Stockholm University

Transport Level Security

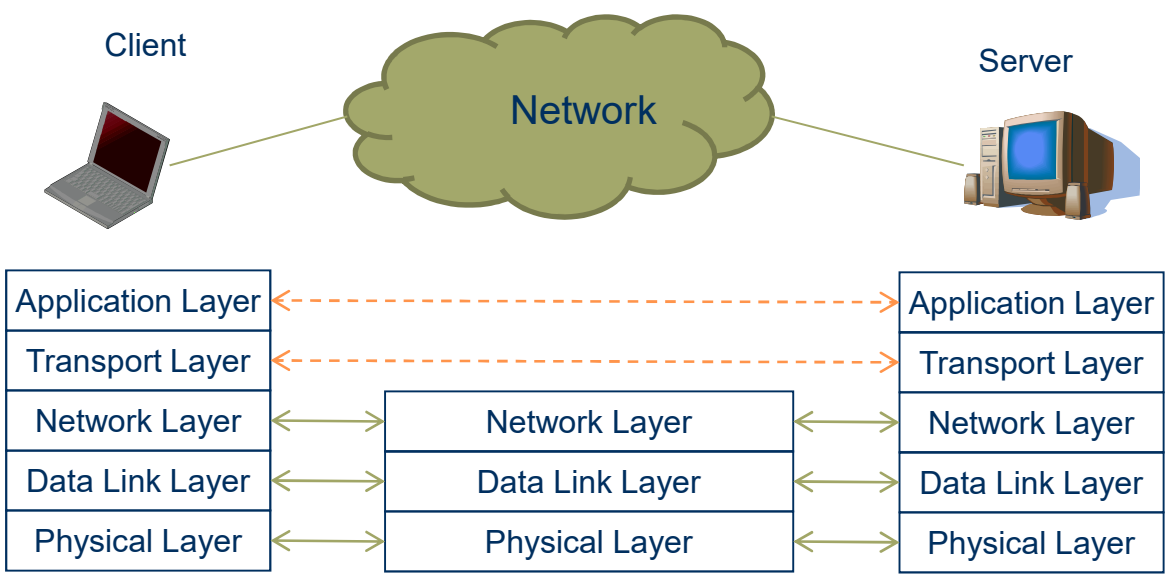
Network Security (NETSEC)

Yuhong Li

What is Transport Level Security -1

Stockholm University

Networking Reference Model

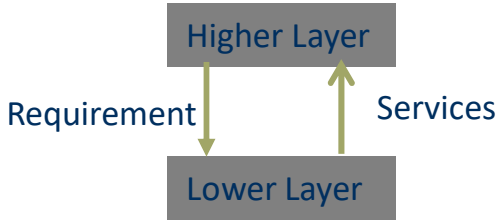


What is Transport Level Security -2

Stockholm University

- TCP/IP Transport layer

Application Layer	Web services, remote login, multimedia streaming …
Transport Layer	Congestion control, flow control, error control, quality of service
Network Layer	Addressing, routing, device location, hand-over
Data link Layer	Authentication, media access, multiplexing, media access control
Physical Layer	Encryption, modulation, interference, attenuation, frequency

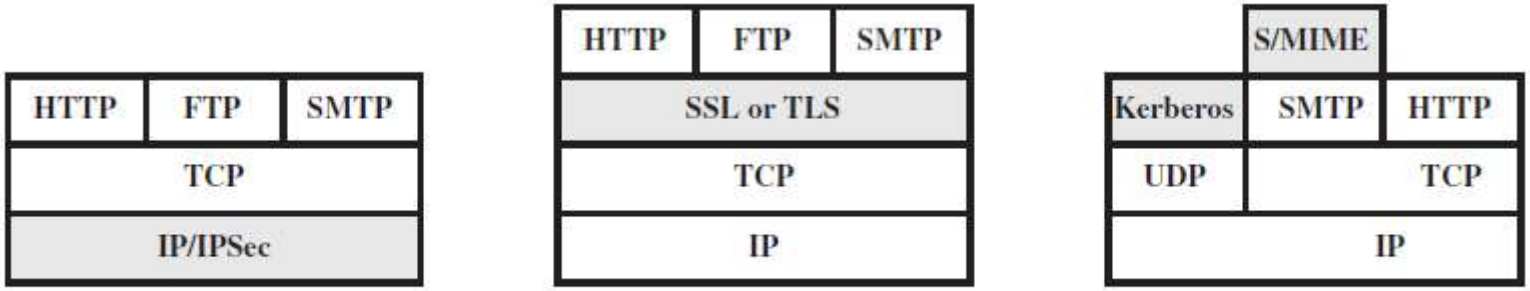


Transport Layer: End-to-end ordered and reliable connection.

What is Transport Level Security -3

Stockholm University

Security mechanisms in TCP/IP model

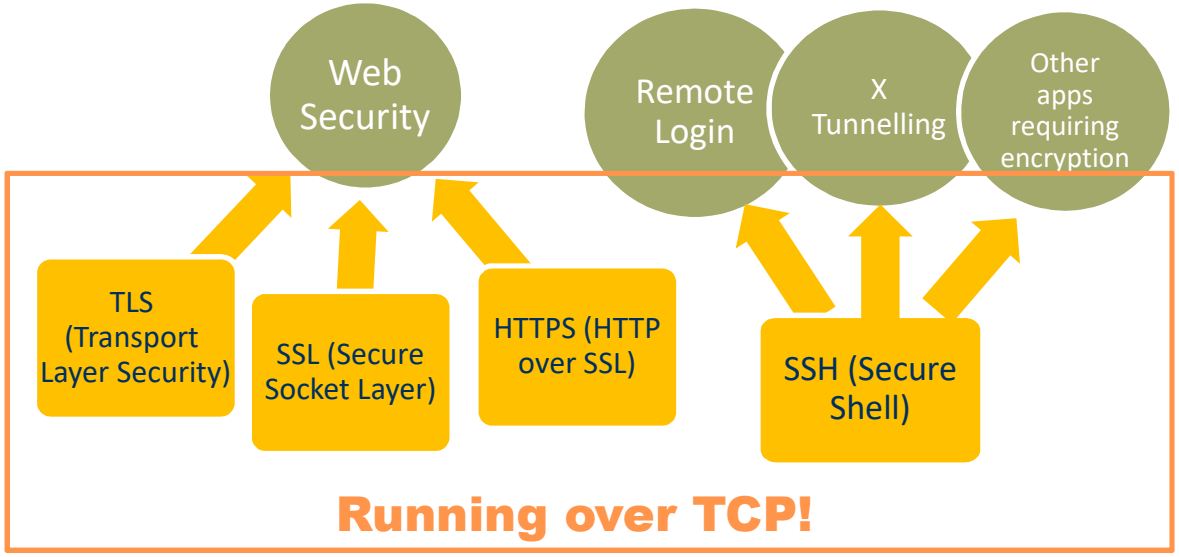


Transport level security: security mechanisms based on (or on top of) transport layer, providing secure services to application layer

Why Transport Level Security?

Stockholm University

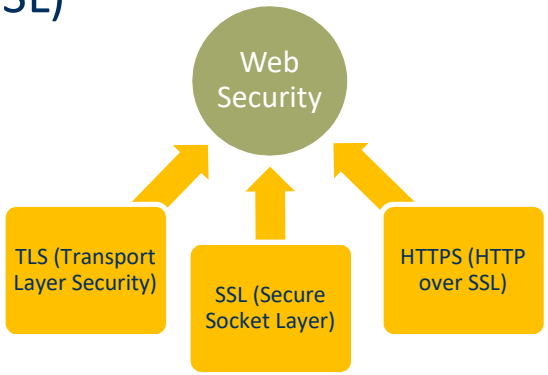
Applications require security on top of reliable and ordered data transmission

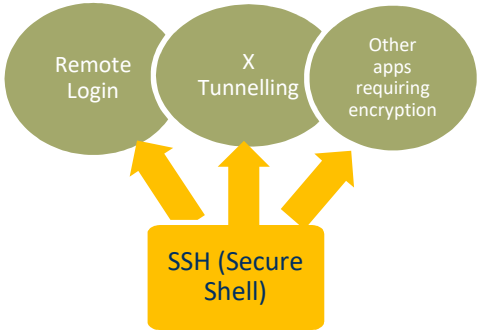


- Application layer:
  - Not convenient, cost
- IPSec
  - has too much overhead!

