Unit - II

Modular Arithmetic

If a is divided by n, then

$a = qn + r \quad 0 \le r < n ; \quad q = \lfloor a/n \rfloor$

where
q - quotient
r - remainder

$\lfloor x \rfloor$ - largest integer $\le x$.

Examples

$11/7 \Rightarrow a = 11, n = 7 \Rightarrow \underline{1} \times 7 + \underline{4} = 11 \Rightarrow r = 4$

$-11/7 \Rightarrow a = -11, n = 7 \Rightarrow \underline{-2} \times 7 + \underline{3} = -11 \Rightarrow r = 3$

$\therefore a = \lfloor a/n \rfloor \times n + (a \bmod n)$

Two integers are said to be congruent modulo n if

$a \bmod n = b \bmod n$

$\therefore a \equiv b \bmod n$

$4 \bmod 23 = 4$

Ex 1

$73 \bmod 23$

$\underline{3} \times 23 + \underline{4} = 73$

$\Downarrow$

(4)

$\therefore 73 \equiv 4 \bmod 23$

Ex 2

$21 \bmod 10 = 1$

$-9 \bmod 10 \Rightarrow$

$\underline{-1} \times 10 + \underline{1} = -9$

$\Rightarrow +1$

$\therefore 21 \equiv -9 \bmod 10$

Divisors

A nonzero value b divides a if $a = mb$ for some m where a, b, m are integers i.e, there are no remainders upon division. 'b' is called divisor of 'a'.

Positive divisors of 24 are 1, 2, 3, 4, 6, 8, 12 and 24.

1) If $a|1$, then $a = \pm 1$

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

3) Any $b \neq 0$ divides 0

4) If $b|g$ and $b|h$, then $b|(mg + nh)$ for arbitrary integers m and n.

Ex:- $b = 7, g = 14, h = 63, m = 3, n = 2$

To show: $7|(3 \times 14 + 2 \times 63)$

$7|(3 \times 14 + 2 \times 63) = 7|7(3 \times 2 + 2 \times 9)$

$= 7|21$

Note: If $a \equiv 0 \bmod n$, then $n|a$

Properties of modulo operator

1) $a \equiv b \bmod n$ if $n|(a-b)$

2) $a \equiv b \bmod n \implies b \equiv a \bmod n$

3) $a \equiv b \bmod n$ && $b \equiv c \bmod n \implies a \equiv c \bmod n$.

Proof 1:- If $n|(a-b)$, then $(a-b) = kn \implies a = b + kn$

$\therefore a \bmod n = b \bmod n$

$\therefore a \equiv b \bmod n$

$b \div n = r$

$qn + r = b$

Proof 2:

$a \bmod n = b \bmod n$

$(q_1 n + r) \bmod n = (q_2 n + r) \bmod n$

$(q_1 n) \bmod n = (q_2 n) \bmod n$

Modular Arithmetic Operations

Manoj
Date / /20

mod n operator maps all integers to $\{0, \ldots n-1\}$

Properties

$$(a \bmod n + b \bmod n) \bmod n = (a+b) \bmod n$$

$$(a \bmod n - b \bmod n) \bmod n = (a-b) \bmod n$$

$$(a \bmod n * b \bmod n) \bmod n = (a*b) \bmod n$$

Exponentiation is performed by repeated multiplication

$$11^7 \bmod 13 = ?$$

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

13
11 7 2

$$11^b \bmod 13 = (11^2)^3 \bmod 13 = (4)^3 \bmod 13 = 64 \bmod 13 = 12 \bmod 13$$

13
5 2 0
1

$$11^7 \bmod 13 = (11 \times 11^6) \bmod 13 = (11 \times 12) \bmod 13 = 132 \bmod 13$$

$$= 2 \bmod 13$$

Addition 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 mod 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.

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

Properties of Modular Arithmetic

$\mathbb{Z}_n = \{0, 1, 2, \ldots (n-1)\}$

↓ Residue classes.

