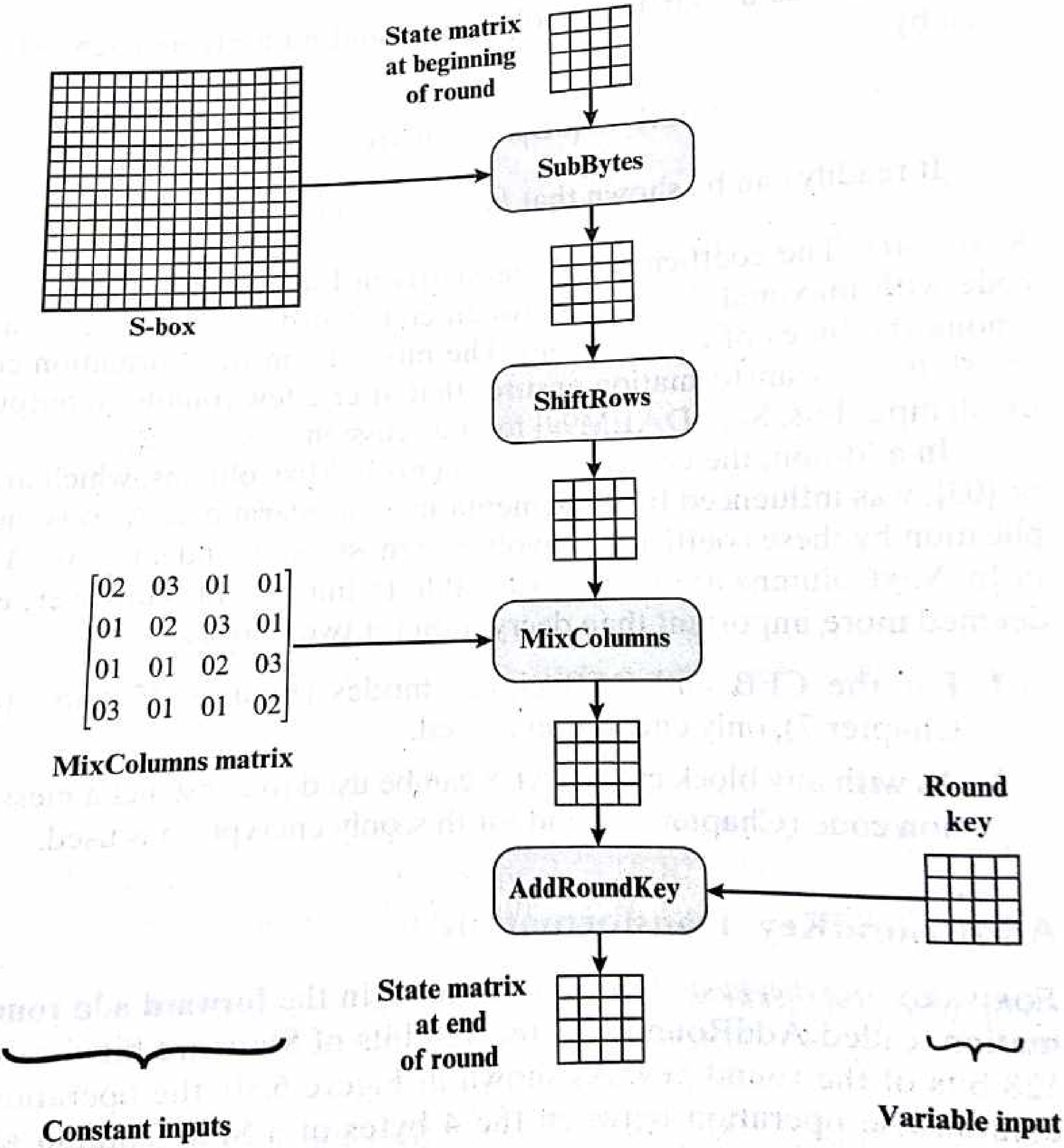
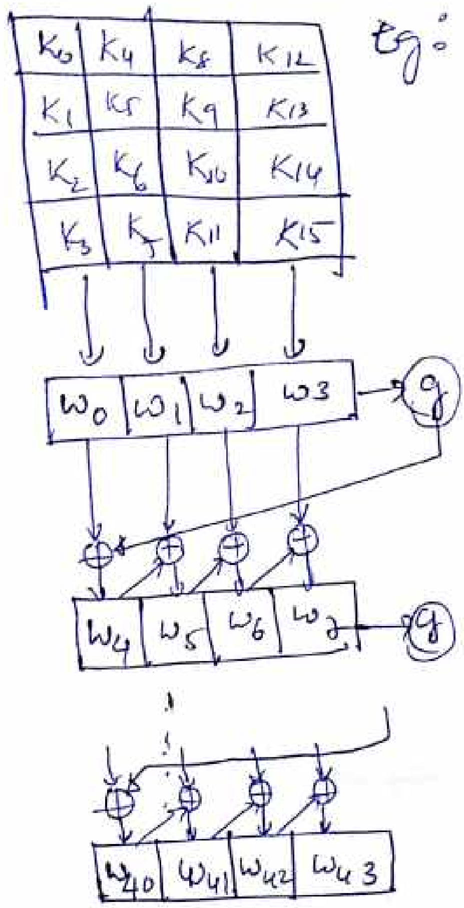


