School of Electrical Engineering and Computing

SENG2250/6250
SYSTEM AND NETWORK SECURITY

(S2, 2020)

SSL/TLS

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Outline

- OSI Security Architecture
- Secure Socket Layer (SSL)
- Web Application Security
  - *HTTPS*
  - *SSH*

What is computer network?

A computer network is a network, where two or more computers are connected together for the purpose of sharing resources or communicating data electronically.

Reference Model

- Reference models describe the layers in a network architecture :
  - *Open System Interconnection (OSI) – 7 layers.*
  - *TCP/IP reference model – 4 layers.*
- Principles for the seven layers.
  - *Layers created for different abstractions.*
  - *Each layer performs well-defined function.*
  - *Function of layer chosen with standards in mind.*
  - *Minimize information flow across layer interfaces.*
  - *Find the optimum number of layers.*

OSI and TCP/IP Models

7. Provides functions needed by users

6. Converts different representations

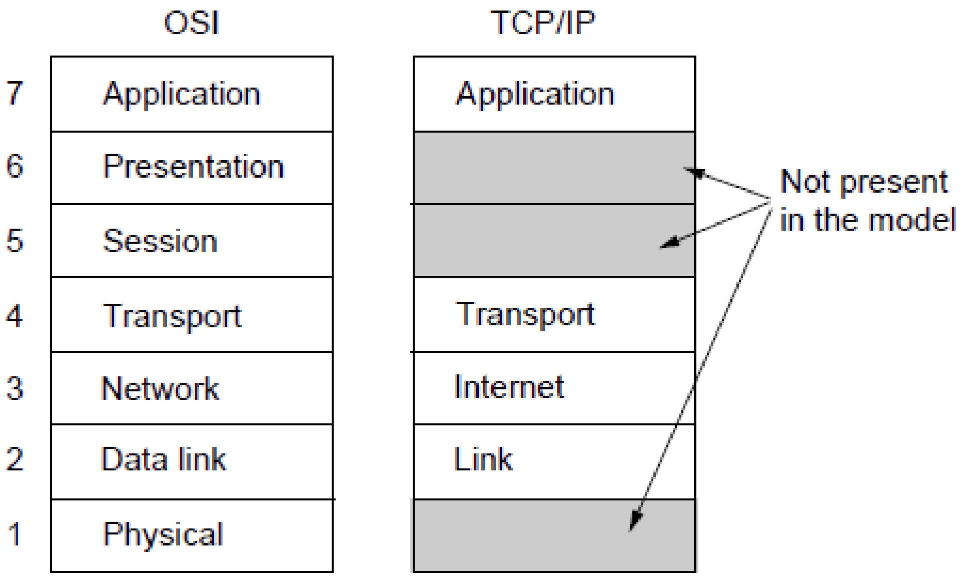
5. Manages task dialogs

4. Provides end-to-end delivery

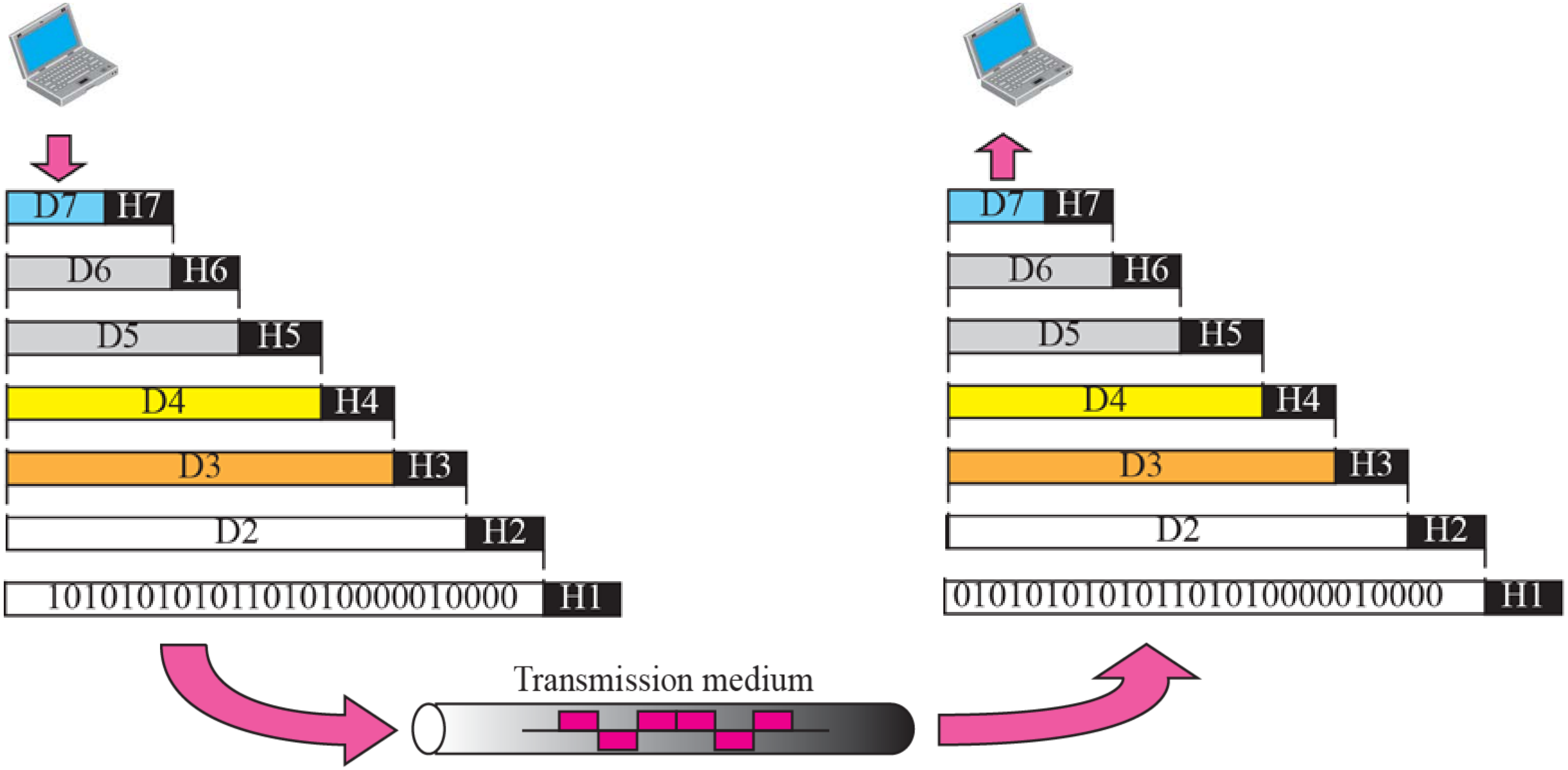
3. Sends packets over multiple links

2. Sends frames of information

1. Sends bits as signals



OSI Layers


D7
H7
D6
H6
D5
H5
D4
H4
D3
H3
D2
H2
1010101010101101010000010000
H1
Transmission medium
D7
H7
D6
H6
D5
H5
D4
H4
D3
H3
D2
H2
01010101010101101010000010000
H1

Network Security

- What are the network security threats?
- What Network Security Services are required?
- Where are these services provided?
- How are they to be managed and implemented?

Network Protocols

- Network protocols are formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a network. Network protocols govern the end-to-end processes of timely, secure and managed data or network communication.
