Unit – IV (Web Security)

Web Security

- Requirement, Secure Socket Layer, Transport Layer Security, and Secure Electronic Transactions.

Web Security refers to the practices, technologies, and measures used to protect websites, web applications, and web services from cyber threats and unauthorized access over the Internet.

It ensures that the data exchanged between users and websites remains confidential, secure, and unaltered.

Web security includes mechanisms like encryption (HTTPS/SSL/TLS), authentication, firewalls, and secure coding practices to prevent attacks such as phishing, malware, SQL injection, and cross-site scripting (XSS). It also focuses on protecting user information like passwords, bank details, and personal data from hackers.

By implementing strong web security, organizations can maintain data integrity, user trust, and safe online transactions, making it essential for e-commerce, banking, and all modern web-based systems.

What is Web Security?

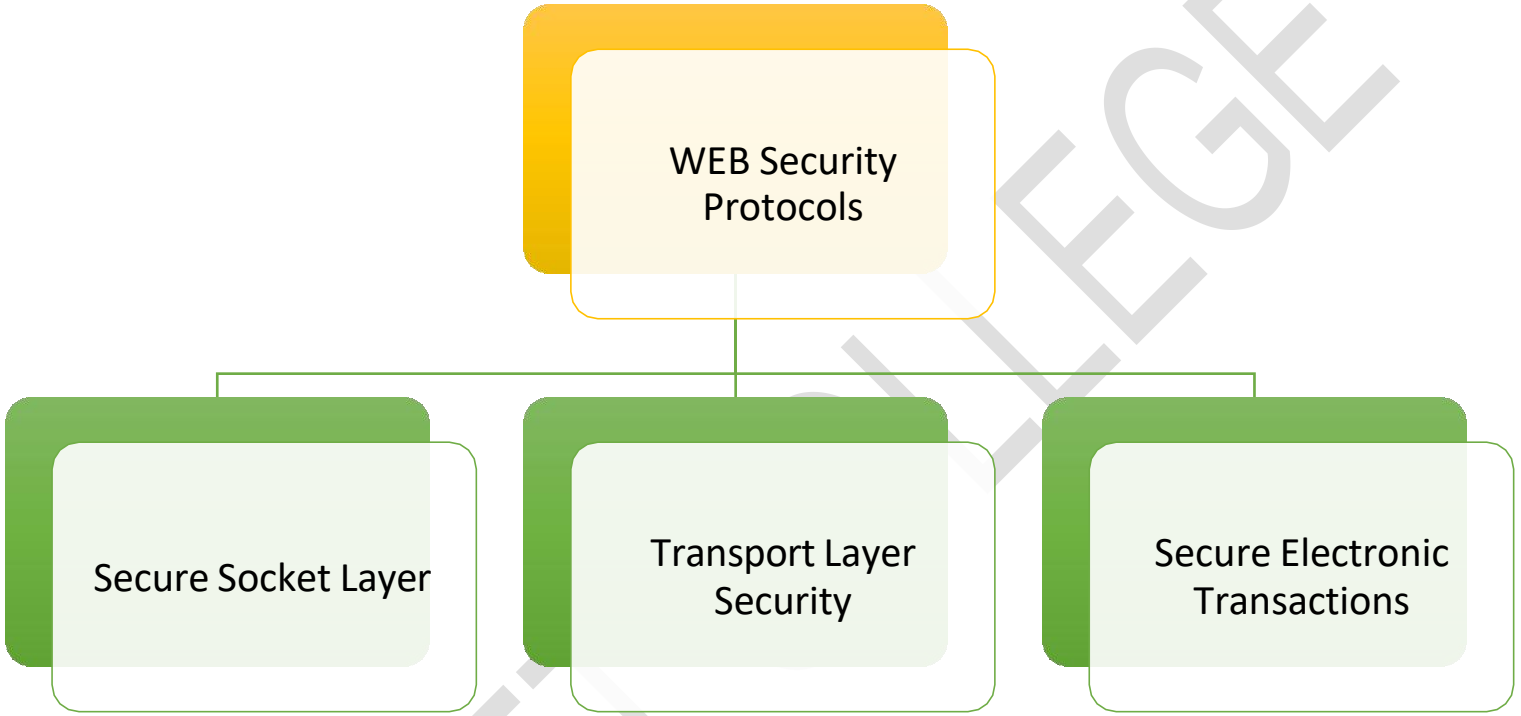
Web security focuses on securing:

- Websites
- Web applications
- Web servers
- User data (login, passwords, payment info)
- Its goal is to ensure:

- **Confidentiality** – data remains private
- **Integrity** – data is not altered
- **Availability** – services are always accessible

Threat	Meaning
Hidden Manipulation	Changing hidden fields
Parameter Tampering	Modifying URL/form data
Cookie Poisoning	Altering cookies
Stealth Commanding	Hidden command execution
Forceful Browsing	Accessing restricted URLs
Backdoor & Debug	Exploiting developer access
Configuration Subversion	Misusing wrong settings
Vendor-Assisted Hacking	Third-party software flaws

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerability to all other threats	• Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	• Encryption • Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	• Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	• Cryptographic techniques


WEB Security Protocols
Secure Socket Layer
Transport Layer Security
Secure Electronic Transactions

1. Secure Socket Layer (SSL)

What is SSL?

