MONASH University
Information Technology

FIT3031 INFORMATION & NETWORK SECURITY

MONASH University
Information Technology

FIT3031 INFORMATION & NETWORK SECURITY

Lecture 5
Web Security

Unit Objectives

- ✓ OSI security architecture
  - **common security standards and protocols for network security applications**
  - **common information risks and requirements**
- ✓ operation of private key encryption techniques
- ✓ operation of public encryption techniques
- ✓ concepts and techniques for digital signatures, authentication and non-repudiation
- ✓ security threats of web servers, and their possible countermeasures
- Wireless Security Issues
- security threats of email systems and their possible countermeasures
- IP security
- intrusion detection techniques for security purpose
- risk of malicious software, virus and worm threats, and countermeasures
- firewall deployment and configuration to enhance protection of information assets
- network management protocol for security purpose

Review of Last Lecture

- **A workstation cannot be trusted for access control purpose**
- **Kerberos authentication protocol provides a centralized authentication server to authenticate users to servers and servers to users**
  - Relies on encryption, v.4 uses DES
- **Two key component of Kerberos are**
  - Authentication Server (AS)
    - > users initially negotiate with AS to identify self
    - > AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
  - Ticket Granting server (TGS)
    - > users subsequently request access to other services from TGS on the basis of users' TGT
- **X.509 is a standard for digital certificate to identify a user's public key and supports authentication service**
- **Certification Authority (CA) & PKIX**
- **Federated Identity management**

Lecture 5: Objectives

- **Outline the security threat of communicating over the Internet**
- **Discuss how security of web server can be compromised**
- **Describe how SSL can be used to make a communication channel secure**
- **Outline the services provided by SSL**
- **Understand SSL record protocol**
- **Understand SSL handshake protocol**
- **Implementation of secure communication between a web browser and a web server using HTTPS.**
- **Understand SSH protocol for secure remote logon and**
- **other client/server facilities.**

Lecture 5: Outline

- **Web security threats**
- **Secure Socket Layer (SSL) protocol**
  - SSL record protocol
  - SSL handshake protocol
- **HTTPS (HTTP over SSL)**
- **Secure Shell (SSH)**

Web Popularity

- **Today most big business organizations and government agencies have web sites, because**
  - communication is cheaper
  - dissemination of information is rapid
  - provides highly visible outlet for product information
- **Web is growing as an increasingly popular platform for business transaction**
  - use of Internet eliminates the use of proprietary network, reduces cost
  - payment can be anonymous, but identity can be disclosed if needed

Web Security Threats

- **Web is easy to use, but the underlying software is extremely complex**
  - hides potential security flaws
  - numerous attack has been reported
- **Once the web server is compromised, entire organization's network becomes vulnerable**
- **Common users are not aware of security risk**
  - lacks tools or knowledge of effective countermeasures
  - becomes potential victim

Web Security Threats

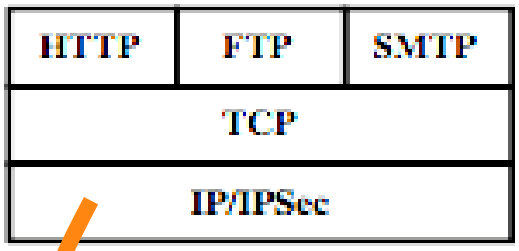
- **Threats and possible countermeasures of web security:**
  - **integrity** – cryptographic checksum (hash)
  - **confidentiality** – encryption, web proxies
  - **denial of service** – difficult to prevent
  - **authentication** - cryptography
- **Web security according to location of the threats**
  - web server, web client
    > falls under system security (covered in later lecturers)
  - network traffic between the client and server
    > eavesdropping on communication, gaining access information to the server (covered in this lecture)

Web Security Threats

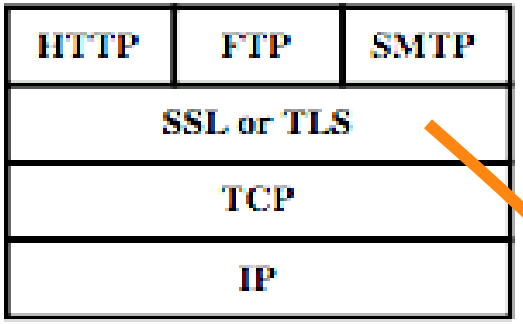
Table 5.1 A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerabilty 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

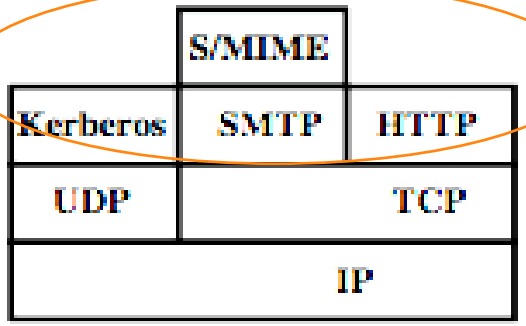
Web Traffic Security Approaches



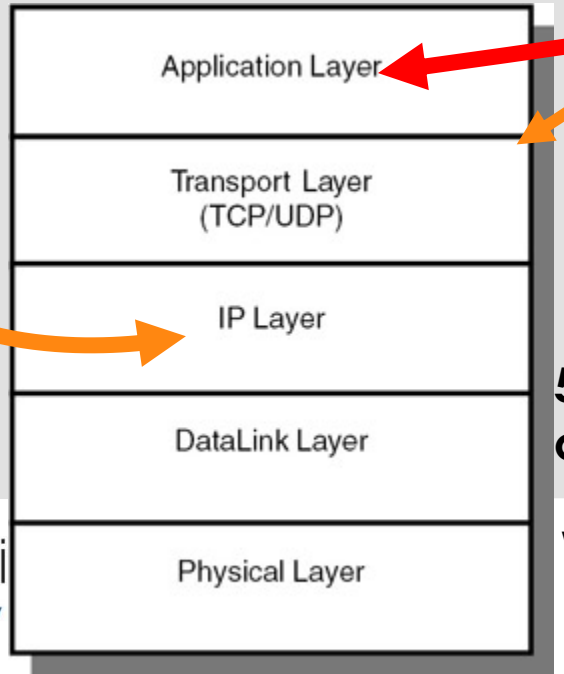
(a) Network Level



(b) Transport Level



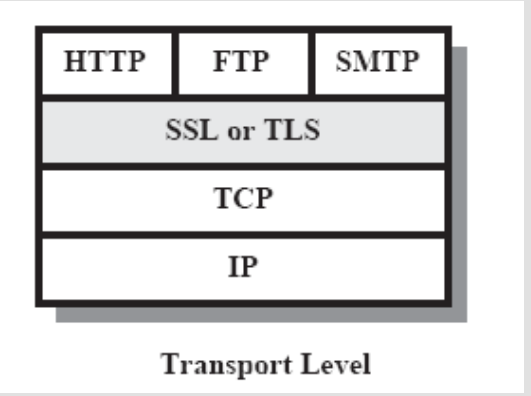
(c) Application Level

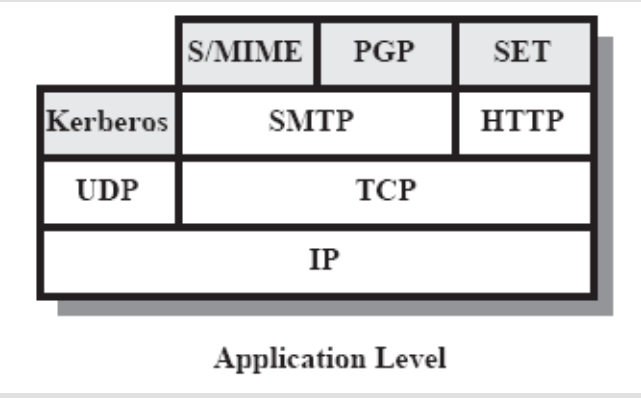


5-Layers of TCP/IP communication Architecture

Web traffic security

- **One approach is to implement security protocol above TCP layer of TCP/IP protocol stack**
- **Two implementation choices above TCP layer**
  - incorporate SSL (secure socket layer) or TLS (transport layer security) in the protocol suite
  - embed SSL in specific packages
    - > Netscape, IE browser are equipped with SSL
- **Another approach is to implement application specific security services embedded within the particular application**
    - > Secure Electronic Transaction for Internet based payment system

