UNIT-I

INTRODUCTION & NUMBER THEORY

Introduction - 1
Services - 7
Mechanisms - 8
Attacks - 4
OSI security architecture - 3
Network security model - 9

Classical encryption techniques - 11
- Symmetric cipher model - 11
- Substitution techniques - 16
- Transposition techniques - 25
- Steganography - 27

Finite fields and number theory - 28

Groups - 29
Rings - 30
Fields - 30

Modular arithmetic - 31
Euclid's algorithm - 33

Finite fields - 34
Polynomial arithmetic - 35
Prime numbers - 37
Fermat's & Euler's theorem - 39

The Chinese remainder theorem - 40
UNIT-I

Basic terminology

Cryptography

- Cryptography is the study of techniques for ensuring secrecy & authentication of information.

- Cryptography - study of design of techniques

- Crypt analysis - this deals with the concept of defeating cryptography

Network security

It covers the use of cryptographic algorithms in network protocols and new apps.

- Computer security - refers to the security of computers against intruders & malicious 8lw.

- Information security - Information needs to be secured. The security of info needs to be against physical damage & administrative damage.
1. **Computer Security**

   It is the collection of tools to protect data and thwart hackers called computer security.

2. **Network Security**

   Used to protect the data during the transmission across the network.

3. **Internet Security**

   Security against the data when it transmitted across the Internet.

4. **OSI Security Architecture**

   OSI architecture provides a way to organize the security:
   - Security Attack
   - Security Mechanism
   - Security Service

**Threat** - It is a possible danger that can exploit vulnerability.

**Attack** - It is an intelligent act (e.g., deliberate to evade security & violate the security policy of a system).
8. Security Attack:
   * Attack is defined as an action that compromises the security of un-owned by the org.
   * It can be classified as:
     - Passive attack
     - Active attack

   Passive attack:
   * The opponent wants to obtain the info (e) being transmitted across the net involves no alteration.

   Characteristics:
   - Difficult to detect
   - Possible to prevent by encryption

   Classification:
   - Release of message contents
   - The msg to be transmitted should be prevented from eaves-dropping

   Traffic Analysis:
   - Here, the intruder watches the frequency, length of msg exchanged between the two Principals.

   Active Attacks:
   - Message altered or introduce to the
Characteristics:

- Difficult to prevent
- Detection is feasible & can be recovered from the causes.

Classification:

- Masquerade

- Replay
  - Modification of msgs
  - Denial of service
  - S/W attack

* Masquerade:

```
Bob       Internet
        ^
    msg from x that appears to be from Bob
```

when one entity pretends to be a different entity.

* The attacker captures the authentication & impersonates the sender.

* Replay

```
Internet
        ^
    capture msg from Alice later
```

www.studentsfocus.com
* The attacker captures the msg & retransmits the msg without any modification to produce unauthorized effect.

* Modification of messages:

Bob \(\rightarrow\) Internet \(\rightarrow\) Alice.

Bob modifies msg from Bob to Alice.

* The attacker captures the msg & retransmits the msg with modification (or) delays (or) reorders the msg to produce unauthorized effect.

* Denial of service:

X \(\rightarrow\) disrupt service provided by sewer

In

The service has specific target like suppress all the msgs directed to a user (or) disable the n/w, degrade the performance.

* Slow Attack:

X \(\rightarrow\) slow attacks are those which can be introduce into the systems (or)
Security Services:

1. Security service is a service provided by the protocol layer, which ensures the protection of systems or data transfer.

2. Services:

   - Authentication
     - It is the assurance that the claimed entity is the one that it claims to be.
   - Access control
     - This prevents unauthorized use of a resource.
   - Data confidentiality
     - Protection of data from unauthorized disclosure.
   - Data Integrity
     - This gives the assurance that data received are not modified, deleted, or updated.
   - Non-Repudiation
     - This provides the protection against the denial by one of the principals involved in the communication.
A **Availability**
  * Resource accessible / usable.

RFC 1828:
  * A processing or common service provided by a system to give a specific kind of protection to system resources.

**Security Mechanism**:
  * Feature designed to detect, prevent (or) recover from a security attack.
  * No single mechanism that will support all services required.
  * However, one particular element underlies many of the security mechanisms in use.
  * Cryptographic techniques.

