Web Security

Web Security Overview

Why Is Web Security Important?

- Users expect their **private data** to be protected.
- Users trust that service providers will **not misuse or share** their data.
- Service providers must:
    - Prevent attackers from stealing sensitive information
    - Block unauthorized access or changes to data
    - Maintain service availability to avoid financial losses

Key Areas of Web Security

1. Client Security

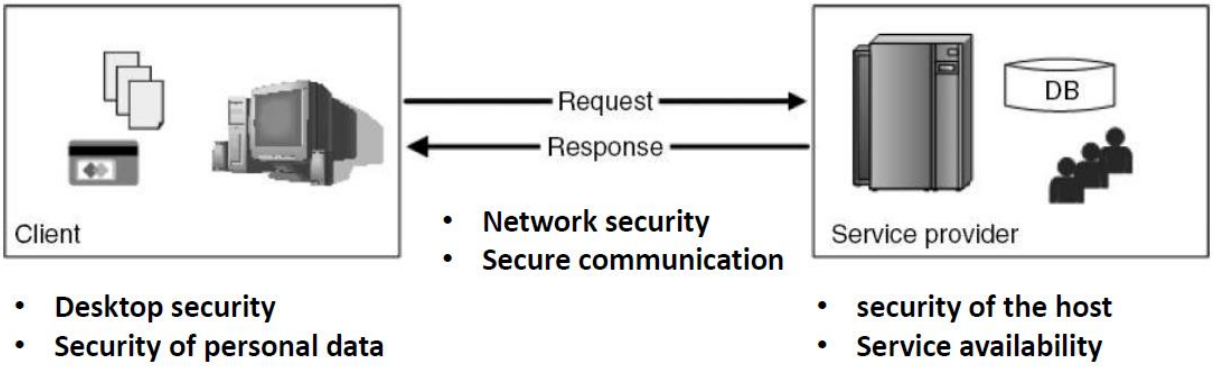
- Protect the user's computer from threats
- Secure personal data stored on the device

2. Network Security

- Ensure data is safely transferred
- Prevent interception or tampering during communication

3. Server Security

- Protect server infrastructure and databases
- Ensure continuous access to the web service



3 key areas where security is needed:

1. **Client Side (User's Device)**
    - Desktop security
    - Protection of personal data (e.g., passwords, credit card info)
2. **Network (Between Client and Server)**
    - Network security
    - Secure communication (like HTTPS encryption)
3. **Service Provider (Server Side)**
    - Server/host protection
    - Ensuring service availability (avoid crashes or downtime)

Security Aspects (Core Concepts)

Confidentiality

- Ensures communication between client and server is private
- Prevents third parties from reading sensitive information
- Common technique: Data encryption (e.g., SSL/TLS)

Integrity

- Guarantees that data is not changed during transmission
- Ensures information is delivered exactly as sent

Non-repudiation

- Prevents users or systems from denying actions they performed
- Example: A user placing an online order cannot deny doing so

Authentication

- Verifies the identity of a user or system
- Typically done using login credentials like username and password

Authorization

- Determines what actions an authenticated user can perform
- Controls access to different parts of the system based on roles

Availability

- Ensures the system is always accessible when needed
- Downtime can lead to service disruption and financial loss

Privacy

- Requires proper handling and protection of user data
- Prevents unauthorized sharing or misuse of personal information

Data Encryption and Decryption

What is Encryption?

- **Encryption** is a method to **protect data** so that only the **intended person** can read it.
- It **transforms original data** (called **plain text**) into an unreadable format (called **cipher text**).
- This is done using **mathematical functions or keys**.

Why is Encryption Used?