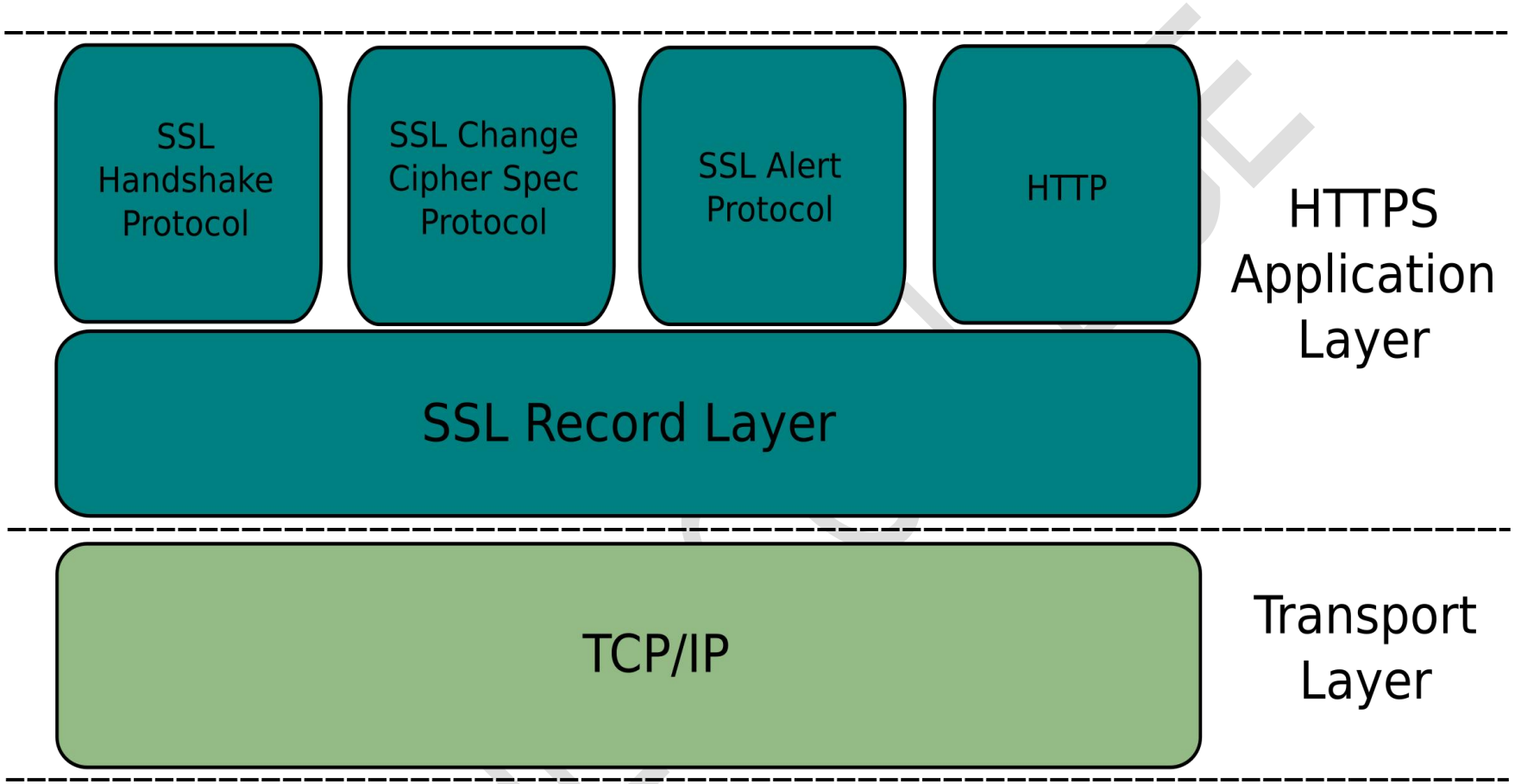
- **SSL** is a security protocol used to **encrypt data** between a web browser and web server.
- It ensures **confidentiality, integrity, and authentication**.