Figure 6.8 Inputs for Single AES Round

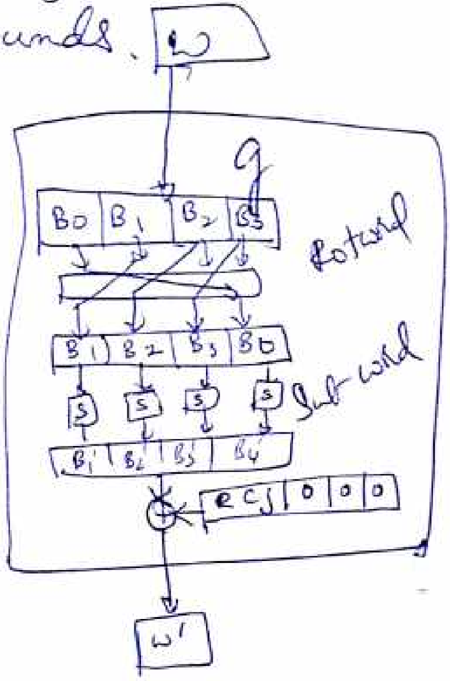
AES Key Expansion

4 word (16 bytes) produces a linear array of 44 words (176 bytes)

Eg: K = 16 bytes
10 rounds.



Overall algorithm.



function g.

① RotWord → left one shift

② SubWord → Substitution using S Box (same AES S-Box used in SubBytes)

③ Result is XORed with a Round constant Rcon[j]

→ Round constant is a word in which 3 right most bytes are '0'

→ Effect of Rcon → only on left most byte of the word.

→ Rcon is different for each round & is defined as

$Rcon[j] = (RC[j], 0, 0, 0)$

with $RC[1] = 1$

$RC[j] = 2 \cdot RC[j-1]$ multiplication defined over $GF(2^8)$.

1. RotWord performs a one-byte circular left shift on a word. This means that an input word $[B_0, B_1, B_2, B_3]$ is transformed into $[B_1, B_2, B_3, B_0]$.
2. SubWord performs a byte substitution on each byte of its input word, using the S-box (Table 6.2a).
3. The result of steps 1 and 2 is XORed with a round constant, Rcon[j].

The round constant is a word in which the three rightmost bytes are always 0. Thus, the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word. The round constant is different for each round and is defined as $\text{Rcon}[j] = (\text{RC}[j], 0, 0, 0)$, with $\text{RC}[1] = 1$, $\text{RC}[j] = 2 \cdot \text{RC}[j-1]$ and with multiplication defined over the field $GF(2^8)$. The values of RC[j] in hexadecimal are

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

For example, suppose that the round key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then the first 4 bytes (first column) of the round key for round 9 are calculated as follows:

i (decimal)	temp	After RotWord	After SubWord	Rcon (9)	After XOR with Rcon	w[i − 4]	w[i] = temp ⊕ w[i − 4]
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3

Module 4 (2)

Bruce Schneier "Applied Cryptography Protocols, Algorithms & Source code in C" 2nd Ed. Wiley Publications (1)

4 Pseudo-Random Sequence Generators & Stream Ciphers

Linear Congruential Generators, Linear Feedback Shift Registers, Design & analysis of Stream ciphers, Stream ciphers using LFSRs.

