

CNS Assignment 2

1. Explain in detail about Euler's Totient function?

Ans.1)

Euler's Totient Function, often denoted by $\phi(n)$, is a mathematical function that counts the number of positive integers less than or equal to n that are coprime (relatively prime) to n . The term "coprime" refers to numbers that have no common factors other than 1.

The Euler's Totient Function is named after the Swiss mathematician Leonhard Euler, who made significant contributions to number theory.

Euler's Totient Function Formula:

$$\phi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdot \dots \cdot \left(1 - \frac{1}{p_k}\right)$$

Here,

n is a positive integer, and p_1, p_2, \dots, p_k are the distinct prime factors of n .

Properties and Observations:

1. Euler's Product Formula:

- The formula expresses $\phi(n)$ as a product over the prime factors of n .

2. Coprime Relationship:

- $\phi(n)$ gives the count of positive integers coprime to n .

3. Special Cases:

- For a prime number $\phi(p)=p-1$, as all numbers less than p are coprime to p .
- For $n=p \cdot q$ (product of two distinct primes), $\phi(n)=(p-1) \cdot (q-1)$.

4. Euler's Theorem:

- Euler's Totient Function is used in Euler's Totient Theorem, which states that

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

if a and n are coprime.

2.a) In public key system using RSA you intercept the cipher text $C = 10$ sent to the user whose public key is $e = 5$, $n = 35$, what is the plaintext M ?

Ans.2.a)

In the RSA algorithm, the encryption and decryption processes are based on the modular exponentiation operation. The encryption operation is defined as $C \equiv M^e \pmod{n}$, where C is the ciphertext, M is the plaintext, e is the public exponent, and n is the modulus.

Given $C=10$, $e=5$, and $n=35$, the goal is to find the plaintext M .

The decryption operation is defined as $M \equiv C^d \pmod{n}$, where d is the private exponent. In RSA, d is calculated as the modular multiplicative inverse of e modulo (n) , where $\phi(n)$ is Euler's totient function.

To find d , we need to calculate $\phi(n)$. For $n=35$, $\phi(35)$ is calculated as follows:

$$\phi(35) = \phi(5 \times 7) = (5-1) \times (7-1) = 4 \times 6 = 24$$

Now, find d such that $5 \times d \equiv 1 \pmod{24}$.

In this case, $d=5$ because $5 \times 5 \equiv 1 \pmod{24}$.

Now, use d to decrypt the ciphertext $C=10$ and find the plaintext M :

$$M \equiv C^d \pmod{n}$$

$$M \equiv 10^5 \pmod{35}$$

Calculate M :

$$M \equiv 100000 \pmod{35}$$

$$M \equiv 5$$

So, the plaintext M is 5.

2.b) Demonstrate the working of Diffie-Hellman key exchange algorithm with suitable example.

Ans.2.b)

The Diffie-Hellman key exchange algorithm is a method for two parties to securely agree on a shared secret key over an insecure communication channel. The security of the algorithm relies on the difficulty of the discrete logarithm problem.

Here's a simplified example of the Diffie-Hellman key exchange algorithm:

Steps:

1. Initialization:

- Choose a large prime number p and a primitive root modulo p , denoted as g . These values are public and agreed upon by both parties.

Let's choose $p=23$ and $g=5$ for this example.

2. Public Key Exchange:

- Alice and Bob publicly share their choices of p and g .
- Alice chooses a private key a (a random number), and Bob chooses a private key b .

Alice's private key: $a=6$

Bob's private key: $b=15$

3. Compute Partial Keys:

- Both Alice and Bob independently compute their partial keys using the public values p and g and their private keys a and b .

Alice's partial key: $A = g^a \bmod p = 5^6 \bmod 23 = 8$

Bob's partial key: $B = g^b \bmod p = 5^{15} \bmod 23 = 19$

Now, both Alice and Bob have computed the same shared secret key ($s=2$) without explicitly exchanging it over the insecure channel. This shared secret can be used as a symmetric key for encryption and decryption in subsequent communication.

3. Explain in detail about MD5 Message Digest Algorithm.

Ans.3)

Message Digest 5:- (MD5)

- ① There are a no. of popular message digest algorithms known as MD_n for various values of n.
- ② MD5 is the most popular and is fifth in a series of message digests design by Ronald Rivest

4. Explain the Hash Functions and their Security.

Ans.4)

A hash function is a mathematical function that takes an input (or 'message') and produces a fixed-size string of characters, which is typically a hash value or hash code. The output, often a 'digest,' is unique to the specific input, and even a small change in the input should produce a substantially different hash.

Properties of Hash Functions:

1. Deterministic:
 - For the same input, the hash function will always produce the same output.
2. Fixed Output Length:
 - The output of the hash function is of a fixed size, regardless of the input size.
3. Efficient to Compute:
 - It should be computationally efficient to calculate the hash value for any given input.
4. Pre-image Resistance:
 - Given a hash value, it should be computationally infeasible to reverse the process and find an input that produces that hash.
5. Collision Resistance:
 - It should be unlikely that two different inputs produce the same hash value.

Security of Hash Functions:

1. Data Integrity:
 - Hash functions are widely used to ensure data integrity. By comparing hash values before and after transmission or storage, one can verify if the data has been altered.
2. Password Hashing:
 - Hash functions are used to store passwords securely. Rather than storing plaintext passwords, systems store the hash of the password. This way, even if the hash is compromised, the original password is not easily obtainable.
3. Digital Signatures:
 - In digital signatures, hash functions are used to generate a fixed-size representation of a message, which is then signed. The recipient can use the sender's public key to verify the signature.
4. Cryptographic Applications:
 - Hash functions are essential in various cryptographic protocols, including HMAC (Hash-based Message Authentication Code), digital certificates, and blockchain.
5. Blockchain Technology:
 - Hash functions play a crucial role in creating a chain of blocks in blockchain. The hash of a block is included in the subsequent block, ensuring the integrity of the entire chain.