Outline

- Web security considerations
  - Web security threats
  - Web traffic security approaches
- Secure Socket Layer (SSL)
- Transport Layer Security (TSL)
- HTTP over SSL (HTTPS)
- Secure Shell (SSH)





Web Service Considerations

Stockholm University

- Web service: a client/server application, running over the Internet and TCP/IP based Intranets
- Needs for tailored security tools:
  - May hide many potential security flaws
    - Web servers are relatively easy to configure and manage
    - Web content is increasingly easy to develop
    - The underlying software is extraordinarily complex
  - A Web server play an important role: e.g., can be exploited as a launching pad into the corporation's or agency's entire computer complex
  - Casual and untrained (in security matters) users are common clients for Web-based services
    - Not aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures

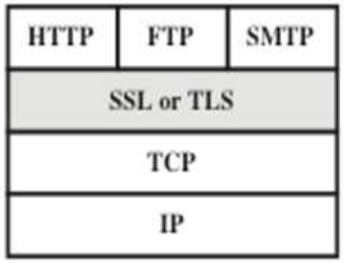
Stockholm University

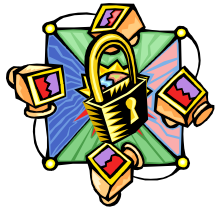
Secure Socket Layer (SSL)

Secure Sockets Layer (SSL)

Stockholm University

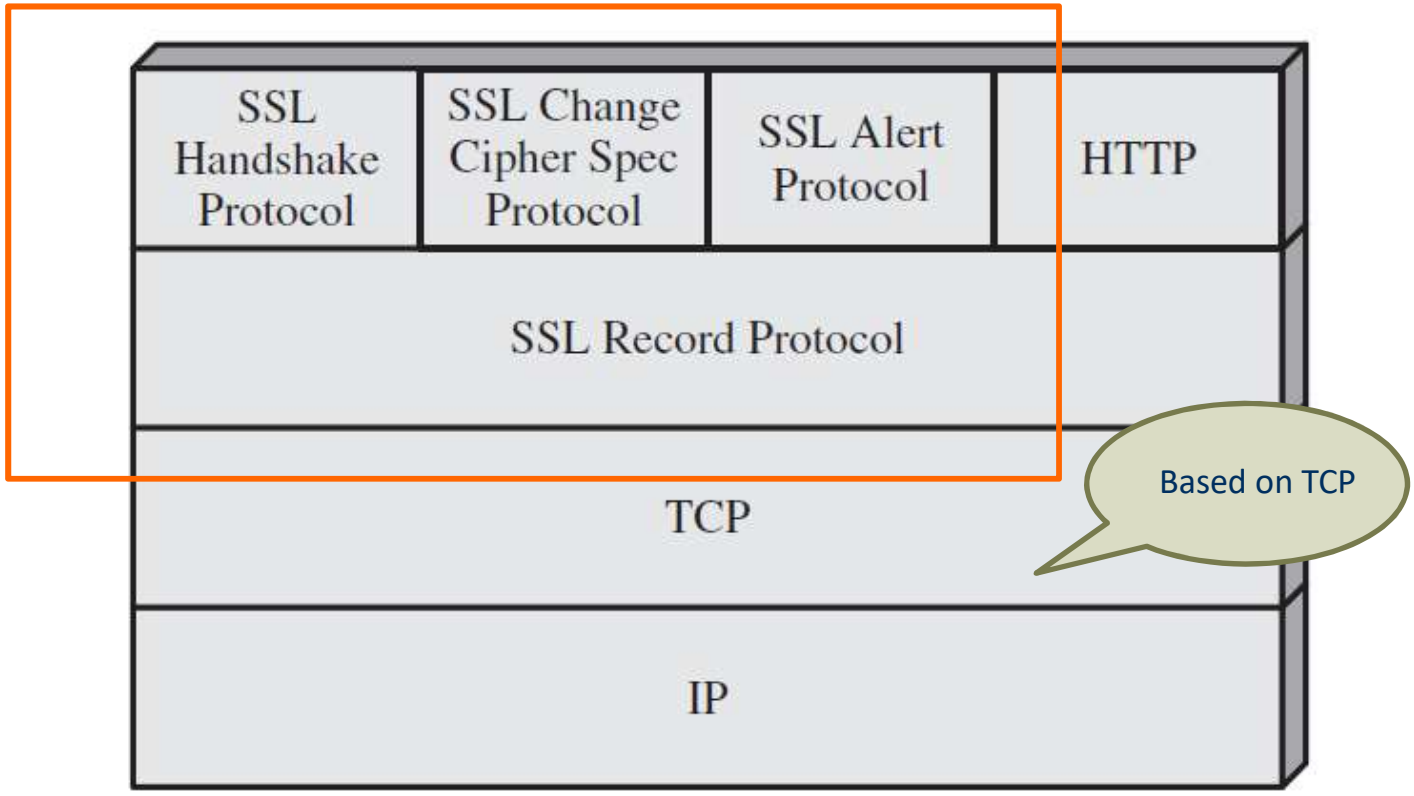
- A general purpose service, implemented as **a set of protocols** that rely on TCP
  - Could be provided as part of the underlying protocol suite and therefore be transparent to applications
  - Can be embedded in specific packages, e.g., integrated in a browser
- One of the most widely used security services





SSL Protocol Stack

Stockholm University



SSL Architecture

Stockholm University

Two important SSL concepts:

SSL connection

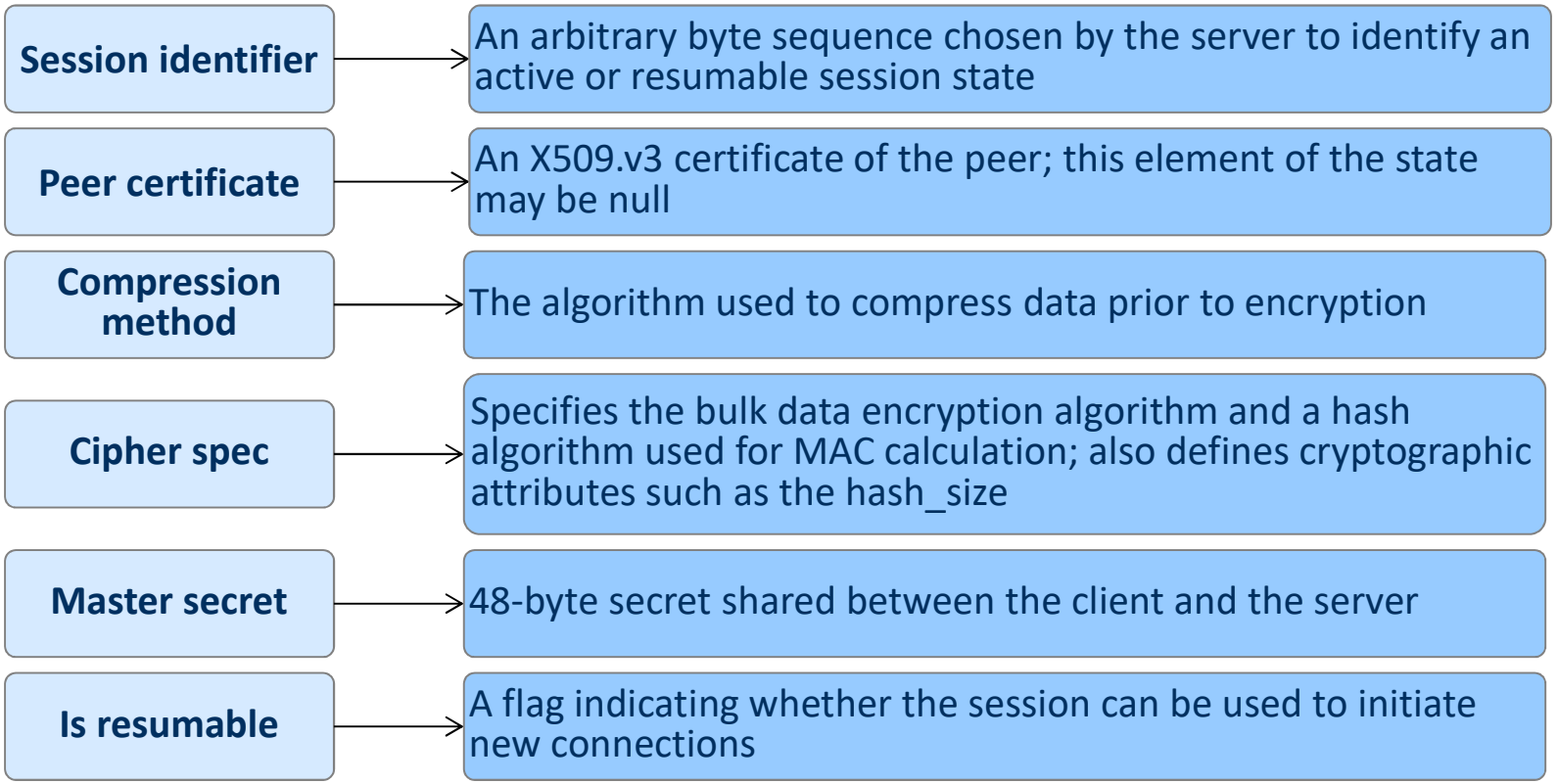
- A transport that provides a suitable type of service (for SSL such connections are peer-to-peer relationships)
- Connections are transient
- Every connection is associated with one session

SSL session

- An association between a client and a server
- Created by the Handshake Protocol
- Define a set of cryptographic security parameters which can be shared among multiple connections
- Are used to avoid the expensive negotiation of new security parameters for each connection

Parameters Defining a Session State

Stockholm University

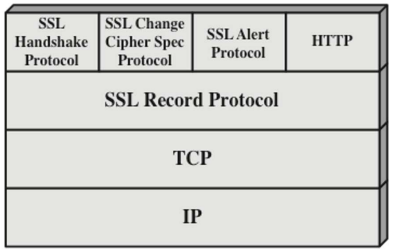


Parameters Defining a Connection State

Stockholm University

Server & client random	• Byte sequences that are chosen by the server and client for each connection
Server write MAC secret	• The secret key used in MAC operations on data sent by the server
Client write MAC secret	• The secret key used in MAC operations on data sent by the client
Server write key	• The secret encryption key for data encrypted by the server and decrypted by the client
Client write key	• The symmetric encryption key for data encrypted by the client and decrypted by the server
Initialization vectors	• When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key • This field is first initialized by the SSL Handshake Protocol • The final ciphertext block from each record is preserved for use as the IV with the following record
Sequence numbers	• Each party maintains separate sequence numbers for transmitted and received messages for each connection • When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero • Sequence numbers may not exceed $2^{64} - 1$

SSL Record Protocol



The SSL Record Protocol provides two services for SSL connections

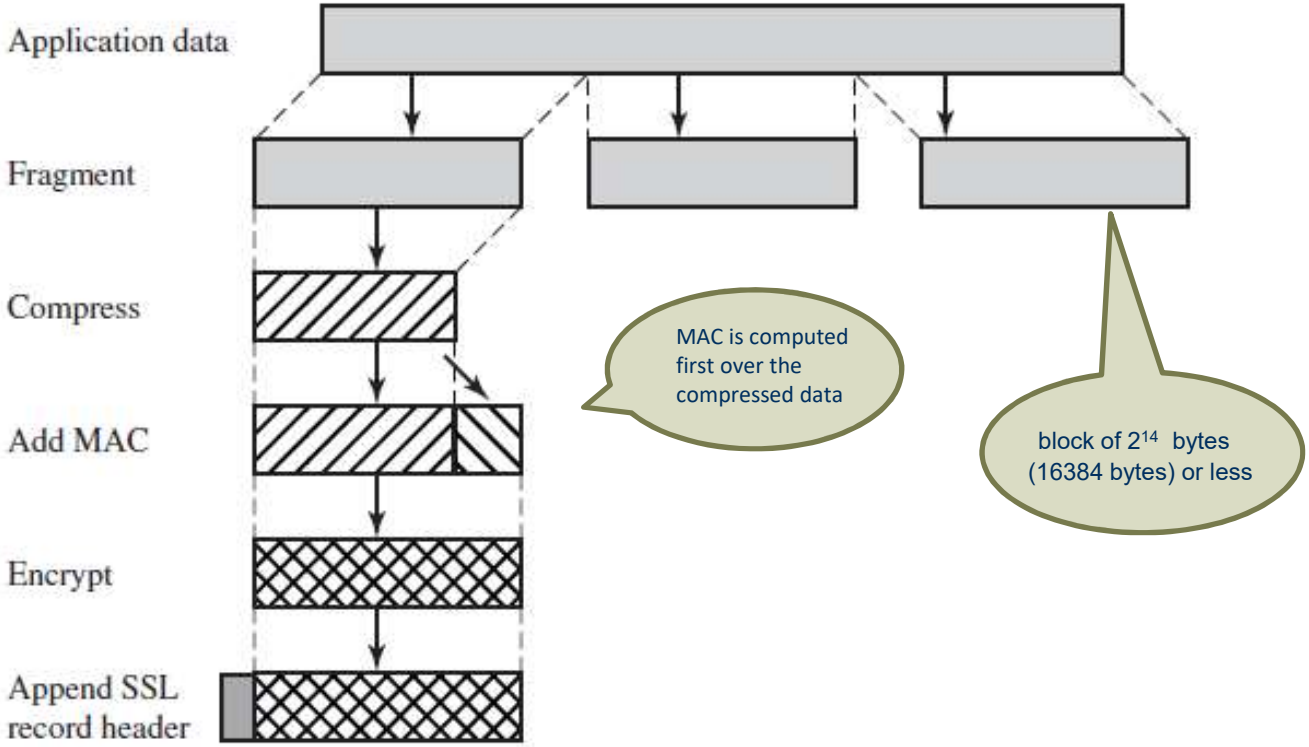
Confidentiality

The Handshake Protocol defines a shared **secret key** that is used **for** conventional **encryption of** SSL **payloads**

Message integrity

The Handshake Protocol also defines a shared **secret key** that is used to form a message **authentication code (MAC)**