X.800: Specific security mechanisms
  * Pervasive
  * Specific security mechanisms may be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

www.studentsfocus.com
- Encryption
  > Authentication
  > Digital signature
  > Traffic padding
  > Access control
  > Routing control
  > Data integrity
  > Notarization

* Pervasive Security Mechanisms:

* Mechanisms that are not specifically tied to any particular OSI security service (or protocol layer):
  > Trusted functionality
  > Security label
  > Event detection
  > Security audit trail
  > Security recovery

**Network Security Model:**

![Diagram of network security model with trusted third party (e.g., arbitrator, distributed, of secret info) between sender and receiver.]
Classical Encryption Techniques

Symmetric Encryption - [Symmetric cipher method]

* Symmetric encryption is a form of encryption in which encryption and decryption are performed using the same key.
* Symmetric encryption transforms plaintext into cipher text using a secret key and an encryption algorithm. Using the same key as a decryption algorithm, the plaintext is recovered from the cipher text.

It is also named as conventional.

<table>
<thead>
<tr>
<th>Private-Key</th>
<th>Single-Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Sender &amp; recipient share a key.</td>
<td></td>
</tr>
</tbody>
</table>

Ingredients:

- Plain text - original msg
- Cipher text - coded msg
- Cipher - algorithm for transforming plaintext to cipher

Key - info used in cipher, known only to sender/ receiver.
> **Encipher (Encrypt)** - converting plaintext to cipher text.
> **Decipher (Decrypt)** - reversing ciphertext from plaintext.

> **Cryptography** - study of encryption principles/methods.

> **Cryptanalysis** (code breaking) - study of principal methods of deciphering ciphertext without knowing key.

> **Cryptology** - field of both cryptography & cryptanalysis.

**Symmetric Cipher Model:**

- Plaintext
- Secret key (shared by sender & recipient)
- Encryption algorithm (AES)
- Ciphertext

**Requirements:**

- Two requirements for secure use of symmetric encryption:
  - A strong encryption algorithm.
  - A secret key known only to sender/receiver.
\[ Y = E(k, x) \]
\[ x = D(k, y). \]

Cryptography:

- It can characterize cryptographic type of encryption or not used
  - Substitution
  - Transposition
  - Product

- Number of keys used
  - Single-key (or) private
  - Two-key (or) public

- Way in which plaintext is processed
  - Block
  - Stream

Cryptanalysis:

- Objective to recover key not to

- All have general approaches
  - Cryptanalytic attack
  - Brute-force attack

If not, then succeed all key used compromised.
Crypt Analysis:

1. The crypt analytic attack depends upon the nature of the algo & little knowledge about the general characteristics of the plain text & plain & cipher text pairs.

2. Brute force attack:
   - Here the attacker tries to find out the key used for the transformation.
   - The attacker tries every possible key until an intelligible translation of cipher text into plain text is obtained.

Cryptanalytic Attacks:

* Cipher text only
  - Only know algo & cipher text, only know algo & cipher text, 

* Known Plain text
  - Known Plain text & suspect plain text & cipher text.

* Chosen Plain text
  - Choose plain text & obtain cipher text

* Chosen Cipher text
  - Select cipher text & obtain plain text
* Chosen text
* Select plaintext (or) ciphertext

to encrypt / decrypt

Requirements of Encryption algo:
1. Unconditional Security
   * No matter how much computing (or) time is available, the cipher can be broken since the ciphertext provides insufficient info to uniquely determine the corresponding plaintext.

2. Computational Security
   * Given limited computing resources (e.g., time needed for calculations is greater than age of universe), the cipher cannot be broken.

Brute force search:

Ex. Book 62 & PPT also.

* Always possible to simply by choosing one of the most basic attacks: proportional key size.
* Assuming either known / recognize plaintext.
Substitution Cipher Techniques:

*Where letters of plaintext are replaced by other letters (or) by numbers (or) symbols.*

*If plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.*

- Caesar Cipher
- Monoalphabetic
- Playfair
- Hill
- Polyalphabetic
- Autocry
- Vigenère
- One-time Pad

Caesar Cipher: (by Julius Caesar)

*First allied use in military affairs.*

*Replaces each letter by 3rd letter on.

