

Cryptography and Network Security

Chapter 16

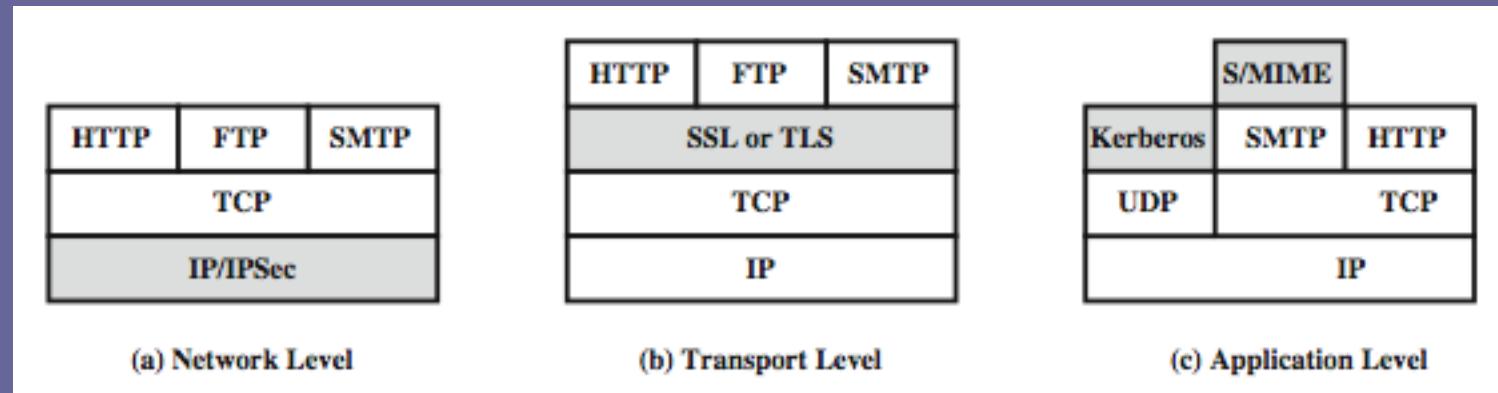
Fifth Edition
by William Stallings



Web Security

- Web now widely used by business, government, individuals
- but Internet & Web are vulnerable
- have a variety of threats
 - integrity
 - confidentiality
 - denial of service
 - authentication
- need added security mechanisms

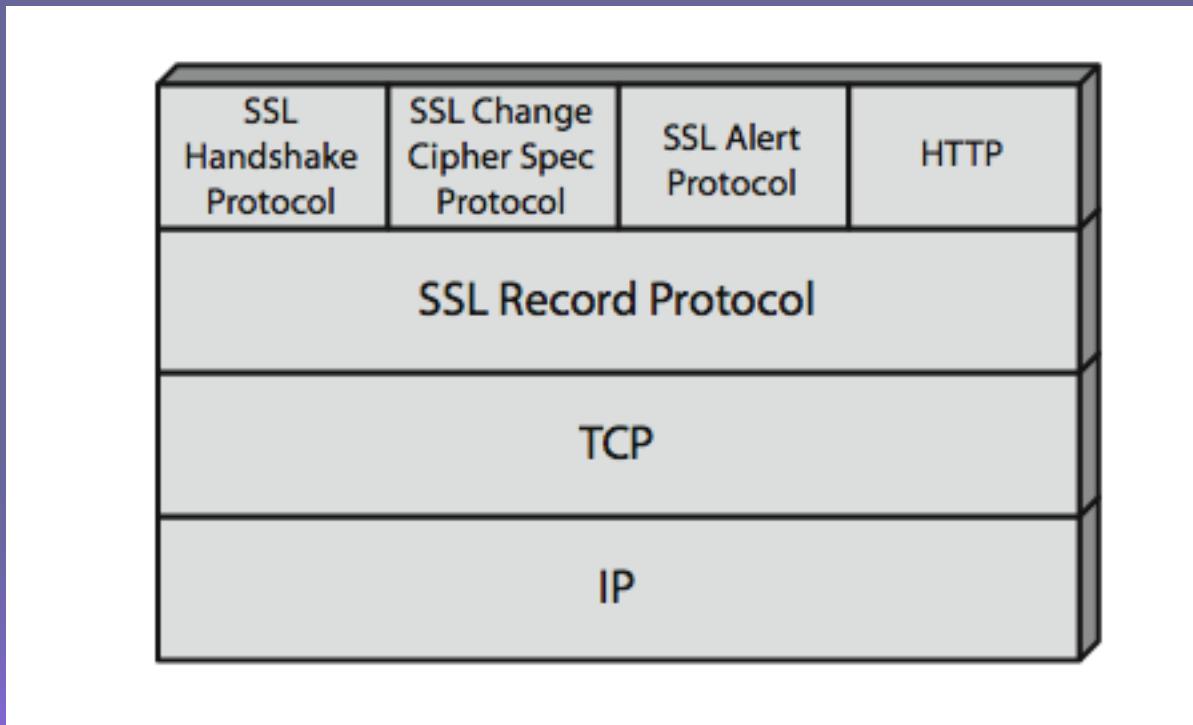
Web Traffic Security Approaches



SSL (Secure Socket Layer)