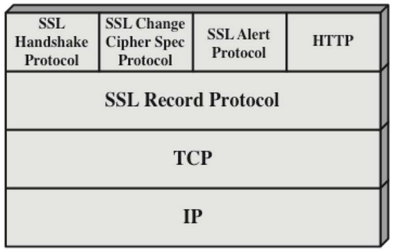
SSL Record Protocol Operation

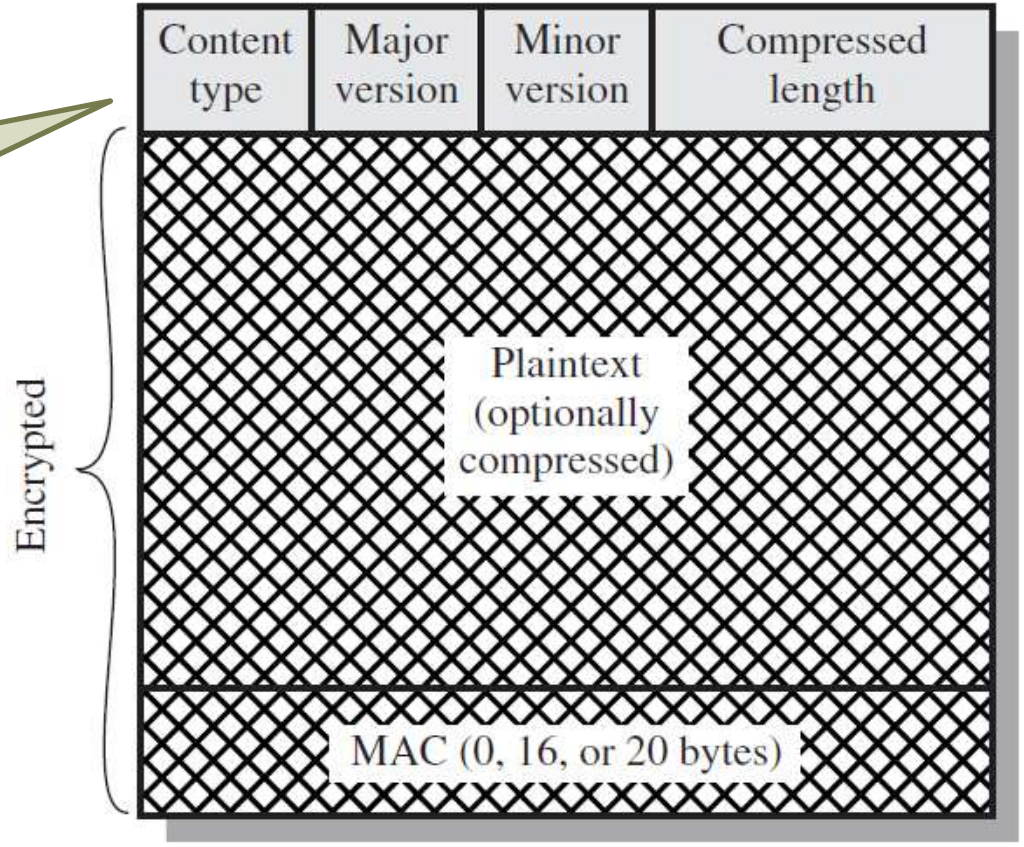


SSL Record Format

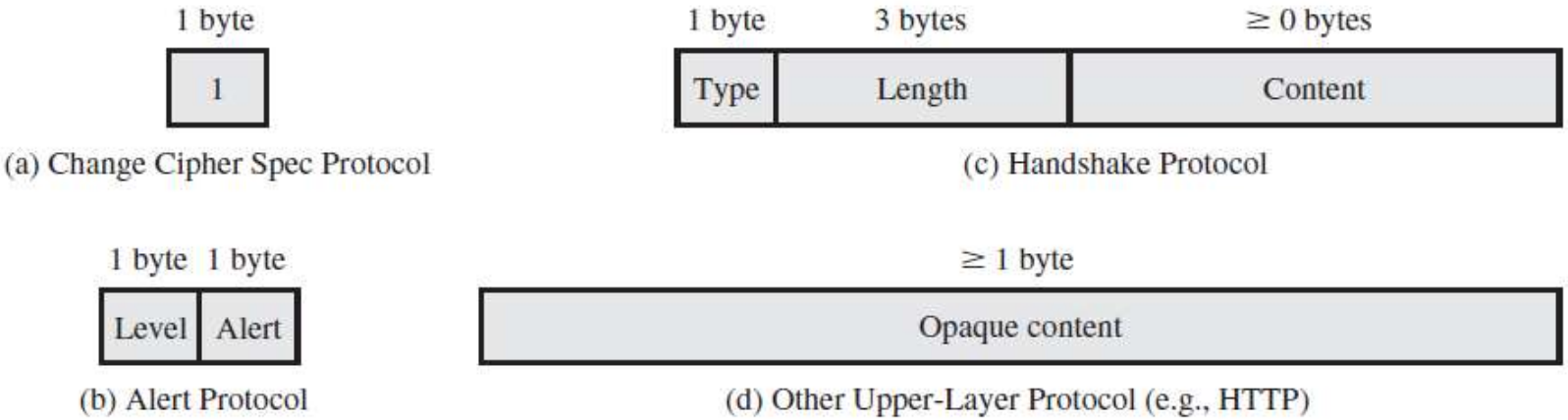
Stockholm University

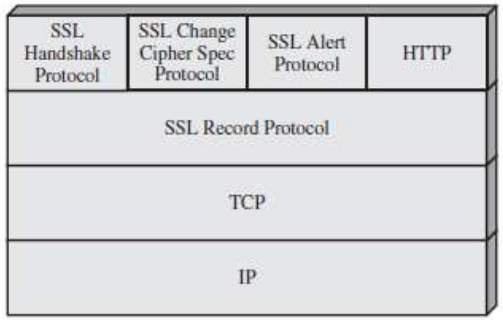
a)change_cipher_spec
b)alert
c) handshake
d) application_data





SSL Record Protocol Payload

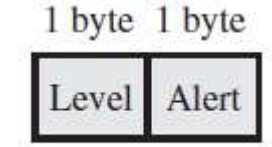




SSL Alert Protocol

Stockholm University

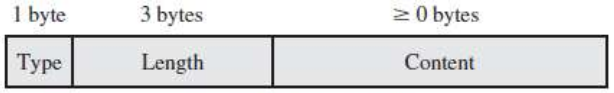
- Used to convey SSL-related alerts to the peer entity
  - Compressed and encrypted
- Level: the level of the severity of the message
  - Warning
  - Fatal: SSL immediately terminates the connection
- Alert:
  - unexpected_message; bad_record_mac; decompression_failure; handshake_failure; illegal_parameter; ….



(b) Alert Protocol

SSL Handshake Protocol

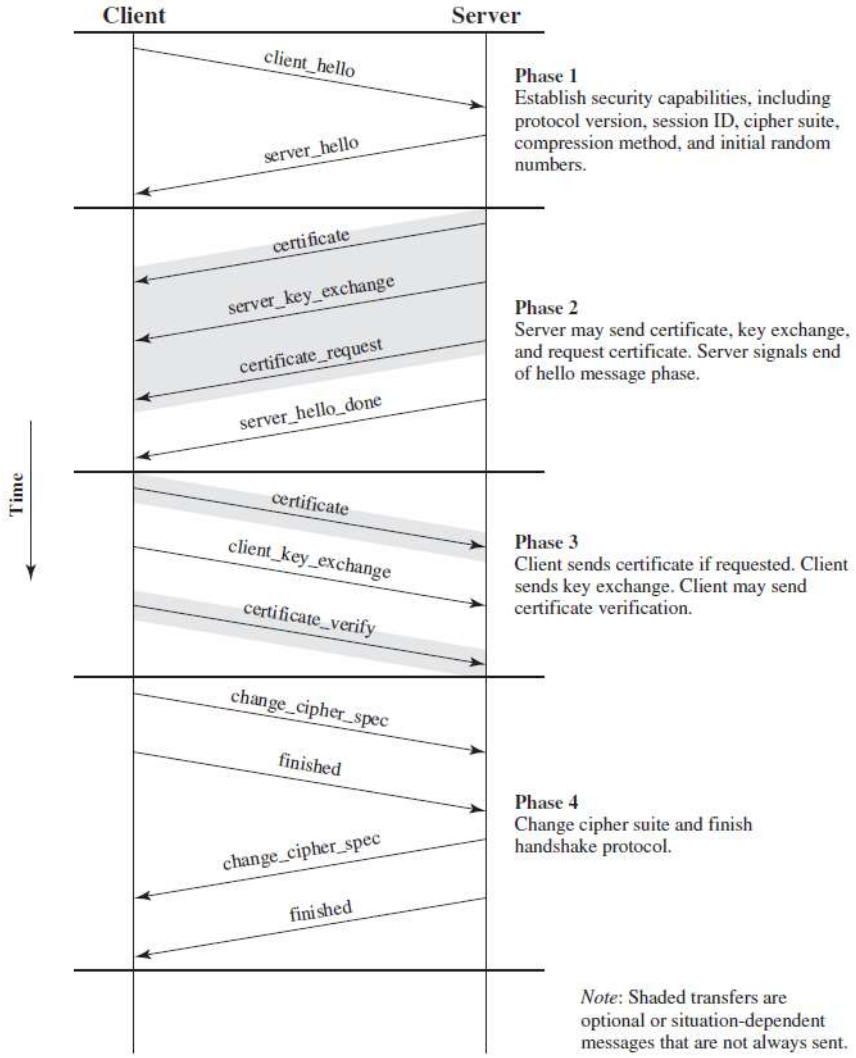
- Allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.
- Used before any application data is transmitted