- To **secure communication**
- To protect data from **hackers or third parties**

Types of Encryption

1. One-Way Encryption

- Also called **hashing**
- Converts data to a unique code that **cannot be reversed**
- Commonly used to store **passwords**
- Example: **SHA-256**, **MD5**

2. Two-Way Encryption

- Can be **reversed** back to the original data using a key
- Two types:

  Symmetric Encryption

  - **Same key** is used to encrypt and decrypt
  - Fast and efficient
  - Example: **AES**, **DES**

  Asymmetric Encryption

  - Uses **two keys**: one **public** and one **private**
  - Public key encrypts, private key decrypts
  - Example: **RSA**

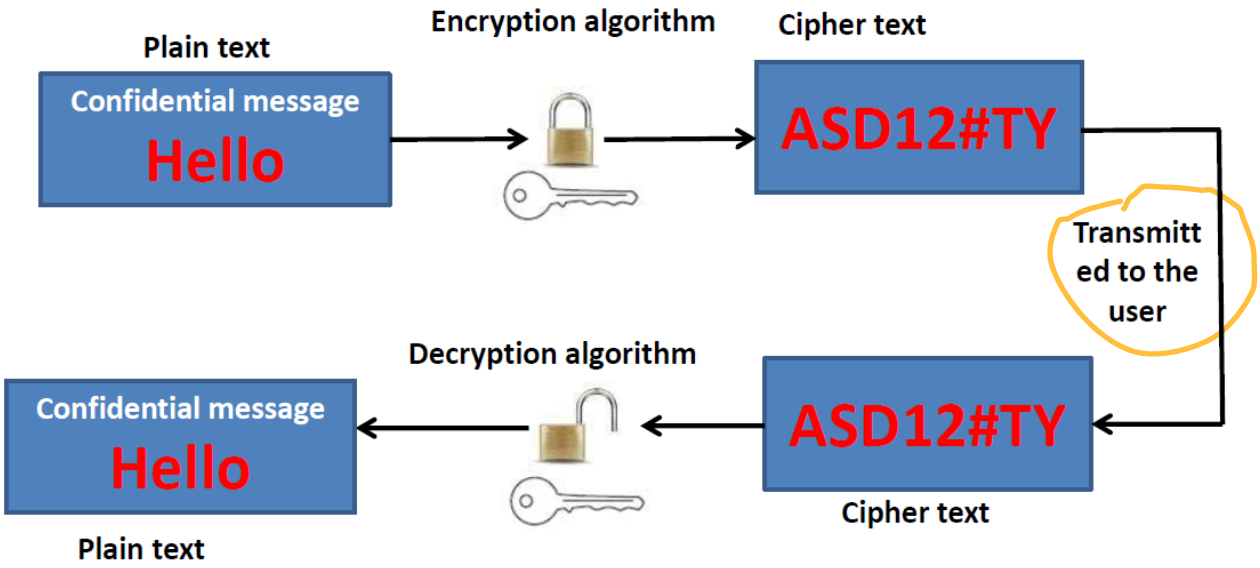
What is Decryption?

- **Decryption** is the **reverse process** of encryption.

- It **changes the secret (encrypted) message** (cipher text) **back into its original form** (plain text).
- Only someone with the **correct key** can do this.

Example:

- Encrypted message: XHU#32@JK
- After decryption: Hello123



Encryption and Decryption Process

Step 1: Plain Text (Original Message)

- The message starts as readable text.
- Example: "Hello" (confidential message)

Step 2: Encryption

- The message goes through an **encryption algorithm**.
- A **key** is used to lock (secure) the message.
- Result: Encrypted message called **cipher text** (e.g., "ASD12#TY")
- Now the message is **not readable** by others.

Step 3: Transmission

- The encrypted message is **sent over the network**.
- Even if someone intercepts it, they **can't understand** the message without the key.

◇ Step 4: Decryption

- The **cipher text** is received by the intended user.
- Using the **correct decryption key**, the cipher text is **unlocked**.
- Result: Original message "Hello" is **restored**.