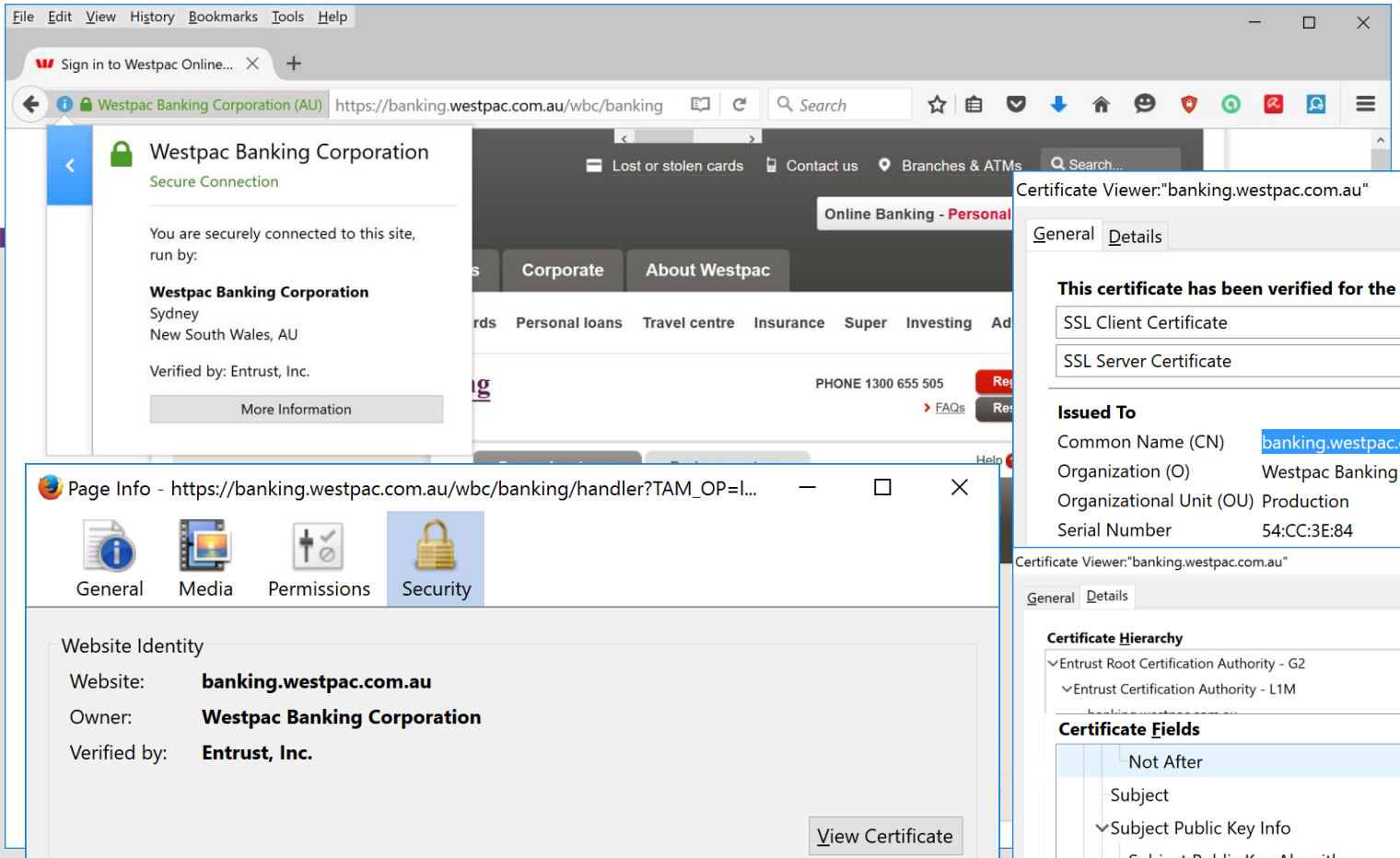


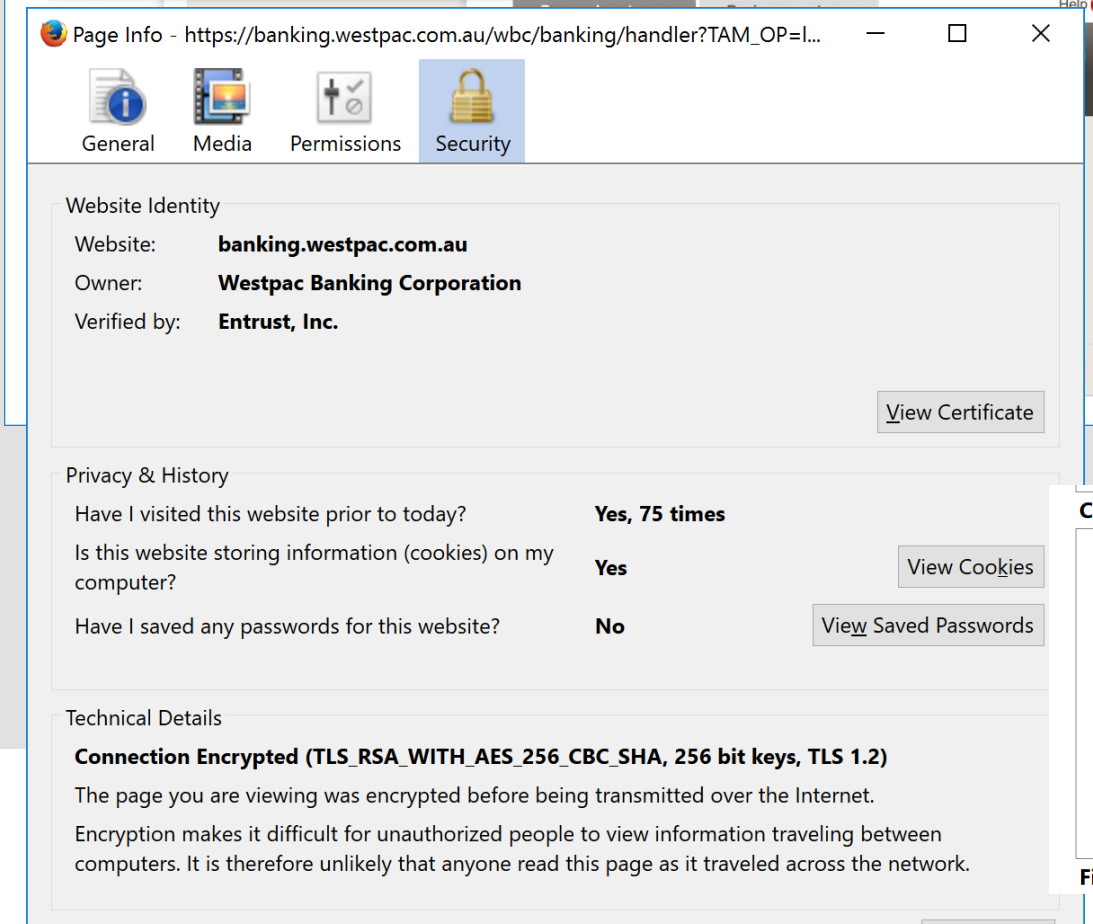


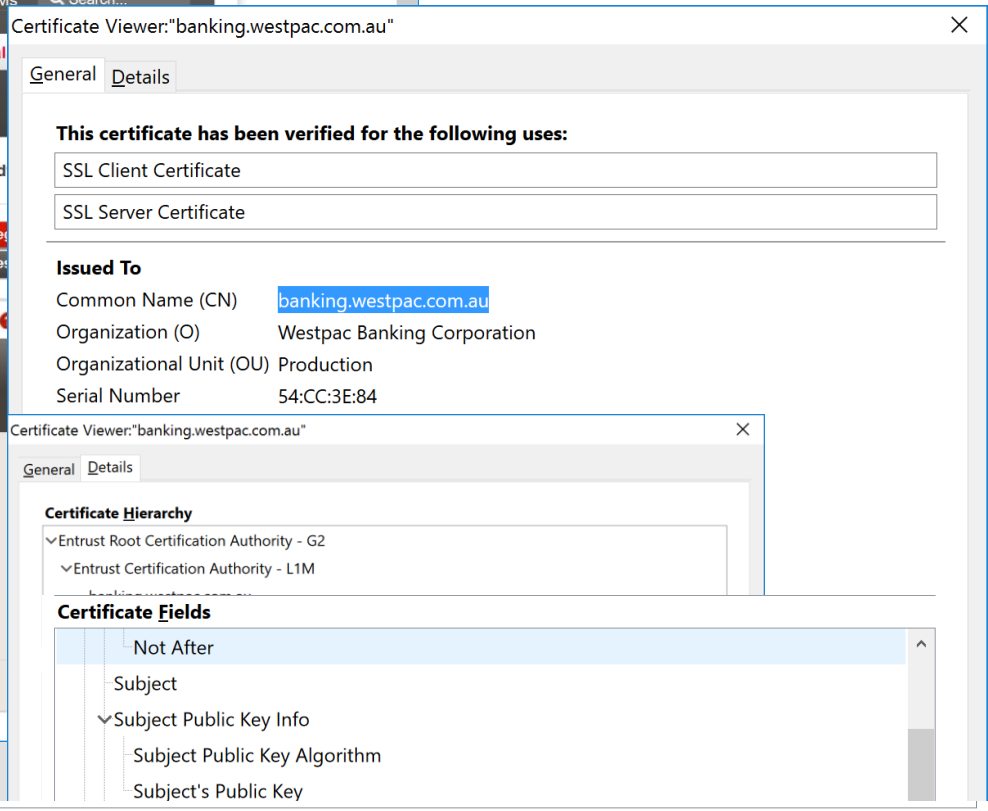
SSL/TLS

- **A new layer inserted between transport layer and application layer**
  - therefore capable of protecting communication from any application protocol above TCP
- **Originally developed by Netscape**
- **Version 3 was designed with public input**
- **Subsequently became Internet standard known as TLS (Transport Layer Security)**
- **The first version of TLS is essentially SSLv3.1**
  - it evolved into TLS specified in RFC 2246
  - very close to and backward compatible with SSLv3