(c) Handshake Protocol

Message Type	Parameters
`hello_request`	null
`client_hello`	version, random, session id, cipher suite, compression method
`server_hello`	version, random, session id, cipher suite, compression method
`certificate`	chain of X.509v3 certificates
`server_key_exchange`	parameters, signature
`certificate_request`	type, authorities
`server_done`	null
`certificate_verify`	signature
`client_key_exchange`	parameters, signature
`finished`	hash value

SSL Handshake Protocol Action



Cipher suite:
CiperSpec (Cipher algorithm, MAC algorithm, Cipher type, hash size, IV size, key material, isExportalbe)
Key exchange method (RSA; Fixed, Ephemeral, Anonymous Diffie-Hellman; Fortezza);.

Cryptographic Computations

Stockholm University

- Two further items are of interest:
  - Different kinds of keys ->generated from the master secret
    - CipherSpecs require a 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
    - These parameters are generated from the master secret by hashing the master secret into a sequence of secure bytes of sufficient length for all needed parameters
  - The creation of a shared master secret by means of the key exchange
    - The shared master secret is a one-time 48-byte value generated for this session by means of secure key exchange

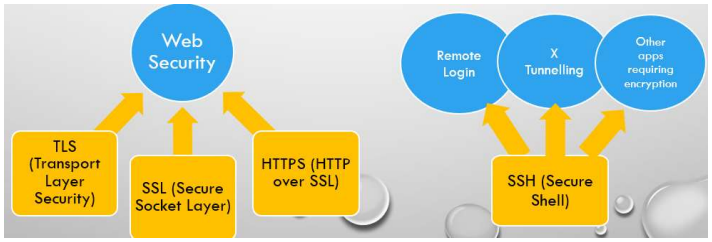
Stockholm University

Transport Layer Security (TLS)
HTTPS (HTTP over SSL)

Transport Layer Security (TLS)

Stockholm University

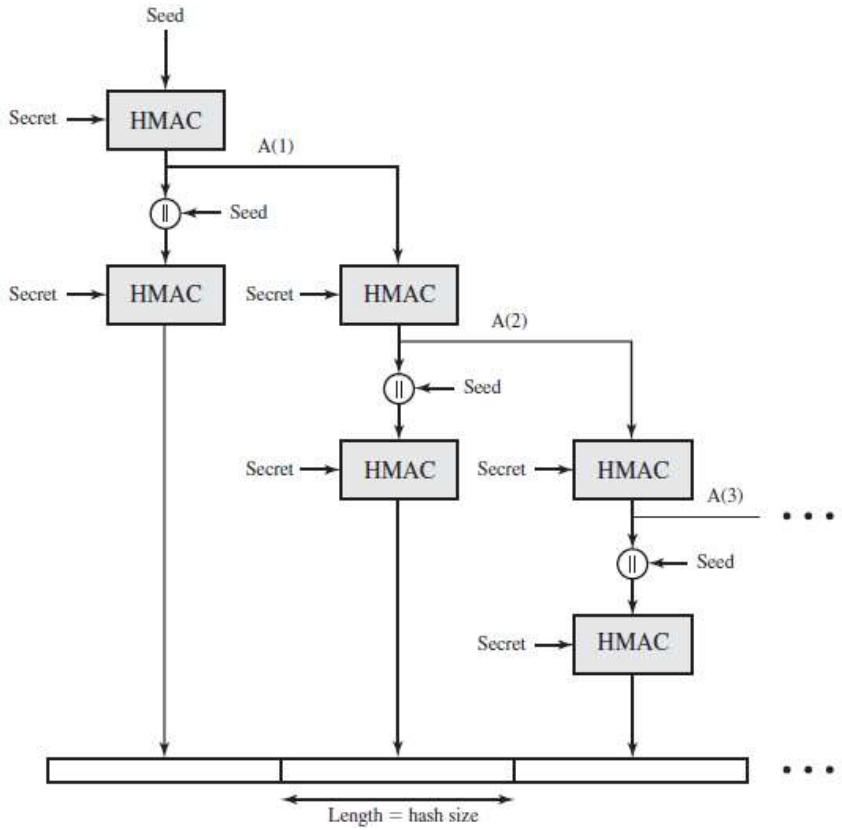
- An IETF standardization initiative whose goal is to produce an Internet standard version of SSL
- Is defined as a Proposed Internet Standard in RFC 5246
  - RFC 5246 is very similar to SSLv3



Differences include:

- Version number
- Message Authentication Code
- Pseudorandom function
- Alert codes
- Cipher suites
- Client certificate types
- Certificate_verify and Finished Messages
- Cryptographic computations
- Padding

TLS Function P_hash (secret, seed)



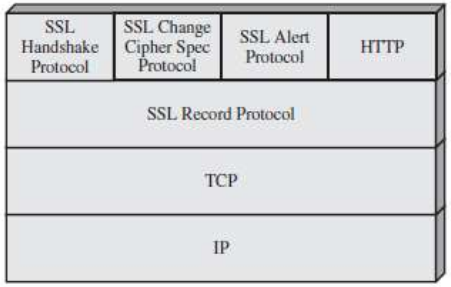
- PRF(Pseudorandom function) to expand keys into blocks of data for key generation or validation
- P_hash: data expansion function
  - Input: secret, seed
  - Output: >=required key length
  - Iteration: HMAC_hash (secret, A(x)+seed), e.g., HMAC_MD5(secret, data), HMAC_SHA256(secret, data)
  - P_hash(secret, seed) = HMAC_hash(secret, A(1) + seed) + HMAC_hash(secret, A(2) + seed) + HMAC_hash(secret, A(3) + seed) + ...