Security Concerns:

1. Collision Attacks:
 - A collision occurs when two different inputs produce the same hash output. While hash functions aim to minimize the likelihood of collisions, the existence of collision-resistant hash functions is a topic of ongoing research.
2. Length Extension Attacks:
 - Some hash functions are vulnerable to length extension attacks, where an attacker can extend the hash value without knowing the original input.
3. Algorithmic Vulnerabilities:
 - Cryptographic hash functions need to resist various attacks, such as birthday attacks and differential cryptanalysis.

Common Hash Functions:

1. MD5 (Message Digest Algorithm 5):
 - Previously widely used but now considered insecure due to vulnerabilities.
2. SHA-1 (Secure Hash Algorithm 1):
 - Also deprecated due to vulnerabilities; SHA-256 and SHA-3 are more secure alternatives.
3. SHA-256 (Secure Hash Algorithm 256-bit):
 - Part of the SHA-2 family, commonly used in blockchain and other security protocols.
4. SHA-3 (Secure Hash Algorithm 3):
 - The latest member of the Secure Hash Algorithm family, designed to provide the same security as SHA-2.

5. Explain in detail about Application Gateway firewall.

Ans.5)

An application gateway is a network device or service that provides application-layer functions, such as protocol translation, content filtering, and load balancing. It operates at the application layer (Layer 7) of the OSI model and can handle various protocols and applications.

Key functions of an application gateway may include:

1. Protocol Conversion:
 - Translating between different network protocols to enable communication between systems that use different protocols.
2. Load Balancing:
 - Distributing incoming network traffic across multiple servers to ensure no single server is overwhelmed, improving reliability and performance.
3. SSL Termination:
 - Handling the encryption and decryption of SSL/TLS traffic, offloading this process from backend servers.
4. Caching:
 - Storing frequently accessed data in a cache to reduce latency and improve response times.
5. Content Filtering:
 - Inspecting and filtering content based on predefined rules or policies.

Firewall in Computer Networks:

A firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predetermined security rules. The primary goal of a firewall is to establish a barrier between a trusted internal network and untrusted external networks, such as the internet.

Key functions of a firewall may include:

1. Packet Filtering:
 - Examining packets of data and allowing or blocking them based on predefined rules. This is often done at the network layer (Layer 3) using rules based on source and destination IP addresses and ports.
2. Stateful Inspection:
 - Keeping track of the state of active connections and making decisions based on the context of the traffic. Stateful firewalls operate at both the network and transport layers (Layers 3 and 4).
3. Proxy Services:
 - Acting as an intermediary between clients and servers, forwarding requests and responses to enhance security and privacy.
4. Network Address Translation (NAT):
 - Modifying network address information in packet headers to allow multiple devices in a private network to share a single public IP address.

6. Explain in detail about SQL injection?

Ans.6)

SQL injection is a type of cyberattack that occurs when an attacker is able to insert or manipulate malicious SQL code into a query, typically through user inputs in web applications or other software that interacts with a database. The goal of SQL injection attacks is to manipulate the SQL queries executed by the application's database, often leading to unauthorized access, data manipulation, or other malicious activities.

How SQL Injection Works:

1. User Input Vulnerability:
 - SQL injection usually occurs when a web application doesn't properly validate or sanitize user inputs before including them in SQL queries.
2. Malicious Input:
 - An attacker submits specially crafted input, often in the form of SQL code, into input fields or parameters expected by the application.
3. Injection Points:
 - The attacker aims to exploit injection points where user input is directly concatenated into SQL queries.
4. Manipulating SQL Queries:
 - The injected SQL code becomes part of the query executed by the database server, leading to unintended and potentially harmful consequences.

Types of SQL Injection:

1. Classic SQL Injection:
 - Occurs when attackers inject malicious SQL code into user input fields, such as login forms or search boxes.
2. Blind SQL Injection:
 - Attackers infer information from the database by injecting queries that result in true or false conditions. The application's response helps them deduce the structure or content of the database.
3. Time-Based Blind SQL Injection:
 - Similar to blind SQL injection, but the attacker induces the server to wait for a specified time before responding, revealing information based on the delay.
4. Union-Based SQL Injection:
 - Involves injecting a SQL UNION statement to combine the results of the original query with results from another query, allowing attackers to extract data.
5. Error-Based SQL Injection:
 - Exploits SQL errors generated by the application to extract information about the database structure or content.

Preventing SQL Injection:

1. Parameterized Queries:
 - Use parameterized queries or prepared statements that separate SQL code from user input.
2. Input Validation and Sanitization:
 - Validate and sanitize user inputs to ensure they conform to expected formats and don't include malicious code.
3. Least Privilege Principle:
 - Limit database user permissions to the minimum necessary for the application to function, reducing the potential impact of an SQL injection attack.
4. Use ORM (Object-Relational Mapping) Libraries:
 - ORM libraries often provide abstraction layers that help prevent direct SQL injection by handling SQL queries behind the scenes.
5. Stored Procedures:
 - Use stored procedures with parameterized inputs to encapsulate SQL logic and minimize the risk of injection.
6. Web Application Firewalls (WAF):
 - Implement a WAF to filter and monitor HTTP traffic, blocking known SQL injection patterns.
7. Regular Security Audits:
 - Regularly audit and test web applications for security vulnerabilities, including SQL injection, to identify and address potential risks.