Example:

Computer

FRPS WHU
It can define transformation as,

\[
\begin{align*}
\text{a} &\rightarrow \text{g} \\
\text{b} &\rightarrow \text{h} \\
\text{c} &\rightarrow \text{i} \\
\text{d} &\rightarrow \text{j} \\
\text{e} &\rightarrow \text{k} \\
\text{f} &\rightarrow \text{l} \\
\text{g} &\rightarrow \text{m} \\
\text{h} &\rightarrow \text{n} \\
\text{i} &\rightarrow \text{o} \\
\text{j} &\rightarrow \text{p} \\
\text{k} &\rightarrow \text{q} \\
\text{l} &\rightarrow \text{r} \\
\text{m} &\rightarrow \text{s} \\
\text{n} &\rightarrow \text{t} \\
\text{o} &\rightarrow \text{u} \\
\text{p} &\rightarrow \text{v} \\
\text{q} &\rightarrow \text{w} \\
\text{r} &\rightarrow \text{x} \\
\text{s} &\rightarrow \text{y} \\
\text{t} &\rightarrow \text{z}
\end{align*}
\]

Then,

\[
C = E(k, P) = (P + k) \mod (26)
\]

\[
P = D(k, C) = (C - k) \mod (26)
\]

Types of attack:

- Brute-force attack

Cryptanalysis of Caesar's cipher:

- Only have 26 possible ciphers
- A maps to A, B, C... Z
- A brute-force search attack

Given ciphertext, just try all 26 letters.
Do not need to recognize when the plaintext:

Mono-alphabetic cipher:

Rather than just shifting the

Could shuffle (jumble) the
Each plaintext letter maps to a different random ciphertext letter.

Hence key is 36 letters long for security.

Total of $26^3 = 4 \times 10^{26}$ keys. With so many keys, might think is secure, but would be wrong. Here, the plbm is language characteristics: language redundancy & cryptanalysis.

- Human language is redundant.
- Example: with 1st b m and 8th phd...
- Letters are not equally commonly used.
- In English E is by far the most common letter, followed by T, R, N, I, O, A, S.
- Other letters like Z, J, K, Q, W are fairly rare.
- Have tables of single, double, triple letter frequencies for various lang.

Table in book 66 & PPT.
Use in cryptanalysis:

* Key concept - monoalphabetic substitution ciphers do not change relative letter frequencies.
* Calculate letter frequencies for ciphertext.
* Compare counts/plots against known values.
* If Caesar cipher, look for common peaks/throughs.

\[ \Rightarrow \] Peaks at: A-E-I triple, NO PAIR, RST triple.

\[ \Rightarrow \] Letters at: 5K, X-Z.

* For monoalphabetic must identify each letter - tables of common double/triple letters help.

Playfair cipher:

\[ \Rightarrow \] Not even the large no. of keys in a monoalphabetic cipher provides sec.

* One approach to improving security was to encrypt multiple times.
The Playfair cipher is an example invented by Charles Wheatstone in 1854.

Rules of Encryption:

1. When

2. Playfair key matrix:

   - A 5x5 matrix of letters based on a keyword.

   - Fill in letters of keyword.

   - Fill rest of matrix with other letters.

   - Ex: MONARCHY

   

<table>
<thead>
<tr>
<th>M</th>
<th>O</th>
<th>N</th>
<th>A</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>H</td>
<td>Y</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>G</td>
<td>I</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>P</td>
<td>Q</td>
<td>S</td>
<td>T</td>
</tr>
<tr>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Z</td>
</tr>
</tbody>
</table>

   Encrypting & Decrypting:

   - Plaintext is encrypted using letters at a time.

   - If a pair is a repeated letter, repeat the letter like K K.
(a) If both letters fall in the same row, replace each with letter to right [wrapping back to start from end].

(b) If both letters fall in the same column, replace each with letter to right below it [wrapping to top from below].

Otherwise, each letter is replaced by the letter in the same row & in the column of the other letter of the word.

Example:

```
Ba 1 x 00 n

Ba 1 x 10 on [replaced by x]
```

```
<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

```

```
I/S B
8 U
P M
N A
```

www.studentsfocus.com
Matrix combination \(26 \times 26 = 676\) 

Relative frequency is not the same. Hence, frequency analysis is difficult.

Dis-adv:

- Easy to break because it has the structure of the resemblance of the plain text long.

Hill cipher:

The Hill cipher is a multi-letter cipher. Developed by Lester Hill.

* the algm takes \(m\) successive plain text letters & substitutes for \(m\) cipher text letters.