- transport layer security service
- originally developed by Netscape
- version 3 designed with public input
- subsequently became Internet standard known as TLS (Transport Layer Security)
- uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



SSL Architecture

➤ SSL connection

- a transient, peer-to-peer, communications link
- 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 Record Protocol Services

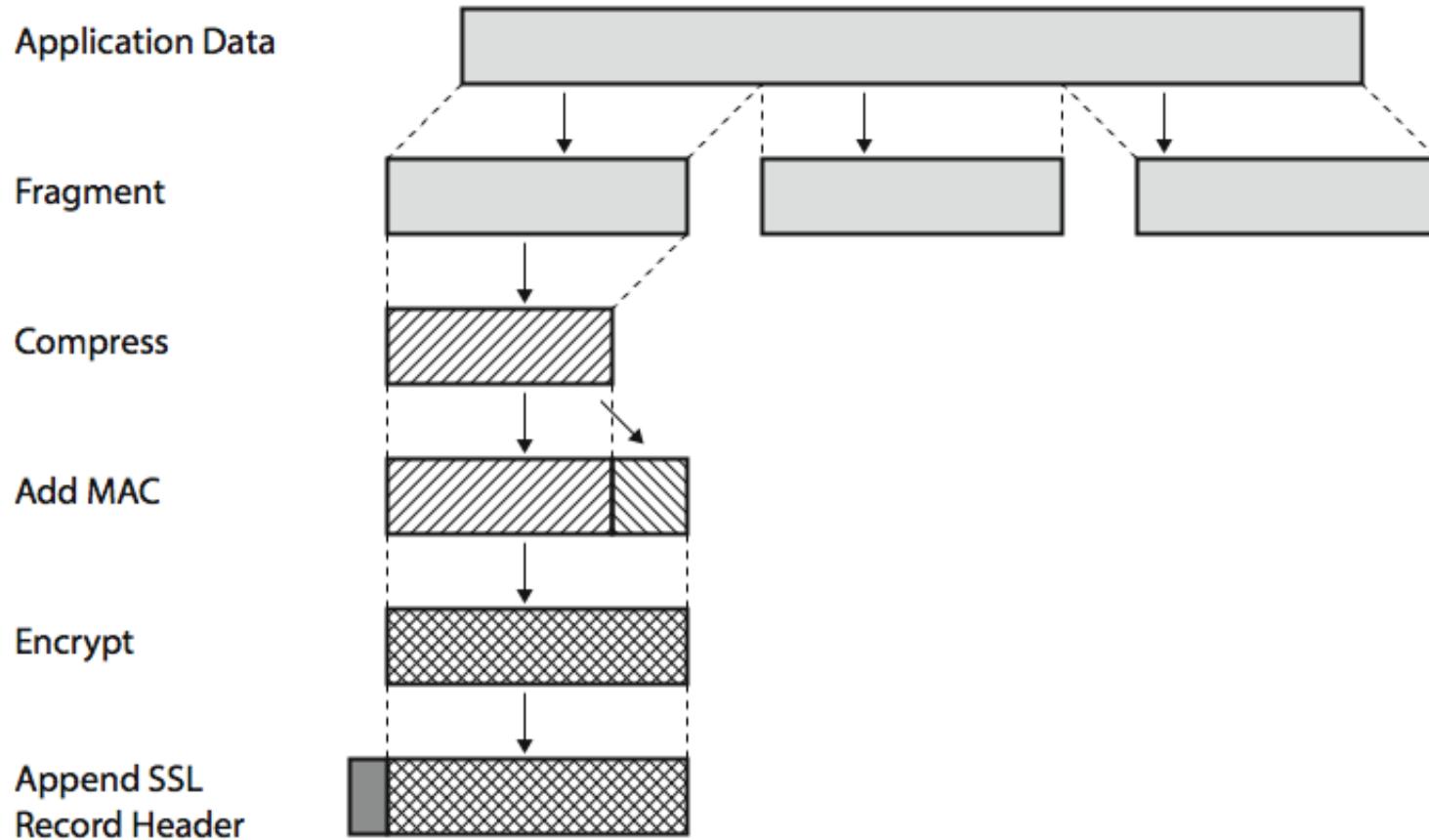
➤ confidentiality

- 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

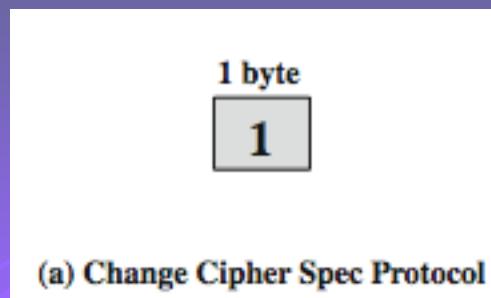
- using a MAC with shared secret key
- similar to HMAC but with different padding

SSL Record Protocol Operation



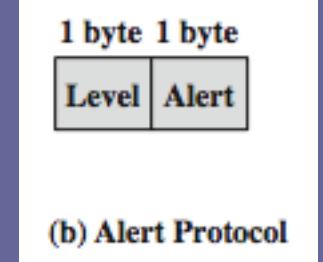
SSL Change Cipher Spec Protocol

- 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



SSL Alert Protocol

- conveys SSL-related alerts to peer entity
- severity
 - warning or fatal
- specific alert
 - 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