Caesar Cipher

- **One of the oldest encryption techniques**
- **Used by Julius Caesar** to send secret military messages

How it Works:

- Each letter in the message is **shifted 3 positions forward** in the alphabet

For example:
A → D
B → E
C → F
D → G
E → H
F → I
G → J
H → K
...and so on

- Example:

Original Message:
BURN THE BRIDGE

Encrypted Message:
EXUQ WKH EUKFIG

Each letter is shifted:

- B → E
- U → X
- R → U
- N → Q
- ...and so on

Cryptographic Algorithms

What are Cryptographic Algorithms?

- These are **mathematical methods** used to **encrypt (lock)** and **decrypt (unlock)** data.
- They **rely on keys** — a key is a **secret code** needed to secure or open the message.

Importance of Keys

- Without the correct key, it is **nearly impossible** to break the encryption.
- A **strong algorithm** is one where the **only way to break it** is by trying **every possible key** (called a **brute force attack**).

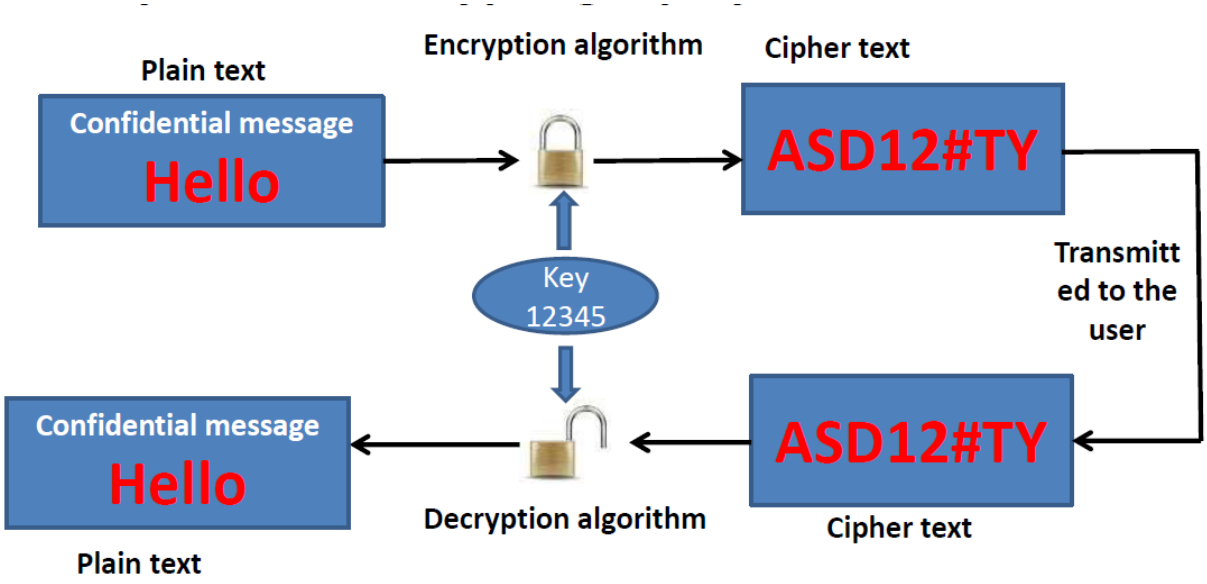
Symmetric Cryptography

Features:

- It uses **two-way encryption**.
- The **same key** is used for both **encryption** and **decryption**.
- That's why it's also called **private key cryptography**.