Here, \(m=3\):

\[
\begin{pmatrix}
    c_1 \\
    c_2 \\
    c_3
\end{pmatrix}
= \begin{pmatrix}
    k_{11} & k_{12} & k_{13} \\
    k_{21} & k_{22} & k_{23} \\
    k_{31} & k_{32} & k_{33}
\end{pmatrix}
\begin{pmatrix}
    p_1 \\
    p_2 \\
    p_3
\end{pmatrix}
\mod 26
\]

\[
e = kp \mod 26
\]

\[
\begin{pmatrix}
    14 & 17 & 5 \\
    21 & 18 & 21 \\
    2 & 2 & 1
\end{pmatrix}
\begin{pmatrix}
    15 \\
    25 \\
    21
\end{pmatrix}
\mod 26
\]
\[
\begin{align*}
\frac{235 + 120}{315 + 504} & \quad \text{mod} \quad 26 \Rightarrow \frac{375}{819} \\
& \quad \text{mod} \quad 26 \\
& \Rightarrow 18
\end{align*}
\]

Decryption: \( P = k^{-1} C \mod 26 \).

Adv & Disadv:

> Completely hides single letter & a frequency info.
> Easily attacked with known plain attack.

Polyalphabetic Ciphers

* It is substitution ciphers.
* It improves security using multiple cipher alphabets.
* It makes cryptanalysis harder.

More alphabets to guess & flatten frequency distribution.

* Use a key to select which alphabet is used for each letter of alphabet.
* Use each alphabet in turn.
* Repeat from start after end.
Plain: Computer

Ciphertext: FSOJMNNMW

One-Time Pad:

* If a truly random key as long as the msg is used, the cipher will be secure, called a one-time pad.
* It is unbreakable since ciphertext bears no statistical relationship to the plaintext.
* Since for any plaintext & any ciphertext there exists a key mapping one to other. It can only use the key once through.

Adv: Random op is produced for each msg
Not easy to break.

Disadv: Practically impossible to generate a random key as to the length of the msg. Key distribution is key protection.
Transposition techniques

1. It is one cipher that is for the permutation of plain text letters.
   These hide the msg by rearranging the letter order.

2. Rail Fence cipher:
   Write msg letters out diagonally a number of rows. Then read off row by row.
   Example:
   Computer science
   "Computer science"
   Adv: Cryptanalysis is difficult.

* Row Transposition ciphers:
* It is more complex. The key specifies the order in which the scrambling to be done.

Ex:
Key: 4 3 1 2
Plain-text: Computer Science
3rd column written in sequence.

Ciphertext: merpnea oecceusn.

Adv & adv:

- Easily recognized because the frog is same in both plain text & ciphertext.
- Can be made sense by more number of transposition.

Rotor Machines:

- The machine has independently rotating cylinders through which electrical pulses can flow. Each cylinder has got 91 p Pluto & 81 p Pluto with internal wiring. The 91 p wire is connected to an unique 81 p Pluto.

x. If we associate each 81 Pluto 91 Pluto with a letter of the alphabet, a single cylinder is a monoalphabetic substitution.
when each ILP key is depressed, the cylinder rotates one position. The internal connections are shifted. The wrap around is followed after 26th letter.

As the rotor machine is advantageous only when we have multiple cylinders, the ILP pin of one cylinder is connected to the ILP pin of the next.

As the cylinder which is closer to the Opr is the ILP cylinder. The ILP rotates one pin position for each key strike.

As the inner cylinder gives the ILP to the middle cylinder which is rotated by one position. The middle cylinder rotates the outer cylinders by one pin position. The DeC is uses the concept of rotor machine.

Steganography:

In steganography, the plain text is hidden. The existence of the msg is concealed.

Methods:

The text is to be stegged is read in a msg to form the sequence of
Character marking:
- Selected letters of printed (or) typed text is over written with pencil. These marks are not visible. They can be seen when the paper is held at an angle to bright light.

- Invisible Ink
- Pin Punctures
- Type Writer Correction Ribbons.

Adv:
- It is can obscure encryption use.

Dis-adv:
- High overhead to hide relatively few info bits.

FINITE FIELDS & NUMBER THEORY:

* will now introduce finite fields of increasing importance in cryptography.
- AES, Elliptic curve, IDEA, public key

* It concerns opens on "numbers".
x. Group:

x. Group is a set of elements (or) numbers denoted by \( (G, *) \) with binary opn.

x. It may be finite (or) infinite, with some opn whose result is also in the set (closure).