- Network communication protocols: Basic data communication protocols, such as TCP/IP and HTTP.
- Network security protocols: Implement security over network communications and include HTTPS, SSL, IPSec and SFTP.
- Network management protocols: Provide network governance and maintenance and include SNMP and ICMP.

Network Protocols

	OSI Layer	Protocols (example)
Software	Application	HTTP, HTTPS, SMTP, FTP
	Presentation	SSL
	Session	
	Transport	TCP, UDP, SSL/TLS
	Network	IPv4, IPv6, IPSec, ARP
Hardware	Data Link	Ethernet, 802.1x, L2TP/PPTP
	Physical	

OSI Security Architecture

- ITU-T Recommendation X.800, a standard.
- It offers a systematic way of defining security requirements and characterizing the approaches to achieve these requirements
- For our purposes, it provides a useful, if abstract, overview of security concepts and measures.

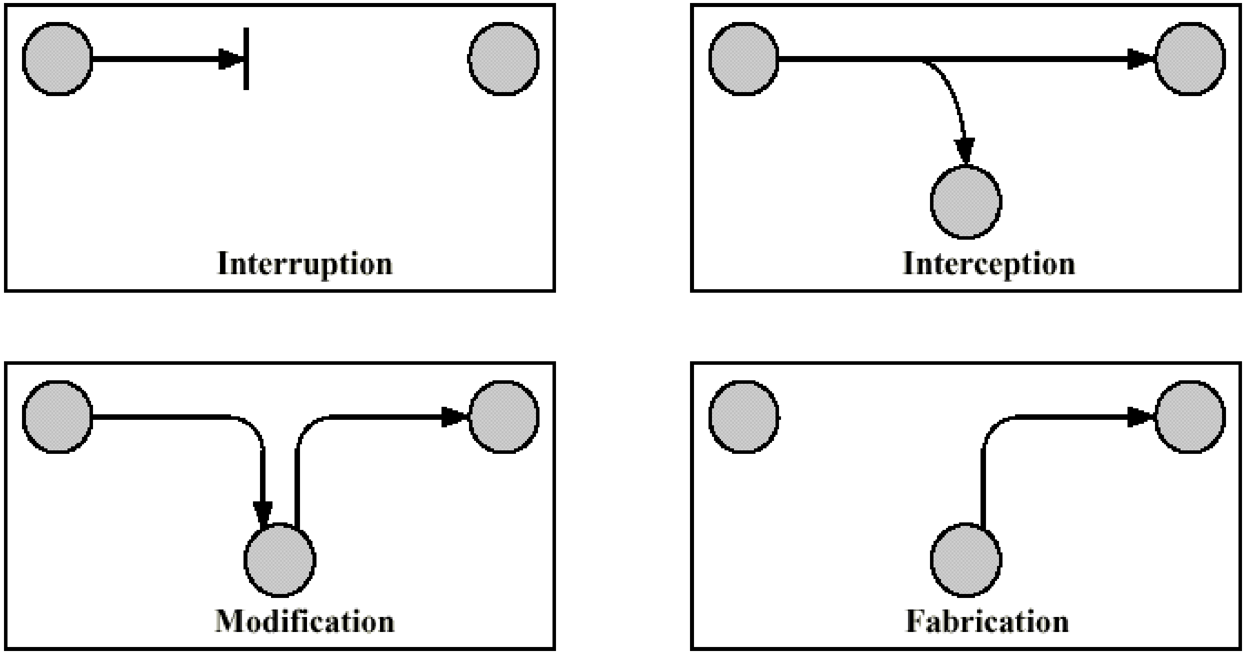
OSI Security Architecture

- Security attacks
  - *Reasons*
- Security mechanisms
  - *Tools*
- Security services
  - *Our goal*

OSI Security Architecture

- Security Services
  - *A processing or communication service that enhances the security of data processing and information transmission.*
- Security Mechanisms
  - *A process (algorithm, protocol or device) that is designed to detect, prevent, or recover from a security attack.*
- Security Attack
  - *Any action that compromises the security of information owned by an organisation or individual.*

Recall Attacks and Threats



Recall Attacks and Threats

- Passive Attacks
  - *Eavesdropping communications*
  - *Traffic analysis on identity, location, …*
- Active Attacks
  - *Replay*
  - *Impersonation*
  - *DoS*
  - *…*