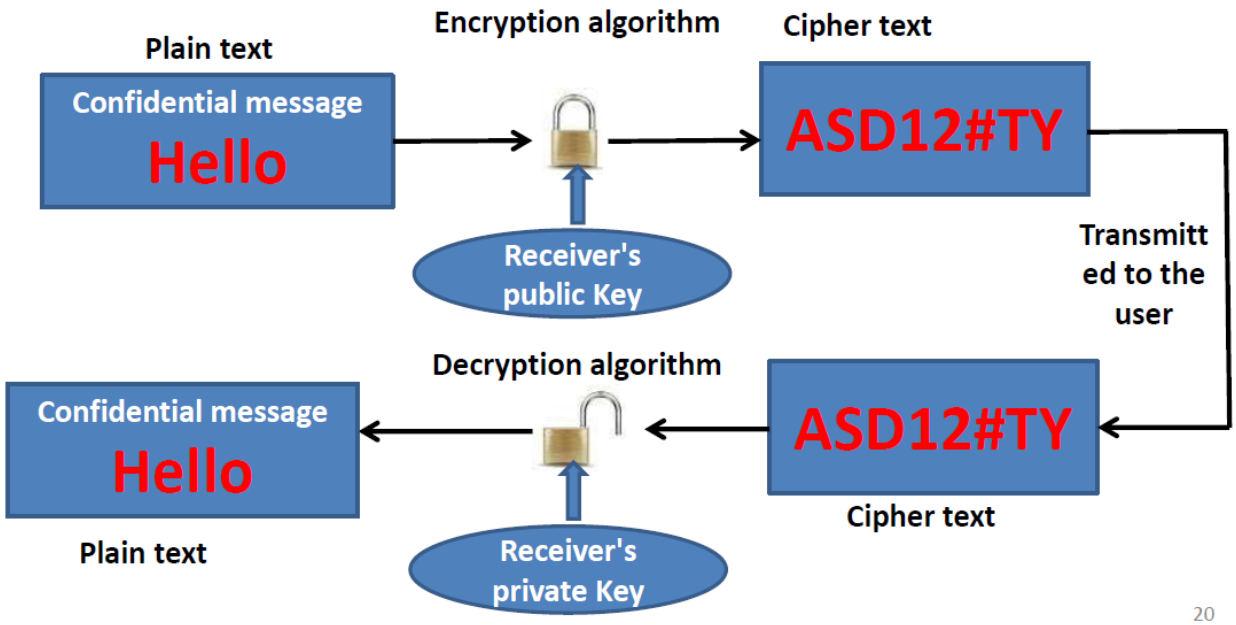
◇ Examples:

- **DES (Data Encryption Standard)**
- **AES (Advanced Encryption Standard)**



Asymmetric Cryptography (Public Key Cryptography):

- Uses **two keys**:
  - **Public Key:** Known to everyone, freely shared.
  - **Private Key:** Known only to the message recipient, kept secret.
- Enables secure communication without sharing a secret key beforehand.
- Example algorithm: **RSA**.



Hashing Algorithms

- **One-way process:** You convert data into a hash (digest), but it's very difficult or impossible to reverse back to the original data.
- **Hash (digest):** A unique "signature" that represents the content of the data. Even a small change in data produces a different hash.

Digital Signatures

- Purpose: To ensure an electronic document is **authentic**, maintaining:
  - **Integrity:** The document hasn't been changed.
  - **Non-repudiation:** The sender cannot deny sending it.