SSL ARCHITECTURE

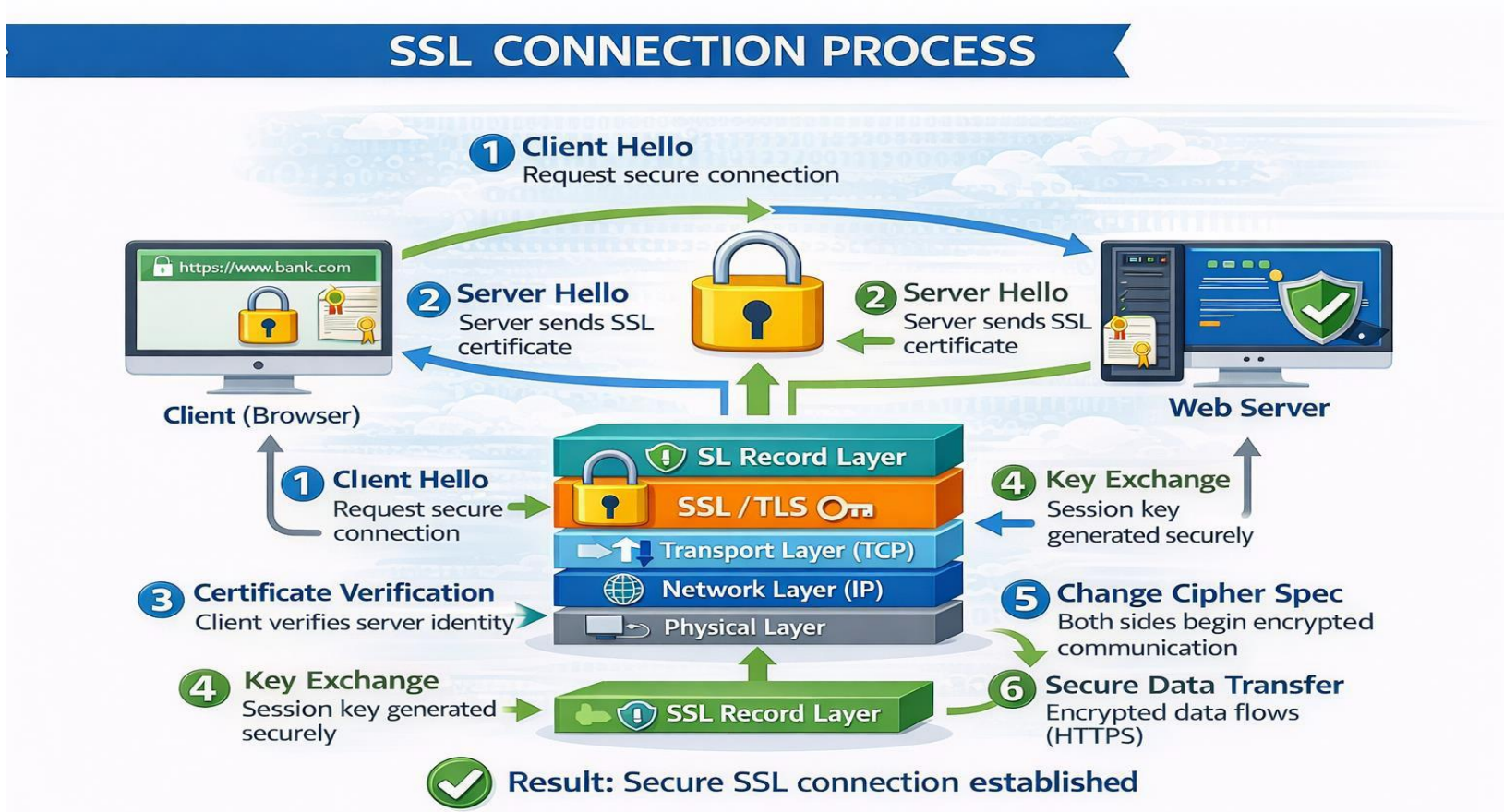
- **Position of SSL**
- **SSL lies between the Application Layer (HTTP)**and**Transport Layer (TCP/IP)**
- It provides**security services**to application protocols like HTTP, FTP, SMTP.

Components:

- **SSL Handshake Protocol:**Authentication & key exchange
- **Change Cipher Spec:**Starts encrypted communication
- **Alert Protocol:**Error & warning messages
- **SSL Record Layer:**Encrypts, decrypts, ensures integrity
- **TCP/IP:**Data transmission


SSL Handshake Protocol
SSL Change Cipher Spec Protocol
SSL Alert Protocol
HTTP
SSL Record Layer
TCP/IP
HTTPS Application Layer
Transport Layer

SSL CONNECTION PROCESS



SSL CONNECTION PROCESS

1. **Client Hello:-** Browser requests secure connection.
2. **Server Hello :-** Server sends **SSL certificate**.
3. **Certificate Verification :-** Client verifies server identity.
4. **Key Exchange :-** Session key is generated securely.
5. **Change Cipher Spec :-** Encryption is activated.
6. **Secure Data Transfer :-** Encrypted data flows (HTTPS).

SSL RECORD PROTOCOL

SSL Record Protocol is the **core protocol of SSL** that provides **secure data transfer**.