Security Mechanisms

- Specific Security Mechanisms
  - *Encryption, digital signatures, access control, authentication, key exchange, traffic padding, routing control, etc.*
- Pervasive Security Mechanisms
  - *Trusted functionality, security label, detection, auditing, recovery, etc.*

Security Services

- Authentication
- Access Control
- Data Confidentiality
- Data Integrity
- Non-repudiation
- Availability

Authentication

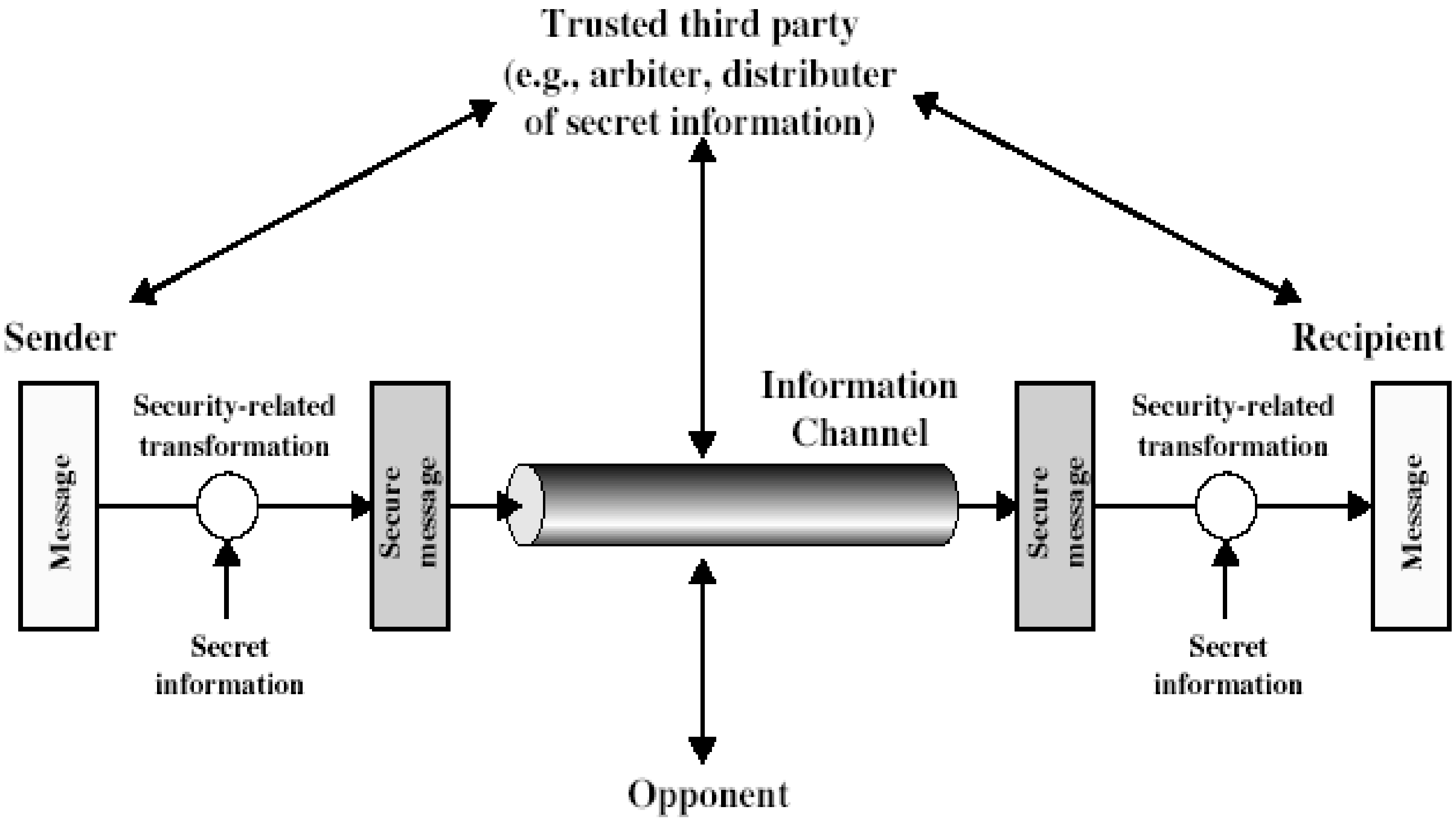
- Provide corroboration of the peer entity/data from N-layer to (N+1)-layer.
- At the establishment of a connection or during data transfer phase of a connection

Non-Repudiation

- Non-Repudiation of Origin
    - *Recipient provided with proof of the origin of data*

- Non-Repudiation of Delivery
    - *Sender provided with proof of a delivery of data*