Linear congruential Generators: LCG.

LCGs are pseudo-random sequence generators of the form $X_n = (aX_{n-1} + b) \bmod m$.

$X_n \to$ n^{th} no. of sequence.

a, b & $m \to$ constants

$X_{n-1} \to$ previous no. of the seq.

$a \to$ multiplier, $b \to$ increment

$X_0 \to$ key / seed.

$m \to$ modulus.

The generator has a period $< m$. if period $= m$ then generator will be a max. period generator (max length)

LCG $\to$ are fast & requires few operations per bit, but cannot be used for cryptography — as they are predictable.

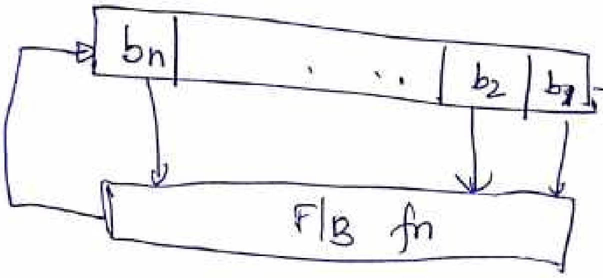
$X_n = (aX_{n-1}^2 + bX_{n-1} + c) \bmod m \to$ Quadratic generator

$X_n = (aX_{n-1}^3 + bX_{n-1}^2 + cX_{n-1} + d) \bmod m \to$ Cubic generator

Linear Feedback Shift Registers

→ are used for both cryptography & coding theory.

→ Stream ciphers based on shift registers → popular in military cryptography since the beginning of electronics.



2 parts
Shift registers
&
f/b function

All the bits are shifted to right by 1 bit & new left most bit is computed as a f^n of other bits in the register. o/p → least significant bit.

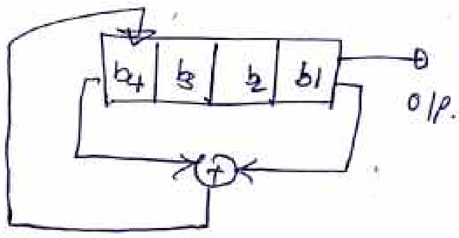
The period of the shift register is the length of the o/p seq before it starts repeating.

Eg:- If a 4 bit shift register has 1111 as initial bits then

$1 \oplus 1 = 0$

$0 \oplus 1 = 1$

b_4	b_3	b_2	b_1
1	1	1	1
	1	1	1
0	1	1	1
1	0	1	1
0	1	0	1
1	0	1	0
1	1	0	1
0	1	1	0
0	0	1	1
1	0	0	1
0	1	0	0
0	0	1	0
⋮		1	0



Simple → XOR fn.

o/p of seq is

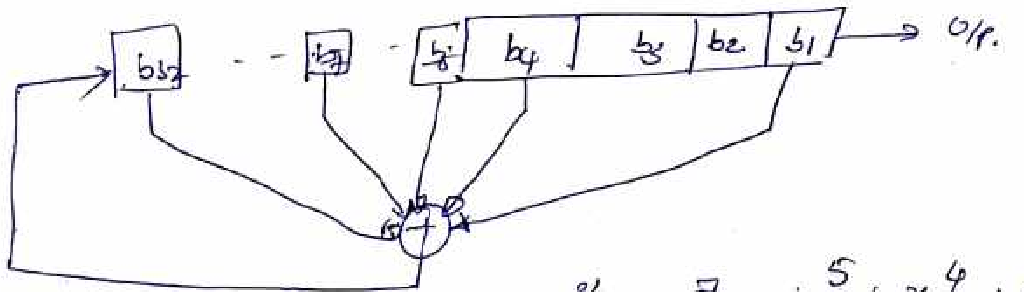
$\underbrace{1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \cdots}_{\text{period}}$

n bit LFSR → $2^n - 1$ states

4 bit → $2^4 - 1 = 15$ = period - m.