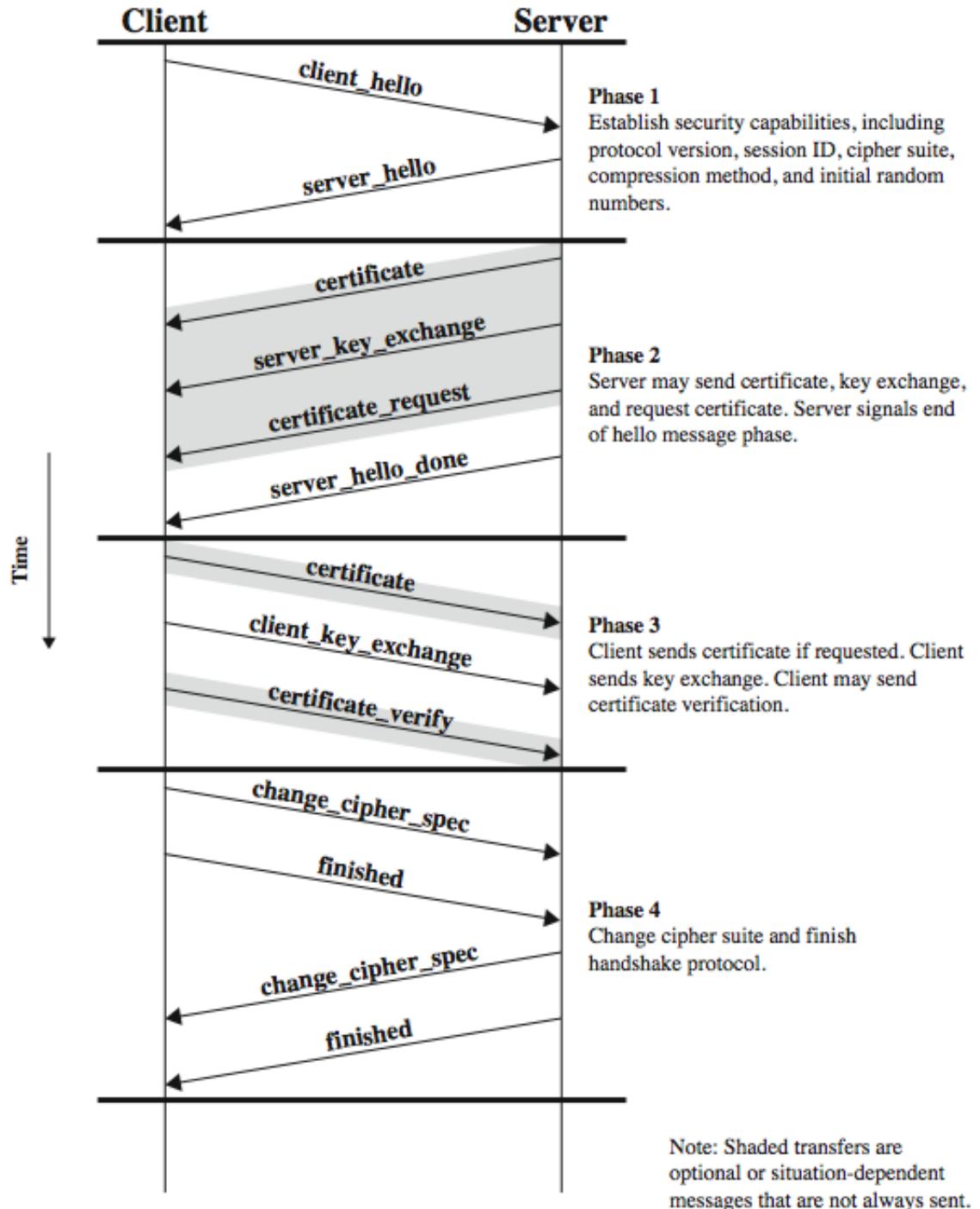


SSL Handshake Protocol

- 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
 1. Establish Security Capabilities
 2. Server Authentication and Key Exchange
 3. Client Authentication and Key Exchange
 4. Finish

1 byte	3 bytes	≥ 0 bytes
Type	Length	Content
		(c) Handshake Protocol

SSL Handshake Protocol



Cryptographic Computations

- master secret creation
 - a one-time 48-byte value
 - generated using 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, a 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

- HTTPS (HTTP over SSL)
 - combination of HTTP & SSL/TLS to secure communications between browser & server
 - documented in RFC2818
 - 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