Models for Network Security

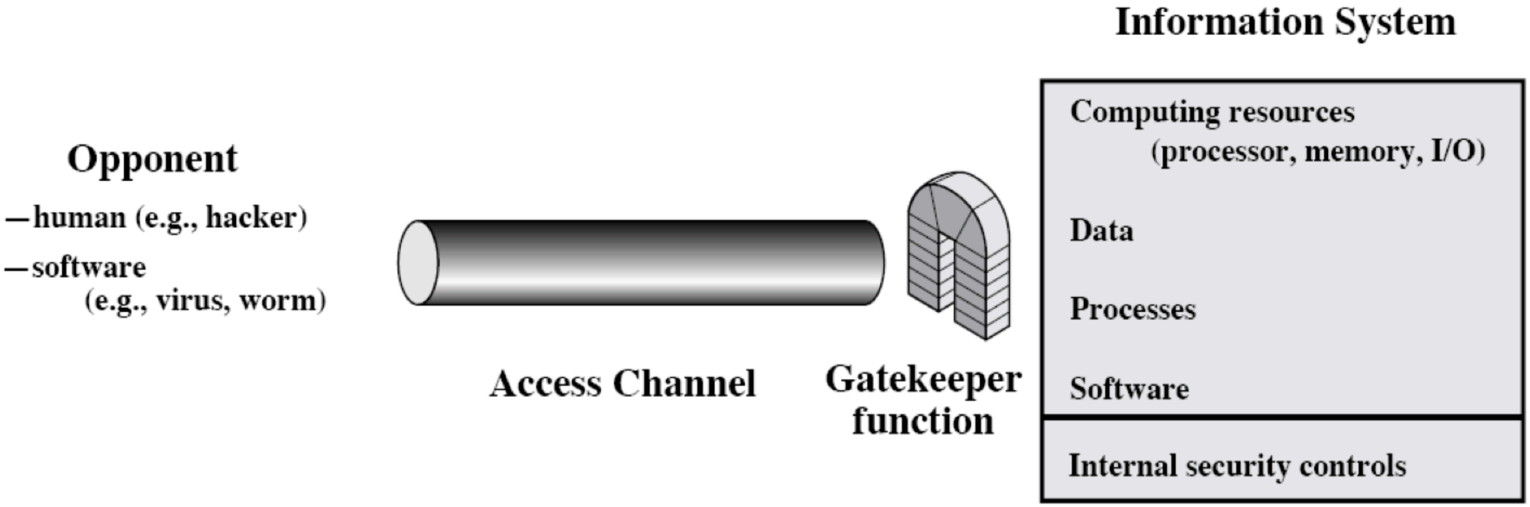


Models for Network Security

Basic tasks in security service design

- Design a secure algorithm for the security transformation.
- How to generate, distribute and keep the secret information (keys) securely.
- Specify a protocol to achieve the security service by using the security algorithm and secret information.
- Implement the whole system properly.

Network Access Security Model



Network Access Security Model

- The goal is to prevent information from unwanted access.
- The gatekeeper can be a login procedure, a firewall, a virus killing tool etc.
- Trusted computer systems may be helpful to implement this model.

Web Security Considerations

- Web is very visible.
- Complex software may hide many security flaws.
- Web servers are easy to configure and manage.
- Users may not be aware of risks.

Web Application Threats

- Confidentiality
- Integrity
- Authentication
- Availability (Denial of Service, etc.)

Confidentiality

- Threats
  - *Eavesdropping on the Net*
  - *Stealing of information from server*
  - *Stealing of data from client*
  - *Information about which client talks to server*
- Countermeasures
  - *Encryption*
  - *Web proxies*

Integrity

- Threats
  - *Modification of user data*
  - *Trojan horse browser*
  - *Modification of message traffic in transit*
- Countermeasures
  - *Cryptographic checksums*

Authentication

- Threats
  - *Impersonation of legitimate users*
  - *Data forgery*
- Countermeasures
  - *Cryptographic techniques*

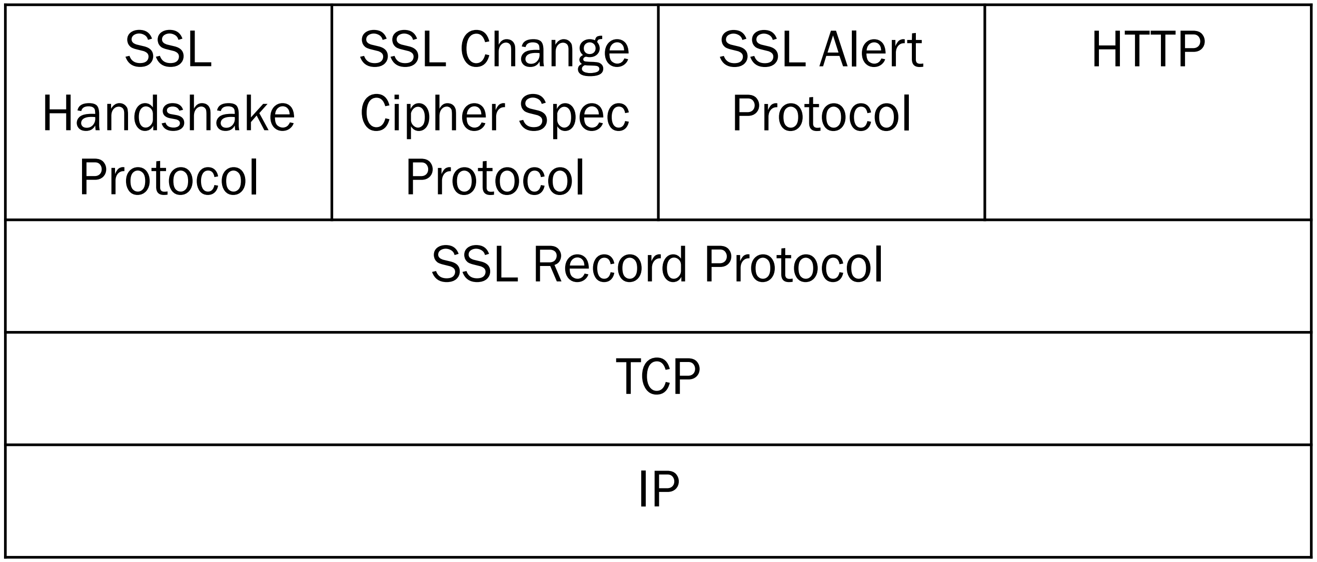
Security in Difference Layers

	OSI Layer	Protocols (example)
Software	Application	HTTP, HTTPS, SMTP, FTP, Kerberos
	Presentation	SSL
	Session	
	Transport	TCP, UDP, SSL/TLS
	Network	IPv4, IPv6, IPSec, ARP
Hardware	Data Link	Ethernet, 802.1x, L2TP/PPTP
	Physical	

SSL and TLS

- SSL - Secure Socket Layer
- TLS – Transport Layer Security
- SSL was originated by Netspace.
- TLS working group was formed within IETF.
- First version of TLS can be viewed as SSLv3.1.