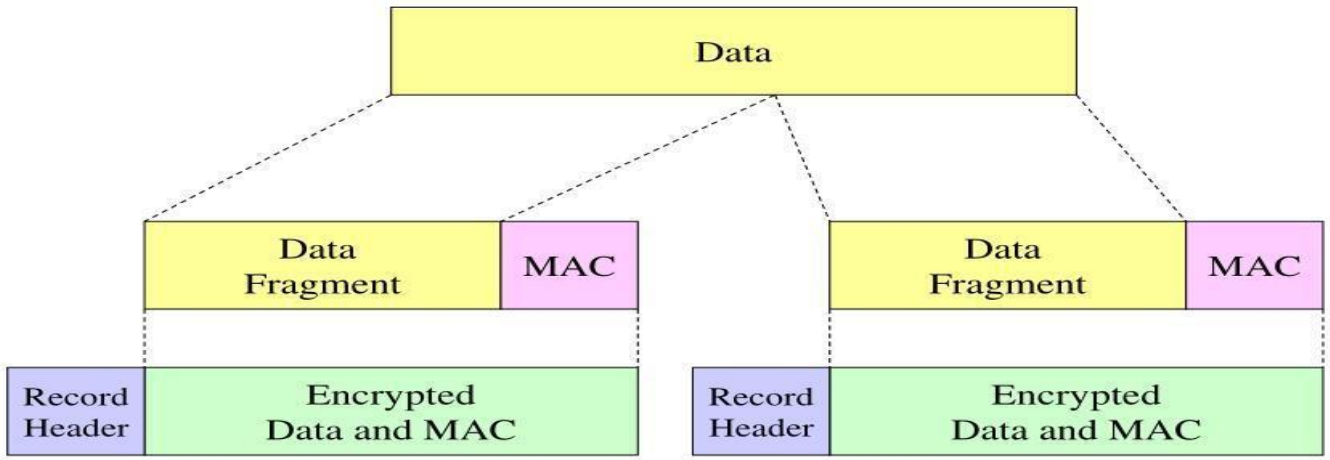
- **Functions:**

•**Fragmentation** – breaks data into blocks
•**Compression** – reduces data size (optional)
•**Encryption** – protects data confidentiality
•**Message Integrity** – uses MAC (message Authentication code)to detect changes

- **Position:**

•Works **above TCP** and **below SSL Handshake / HTTP**

SSL Record Protocol



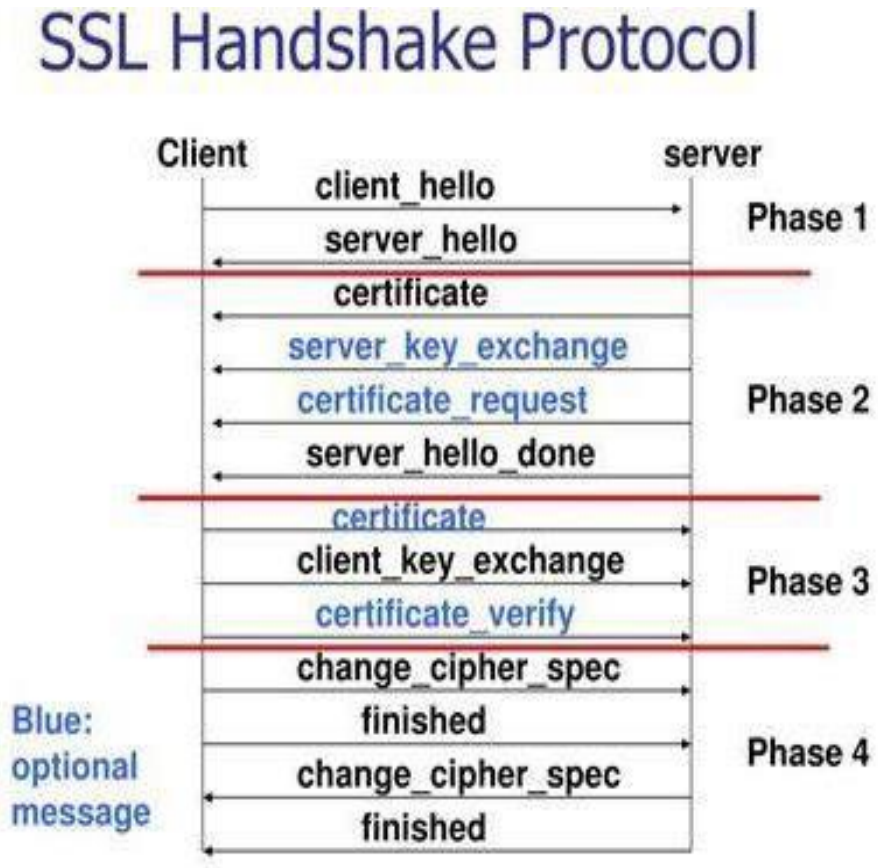
Record Header: Content type; version; length

MAC: includes sequence number, MAC key M_x

Fragment: each SSL fragment 2^{14} bytes (~16 Kbytes)

SSL RECORD PROTOCOL

- Divides data into **fragments (≤ 16 KB)**
- Adds **MAC (Message Authentication Code)** for integrity
- **Encrypts** data + MAC
- Adds **Record Header** (content type, version, length)
- Ensures **secure data transfer over TCP**



SSL Handshake Protocol, showing the sequence of messages exchanged between a **client** and a **server**.
Let's go **phase by phase**:

Phase 1: Hello
- **Client Hello** → Server: proposes protocol & cipher.
- **Server Hello** → Client: agrees on protocol & cipher.