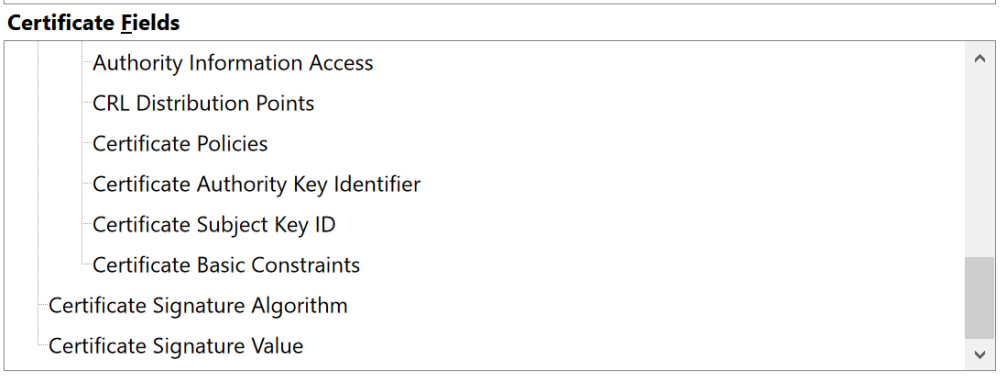
SSL/TLS


File Edit View History Bookmarks Tools Help
Sign in to Westpac Online...
Westpac Banking Corporation (AU)
https://banking.westpac.com.au/wbc/banking
Search
Westpac Banking Corporation
Secure Connection
You are securely connected to this site, run by:
Westpac Banking Corporation
Sydney
New South Wales, AU
Verified by: Entrust, Inc.
More Information
Lost or stolen cards
Contact us
Branches & ATMs
Search...
Online Banking - Personal
Corporate
About Westpac
Personal loans
Travel centre
Insurance
Super
Investing
PHONE 1300 655 505
FAQs


Page Info - https://banking.westpac.com.au/wbc/banking/handler?TAM_OP=l...
General
Media
Permissions
Security
Website Identity
Website: banking.westpac.com.au
Owner: Westpac Banking Corporation
Verified by: Entrust, Inc.
View Certificate
Privacy & History
Have I visited this website prior to today? Yes, 75 times
Is this website storing information (cookies) on my computer? Yes
View Cookies
Have I saved any passwords for this website? No
View Saved Passwords
Technical Details
Connection Encrypted (TLS_RSA_WITH_AES_256_CBC_SHA, 256 bit keys, TLS 1.2)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.


Certificate Viewer:"banking.westpac.com.au"
General
Details
This certificate has been verified for the following uses:
SSL Client Certificate
SSL Server Certificate
Issued To
Common Name (CN) banking.westpac.com.au
Organization (O) Westpac Banking Corporation
Organizational Unit (OU) Production
Serial Number 54:CC:3E:84

Certificate Viewer:"banking.westpac.com.au"
General
Details
Certificate Hierarchy
Entrust Root Certification Authority - G2
Entrust Certification Authority - L1M
Certificate Fields
Not After
Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key


Certificate Fields
Authority Information Access
CRL Distribution Points
Certificate Policies
Certificate Authority Key Identifier
Certificate Subject Key ID
Certificate Basic Constraints
Certificate Signature Algorithm
Certificate Signature Value
Field Value

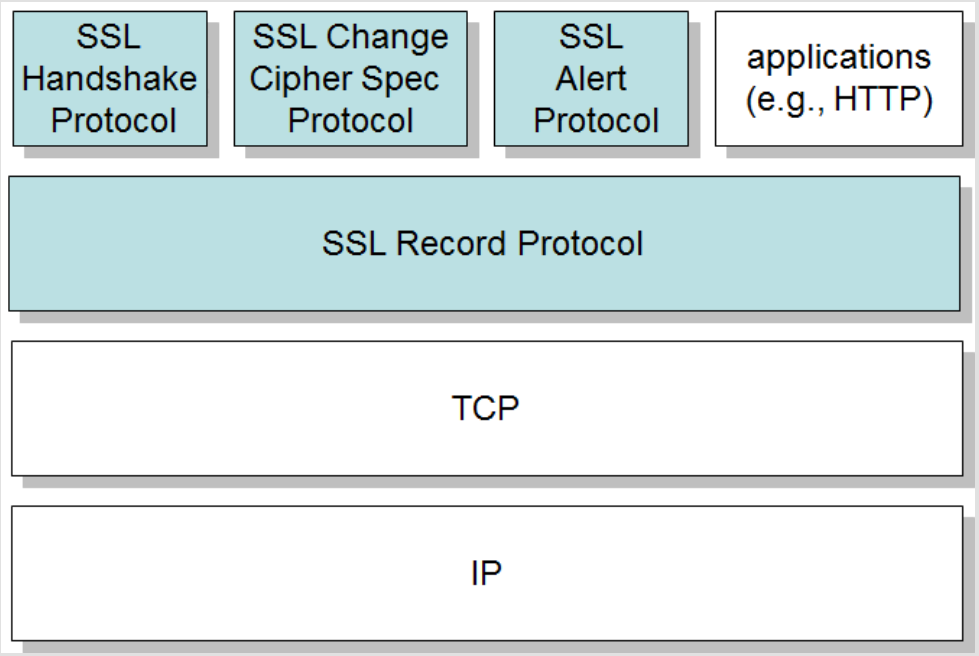
SSL Services

- **The following services are provided by SSL:**
  - **Server authentication:** server's identity is confirmed to the client, by demonstrating valid certificate or public key
    > Specially important for financial transaction
  - **Confidentiality:** data items transferred in the session are encrypted to protect against eavesdropping
  - **Integrity:** MAC is attached to the message
  - **Client authentication**: user's identity is confirmed to the server
    > Important in internet banking/general contracting when the server needs to be sure about client identity

SSL Architecture

- **SSL connection**
  - a transient, peer-to-peer, communications link
  - A connection is a network transport that provides a suitable type of service
  - associated with 1 SSL session
- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections

SSL PROTOCOL STACK



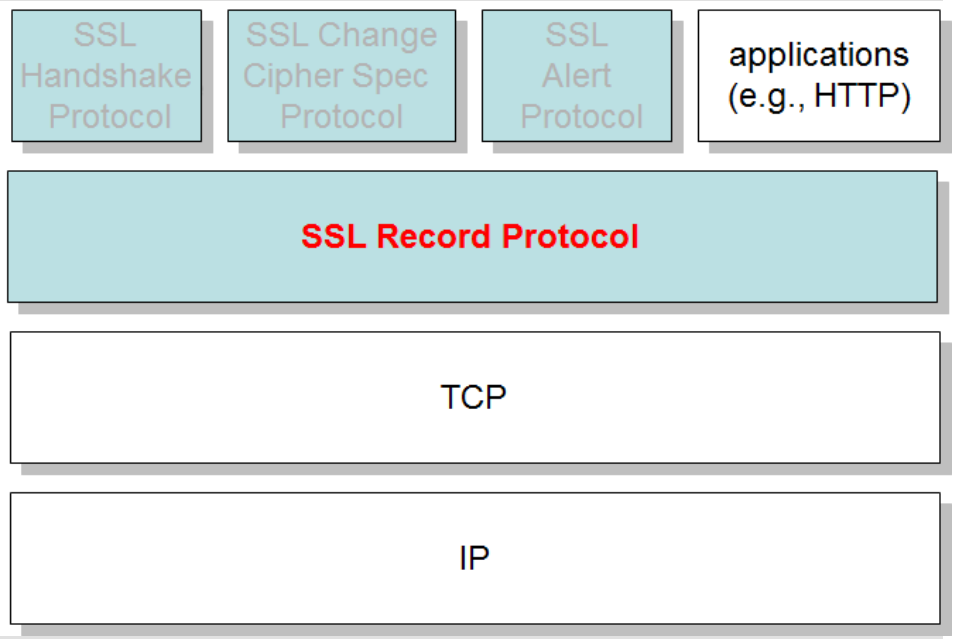
- **SSL has two sub layers of protocols**
- **Two sub-layers**
  - **bottom layer**
    - **SSL record protocol**
  - **upper layer**
    - **SSL handshake protocol**
    - **SSL change cipher Spec protocol**
    - **SSL alert protocol**

SSL components

- **SSL Handshake Protocol**
  - negotiation of security algorithms and parameters
  - key exchange
  - server authentication and optionally client authentication
- **SSL Alert Protocol**
  - error messages (fatal alerts and warnings)
- **SSL Change Cipher Spec Protocol**
  - a single message that indicates the end of the SSL handshake
- **SSL Record Protocol**
  - fragmentation
  - compression
  - message authentication and integrity protection
  - encryption

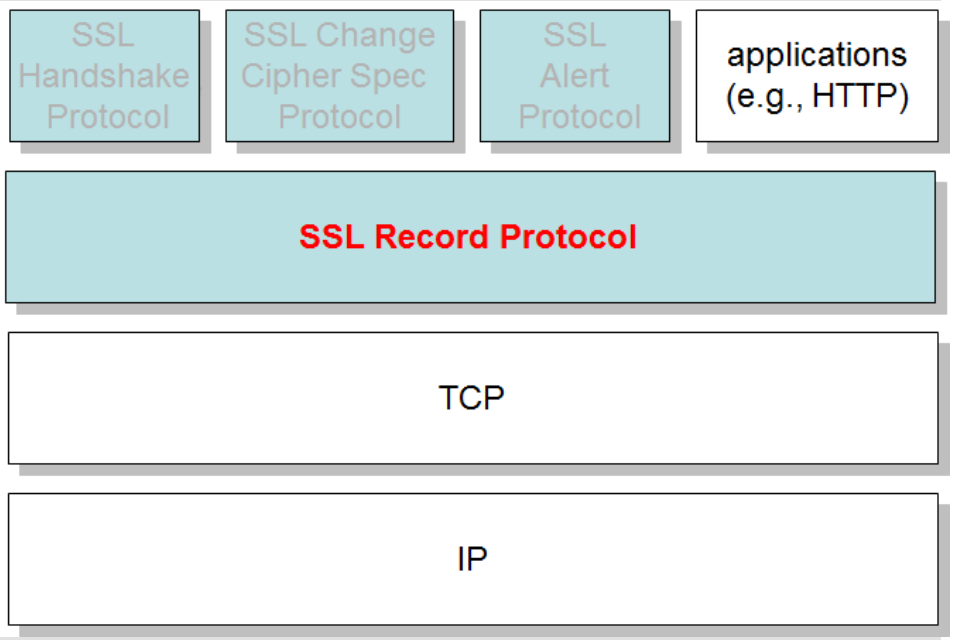
SSL Record Protocol

- The SSL Record Protocol actually transfer the data
- Provides confidentiality and message integrity
- Defines a set of formats and procedures by which message are handed down from the application layer
- Takes data from application layer, encapsulates into appropriate headers and creates an object called record
- Encrypted records are forwarded to TCP layer