SSL Architecture



SSL Handshake Protocol

SSL Change Cipher Spec Protocol

SSL Alert Protocol

HTTP

SSL Record Protocol

TCP

IP

SSL Protocol Stack

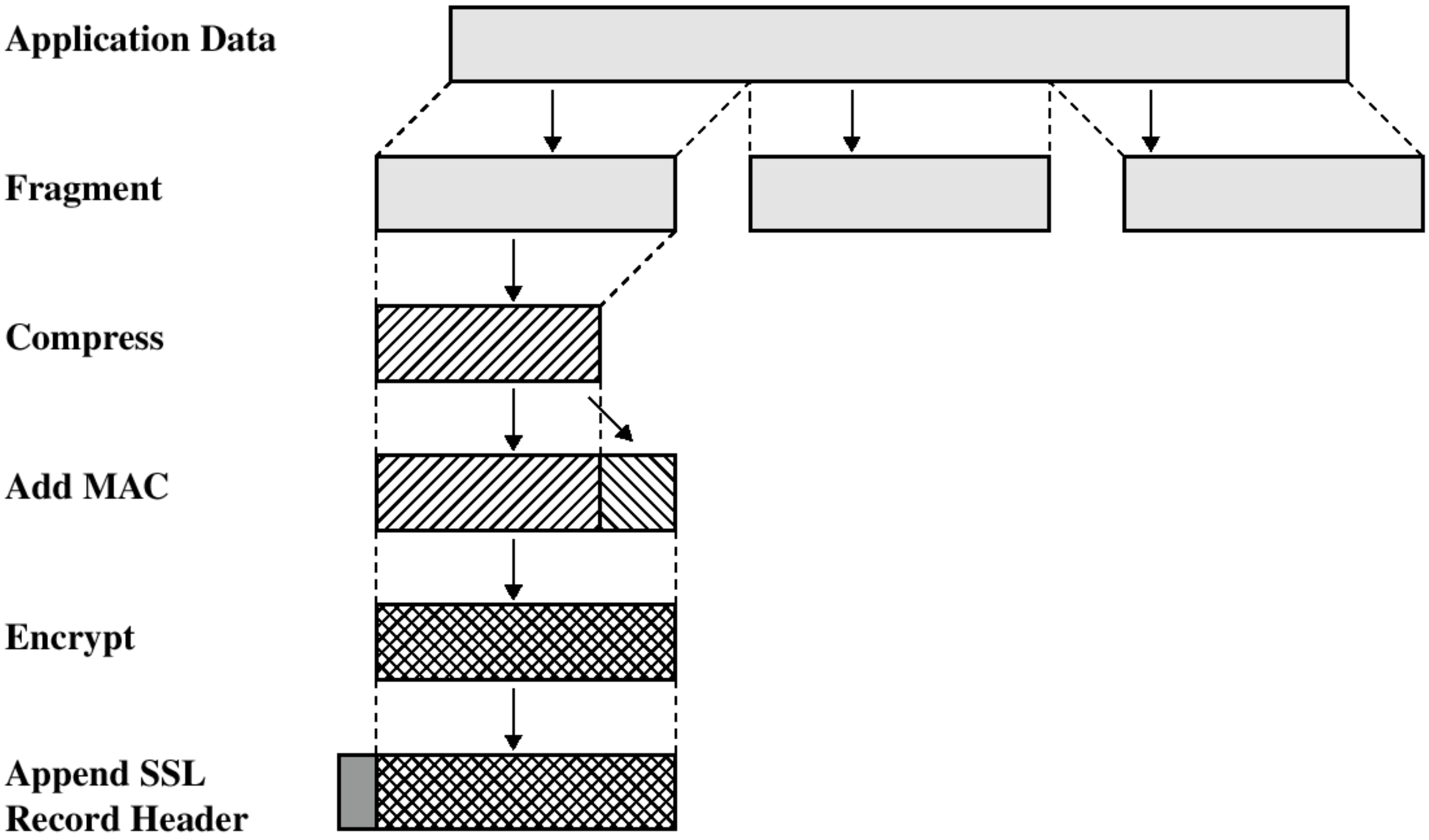
SSL Protocol Stack

- SSL Record Protocol: provides basic security services to various higher-layer protocols.
- SSL exchange management
  - *Handshake protocol*
  - *Change cipher spec protocol*
  - *Alert protocol*

SSL Connection and Session

- SSL Session
  - *An association between a server and a client.*
  - *Created by the handshake protocols.*
  - *Defines a set of cryptographic security parameters.*
  - *A session could be shared by multiple connections.*
- SSL Connection
  - *Peer-to-peer relationship, transient.*
  - *Every connection is associated with one session.*

SSL Record Protocol Operation



SSL Record Protocol

- SSL Record Protocol provides two security services for SSL connections
  - *Confidentiality*
    - Handshake protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
  - *Message integrity*
    - Handshake protocol defines a shared secret key that is used to form a message authentication code.

Change Cipher Spec Protocol

- This protocol contains a single message which consists of a single byte with the value 1.
- Purpose of the message
  - *To cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.*
- This protocol allows to update cipher suite to be used without renegotiate the connection.