Phase 2: Server Auth
Server Certificate → Client: proves identity.
Optional: **Server Key Exchange**, **Certificate Request**.
Server Hello Done → Client: server finished messages.

Phase 3: Client Key Exchange
Optional **Client Certificate** → Server.
Client Key Exchange → Server: share session key.
Optional **Certificate Verify** → Server.

Phase 4: Secure Connection
Change Cipher Spec → switch to encrypted mode.
Finished → confirms handshake.

SSL Alert Protocol

The **SSL Alert Protocol** is a part of the SSL/TLS protocol suite and is used to convey information about errors or warnings during a secure communication session.

Structure of an SSL Alert

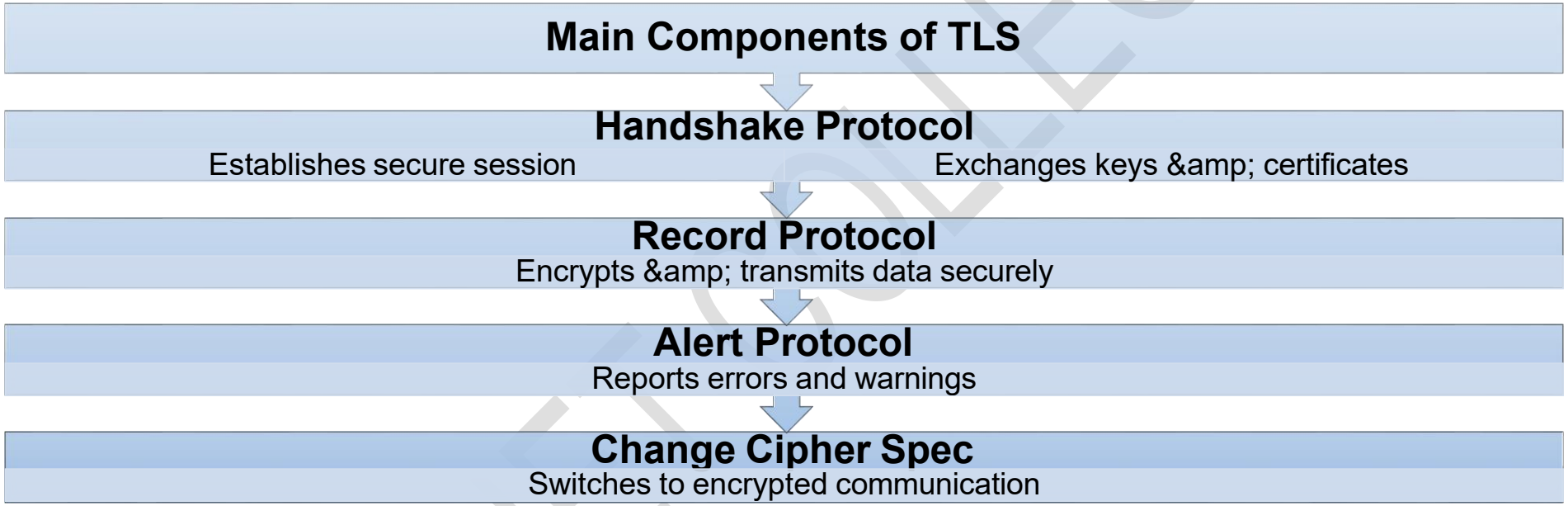
An SSL Alert message is very simple. It consists of **two bytes**:

1. **Alert Level (1 byte)**
    1. **Warning (1)** – The connection may continue.
    2. **Fatal (2)** – The connection must be terminated immediately.