+ means concatenation

HTTPS (HTTP over SSL)

Stockholm University

- Combination of HTTP and SSL, to implement secure communication between a Web browser and a Web server
  - The HTTPS capability is built into all modern Web browsers
  - A user of a Web browser will see URL addresses that begin with **https:// rather than http://**
  - If HTTPS is specified, port 443 is used, which invokes SSL (HTTP: 80)
- Documented in RFC 2818, *HTTP Over TLS*
  - No fundamental change in using HTTP over either SSL or TLS, both implementations are referred to as HTTPS
- When HTTPS is used, the following elements of the communication are encrypted:
  - URL of the requested document
  - Contents of the document
  - Contents of browser forms
  - Cookies sent from browser to server and from server to browser
  - Contents of HTTP header



Connection Initiation

Stockholm University

The agent acting as the HTTP client also acts as the TLS client

- The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake
- When the TLS handshake has finished, the client may then initiate the first HTTP request
- All HTTP data is to be sent as TLS application data

Three levels of connections in HTTPS

- **HTTP connection (HTTP level):** an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer
  - Typically the next lowest layer is TCP, but it may also be TLS/SSL
- **TLS/SSL session/connections (TLS/SSL level):** a session is established between a TLS client and a TLS server
  - This session can support one or more connections at any time
- **TCP connection (TCP level):** A TLS request to establish a connection begins with the establishment of a TCP connection between the TCP entity on the client side and the TCP entity on the server side

Connection Closure

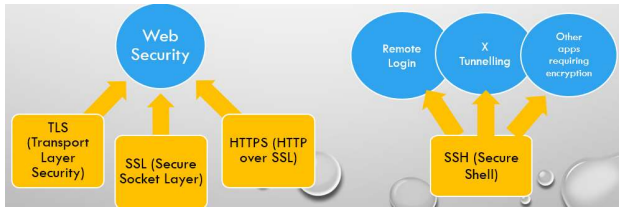
Stockholm University

- An HTTP client or server can indicate the closing of a connection by including the line **Connection: close** in an HTTP record
- The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection
- TLS implementations must initiate an exchange of closure alerts before closing a connection
  - A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an “incomplete close”
- An unannounced TCP closure could be evidence of some sort of attack so the HTTPS client should issue some sort of security warning when this occurs

Stockholm University

Secure Shell (SSH)

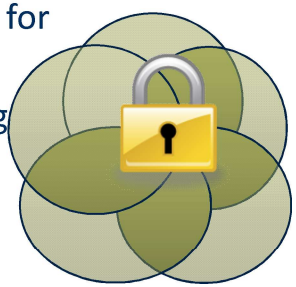
Secure Shell (SSH)



Designed to be simple and inexpensive to implement

SSH client/server applications widely available for most operating systems

- The method for remote login and X tunneling
- One of the most pervasive applications for encryption technology outside of embedded systems



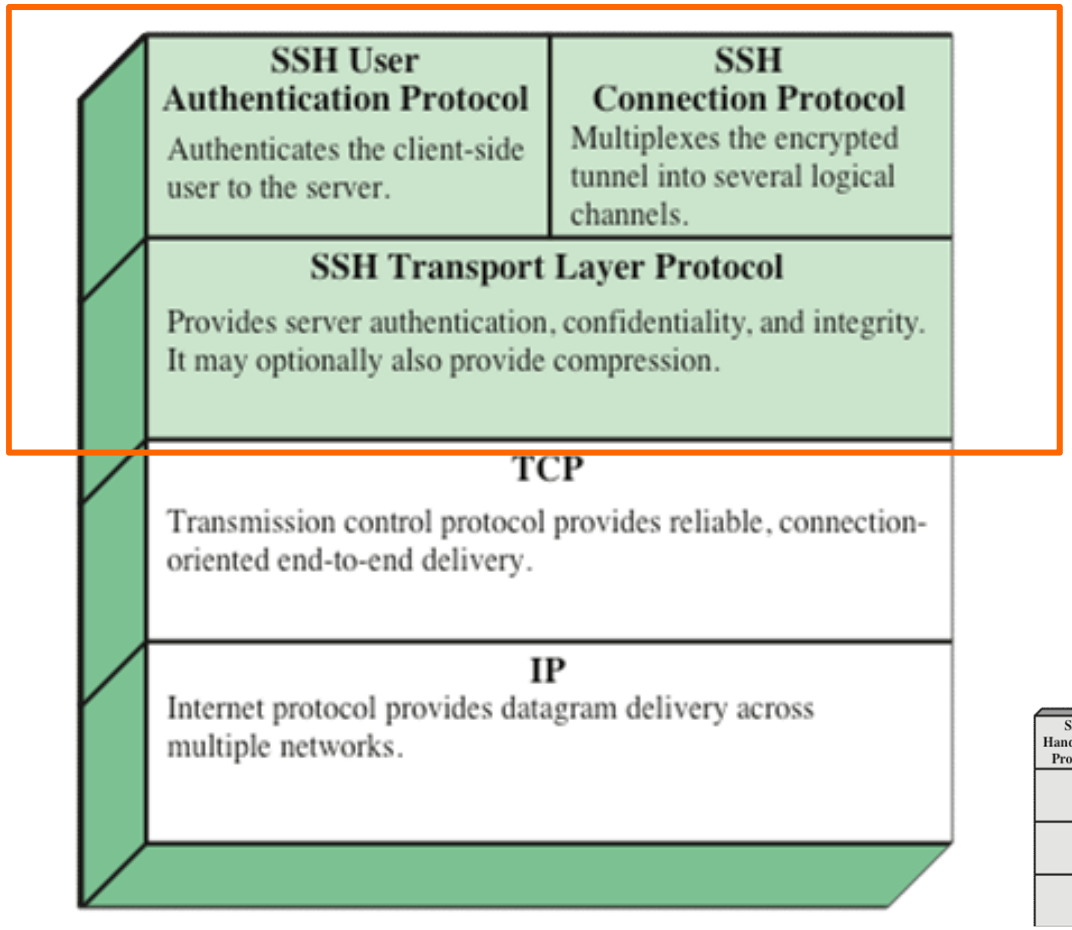
The initial version, SSH1 was focused on providing a secure remote login facility to replace TELNET and other remote login schemes that provided no security

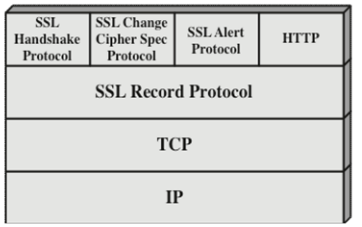
SSH2 fixes a security flaws in the original scheme

- Documented as a standard in IETF RFCs 4250 through 4256

SSH also provides a more general client/server capability and can be used for applications, e.g., file transfer and e-mail