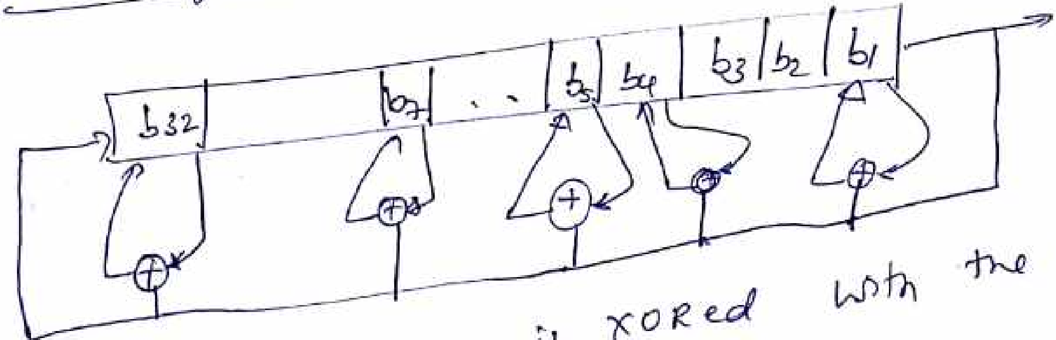
m sequence. — o/p seq.

32 bit long max-length LFSR.



polynomial : $x^{32} + x^7 + x^5 + x^4 + 1$

Galois Configuration :-



Each bit in the seq is XORed with the o/p of the generator & replaced

It is faster in h/w — especially VLSI implementation.

Design & Analysis of Stream Ciphers

LFSRs → most practical stream-cipher designs.

Gives lot of security with only a few logic gates.

H/w - efficient but inefficient in S/w.

Sparse f/b polynomials (a few coeffs) — weakness easily breakable

dense primitive polynomials → with lot of coeffs.
→ ~~few~~ shorter LFSRs.

Single iteration in DES can encrypt → 64 times iterated stream cipher.

Linear Complexity: metric used to analyze.
Analyzing stream ciphers is often easier than block cipher.

Linear complexity — imp. metric & is defined as length n of the shortest LFSR that can mimic the o/p.

Algorithm — Berlekamp-Massey algorithm → generate LFSR & break the stream cipher

Linear complexity profile — measures the linear complexity of the seq as it gets longer & longer.

High linear complexity → a secure generator (does not guarantee)

low —"— → insecure generator. (guarantee)

Correlation immunity: identify some correlation b/n o/p of generator & o/p of one of its internal pieces.

Other attacks: — linear consistency test, meet in the middle consistency attack, best affine approx. attack & derived seq. attack.

DIFFIE - HELLMAN KEY EXCHANGE

Algorithm used to establish a shared secret b/n 2 parties. It is used to exchange cryptography keys for use in symmetric algorithms (AES)

The algorithm itself is limited to the exchange of secret values.

D-H algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.

Primitive root : Primitive root of a prime no. p is one whose powers modulo p generate all the integers from 1 to $p-1$.

i.e if a is a primitive root of prime no. p, then the nos. $a \bmod p$, $a^2 \bmod p$ . . . $a^{p-1} \bmod p$ are distinct & consists of the integers from 1 thro $p-1$ in some permutation.

Then $b \equiv a^i \pmod p$ $i \rightarrow$ exponent $0 \leq i \leq p-1$

i is the discrete logarithm of b for the base a mod p.