2. Alert Description (1 byte)

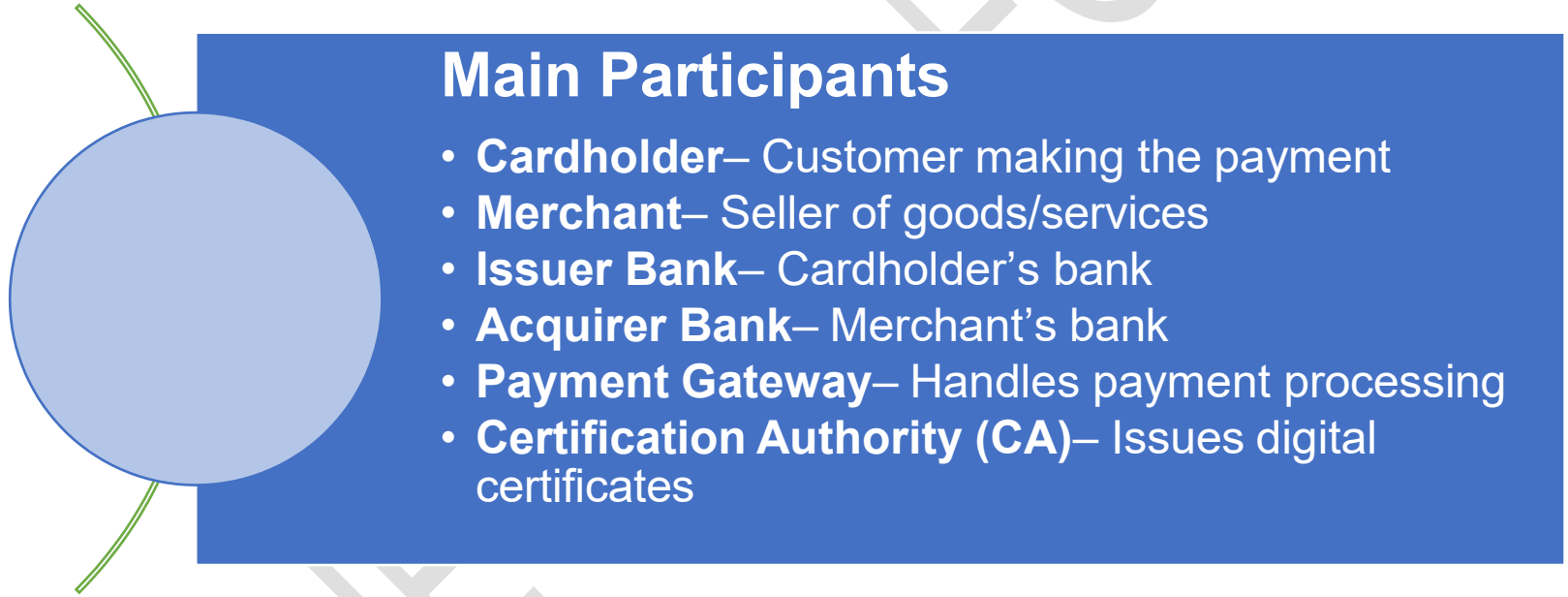
1. Specifies the type of error or warning.
2. Examples:
    1. Close notify – Connection closure notice.
    2. Unexpected message – Message received was not expected.
    3. Bad _ record_ mac – An incorrect MAC was received.
    4. Handshake failure – Problem during handshake.
    5. Protocol version – Unsupported SSL/TLS version.
    6. Certificate expired – Certificate expired.

TLS (Transport Layer Security)
TLS is a **security protocol** used to provide **secure communication** over a network (mainly the Internet). It is the **successor of SSL**.