SSH Protocol Stack





Transport Layer Protocol

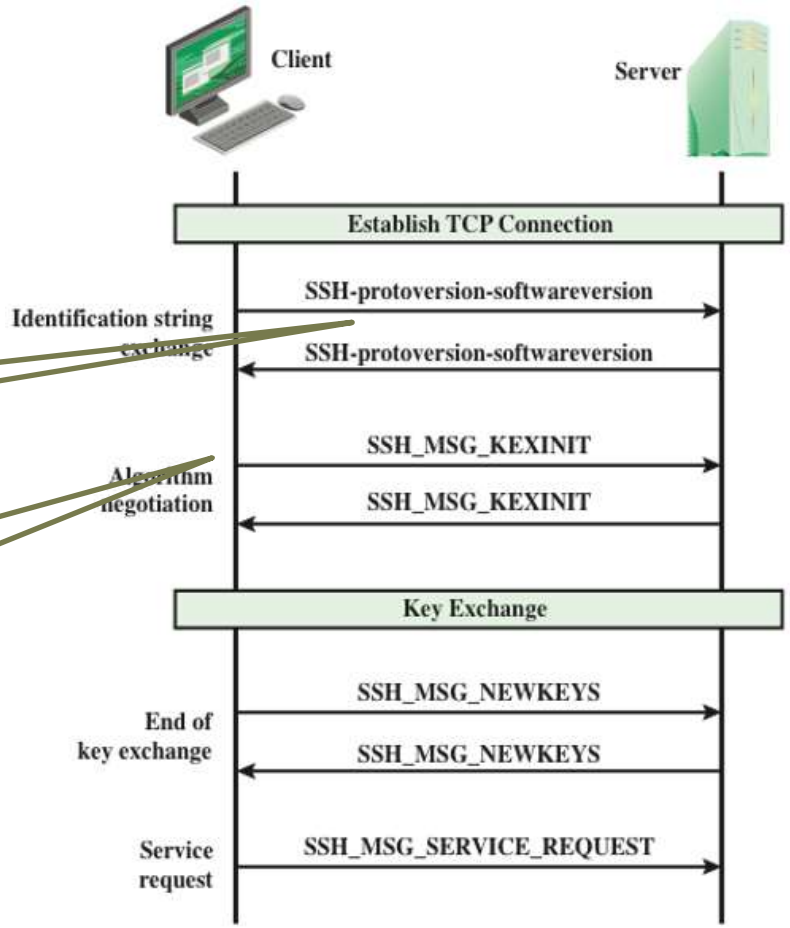
Stockholm University

- server authentication, data confidentiality & integrity
- Host keys: a public/private key pair
  - The server host key is used during key exchange to authenticate the identity
- RFC 4251 dictates two alternative trust models:
  - The client has a local database that associates each host name with the corresponding public host key
  - The host name-to-key association is certified by a trusted certification authority (CA); the client only knows the CA root key and can verify the validity of all host keys certified by accepted CAs

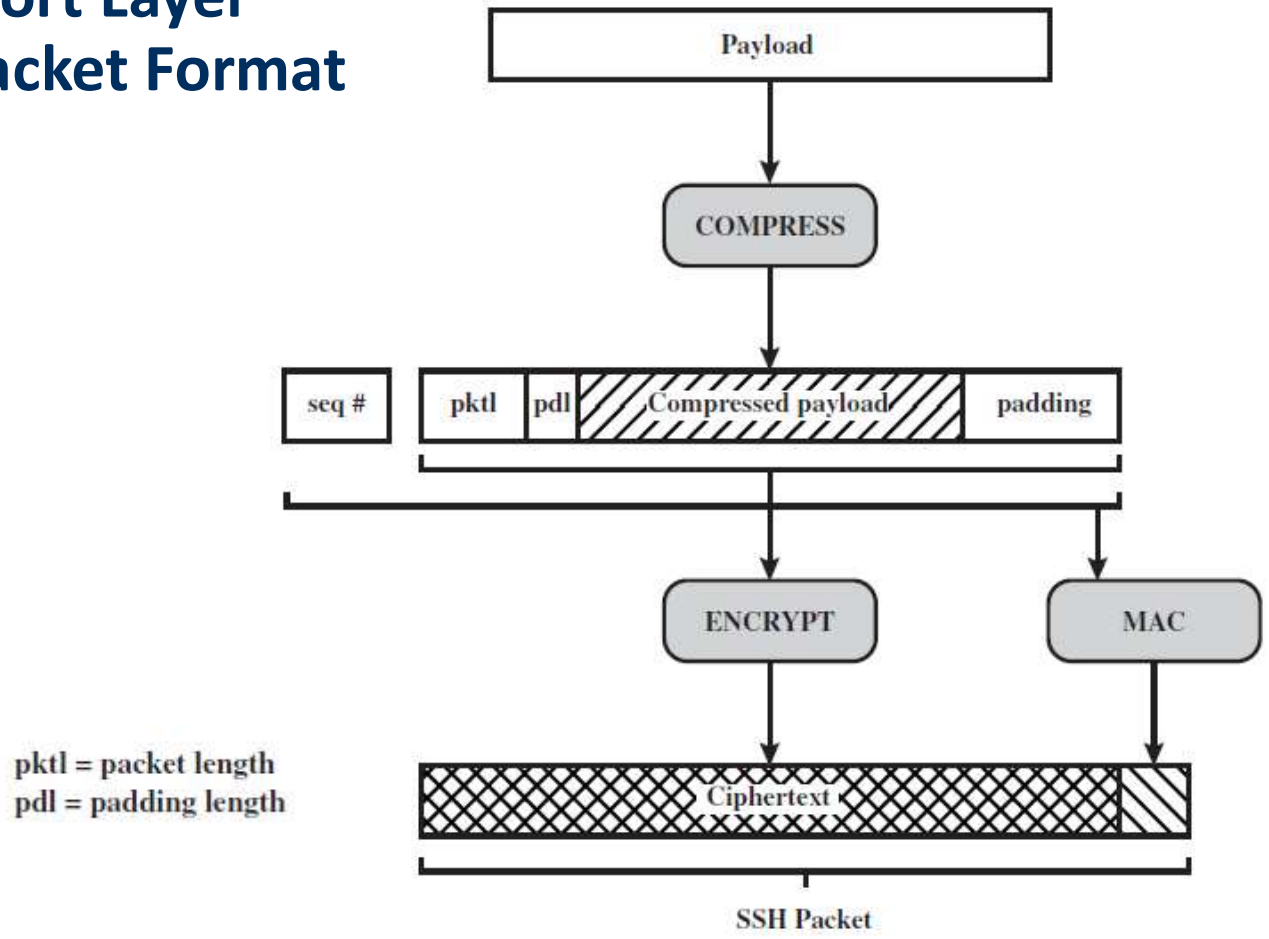
SSH Transport Layer Protocol - Packet Exchanges

SSH-2.0-billsSSH_3.6.3q3<CR><LF>

key exchange,
encryption,
MAC algorithm,
compression algorithm



SSH Transport Layer Protocol Packet Format



Cipher	
3des-cbc*	Three-key 3DES in CBC mode
blowfish-cbc	Blowfish in CBC mode
twofish256-cbc	Twofish in CBC mode with a 256-bit key
twofish192-cbc	Twofish with a 192-bit key
twofish128-cbc	Twofish with a 128-bit key
aes256-cbc	AES in CBC mode with a 256-bit key
aes192-cbc	AES with a 192-bit key
aes128-cbc**	AES with a 128-bit key
Serpent256-cbc	Serpent in CBC mode with a 256-bit key
Serpent192-cbc	Serpent with a 192-bit key
Serpent128-cbc	Serpent with a 128-bit key
arcfour	RC4 with a 128-bit key
cast128-cbc	CAST-128 in CBC mode

MAC algorithm	
hmac-sha1*	HMAC-SHA1; digest length = key length = 20
hmac-sha1-96**	First 96 bits of HMAC-SHA1; digest length = 12; key length = 20
hmac-md5	HMAC-MD5; digest length = key length = 16
hmac-md5-96	First 96 bits of HMAC-MD5; digest length = 12; key length = 16

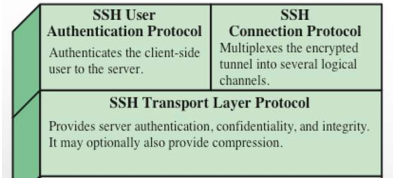
Compression algorithm	
none*	No compression
zlib	Defined in RFC 1950 and RFC 1951

* = Required
** = Recommended

Stockholm University

SSH Transport Layer Cryptographic Algorithms

Client Authentication Methods



Stockholm University

Public key

- The client sends a message to the server that contains the client's public key, with the message signed by the client's private key
- When the server receives this message, it checks whether the supplied key is acceptable for authentication and, if so, it checks whether the signature is correct

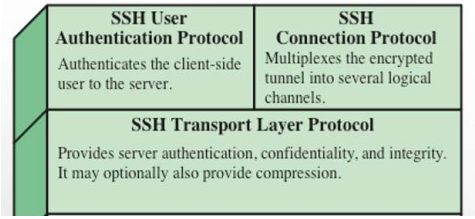
Password

- The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol

Host based

- Authentication is performed on the client's host rather than the client itself
- This method works by having the client send a signature created with the private key of the client host
- Rather than directly verifying the user's identity, the SSH server verifies the identity of the client host