Axôms r obeyed:

> Associative law: \( (a \cdot b) \cdot c = a \cdot (b \cdot c) \)
> Identity element \( e \): \( e \cdot a = a \cdot e = a \)
> Inverse element \( a^{-1} \): \( a \cdot a^{-1} = e \)
> Commutative \( a \cdot b = b \cdot a \).

x. Then forms an abelian group.

Cyclic Group:

x. Define exponentiation as repeated appln of opn.

ex.: \( a^3 = a \cdot a \cdot a \)

x. Identity be: \( e = a^0 \)

x. A group is cyclic if every element is power of some fixed element. Some \( a \) is every

x. To be a generator of the
Rings:

* Rings are a set of "numbers" with two opns (addition & multiplication).
  * An abelian group with addition opn.
  * And multiplication:
    - has closure
    - is associative
    - distributive over addition
    - \( a(b+c) = ab + ac \)
  * If multiplication opn is commutative, it forms a "commutative ring".
  * If multiplication opn has an identity & no zero divisors, it forms an "integral domain".

Fields:

* A set of numbers with 2 opns,
  * abelian group for addition
  * group for multiplication
  * have hierarchy with more axioms/laws
  * group \( \rightarrow \) ring \( \rightarrow \) field
Group

Ring

Field

Integral domain

Commutative ring

Abelian group

Field: 

(A1) Closure under addition

(A2) Associativity of +

(A3) Additive identity

(A4) Additive inverse

(A5) Commutativity of addition

(M1) Closure under multiplication

(M2) Associativity of *

(M3) Distributive law

(M4) Commutativity of multiplication

(M5) Multiplicative identity

(M6) No zero divisors

(M7) Multiplicative inverse

Modular Arithmetic:

 Define modulo operation "a mod n" to be the remainder when a is divided by n.

\[ a = qn + r \]

\[ 0 \leq r < n \]

* Modulo reduction

Ex: 

\[ -12 \mod 7 \]

\[ = -5 \mod 7 \]

\[ = 2 \mod 7 \]

\[ = 5 \mod 7 \]

www.studentsfocus.com
If $a \equiv b \pmod{n}$, then:

$$a \mod n = b \mod n$$

* when divided by $n$, $a$ & $b$ have same remainder.

* Ex: $100 \equiv 34 \mod 11$.

**Operations:**

* Can perform arithmetic with residues.
* Uses a finite no. of values, & loops back from either end.
* $\mathbb{Z}_n = \{0, 1, \ldots, (n-1)\}$.

* When do addition & multiplication, & modulo reduces answer. Can do reduction at any point, i.e.,

$$a+b \mod n = \left( a \mod n + b \mod n \right) \mod n$$

**Properties:**

1. \[ \left[ (a \mod n) + (b \mod n) \right] \mod n = (a+b) \mod n \]

* Ex: \[ (11 \mod 8) + (15 \mod 8) \mod 8 \]

= $\Rightarrow 2 \mod 8 \Rightarrow 2 (11+15) \mod 8$
2. \[ (a \mod n) - (b \mod n) \mod n \]
   \[= (a - b) \mod n. \]

\[ (1 \mod 8) - (15 \mod 8) \mod 8 \]
\[= -4 \mod 8 \iff 4 \mod 8 \]
\[= -4 \mod 8 = 4. \]

3. \[ (a \mod n) \times (b \mod n) \mod n \]
   \[= (a \times b) \mod n. \]

\[ (11 \mod 8) \times (15 \mod 8) \mod 8 \]
\[= 11 \mod 8 \iff 3 \times 7 \mod 8 \]
\[= 160 \mod 8 = 5. \]

---

Euclid's algorithm

A more efficient way to find the GCD(a, b):

\[ \text{GCD}(a, b) = \text{GCD}(b, a \mod b) \]

Algorithm:

Euclidean Algorithm to compute GCD(a, b):

If \( b = 0 \) then return \( a \);
else return \( \text{GCD}(b, a \mod b) \).
Extended Euclidean Algorithm:

1. Calculates not only GCD but also \( x \) and \( y \):

\[ a \times x + b \times y = d = \gcd(a, b) \]

2. It is useful for later crypto computations.

3. Follow sequence of divisions for GCD but assume at each step \( d \), can find \( x \) and \( y \):

\[ d = a \times x + b \times y \]

4. At end find GCD value \( d \) and also \( x \) and \( y \).

5. If \( \gcd(a, b) = 1 \) these values are inverse.

Finite Fields (Galois):

1. Finite fields play a key role in cryptography.

2. It can show no of elements in a finite field must be a power of a prime \( p^n \). It can be denoted \( \text{GF}(p^n) \).