Eg:- $a = 2$, $p = 11$

$a^1 \bmod p \rightarrow 2 \bmod 11 = 2$

$a^2 \bmod p \rightarrow 4 \bmod 11 = 4$

$a^3 \bmod p \rightarrow 8 \bmod 11 = 8$

$a^4 \bmod p \rightarrow 16 \bmod 11 = 5$

$a^5 \bmod p \rightarrow 32 \bmod 11 = 10$

$a^6 \bmod p \rightarrow 64 \bmod 11 = 9$

$a^7 \bmod p \rightarrow 128 \bmod 11 = 7$

$a^8 \bmod p \rightarrow 256 \bmod 11 = 3$

$a^9 \bmod p \rightarrow 512 \bmod 11 = 6$

$(p-1)$ $a^{10} \bmod p \rightarrow 1024 \bmod 11 = 1$

a is primitive root of p

Alice	Bob
① Alice & Bob share a prime no q & α such that $\alpha < q$ & α is a primitive root of q.	① —— " ——
② Alice generates a private key X_A such that $X_A < q$	② Bob generates a private key X_B such that $X_B < q$
③ Alice calculates public key $Y_A = \alpha^{X_A} \bmod q$	③ Bob calculates public key $Y_B = \alpha^{X_B} \bmod q$.
④ Alice receives Bob's Y_B public key Y_B in plain text	④ Y_A Bob receives Alice's public key Y_A in plaintext.
⑤ Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$	⑤ Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$.

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q. \\
&= \left(\alpha^{X_B} \bmod q\right)^{X_A} \bmod q \\
&= \left(\alpha^{X_B}\right)^{X_A} \bmod q \\
&= \left(\alpha^{X_A}\right)^{X_B} \bmod q \\
&= \left(\alpha^{X_A} \bmod q\right)^{X_B} \bmod q. \\
&= (Y_A)^{X_B} \bmod q.
\end{aligned}
$$

K → Shared key.

↓

Symmetric secret key.

$\alpha = 2$, $q = 11$.

Select $X_A \rightarrow 8$ $8 < 11$

$Y_A = \alpha^{X_A} \bmod q = 2^8 \bmod 11 = 3$.

Select $X_B = 4$ $4 < 11$

$Y_B = \alpha^{X_B} \bmod q = 2^4 \bmod 11 = 5$

Secret key at A, $K = (Y_B)^{X_A} \bmod q$

$= 5^8 \bmod 11$

$\boxed{K = 4.}$

Secret key at B, $K = (Y_A)^{X_B} \bmod q$

$= (3)^4 \bmod 11$

$\boxed{K = 4.}$

with large nos. problem $\rightarrow$ impractical.

$C = E(K, M)$

$P = D.(K, C)$

If Darth $\longrightarrow$ wants to attack (man in the middle Attack)

then K_1 & K_2 are generated using Y_A & Y_B

& $K_1 \neq K_2$.

Victor Miller (IBM)
Neil Koblitz (UoW)

Elliptic Curve Arithmetic 1985 - ECC

Most of the products & standards that use public key cryptography for encryption & digital signatures use RSA. As the key length for secure RSA use has increased over recent years, this has put a heavier processing load on app^ns using RSA. (E commerce) based on difficult logarithm.

Elliptic Curve Cryptography (ECC) offers equal security for a smaller key size, thereby reducing processing overhead.

ECC is more difficult to explain than RSA / DHKE

Elliptic curves

Not ellipses but they are described by cubic eq^ns.

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

where a, b, c, d, e are real n̲os.

x & y $\longrightarrow$ value of real n̲os.

Simple eq^n $\longrightarrow$ $\boxed{y^2 = x^3 + ax + b}$ $\longrightarrow$ eq^n is cubic or degree 3.

$$y = \sqrt{x^3 + ax + b}$$

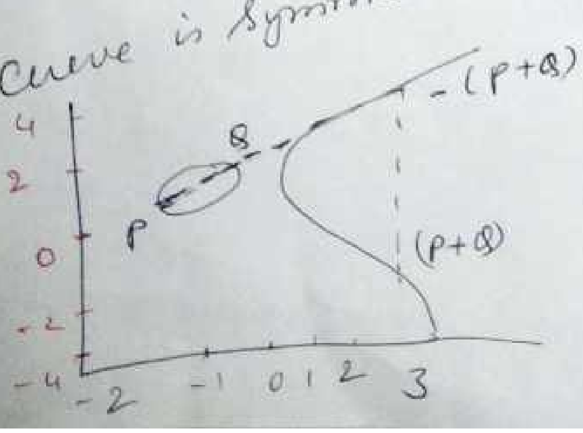
for each value of x, given a & b we can plot values of y +ves / -ve.

curve is symmetric about y = 0.