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

Associative
$((w+x)+y) \bmod n = (w+(x+y)) \bmod n$
$((w \times x) \times y) \bmod n = (w \times (x \times y)) \bmod n$

Distributive
$(w \times (x+y)) \bmod n = ((w \times x) + (w \times y)) \bmod n$
$(w + (x * y)) \bmod n = ((w+x) * (w+y)) \bmod n$

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

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

Mangal
Date / /20

Property 1

If $(a+b) \equiv (a+c) \bmod n$, then $b \equiv c \bmod n$

Proof: Add additive inverse of a to both sides

$$(a+b) \equiv (a+c) \bmod n$$

$$(-a+a+b) \equiv (-a+a+c) \bmod n$$

$$\therefore b \equiv c \bmod n$$

Property 2:

If $(a \times b) \equiv (a \times c) \bmod n$ then $b \equiv c \bmod n$ if a is relatively prime to n.

Two integers are relatively prime if their only common positive integer factor is 1.

Proof: Multiply with multiplicative inverse on both sides

$$(a^{-1} \times a \times b) \equiv (a^{-1} \times a \times c) \bmod n$$

$$b \equiv c \bmod n$$

RSA Algorithm

Block cipher in which PT & CT are integers between 0 and n for some n. Typical size for n is 1024 bits / 309 decimal digits.

Plaintext is divided into blocks of size. --
(Blocksize $\leq \log_2 n$)

Mangal
Date / /20

Encryption: $C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

Sender to $\{e, n\}$ Public key

Receiver to $\{d, n\}$ Private key

Requirements expected in public key encryption

1) It is possible to find values of e, d, n such that $M^{ed} = M \bmod n$ for all $M < n$

2) It is relatively easy to calculate M^e and C^d for all values of $M < n$

3) It is infeasible to determine d given e and n.

RSA

Select p, q two prime numbers (private)

$n = pq$ (public, calculated)

e with $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$ (public)

$d = e^{-1} \bmod \phi(n)$ (private, calculated)

Example

1) Select $p = 17, q = 11$

2) Calculate $n = pq = 187$

3) Calculate $\phi(n) = (17-1)(11-1) = 160$

4) Choose e; $e = 7$

5) Find d;

$d = e^{-1} \bmod \phi(n) = 23.$

$= 7^{-1} \bmod 160.$

Mangal
Date / /20

$M = 88 \quad C = 88^{7} \bmod 187$

$= \left[\left(88^{4} \bmod 187 \right) \times \left(88^{2} \bmod 187 \right) \times \left(88 \bmod 187 \right) \right] \bmod 187.$

$88 \bmod 187 = 88$

$88^{2} \bmod 187 = 7744 \bmod 187 = 77$

$88^{4} \bmod 187 = 77^{2} \bmod 187 = 5929 \bmod 187 = 132$

$\therefore C = (132 \times 77 \times 88) \bmod 187$

$= (894432) \bmod 187 = 11$

$\boxed{C = 11}$

$M = 11^{23} \bmod 187$

$= \left(11^{8} \bmod 187 \times 11^{8} \bmod 187 \times 11^{4} \bmod 187 \times 11^{2} \bmod 187 \times 11 \bmod 187 \right) \bmod 187$

$11 \bmod 187 = 11$

$11^{2} \bmod 187 = 121 \bmod 187 = 121$

$11^{4} \bmod 187 = 121^{2} \bmod 187 = 14641 \bmod 187 = 55$

$11^{8} \bmod 187 = 55^{2} \bmod 187 = 3025 \bmod 187 = 33$

$\therefore M = (33 \times 33 \times 55 \times 121 \times 11) \bmod 187$

$= 79720245 \bmod 187 = 88$

$\boxed{M = 88}$

Euclidean algorithm

Mangal
Date 1/1/20

1) Divide the larger of the given integers by smaller one to get quotient & remainder
2) Divide the second integer by remainder and proceed in same manner until remainder becomes zero. The previous remainder is the desired gcd

Ex:1) gcd (20, 16)

$20 = 1 \times 16 + 4$

$16 = 4 \times 4 + 0$

$\therefore$ gcd = 4

2) gcd (50, 60)