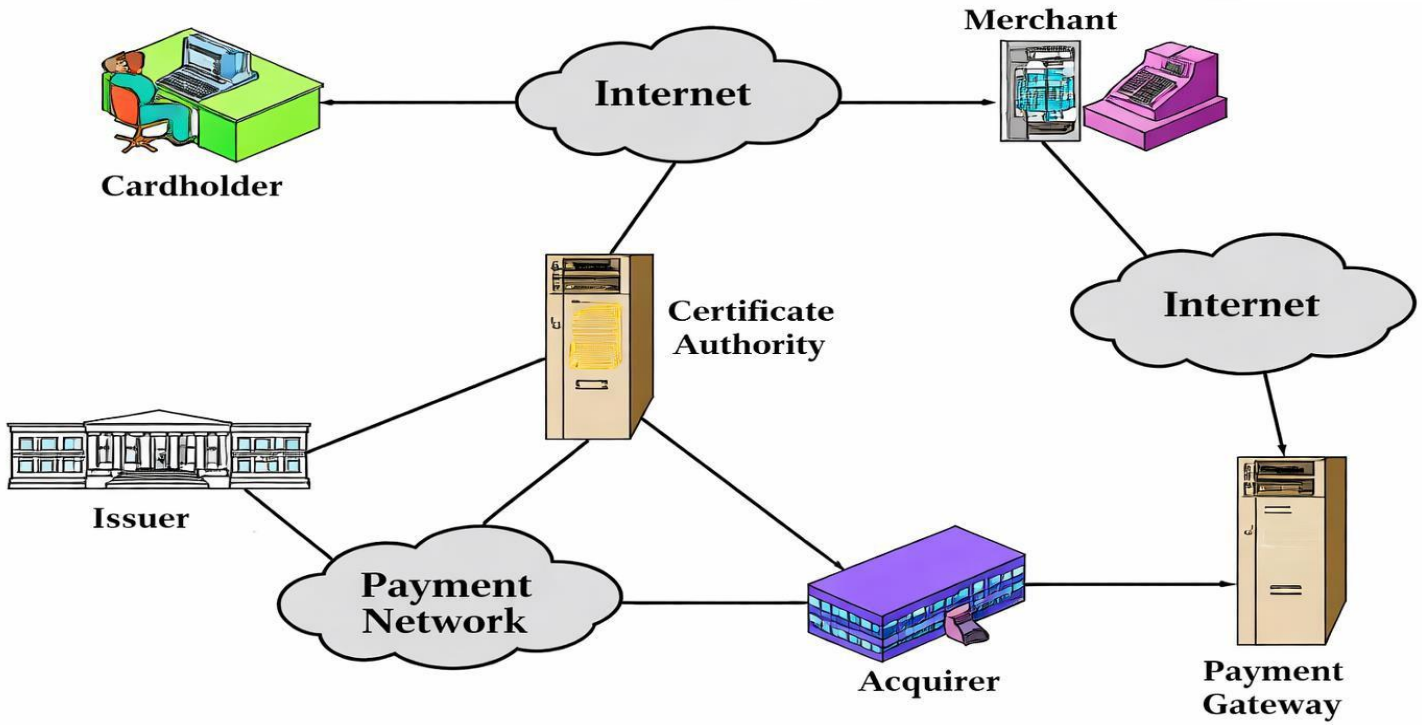


Secure Electronic Transaction (SET)

Secure Electronic Transaction (SET) is a **security protocol** developed by **Visa and MasterCard** to ensure **secure online credit card transactions** over the Internet.



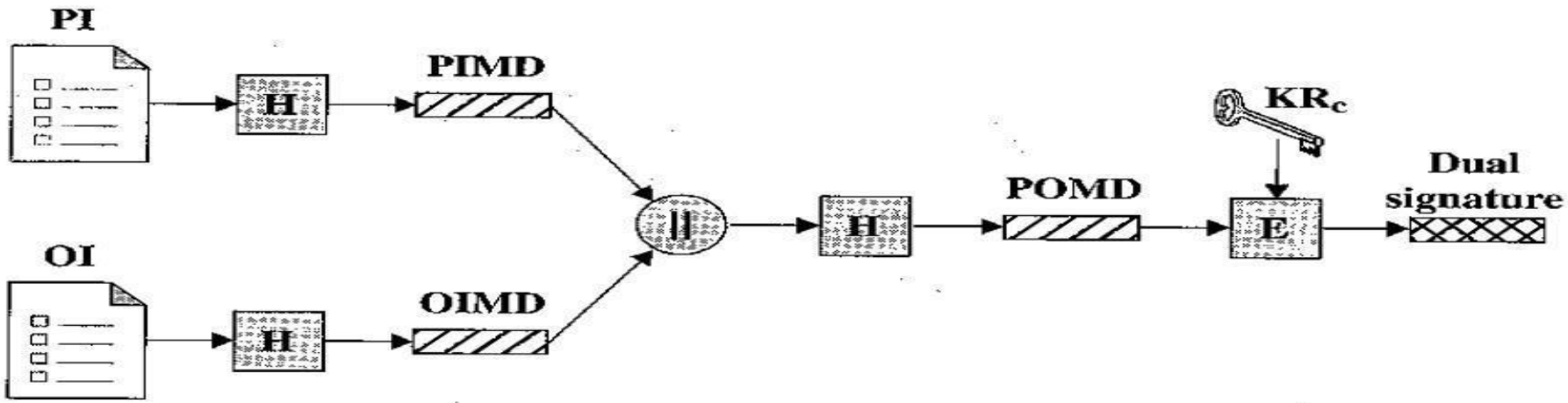
SET Components and Participants



Objectives of SET

1. **Confidentiality** – Card details are kept secret
2. **Integrity** – Data cannot be altered during transmission
3. **Authentication** – Verifies cardholder and merchant
4. **Non-repudiation** – Transaction cannot be denied later

Dual Signature



PI = Payment information
OI = Order information
H = Hash function (SHA-1)
|| = Concatenation

PIMD = PI message digest
OIMD = OI message digest
POMD = Payment order message digest
E = Encryption (RSA)
KR_c = Customer's private signature key

1. Payment Information (PI)
•PI contains **credit card details**
•A **hash function (H – SHA-1)** is applied
•Result → **PIMD (Payment Information Message Digest)**

2. Order Information (OI)
OI contains **order details (items, price)**
Hash function (H) is applied
Result → **OIMD (Order Information Message Digest)**

3. Concatenation
PIMD and OIMD are **concatenated** using ||
This joins payment and order data **without exposing contents**

4. Hashing the Combined Digest
Hash is applied again on the concatenated value
Result → **POMD (Payment Order Message Digest)**

5. Encryption (Signature Creation)
•POMD is **encrypted using customer's private key ($KR_{(C)}$)**
•Encryption algorithm used: **RSA**
•Output is the **Dual Signature**