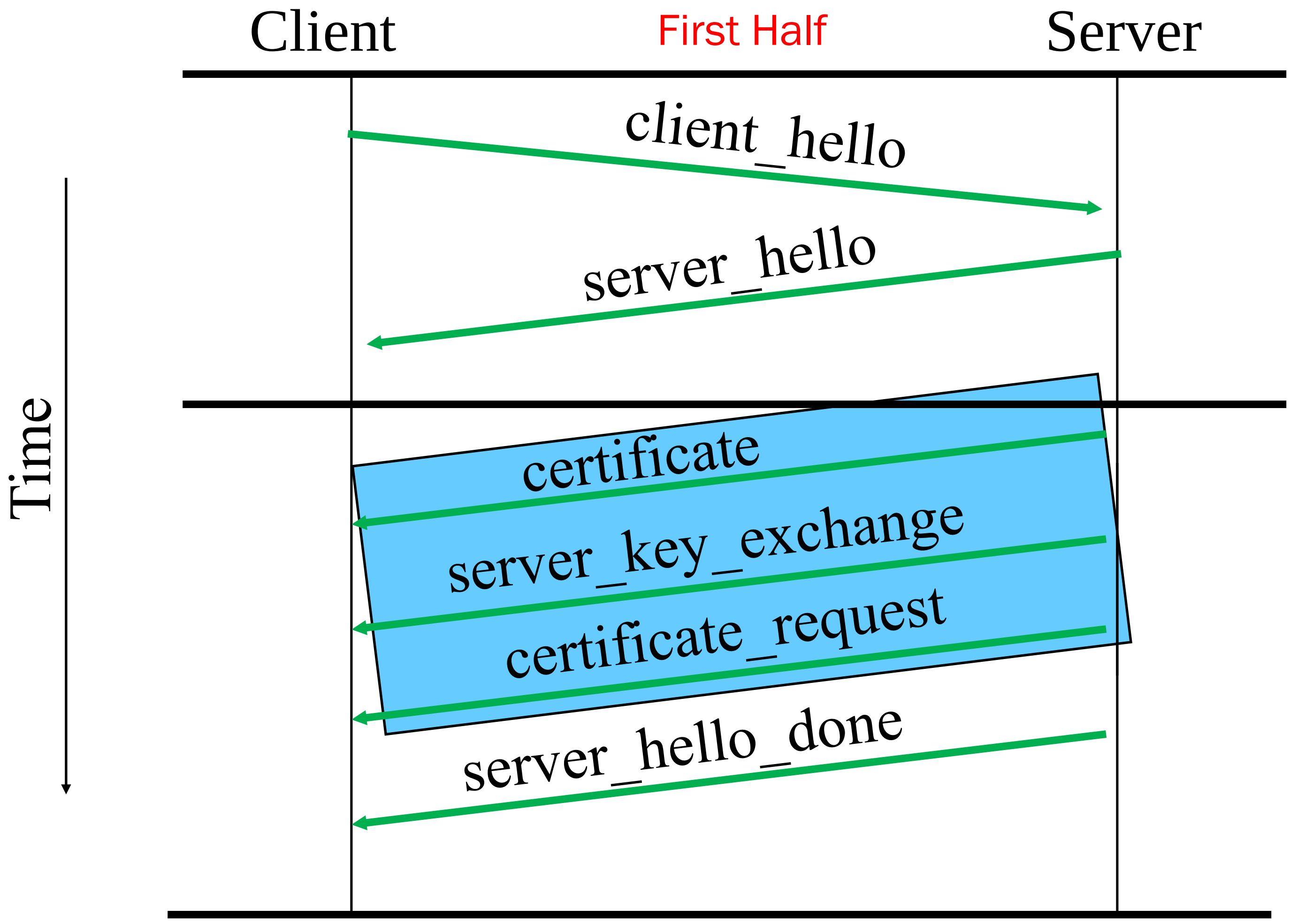
Alert Protocol

- Used to convey SSL-related alerts to the peer entity.
- Alert message are compressed and encrypted.
- The message contains two bytes
  - *1 byte: warning (1) or fatal (2)*
  - *1 byte: status of the certificate and other specific alerts.*

SSL Handshake Protocol

- Allows the server and client to authenticate each other.
- Negotiate encryption, MAC algorithm and cryptographic keys.
- Used before any application data are transmitted.