$E(\underset{a}{-1}, \underset{b}{0})$

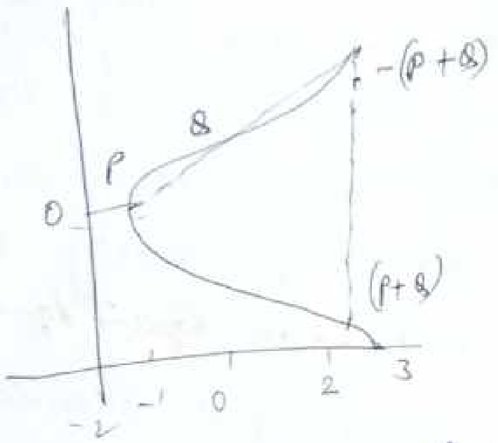
$y^2 = x^3 + ax + b$

$y^2 = x^3 - x + 0$



$\boxed{y^2 = x^3 - x}$

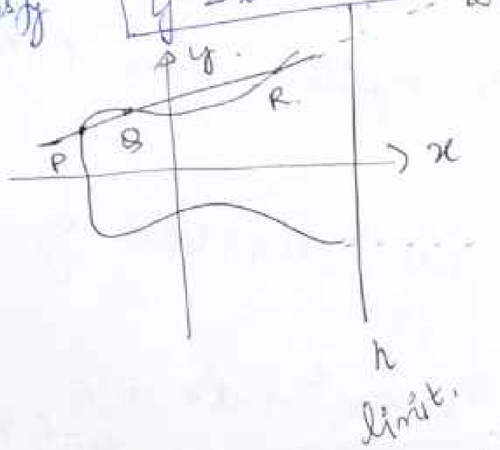
When we draw a line, max. 3 points it can touch.



$y^2 = x^3 + x + 1$.

E(1,1)

E(a,b) → set of points consisting of all of the points (x,y) that satisfy $\boxed{y^2=x^3+ax+b}$



3 points P, Q & R.

Set E(a,b) → abelian group. (A1) to (A5)

Let E(a,b) → EC. then Q = kP & k < n.

Given P & k, Q can be found.

But difficult to find k, if Q & P are known → One way function

or Trap door function.

It is a discrete log. problem.

ECC can be used for Key exchange, Encryption/Decryption & Digital Signatures.

Elliptic Curve Cryptography:

Addition operation in ECC $\longrightarrow$ modular multiplication in RSA.

Multiple addition $\longrightarrow$ modular exponentiation.

Key exchange:- Analog of Diffie-Hellman Key Exchange.

First pick a large integer 'q' such that q is either prime no. or an integer of the form 2^m. Choose a & b to get $E_q(a,b)$.

Next pick a base point $G = (x_1, y_1)$ in $E_p(a,b)$ whose order is n.

Alice	Bob
1) Alice selects an integer n_A less than n $\rightarrow$ private key	1) Bob selects $n_B < n$ private key of Bob.
2) Alice generates a public key $P_A = n_A \times G$.	2) Bob also generates a public key $P_B = n_B \times G$.
3) Alice generates secret key $k = n_A \times P_B$	3) $k = n_B \times P_A$

$\boxed{k \rightarrow \text{same result}}$

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A.$$

Note:- Secret key $\longrightarrow$ pair of nos. $\longrightarrow$ single no.

EC Encryption / Decryption :-

Many methods → Simplest one is discussed here.

The first task is to encode the plaintext message 'm' to be sent as an (x, y) point P_m.

It requires a point G & an elliptic group $E_q(a,b)$ as parameters. Each user A, selects a private key n_A & generates a public key $\boxed{P_A = n_A \times G}$

→ To encrypt & send a message P_m to B, A chooses a random +ve integer k & produces the ciphertext C_m

$$C_m = \{kG, P_m + kP_B\}$$ {A has used B's public key P_B.}

→ To decrypt the ciphertext, B multiplies the first point in the pair by B's private key & subtracts the result from the second point.

$$P_m + kP_B - n_B(kG) = P_m + k(\cancel{n_B} \times G) - \cancel{n_B} kG$$

$$= P_m.$$

Only A knows the value of 'k' → nobody can remove the mask kP_B.