$60 = 1 \times 50 + 10$

$50 = 5 \times 10 + 0$

gcd = 10

3) gcd (24598, 7895)

$24598 = 3 \times 7895 + 913$

$7895 = 8 \times 913 + 591$

$913 = 1 \times 591 + 340$

$591 = 1 \times 340 + 251$

$340 = 1 \times 251 + 89$

$251 = 2 \times 89 + 73$

$89 = 1 \times 73 + 16$

$73 = 4 \times 16 + 9$

$16 = 1 \times 9 + 7$

$9 = 1 \times 7 + 2$

$7 = 3 \times 2 + 1$

$3 = 3 \times 1 + 0$

gcd = 1

4) gcd (56245, 43159)

Mangal
Date / /20

$56245 = 1 \times 43159 + 13086$
$43159 = 3 \times 13086 + 3901$
$13086 = 3 \times 3901 + 1383$
$3901 = 2 \times 1383 + 1135$
$1383 = 1 \times 1135 + 248$
$1135 = 4 \times 248 + 143$

$248 = 1 \times 143 + 105$
$143 = 1 \times 105 + 38$
$105 = 2 \times 38 + 29$
$38 = 1 \times 29 + 9$
$29 = 3 \times 9 + 2$
$9 = 4 \times 2 + 1$
$4 = 1 \times 4 + 0$

$\therefore$ gcd (56245, 43159) = 1

5) gcd (56211, 19385)

$56211 = 2 \times 19385 + 17441$
$19385 = 1 \times 17441 + 1944$
$17441 = 8 \times 1944 + 1889$
$1944 = 1 \times 1889 + 55$
$1889 = 34 \times 55 + 19$
$55 = 2 \times 19 + 17$

$19 = 1 \times 17 + 2$
$17 = 8 \times 2 + 1$
$2 = 2 \times 1 + 0$

gcd = 1

Extended Euclidean Algorithm for finding inverse of a number modulo n.

1) Start at step 0. Let q_i denote quotient at step i.
2) An auxiliary number p_i is calculated at each steps as
$$p_i = (p_{i-2} - p_{i-1} \cdot q_{i-2}) \bmod n$$
Initially $p_0 = 0$, $p_1 = 1$
3) This calculation is done for one step beyond last step of Euclidean algorithm

4) If last non-zero remainder [illegible] then if this remainder is 1, x has an inverse and it is P_{k+2}

Mangal
Date / / 20

Ex1: $15^{-1} \bmod 26$

Step	
0	$26 = 1 \times 15 + 11$
1	$15 = 1 \times 11 + 4$
2	$11 = 2 \times 4 + 3$
3	$4 = 1 \times 3 + 1$
4	$3 = 3 \times 1 + 0$

q	P
	$P_0 = 0$
1	$P_1 = 1$
2	$P_2 = (P_0 - P_1 q_0) \bmod 26$ $= (0 - 1 \times 1) \bmod 26$ $= -1 \bmod 26 = 25$
1	$P_3 = P_1 - P_2 q_1$ $= 1 - 25 \times 1) \bmod 26$ $= -24 \bmod 26 = 2$
3	$P_4 = P_2 - P_3 q_2$ $= 25 - (2 \times 2) \bmod 26$ $= 21$

$P_5 = P_3 - P_4 q_3$
$= 2 - (21 \times 1) \bmod 26$
$= -19 \bmod 26 = 7$

Ex2: $7^{-1} \bmod 160$

Step	
0	$160 = 22 \times 7 + 6$
1	$7 = 1 \times 6 + 1$
2	$6 = 6 \times 1 + 0$

q	P
22	$P_0 = 0$
1	$P_1 = 1$
6	$P_2 = (P_0 - P_1 q_0) \bmod 160 = (0 - 1 \times 22) \bmod 160$ $= 138$