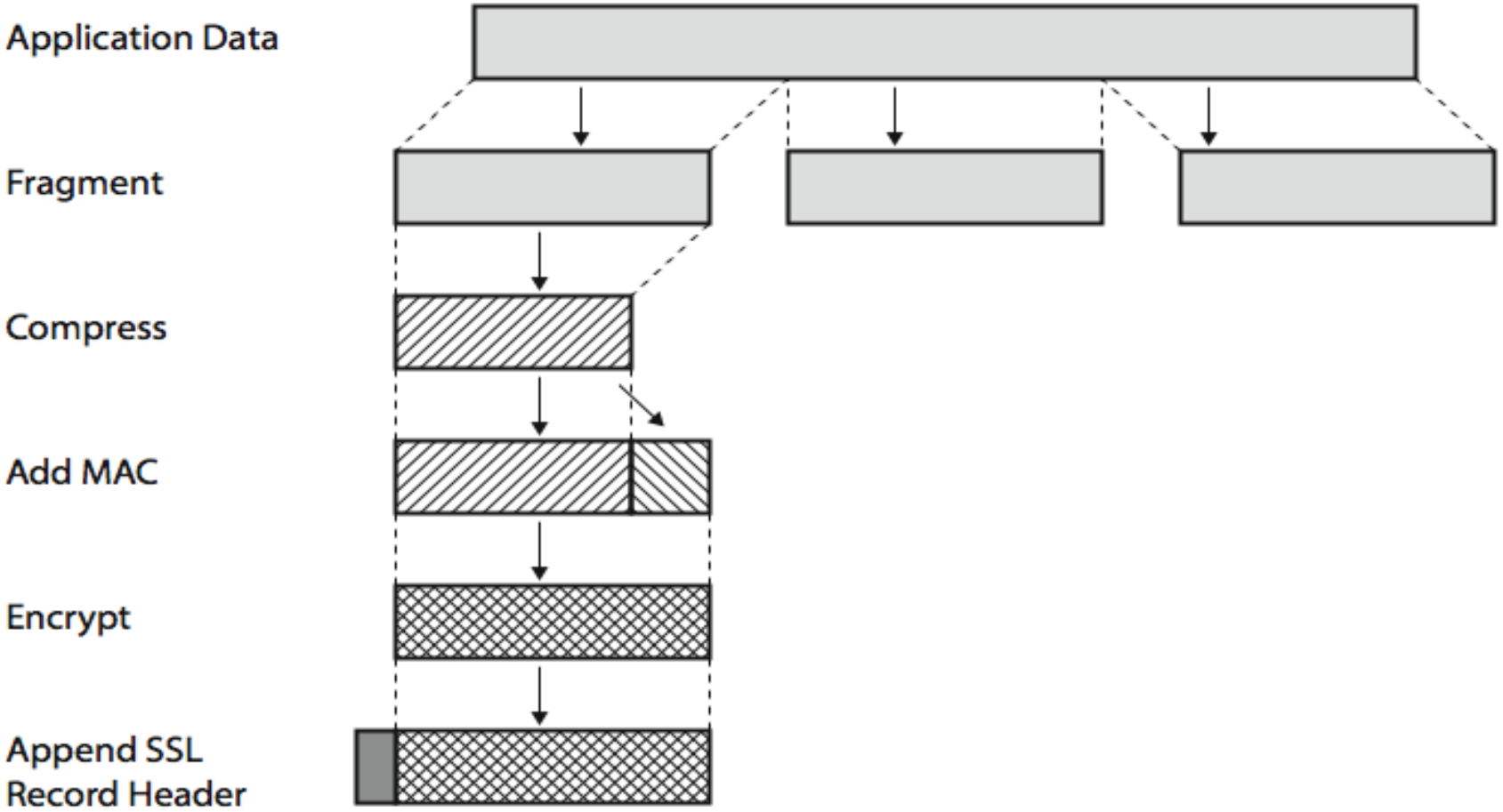


SSL Record Protocol Services

- **Confidentiality**
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
  - **message is compressed before encryption**
- **message integrity**
  - using a MAC with shared secret key
  - similar to HMAC but with different padding



SSL Record Protocol Operation

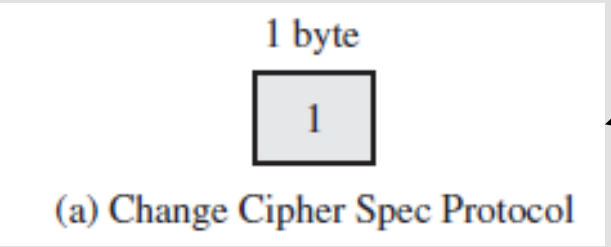


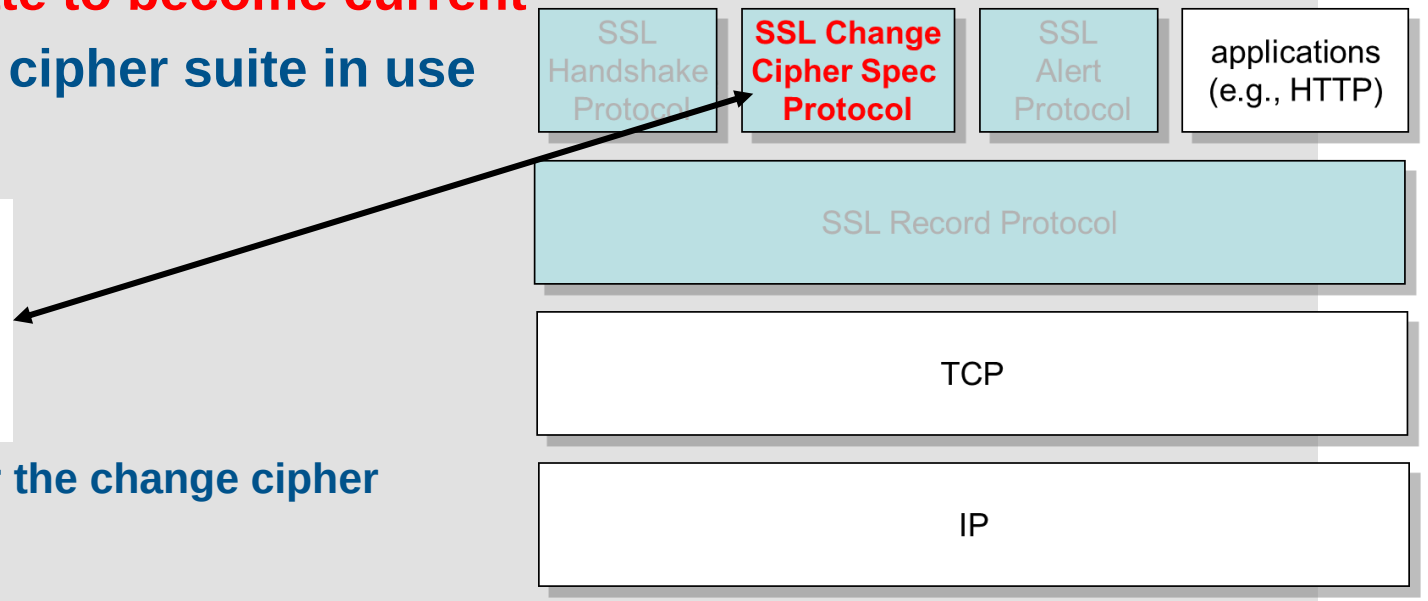
SSL Record Protocol Operation ...

- **SSL Record Protocol operation involves:**
  - **Fragmentation**
    - > fragments the data in manageable block size (16KB or less)
  - **Compression**
    - > optional
    - > must be lossless
    - > SSLv3 (TLS) does not specify any compression algorithm
  - **Integrity protection**
    - > compute MAC on the compressed data using SHA-1,MD5
    - > uses a shared secret key negotiated in handshake protocol
  - **Encryption**
    - > compressed message and MAC are encrypted using symmetric encryption algorithm
    - > Algorithm permitted: AES, IDEA, RC2, RC4, DES, 3DES, Fortezza
  - **– Append SSL record header**

SSL Change Cipher Spec Protocol

- **one of 3 SSL specific protocols which use the SSL Record protocol**
- **a single message**
- **causes pending state to become current**
- **hence updating the cipher suite in use**