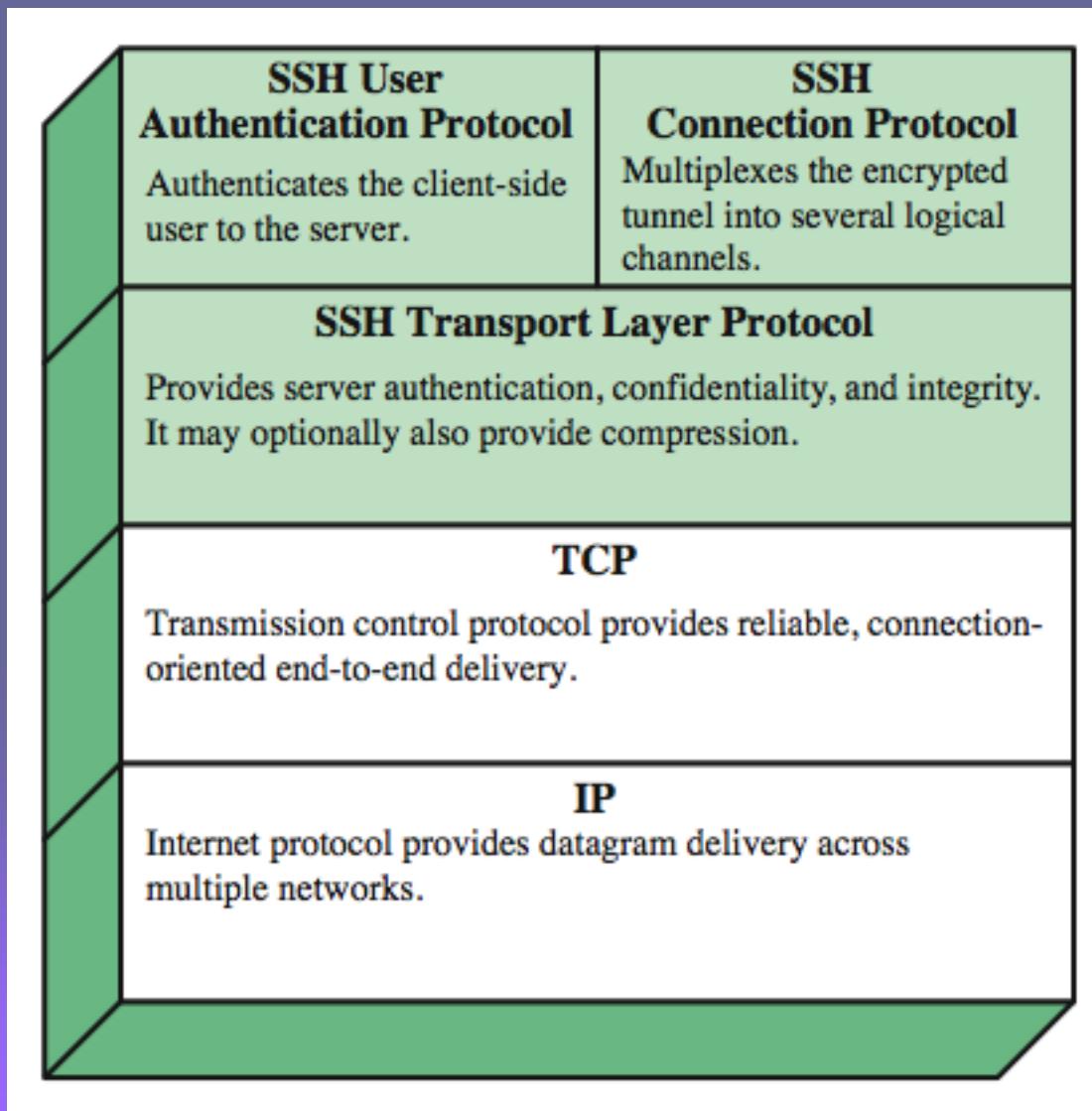
HTTPS Use

- 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

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

- 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