SSL Handshake Protocol

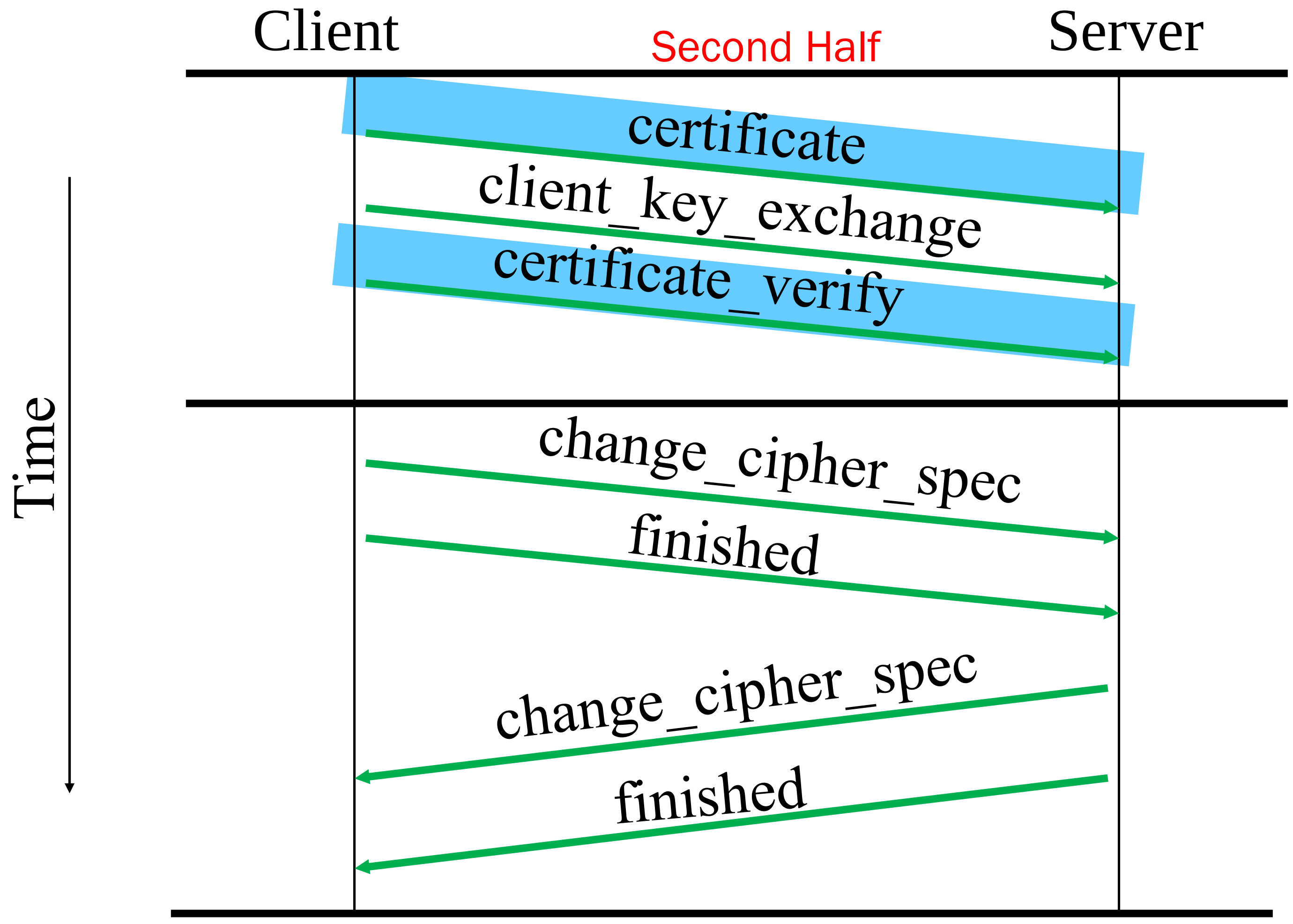


Establish security capabilities, including protocol version, session ID, cipher suite, compression method and initial random numbers.

Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase

NOTE:- Shaded transfers are optional or situation-dependent messages that are not always sent.

SSL Handshake Protocol



Client sends certificate if requested.
Client sends key exchange.
Client may send certificate verification.

Change cipher suite and finish handshake protocol.

NOTE:- Shaded transfers are optional or situation-dependent messages that are not always sent.

SSL Handshake Protocol

- Client Hello
  - *SSL Version*
  - *Timestamp, Nonce*
  - *Session ID (empty or non-empty)*
  - *Cipher Suite: List of algorithms*
  - *Compression Methods: List of Algorithms*

SSL Handshake Protocol

- Server Hello
  - *SSL Version: lower version of that suggested by the client and highest supported by the server.*
  - *Timestamp, Nonce*
  - *Session ID*
    - If client's non-empty, server can use the same session state, if it wishes.
    - If client's empty, server generates a new session identifier.
  - *Selected cipher*
  - *Selected compression method*

SSL Handshake Protocol

- Certificate Message (optional)
  - *If server needs to be authenticated, it sends its certificate (typically X.509) immediately following Server Hello message.*
- Server Key Exchange Message (optional)
  - *Not needed for RSA key exchange and fixed Diffie-Hellman.*
  - *Only in certain cases such as Anonymous Diffie-Hellman, Fortezza (token based key exchange algorithm).*
  - *Sign a short ephemeral public key.*

SSL Handshake Protocol

- Certificate Request (optional)
  - *Server can request a certificate from the client*
  - *Certificate, Type, Certificate Authorities*
- Server Hello Done
  - *Server waits for client's response.*

SSL Handshake Protocol

- Upon receiving Server Hello messages, the client
  - *Checks whether the security parameters are accepted.*
  - *Verifies the server certificate, if needed.*
  - *Sends client certificate if requested.*

SSL Handshake Protocol

- Client Key Exchange
  - *Depends on the type of key exchange*
  - *RSA*
    - Pre-master secret key (48 bytes)
    - Encrypts under server's public key
- Certificate Verify (optional)
  - *Only send if client certificate provided.*
  - *Provides explicit verification of client's certificate.*
  - *Message contains a signed hash code of the preceding messages.*

SSL Handshake Protocol

- Client Change Cipher Spec
    - *Makes the cipher spec agreed current.*
- Finished
    - *Verifies that the key exchange and authentication.*
    - *Signed hash code using master secret calculated from pre-master secret*
        - Function: Combined MD5 and SHA using master secret.
- Server changes its cipher spec and responds its Finished message.

Key Exchange Methods

- Supported key exchange methods
  - *RSA*
  - *Fixed Diffie-Hellman*
  - *Ephemeral Diffie-Hellman (most secure)*
  - *Anonymous Diffie-Hellman*
    - No authentication
    - Suffers from man-in-the-middle attacks

TLS

- To produce an Internet standard version SSL (RFC 5246).
- MAC
  - *TLS uses HMAC*
  - *SSL v3: the key is concatenated rather than XORed .*
- Cipher Suites
  - *Key exchange: TLS supports all of key exchange techniques of SSL v3 except Fortezza.*
  - *Symmetric encryption: TLS supports all in SSL v3 excepts Fortezza.*
- Some minor differences on certificate types.