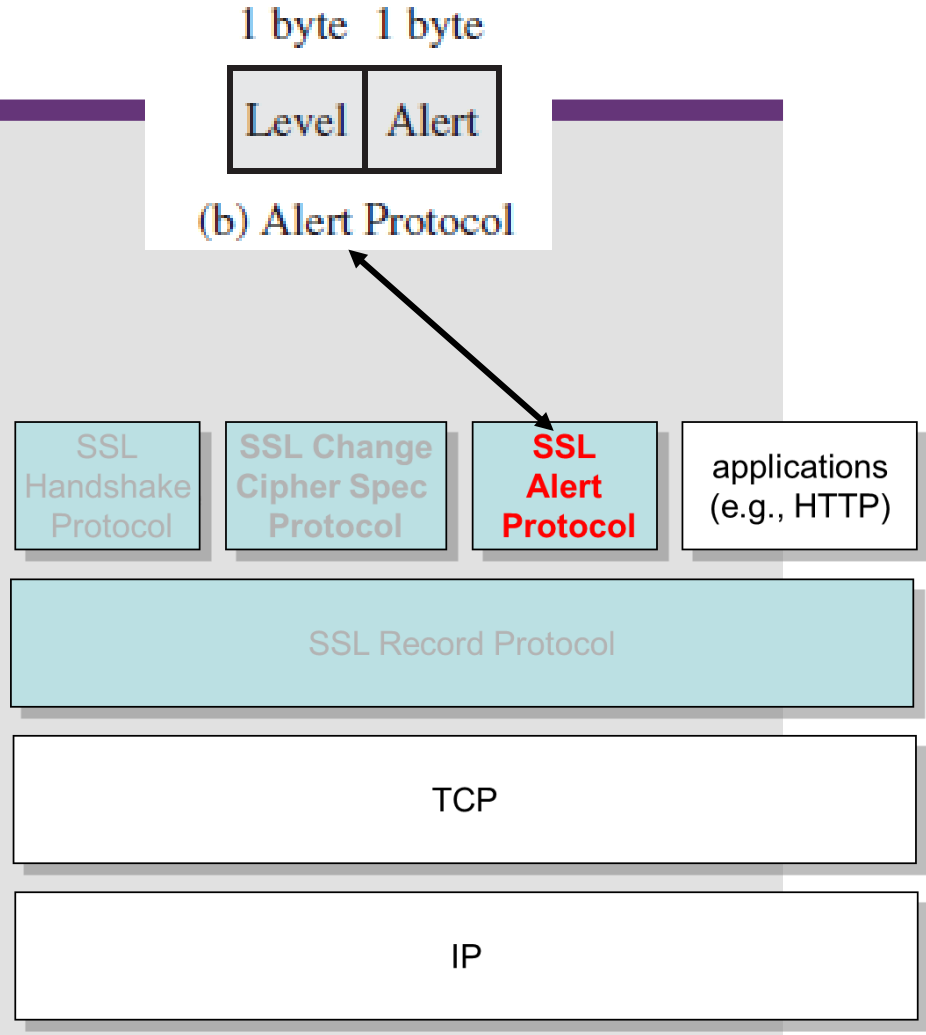




- **There are two states for the change cipher spec message.**
  - Read Current
  - Read Pending

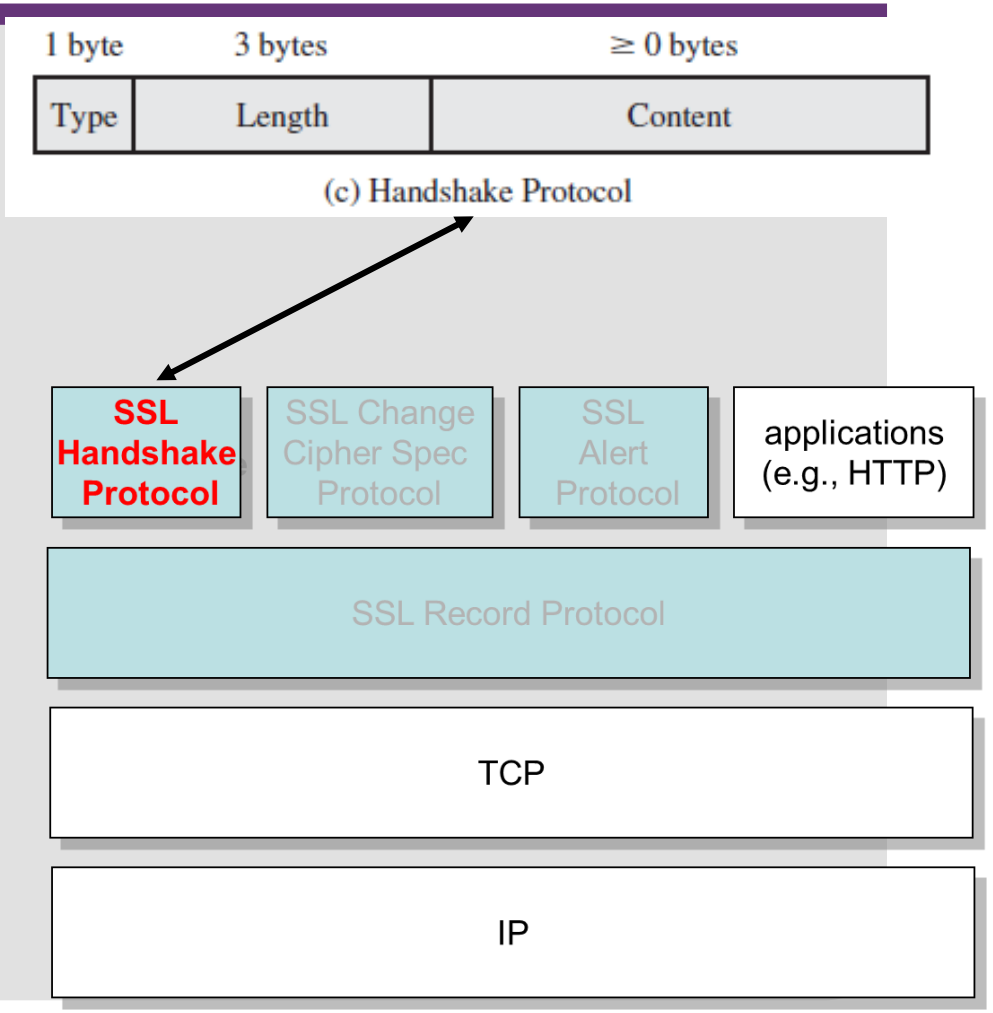
SSL Alert Protocol

- **conveys SSL-related alerts to peer entity**
- **Severity level**
  - **Warning (1) or fatal (2)**
- **specific alert codes**
  - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- **compressed & encrypted like all SSL data**



SSL Handshake Protocol

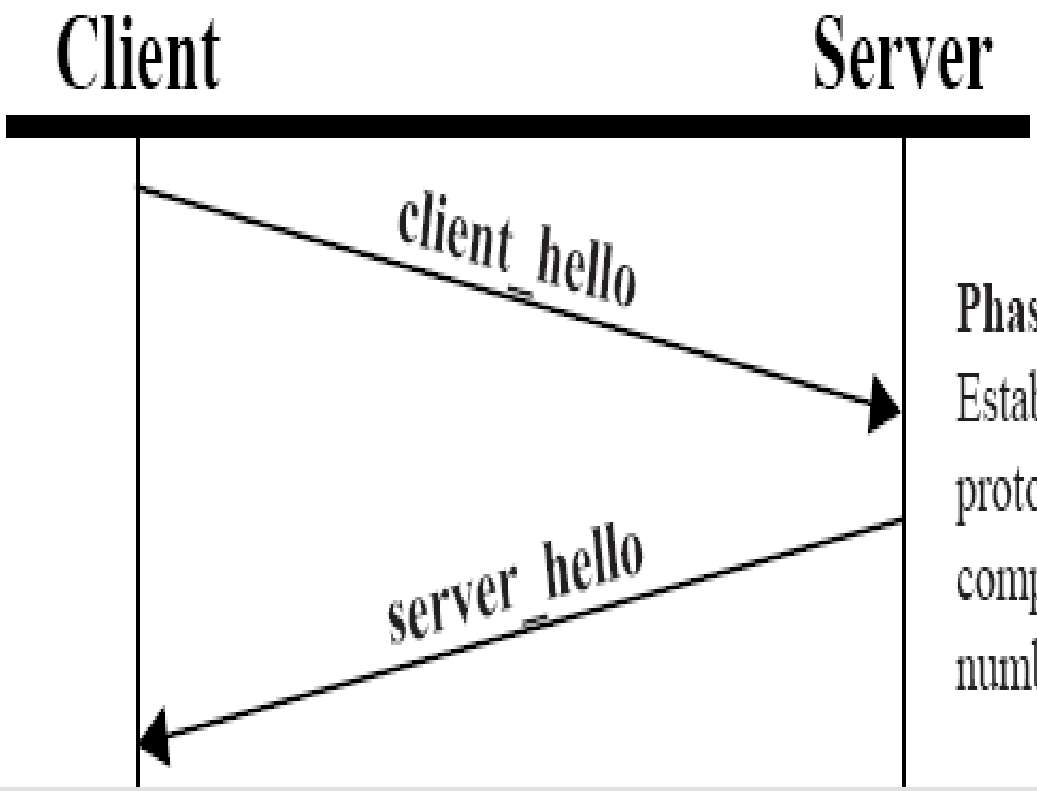
- **allows server & client to:**
  - **authenticate each other**
  - **to negotiate encryption & MAC algorithms**
  - **to negotiate cryptographic keys to be used**
- **comprises a series of messages in phases**
  - **Establish Security Capabilities**
  - **Server Authentication and Key Exchange**
  - **Client Authentication and Key Exchange**
  - **Finish**



SSL Handshake Protocol – Phase 1

- **Initiates a logical connection**
- **Establishes the security attribute of the connection**
- **Exchange of two messages takes place:**
  - **client sends a client_hello message which includes:**
    - **highest SSL version supported by client**
    - **28 byte random number and 32-bit timestamp**
    - **session ID**
    - **cipher suite – cryptographic algorithms supported by the client in order of preference**
    - **compression method supported by client**
  - **Server send server_hello message which contains the same parameters**
    - **version supported by server**
    - **random number generated by the server, independent of client**
    - **same or new session ID**
    - **cipher suite – single cipher suite selected by the server from those proposed by client**
    - **compression method selected by the server from those proposed by client**