$P_3 = (P_1 - P_2 q_1) \bmod 160$
$= (1 - 138 \times 1) \bmod 160 = 23$

Mangal
Date / /20

Ex 3: $53^{-1} \bmod 154$

Step	
0	$154 = 2 \times 53 + 48$
1	$53 = 1 \times 48 + 5$
2	$48 = 9 \times 5 + 3$
3	$5 = 1 \times 3 + 2$
4	$3 = 1 \times 2 + 1$
5	$2 = 2 \times 1 + 0$

q	p
2	$p_0 = 0$
1	$p_1 = 1$
9	$p_2 = (p_0 - p_1 q_0) = (0 - 1 \times 2) = -2 \bmod 154 = 152$
1	$p_3 = (p_1 - p_2 q_1) = (1 - 152 \times 1) \bmod 154 = -151 \bmod 154 = 3$
1	$p_4 = (p_2 - p_3 q_2) = (152 - 3 \times 9) \bmod 154 = 125$
2	$p_5 = (p_3 - p_4 q_3) = (3 - 125 \times 1) \bmod 154 = (-122) \bmod 154 = 32$

$p_6 = (p_4 - p_5 q_4) \bmod 154$

$= (125 - 32 \times 1) \bmod 154 = 93$

$\therefore 53^{-1} \bmod 154 = 93$

Ex 4: $53^{-1} \bmod 158$

Step	
0	$158 = 2 \times 53 + 52$
1	$53 = 1 \times 52 + 1$
2	$52 = 52 \times 1 + 0$

q	p
2	0
1	1
52	$p_2 = (p_0 - p_1 q_0) \bmod 158$

$= (0 - 2) \bmod 158 = 156$

$p_3 = (p_1 - p_2 q_1) \bmod 158$

$= (1 - 156 \times 1) \bmod 158$

$= -155 \bmod 158$

$= 3$

$\therefore 53^{-1} \bmod 158 = 3$

Computational Aspects

Mangal
Date / /20

Encryption and Decryption

* If exponentiation is done and finally modulo is applied, the values would be very large. Instead, every intermediate value may be applied modulus

* Efficiency of exponentiation :-

Key Generation

* Selection of two large prime numbers will prevent their discovery by exhaustive methods. To select such a large prime number, an odd number is selected at random and tested if it is prime. Miller Rabin algorithm chooses two values n and a and performs calculations to find if n is prime. If n fails the test, n is not prime. If n passes the test, n may be prime / non-prime.

The procedure for picking a prime number is as follows.

1) Pick an odd integer n at random.
2) Pick an integer a<n at random
3) Perform probabilistic primality test. If n fails the test, reject n and goto step 1
4) If n has passed sufficient number of tests, accept n. Otherwise goto 2.

Upon finding p and q, calculation of e and d is done wherein two relatively prime numbers must be found using extended Euclidean algorithm. Probability that two numbers are relatively prime is 0.6.

56

Security of RSA

Mangal
Follows 120

Three possible attacks on RSA are as follows:

- Brute force: Trying all possible private keys
- Mathematical attack: Factoring the product of two primes
- Timing attack: Depends on running time of decryption.

Factoring Problem

* If n is factored into two prime nos, $\phi(n)$ can be calculated and d can be determined
* Without knowledge of p and q, if $\phi(n)$ can be calculated, again d can be found
* Find d directly without finding $\phi(n)$.

Earlier, factorization was done using quadratic sieve method. Later, Generalized Number Field Sieve (GNFS) was used which could factor RSA 129 at only 20% of the computing effort. Now, Special Number Field Sieve (SNFS) can factor the numbers more easily. In near future, RSA key size is expected to be 1024-2048 bits.

In order to prevent finding p and q, the following constraints can be imposed.

- p and q should differ in length by only few digits

* Both $(p-1)$ & $(q-1)$ must have a large prime factor

Mangal
Date / /20

* gcd $(p-1, q-1)$, must be small.

If $e < n$ and $d < n^{1/4}$, d can be determined easily

Timing Attacks

It is based on guessing a private key by keeping track of how long a computer takes to decipher messages

Simple countermeasures include:

- Constant exponentiation time: All exponentiations take same amount of time
- Random delay: Confuse the attack by adding random delay to exponentiation time.
- Blinding: Confuse the attacker by multiplying ciphertext with a random number.