Eg2:- Chinese Remainder Thm

$x \equiv 2 \pmod 3$

$x \equiv 4 \pmod 5$

$x \equiv 5 \pmod 7$

$a_1 = 2, a_2 = 4, a_3 = 5$

$m_1 = 3, m_2 = 5, m_3 = 7$

$\boxed{M = m_1 m_2 m_3 = 105}$

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 \pmod M$

$M_1 = \frac{M}{m_1} = 35$

$M_2 = \frac{M}{m_2} = 21$

$M_3 = \frac{M}{m_3} = 15$

$\therefore M_1 y_1 \equiv 1 \pmod{m_1}$

$35 y_1 \equiv 1 \pmod 3$

$2 y_1 \equiv 1 \pmod 3$

$\cancel{4 y_1 \equiv 2 \pmod 3}$

$3 \overline{)35}$ $11 \cdot 2$

$1 y_1 \equiv 2 \pmod 3$

$2 \times 2 = 1 \bmod 3.$

$\therefore \boxed{y_1 = 2}$

$M_2 y_2 \equiv 1 \pmod{m_2}$

$21 y_2 \equiv 1 \pmod 5$

$1 y_2 \equiv 1 \pmod 5$

$\boxed{y_2 = 1}$

$M_3 y_3 \equiv 1 \pmod{m_3}$

$15 y_3 \equiv 1 \pmod 7$

$1 y_3 \equiv 1 \pmod 7$

$\boxed{y_3 = 1}$

$x \equiv 299 \pmod{105}$ (x, M)

$x = 89 \pmod{105}$

$\boxed{\text{Solution is } x = 89}$

$x = 2 \times 35 \times 2 + 4 \times 21 \times 1 + 5 \times 15 \times 1$

$= 140 + 84 + 75 = 299 - 105 = 194$

$194 - 105 = 089$

Eg3:- $x \equiv 1 \bmod 3$

$x \equiv 1 \bmod 4$

$x \equiv 1 \bmod 5$

$x \equiv 0 \bmod 7$.

$a_1 = a_2 = a_3 = 1$

$a_4 = 0$.

$m_1 = 3;\ m_2 = 4;\ m_3 = 5$

$m_4 = 7$.

$M = m_1 . m_2 . m_3 . m_4 = 3 \times 4 \times 5 \times 7 = 420$.

$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 + a_4 M_4 y_4$.

~~$\equiv 1 \times 140 x$~~

$x = 1 \times 140 \times 2 + 1 \times 105 \times 1 + 1 \times 84 \times 4 + 0 \bmod M$

$x = 721 \bmod 420$.

$\boxed{x = 301}$

$M_1 = \frac{M}{m_1} = \frac{420}{3}$

$M_1 = 140$

$M_2 = \frac{M}{m_2} = \frac{420}{4} = 105$

$M_3 = \frac{M}{m_3} = \frac{420}{5} = 84$

$M_4 = \frac{420}{7} = 60$.

$M_1 y_1 \equiv 1 \bmod 3$

$140 y_1 \equiv 1 \bmod 3$

$2 y_1 \equiv 1 \bmod 3$ $3 \overline{)140}$, $\cancel{46}$ 2

$\boxed{y_1 = 2}$

$M_2 y_2 \equiv 1 \bmod 4$

$105 y_2 \equiv 1 \bmod 4$

$1 y_2 \equiv 1 \bmod 4$ $4 \overline{)105}$, 23-1

$\boxed{y_2 = 1}$

$M_3 y_3 \equiv 1 \bmod 5$

$84 y_3 \equiv 1 \bmod 5$

$4 y_3 \equiv 1 \bmod 5$ $5 \overline{)84}$, 80-4

$4 \times 4 \equiv 1 \bmod 5$

$\boxed{y_3 = 4}$

$M_4 y_4 \equiv 1 \bmod 7$

$60 y_4 \equiv 1 \bmod 7$

$4 y_4 \equiv 1 \bmod 7$ $7 \overline{)60}$, 56-4

$\boxed{y_4 = 2}$