3. \( \text{GF}(p) \)

4. \( \text{GF}(2^n) \).

Galois fields \( \text{GF}(p) \):

1. \( \text{GF}(p) \) is the set of integers \( \{0, 1, \ldots, p-1\} \) with arithmetic operations...
Since have multiplicative inverses

Find inverse with extended Euclidean alg.

Hence, arithmetic is "well-behaved" and can do addition, subtraction, multiplication & division without leaving the field $GF(p)$

<table>
<thead>
<tr>
<th>$GF(7)$</th>
<th>Multiplication</th>
<th>$x^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Polynomial arithmetic

It can compute using polynomials:

$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \bmod P$

Several alternatives:

- Ordinary polynomial arithmetic
- Poly arithmetic with co-ords mod $P$
- And polynomial arithmetic
ordinary polynomial Arithmetic:

1. Add (or) Subtract corresponding co-efficients.
2. Multiply all terms by each other.

\[ f(x) = x^3 + x^2 + 2 \] and \[ g(x) = x^2 - x + 1 \]

\[ f(x) + g(x) = x^3 + 2x^2 - x + 3 \]

\[ f(x) - g(x) = x^3 + x + 1 \]

\[ f(x) \times g(x) = x^5 + 3x^3 - 2x + 2 \]

Polynomial Arithmetic with modulo co-efficients:

1. When computing value of each coefficient, do calculation modulo some value.

forms a polynomial ring.

1. Co-efficients are 0 (or) 1.

\[ f(x) = x^3 + x^2 \] and \[ g(x) = x^2 + x + 1 \]

\[ f(x) + g(x) = x^3 + x + 1 \]

\[ f(x) \times g(x) = x^5 + x^2 \]

Polynomial Division:

1. In the form, \[ f(x) = q(x)g(x) + r(x) \]

1. Can interpret \( r(x) \) as being a remainder.

\[ r(x) = f(x) \mod g(x) \]
* If $f(x)$ divides $g(x)$ and have no remainder, say $g(x)$ divides $f(x)$.

* If $g(x)$ has no divisors other than itself & 1 say it is irreducible (or prime) polynomial.

* Arithmetic modulo an irreducible polynomial forms a field.

**Number theory**

**Prime Numbers:**

* A prime has only have divisors of 1 and itself, they cannot be written as a product of other numbers.

* 1 is prime, but is generally not of interest.

* Prime numbers are central to number theory.

**Ex:** 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37

**Prime Factorisation:**

* To factor a number $n$ is to write it as a product of other numbers.

$$n = a \times b \times c$$

* Factoring a number is relatively hard compared to multiplying the factors together.
* the prime factorisation of a number $n$ is when its written as a product of primes

* $n = \prod_{i=1}^{k} p_i^{a_i}$

Ex: for $n = 10$

- complete set of residues is $\{0, 1, 2, 3, 4\}$
- reduced set of residues is $\{1, 3, 7, 9\}$

- No of elements in reduced set of residues is called the Euler Tottent $\phi(n)$

Ex: $n = 72 = 2^3 \cdot 3^2$

* $a = \prod_{p|a} p^{\alpha}$

Fermat's Theorem

- $a^{p-1} \equiv 1 \pmod{p}$

- $p$ prime & $GCD(a, p) = 1$

- $a^p = a \pmod{p}$

- $a^{\phi(n)} \equiv 1 \pmod{n}$

- When doing arithmetic modulo $n$.

- Complete set of residues $0, 1, \ldots, n-1$

- Reduced set of residues for those not relatively prime to $n$.
To compute \( \varphi(n) \) need to count no of residues to be executed.

- For \( p (p \text{ prime}) \) \( \varphi(p) = p - 1 \)
- For \( p \cdot q (p, q \text{ prime}) \) \( \varphi(p \cdot q) = (p-1)(q-1) \)

\[ \varphi(37) = 36 \]
\[ \varphi(21) = (2-1)(7-1) = 8 \times 6 = 48 \]

Euler's theorem:

A generalization of Fermat's theorem

\[ a^{\varphi(n)} \equiv 1 \pmod{n} \]

For any \( a, n \) where \( \gcd(a, n) = 1 \).