3. SET Protocol

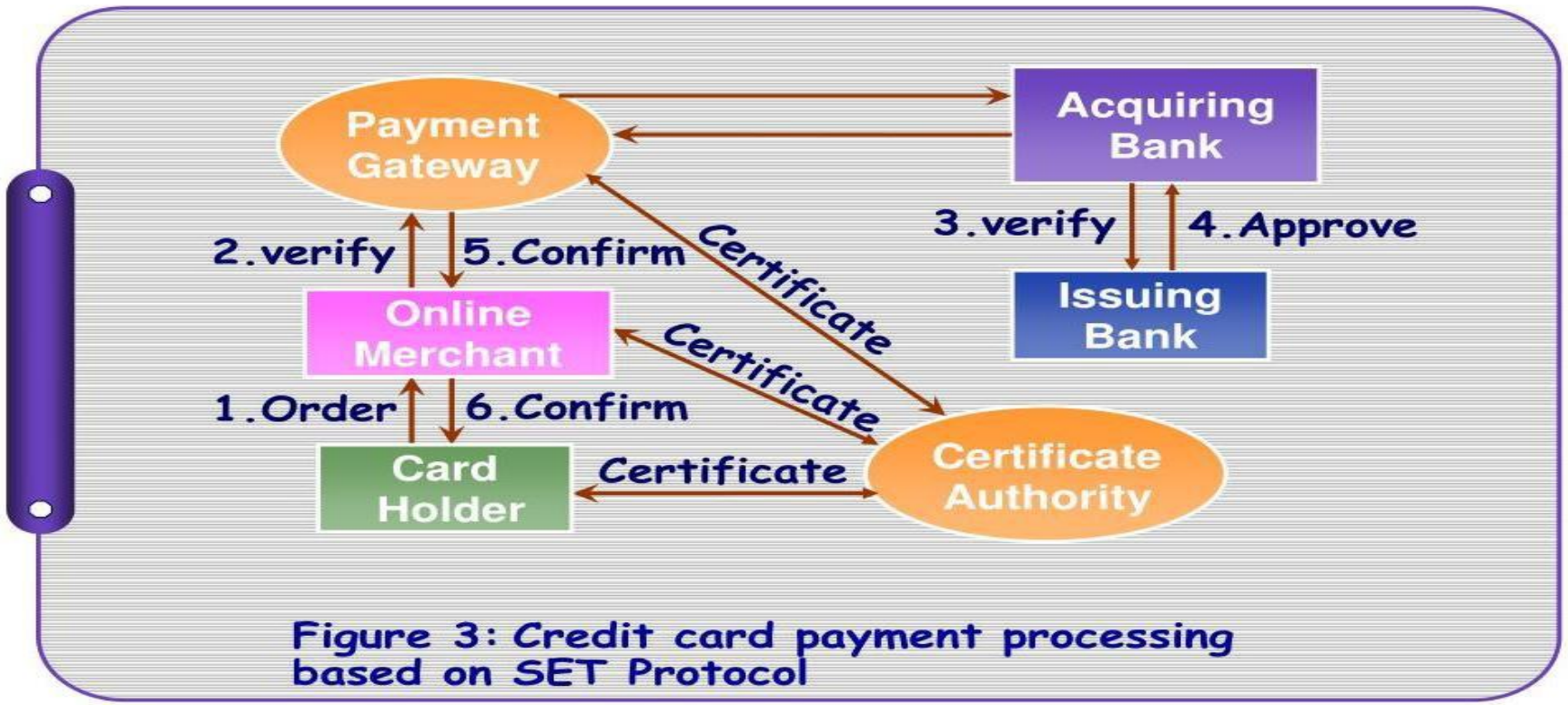


Figure 3: Credit card payment processing based on SET Protocol

- **Card Holder** – Customer who makes the payment
- **Online Merchant** – Seller
- **Payment Gateway** – Processes payment securely
- **Issuing Bank** – Customer’s bank
- **Acquiring Bank** – Merchant’s bank
- **Certificate Authority (CA)** – Issues digital certificates

Very Short Questions

1. What is the main goal of web security?
2. Define Secure Socket Layer (SSL).
3. What does TLS stand for?
4. What is the difference between SSL and TLS?
5. What is Secure Electronic Transaction (SET) used for?
6. Name one requirement of web security.
7. Which protocol provides encryption for HTTP traffic?
8. Does SSL operate at the application layer or transport layer?
9. What is a digital certificate used for in SSL/TLS?
10. Which web security protocol ensures secure online payments?

Short Questions

1. Explain the requirements of web security.
2. How does SSL ensure secure communication over the internet?
3. Describe the main features of TLS.
4. What is the role of SET in electronic commerce?
5. Compare SSL and TLS in terms of security and performance.

Long Questions

1. Discuss in detail the requirements of web security and how they are achieved using SSL/TLS.
2. Explain the working of Secure Socket Layer (SSL), including its handshake process, encryption, and certificate verification.
3. Describe Transport Layer Security (TLS) in detail. How does it improve upon SSL?
4. Explain Secure Electronic Transactions (SET) protocol and its mechanisms for ensuring secure online payments.
5. Discuss the overall web security architecture. Include SSL, TLS, SET, and their roles in protecting data and transactions over the internet.