How Digital Signatures Are Created

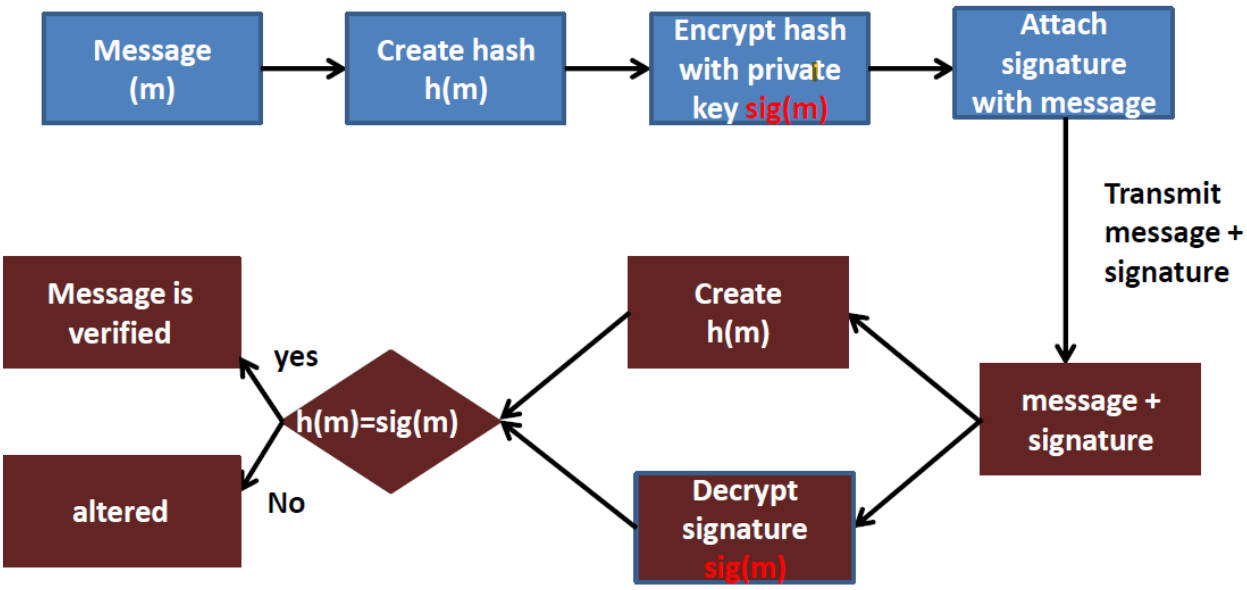
1. Sender creates a hash of the message.

2. Sender encrypts the hash with their **private key** (this encrypted hash is the digital signature).
3. Sender attaches the digital signature with the message.

How Digital Signatures Are Verified

1. Receiver decrypts the digital signature using the sender's **public key** (to get the hash).
2. Receiver creates a new hash from the received message.
3. Receiver compares the decrypted hash and the newly created hash.

- If they match, the message is authentic and unchanged.



Cryptography Ensures:

- **Confidentiality:**

  Only authorized parties can read the information.
- **Integrity:**

  The data is not altered or tampered with during transmission.

- **Availability:**

  The data and systems are accessible when needed.
- **Authenticity:**

  Verifying the identity of the sender or source of the data.
- **Non-repudiation:**

  Preventing the sender from denying they sent the message.

Securing User's Data

- **User Privacy:**
  - After data transmission, users want their data kept **private**.
  - Service providers must carefully protect data from attackers.
- **Establishing Trust:**
  - Providers can specify their **data practices** using the **Platform for Privacy Preferences (P3P)** standard.
  - Users specify their privacy preferences via a **P3P agent**.
  - P3P-enabled browsers warn users if a provider's policies conflict with their preferences.
- **Common Security Threats:**
  - **Phishing:**
    - The most common attack to steal personal info.
    - Attackers impersonate trusted companies via emails to trick users into giving personal data.
  - **Web Spoofing:**
    - Faking websites of famous companies to trick users.
  - **Adware and Spyware:**
    - Adware: Delivers unwanted advertising content.
    - Spyware: Monitors user activity and sends data to attackers.
  - **Remote Access/Backdoors:**
    - Malicious remote access to user machines can steal info, damage files, or control the system.

Viruses

- Can **damage files** or **replicate themselves** to spread.
- Often spread through **email attachments** or **sharing infected files**.

Worms

- Also **replicate themselves**, but usually without needing to attach to other files.
- They **increase network traffic** and **consume processing power**, slowing down systems.

Trojan Horses

- Do **not replicate** themselves.
- Appear as **useful or harmless programs** but perform **malicious actions** secretly.
- Aim to **steal data**, **destroy files**, or **gain unauthorized access** to systems.

Service Providers Issues

- **Goal:**
  Service providers want to **secure their servers** from attackers.

Common Attacks

- **Cross Site Scripting (XSS):**
  - Attackers **inject malicious scripts** into web pages that are dynamically generated.
  - These scripts run in users' browsers to **steal user information** like cookies or session data.
- **SQL Injection:**
  - Attackers insert **malicious SQL commands** into input fields or URLs.
  - This can lead to unauthorized access to the database, data theft, or data manipulation.