Ex:
\[ a = 2; \quad n = 10; \quad \varphi(10) = 4; \]

\[ a^4 \equiv 1 \pmod{10} \]

Hence, \( 2^4 = 16 \equiv 1 \pmod{10} \)

\[ a = 2; \quad n = 11; \quad \varphi(11) = 10; \]

\[ a^{10} \equiv 1 \pmod{11} \]

Hence, \( 2^{10} = 1024 \equiv 1 \pmod{11} \).

Also have: \( a^{\varphi(n)+1} = a \pmod{n} \).
Primality Testing

1. Often need to find large prime nos
   - Traditionally, done using trial division
     - Divide by all nos (primes) in turn less than the square root of the no.
   - Only works for small nos.

2. Alternatively, can use statistical primality tests based on properties of primes.

   - For which all primes nos satisfy property
   - But some composite nos, called pseudo-primes, also satisfy the property.
   - Can use a slower deterministic primality test.

In PPT:
   - Miller-Rabin alg
     - Probabilistic consideration
     - Prime distribution.

Chinese Remainder Theorem

- Used to speed up modulo computations
- If working modulo a product of nos

  Ex: \[ \text{mod } 15 = \text{mod } 3 \cdot 5 \]

* Chinese Remainder Theorem: How can we
  - Use each module \( m_i \) separately
**Chinese Remainder Theorem**

To compute \( A \pmod{M} \) using the Chinese Remainder Theorem (CRT) in several ways:

1. First compute all \( \bar{a}_i = a_i \pmod{m_i} \) separately,
2. Then combine results to get answer.

\[
A = \left( \sum_{i=1}^{k} a_i \cdot c_i \cdot (m_i^{-1}) \pmod{M} \right)
\]

where \( c_i = M / m_i \) have a modulo \( m_i \) for \( 1 \leq i \leq k \).

\( k \) is the smallest power of a known to be a primitive root. Since this is faster than working in the full modulus \( M \).

---

**Primitive Roots**

Consider for \( m = \varphi(n) \), the Euler's theorem have a modi:

\[
\varphi(n) = \prod_{p \mid n} (p - 1)
\]

\( \bar{a}_i \) is called a primitive root of \( n \).

---

**Tober**

- once found a \( m = \varphi(n) \) then \( \bar{a} \) is called a primitive root of \( n \).

---

**GCD**

- there exist for \( m = \varphi(n) \), having \( \gcd(a, m) = 1 \).

---

**Modulo**

- \( a \equiv b \pmod{m} \) always valid.

---

**www.studentsfocus.com**
Discrete Logarithms

1. The inverse problem to exponentiation is to find the discrete logarithm of an element \( a \) modulo \( p \) such that,

\[
\log_p b = x \mod p
\]

written as, \( x = d \log_a b \mod p \).

2. If \( a \) is a primitive root then it always exists, otherwise it may not.

Example:

\[
x = \log_{13} 4 \mod 13 \text{ has no answer}
\]

\[
x = \log_{12} 3 \mod 13 = 4 \text{ by trying successive powers.}
\]

3. Whilst exponentiation is relatively easy, finding discrete logarithms is generally a hard problem.

<table>
<thead>
<tr>
<th>Discrete Logarithms mod ( 19 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base 2</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logarithm</th>
<th>1  2  3  5  6  7  8  9  10 11 12 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_{10} 1 )</td>
<td>1 13 2 11 12 10 9 8 7 6 5 4</td>
</tr>
<tr>
<td>( \log_{10} 2 )</td>
<td>5 12 9 10 7 8 6 11 13 2 1</td>
</tr>
<tr>
<td>( \log_{10} 3 )</td>
<td>13 1 9 2 10 11 8 7 6 5 4 3</td>
</tr>
</tbody>
</table>

www.studentsfocus.com
<table>
<thead>
<tr>
<th>base</th>
<th>log₂n(a)</th>
<th>18</th>
<th>7</th>
<th>1</th>
<th>4</th>
<th>8</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Note: The table represents values of log₂n(a) for different bases, with a specific example for base 3 modulo 19.