SSL Handshake Protocol – Phase 1



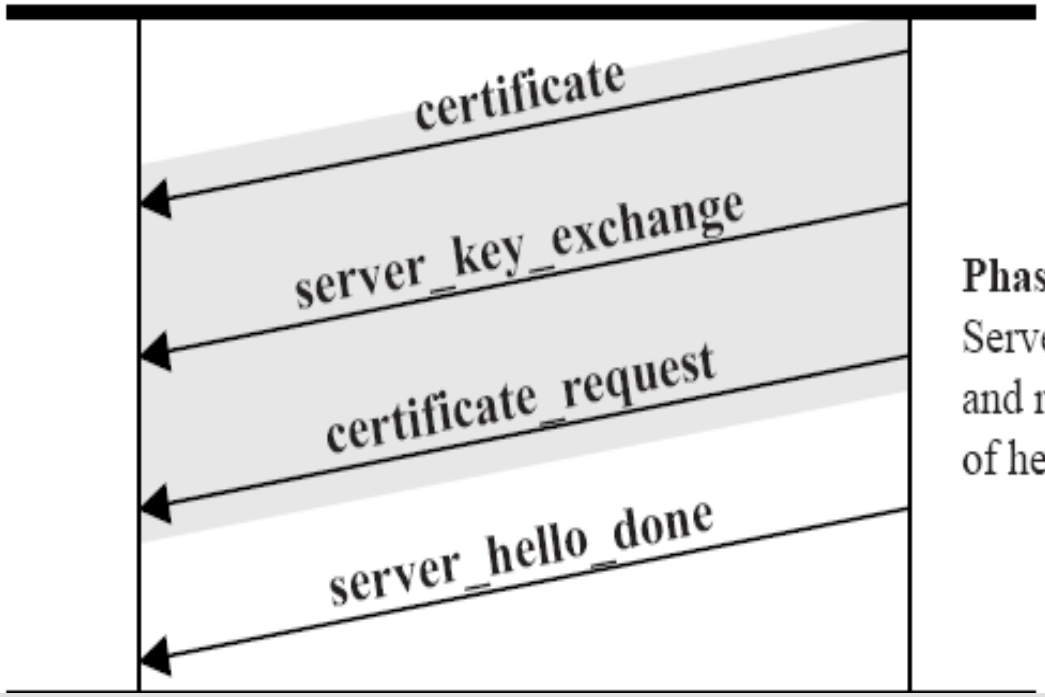
Phase 1

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

SSL Handshake Protocol - Phase 2

- **The server begins this phase by sending its certificate**
- **May send all of the following four messages:**
  - certificate message – required for agreed on key exchange method
  - may send server_key_exchange message – not required for all key exchange method
  - may send certificate_request message requesting a certificate from client
  - finally sends server_done message to indicate the end of the server messages

SSL Handshake Protocol - Phase 2

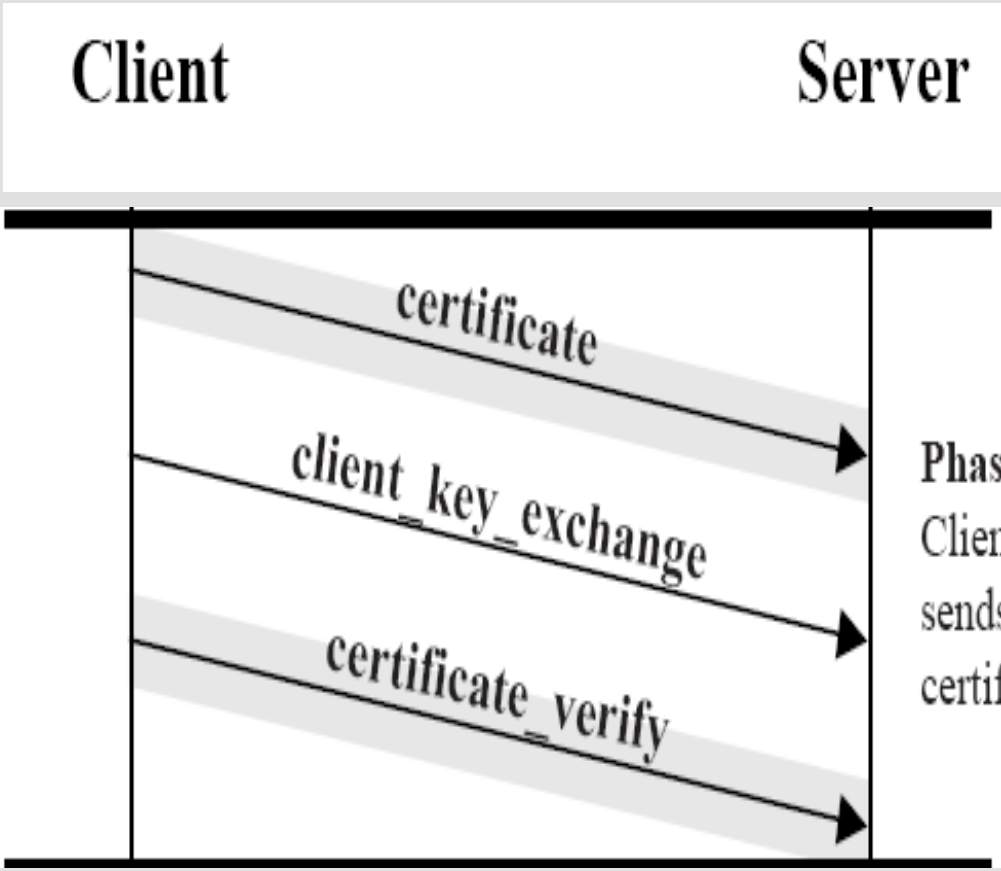


Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

SSL Handshake protocol – Phase 3

- **Client verifies server certificate**
- **Client checks whether server_hello parameters are acceptable**
- **If the above are satisfactory, client may send following messages back to the server**
  - may send certificate message if the server has requested it. no_certificate alert is sent if client does not have any certificate
  - sends client_key_exchange message, which must be sent to deliver the keys - the content of this message depends on the negotiated method of key exchange
  - may send certificate_verify message to provide explicit verification of the client certificate

SSL Handshake protocol – Phase 3



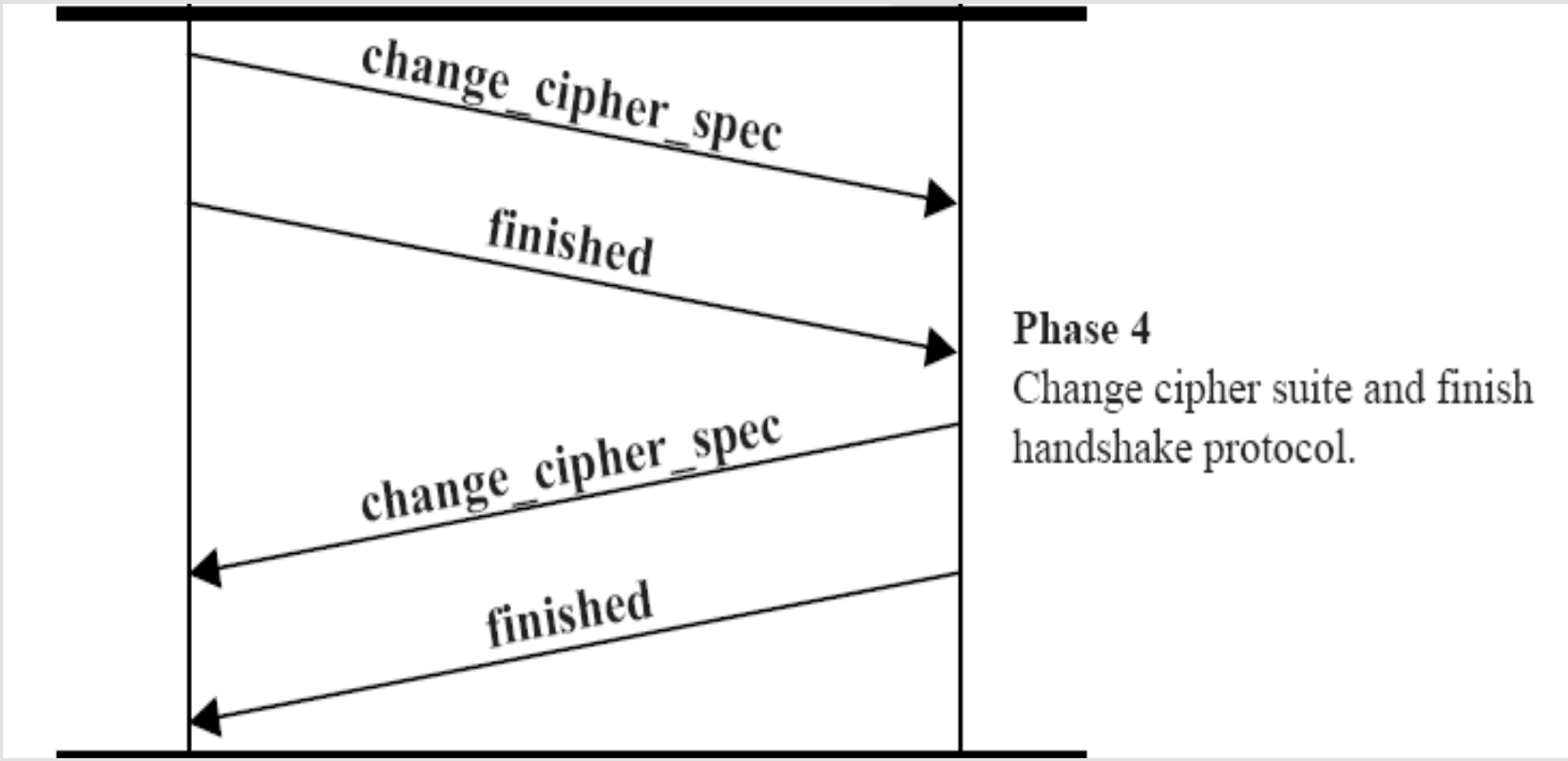
Phase 3

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

SSL Handshake Protocol – Phase 4

- **This phase completes setting up of a secure connection by exchanging messages between client and server**
- **Following messages are send in both directions:**
  - a change_cipher_spec message and copies pending CipherSpec into the current CipherSpec
  - a finished message under the new algorithm and keys
- **If server cannot be successfully authenticated by client on the basis of the delivered certificate, then**
  - handshake terminates
  - Client generates an error message