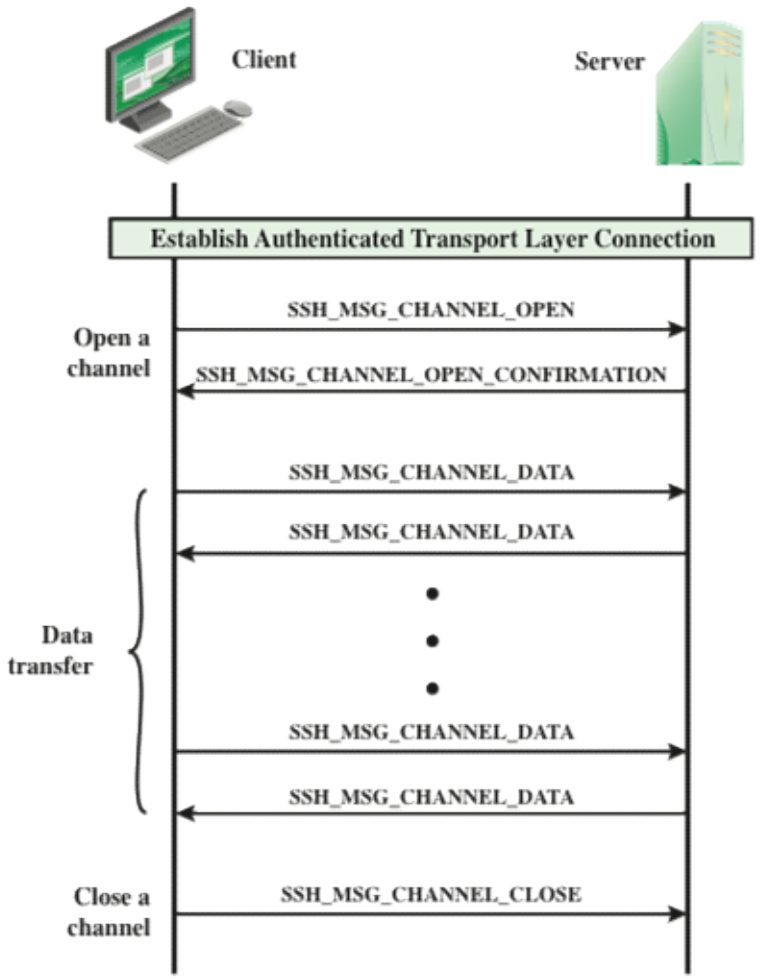
Connection Protocol



Stockholm University

- Runs on top of the SSH Transport Layer Protocol: assumes that a secure authentication connection is in use
  - The secure authentication connection, referred to as a *tunnel,* is used by the Connection Protocol to multiplex a number of logical channels
- Channel mechanism: all types of communication using SSH (e.g., a terminal session) are supported using separate channels
  - Either side may open a channel
  - For each channel, each side associates a unique channel number
  - Channels are flow controlled using a window mechanism
  - No data may be sent to a channel until a message is received to indicate that window space is available
  - The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel

Connection Protocol Message Exchange



Stockholm University

Channel Types

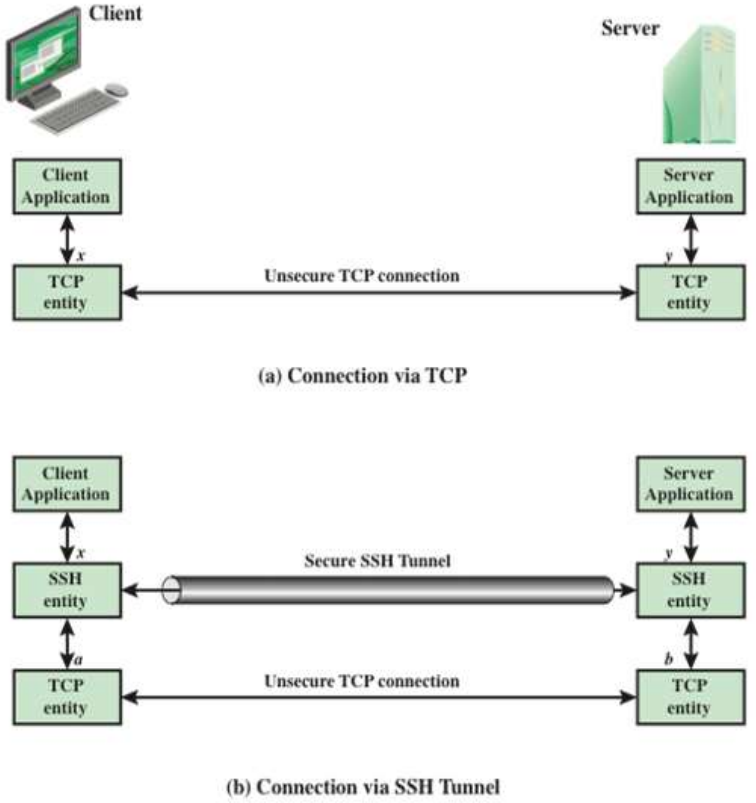
Stockholm University

Session	• The remote execution of a program • The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem • Once a session channel is opened, subsequent requests are used to start the remote program
X11	• Refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers • X allows applications to run on a network server but to be displayed on a desktop machine
Forwarded-TCPIP	• Remote port
Direct-TCPIP	• Local port forwarding

Port Forwarding

- One of the most useful features of SSH
- Provides the ability to convert any insecure TCP connection into a secure SSH connection (also referred to as SSH tunneling)

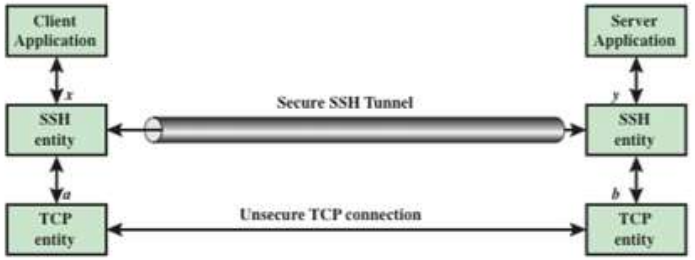
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number (a port is an identifier of a user of TCP)
- An application may employ multiple port numbers



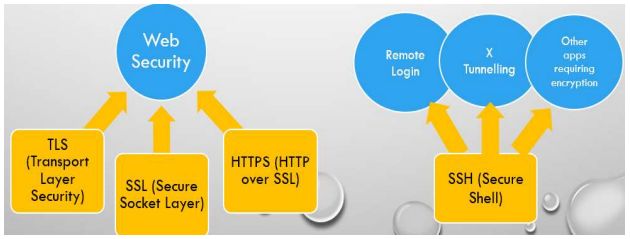
Local and Remote Port Forwarding

Stockholm University

- Local forwarding
  - The SSH entity intercepts selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.
- Remote forwarding
  - A SSH client acts on the server's behalf.



Summary



- Web security considerations
  - Web security threats
  - Web traffic security approaches
- Secure sockets layer
  - SSL architecture
  - SSL record protocol
  - Change cipher spec protocol
  - Alert protocol
  - Handshake protocol
  - Cryptographic computations
- Transport layer security
  - an “updated version” of SSL
- HTTPS
  - Connection initiation
  - Connection closure
- Secure shell (SSH)
  - Transport layer protocol
  - User authentication protocol
  - Communication protocol

Expected Learning Outcomes

Stockholm University

- Understand and explain the principles of transport level security.
- Understand and explain how SSL/TLS, HTTPS, SSH work.
- Understand and describe each protocol in the protocol stack of SSL.
- Understand and describe each protocol in the protocol stack of SSH.
- Understand and explain the parameters defining a session state and a connection state in SSL.

Thank you!