HTTPS

- HTTP over SSL
  - *Secure communication between client and web server.*
  - *https://…*
  - *Use port 443 (https) rather than 80 (http).*
- Encrypted communication
  - *URL of the requested document*
  - *Contents of the document*
  - *Content of browser forms*
  - *Cookies*
  - *Contents of HTTP header*

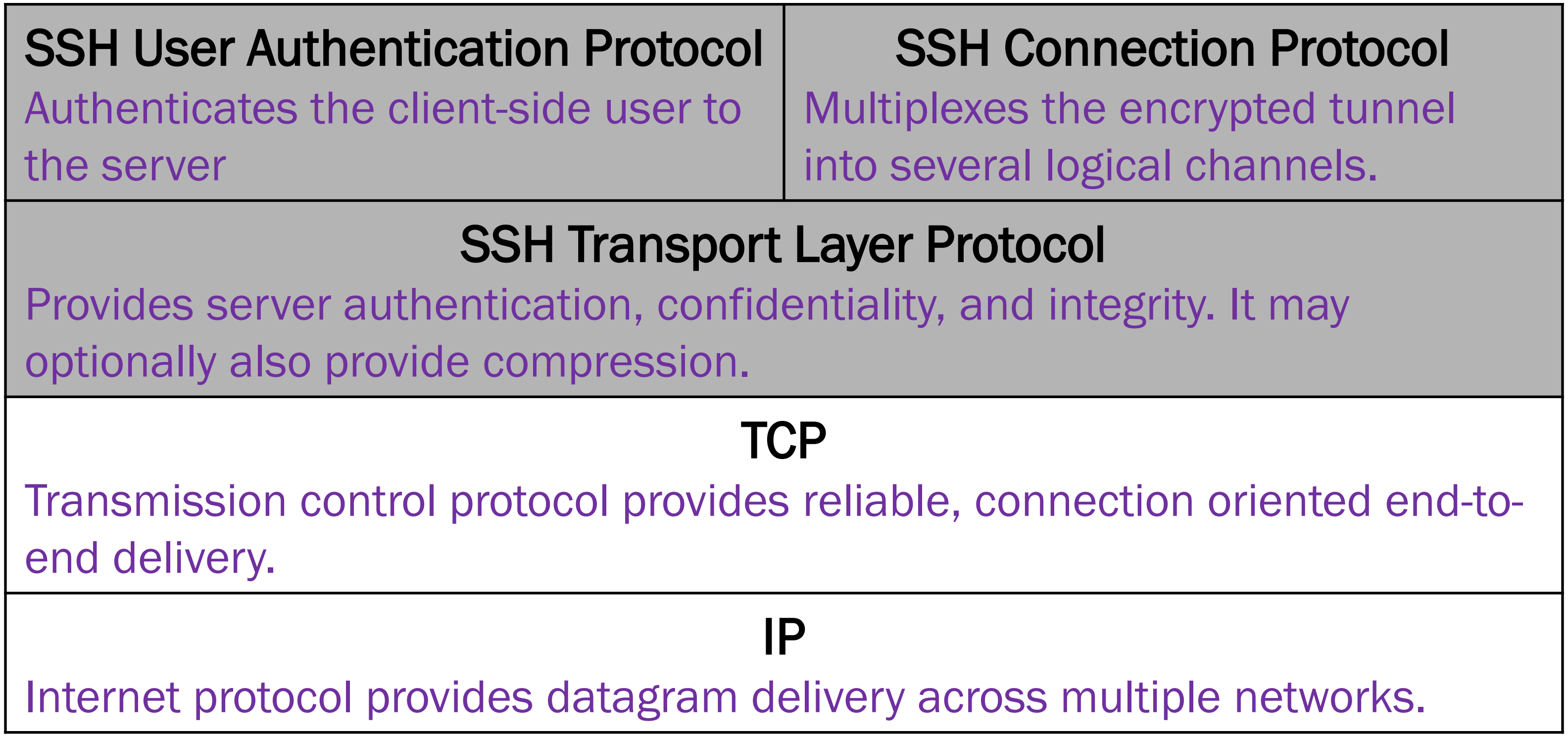
HTTPS

- Connection initiation
  - *TLS handshake then HTTP request.*
- Connection closure
  - *Have “Connection: close” in HTTP record*
  - *TLS level exchange close_notify alerts*
  - *Then close TCP connection*
  - *Handle TCP close before alert exchange sent or completed.*

SSH Overview

- SSH – Secure Shell
  - *Initially designed to replace insecure rsh and telnet utilities.*
  - *Secure remote administration (Unix/Linux etc.)*
  - *Extended to support secure file transfer and email.*
  - *Provide a general secure channel for network applications*
  - *SSH provides security at Application layer.*
  - *Build on top of TCP.*
  - *SSH-2 is better (more secure) than SSH-1 (insecure).*

SSH Protocol Stack



SSH User Authentication Protocol

Authenticates the client-side user to the server

SSH Connection Protocol

Multiplexes the encrypted tunnel into several logical channels.

SSH Transport Layer Protocol

Provides server authentication, confidentiality, and integrity. It may optionally also provide compression.

TCP

Transmission control protocol provides reliable, connection oriented end-to-end delivery.

IP

Internet protocol provides datagram delivery across multiple networks.

SSH-2 Architecture

SSH-2 adopts a three layer architecture:

- SSH Transport Layer Protocol.
    - *Initial connection.*
    - *Server authentication*
    - *Sets up secure channel between client and server.*
- SSH Authentication Protocol
    - *Client authentication over secure transport layer channel.*
- SSH Connection Protocol
    - *Supports multiple connections over a single transport layer protocol secure channel.*
    - *Efficiency (session re-use).*

SSH-2 Security Goals

- Server authenticated in transport layer protocol.
- Client authenticated in authentication protocol.
  - *By public key (DSS, RSA, SPKI, OpenPGP).*
  - *Or simple password for particular application over secure channel.*
- Establishment of a fresh, shared secret.
  - *Shared secret used to derive further keys, similar to SSL and IPSec.*
  - *For confidentiality and authentication in SSH transport layer protocol.*
- Secure cipher suite negotiation.
  - *Encryption, MAC, and compression algorithms.*
  - *Server authentication and key exchange methods.*

SSH-1 versus SSH-2

- Many vulnerabilities have been found in SSH-1 .
  - *SSH-1 Insertion attack exploiting weak integrity mechanism (CRC-32) and unprotected packet length field.*
  - *SSHv1.5 session key retrieval attack (theoretical).*
  - *Man-in-the-middle attacks (using e.g. dsniff).*
  - *DoS attacks.*
    - Overload server with connection requests.
    - Buffer overflows.