SSL Handshake Protocol – Phase 4



Cryptographic Computations

- **master secret creation**
  - a one-time 48-byte secret value is created (384 bits)
  - generated for this session by means of secure key exchange (RSA / Diffie-Hellman) and then hashing info
- **generation of cryptographic parameters**
  - client write MAC secret, a server write MAC secret;
  - a client write key, a server write key;
  - client write IV, and a server write IV
  - generated by hashing master secret

TLS (Transport Layer Security)

- **IETF standard RFC 2246 similar to SSLv3**
- **with minor differences**
  - in record format version number
  - uses HMAC for MAC
  - a pseudo-random function expands secrets
    - > **based on HMAC using SHA-1 or MD5**
  - has additional alert codes
  - some changes in supported ciphers
  - changes in certificate types & negotiations
  - changes in crypto computations & padding

HTTPS: Secure Hypertext Transfer Protocol

- **HTTPS (HTTP over SSL)**
  - combination of HTTP & SSL/TLS to secure communications between browser & server
  - documented in RFC2818, *HTTP over TLS*
  - no fundamental change using either SSL or TLS
- **use https:// URL rather than http://**
  - and port 443 rather than 80
- **encrypts**
  - URL, document contents, form data, cookies, HTTP headers

HTTPS Use

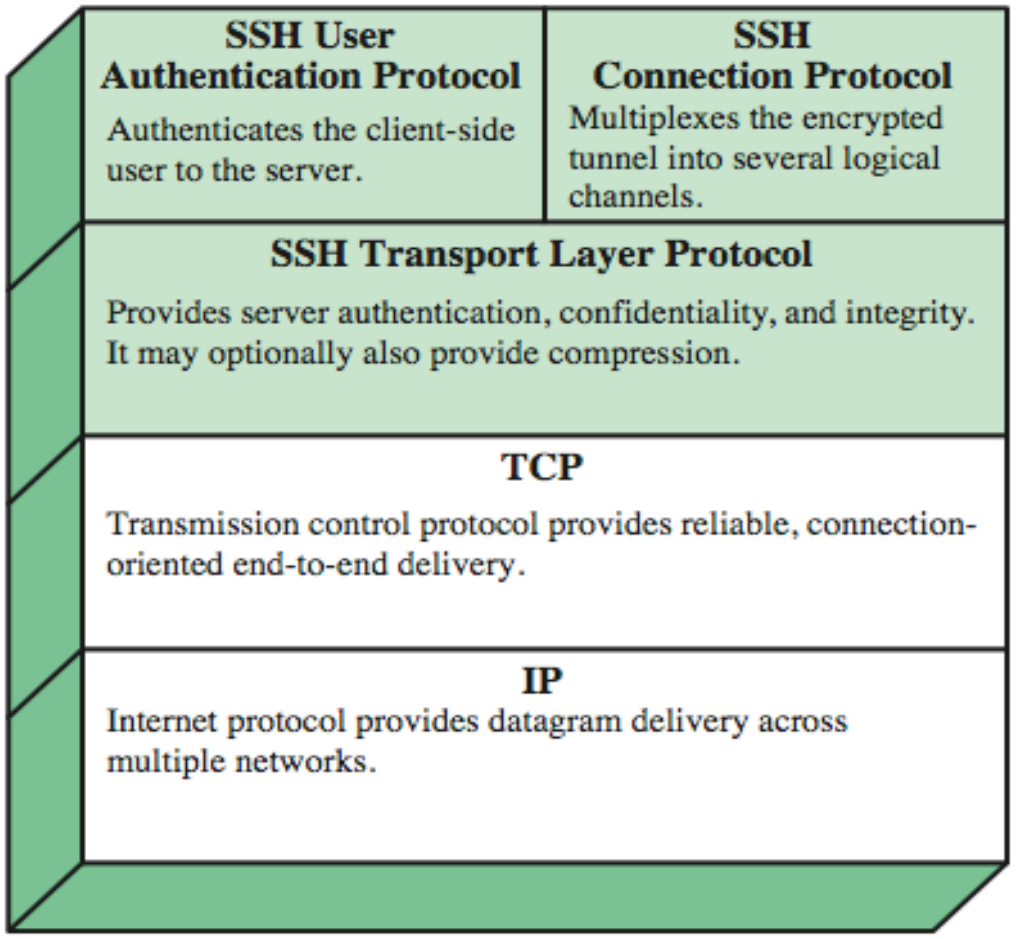
- **connection initiation**
  - TLS handshake then HTTP request(s)
- **connection closure**
  - have “Connection: close” in HTTP record
  - TLS level exchange close_notify alerts
  - can then close TCP connection
  - must handle TCP close before alert exchange sent or completed

SSH Secure Shell Protocol
SSHv1
SSHv2

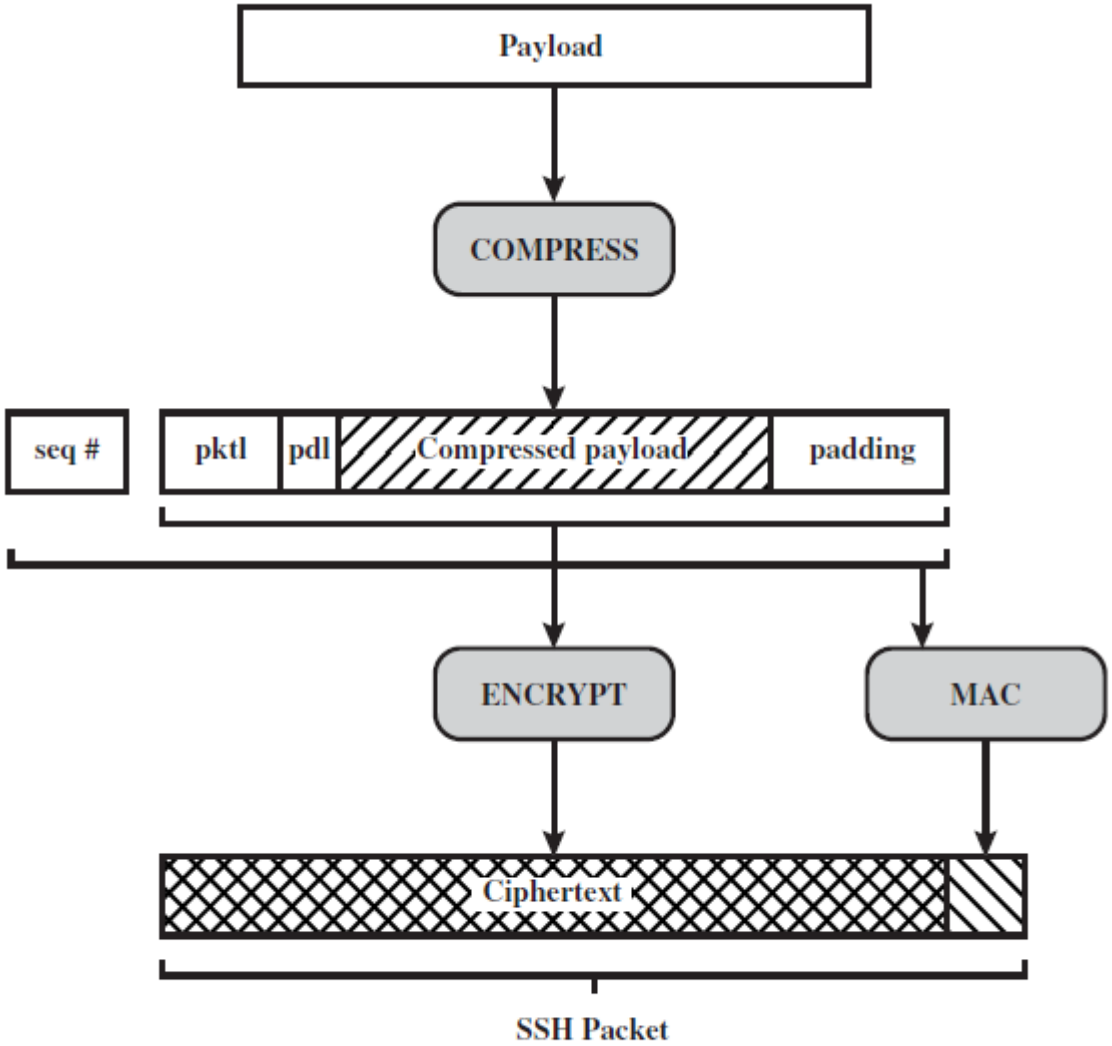
Secure Shell (SSH)

- **protocol for secure network communications**
  - designed to be simple & inexpensive
- **SSH1 provided secure remote logon facility**
  - replace TELNET & other insecure schemes
  - also has more general client/server capability
- **SSH2 fixes a number of security flaws**
- **documented in RFCs 4250 through 4254**
- **SSH clients & servers are widely available**
- **method of choice for remote login/ X tunnels**