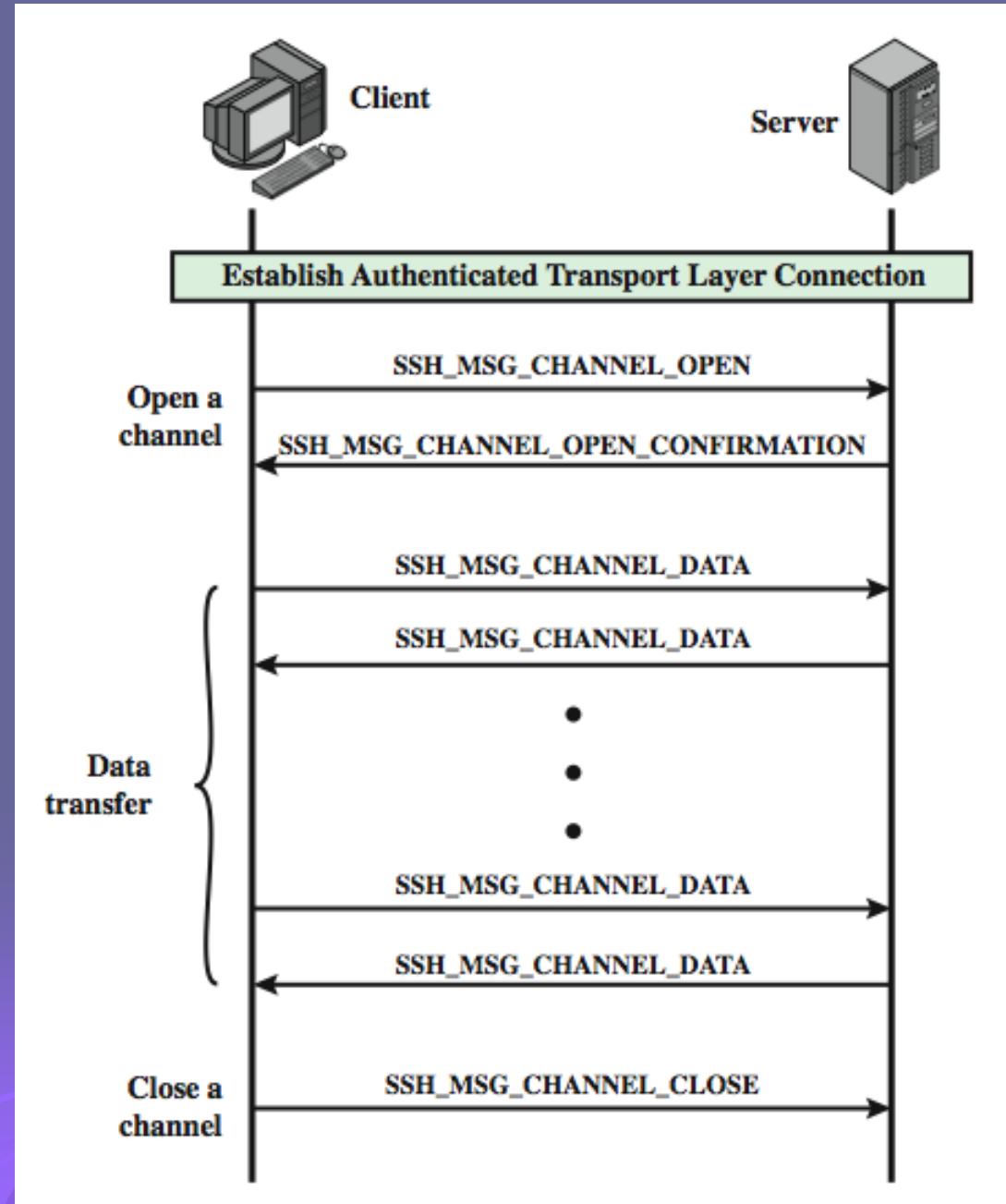
SSH User Authentication Protocol

- 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

Summary

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