SSH Protocol Stack



SSH Transport Layer Protocol Packet Formation



pktl = packet length
pdl = padding length

SSH Transport Layer Protocol

- **server authentication occurs at transport layer, based on server/host key pair(s)**
  - server authentication requires clients to know host keys in advance
- **packet exchange**
  - establish TCP connection
  - can then exchange data
    - > identification string exchange, algorithm negotiation, key exchange, end of key exchange, service request
  - using specified packet format

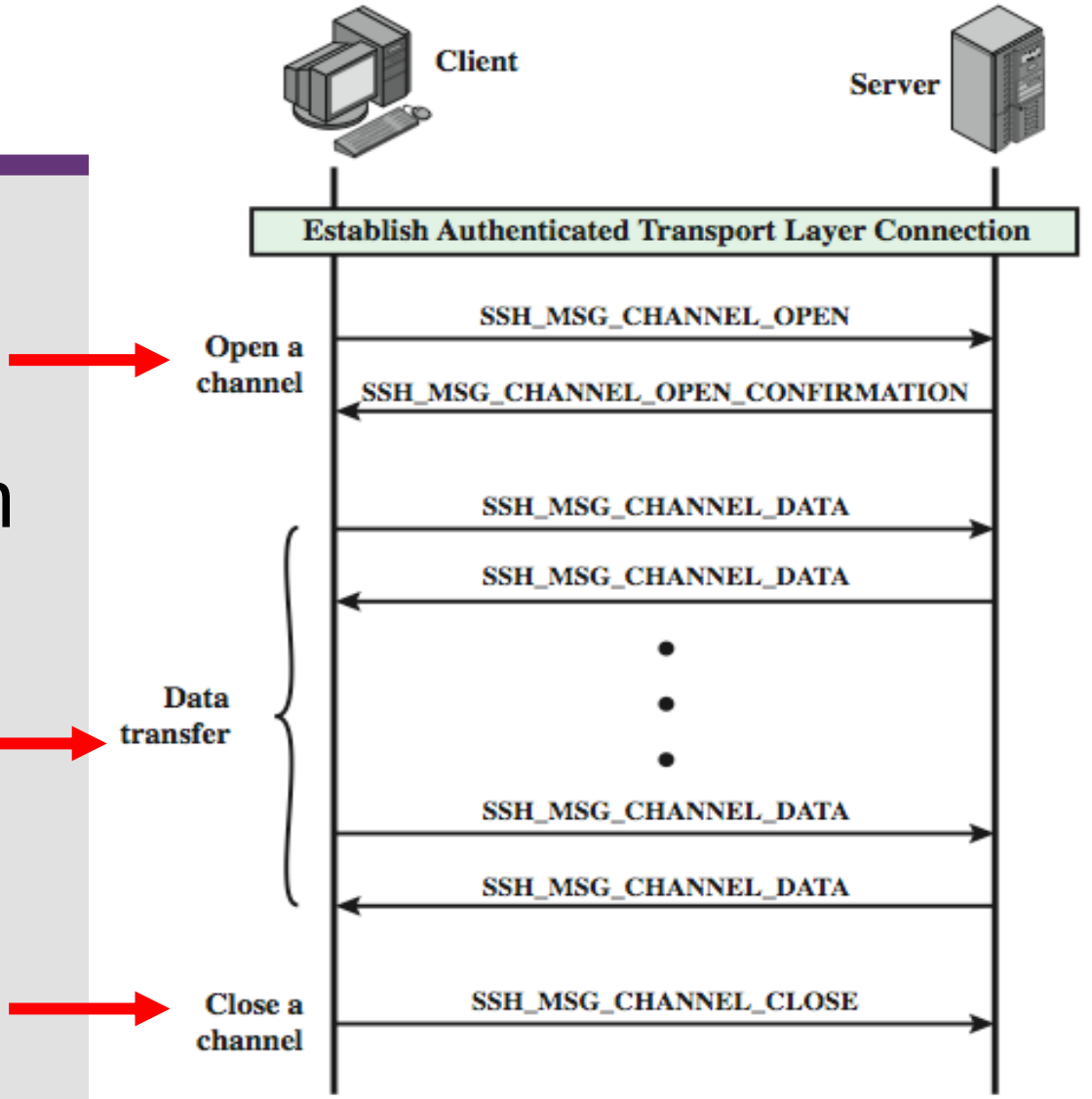
SSH User Authentication Protocol

- **authenticates client to server**
- **three message types:**
  - SSH_MSG_USERAUTH_REQUEST
  - SSH_MSG_USERAUTH_FAILURE
  - SSH_MSG_USERAUTH_SUCCESS
- **authentication methods used**
  - public-key, password, host-based

SSH Connection Protocol

- **runs on SSH Transport Layer Protocol**
- **assumes secure authentication connection**
- **used for multiple logical channels**
  - SSH communications use separate channels
  - either side can open with unique id number
  - flow controlled
  - have three stages:
    - opening a channel, data transfer, closing a channel
  - four types:
    - session, x11, forwarded-tcpip, direct-tcpip.

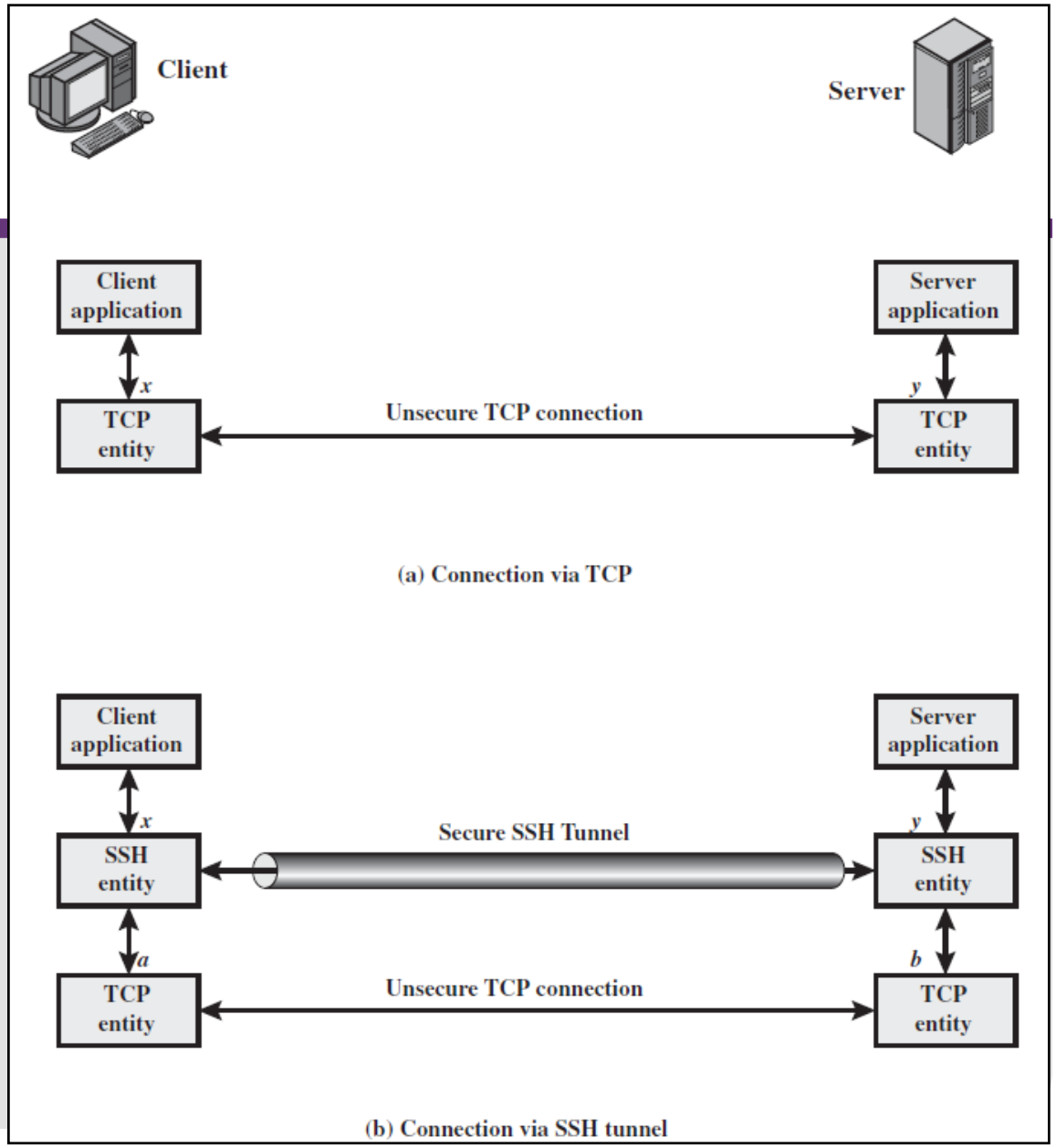
SSH Connection Protocol Exchange



Port Forwarding

- **convert insecure TCP connection into a secure SSH connection**
  - SSH Transport Layer Protocol establishes a TCP connection between SSH client & server
  - client traffic redirected to local SSH, travels via tunnel, then remote SSH delivers to server
- **supports two types of port forwarding**
  - local forwarding – hijacks selected traffic
  - remote forwarding – client acts for server

Port Forwarding



Summary

- **have considered:**
  - need for web security
  - SSL/TLS transport layer security protocols
  - HTTPS
  - secure shell (SSH)

Further Reading

- **Study Guide 5**
- **Chapter 5 of the textbook: Network Security Essentials-Application & Standards" by William Stallings 5th Edition, Prentice Hall, 2013**
- **Additional resources for this week**

- **Acknowledgement: part of the materials presented in the slides was developed with the help of Instructor's Manual and other resources made available by the author of the textbook.**