

# Transport Layer Security

Lennart Franked

Department of Information and Communication Systems (ICS),  
Mid Sweden University, Sundsvall.

2014-10-27

# Overview

- 1 Web Security
  - Web Security Considerations
- 2 SSL/TLS
  - SSL – Secure Socket Layer
  - TLS – Transport Layer Security
- 3 Applications
  - HTTPS
  - SSH
- 4 References

# Literature

The lecture covers chapter 6 “Transport-level Security” in [1]. To make sure you have fully understood the chapter, you should solve problems 6.1, 6.2 and 6.3 in [1].

# Web Security Considerations

- The web is a client/server system.
- Easy to use, easy to develop and easy to setup.
- Complex software.
- Vulnerable to exploits.
- Easy to exploit users.

# Threats on the web

## Web Security Considerations

	Threats	Consequences	Countermeasures
<b>Integrity</b>	<ul style="list-style-type: none"><li>•Modification of user data</li><li>•Trojan horse browser</li><li>•Modification of memory</li><li>•Modification of message traffic in transit</li></ul>	<ul style="list-style-type: none"><li>•Loss of information</li><li>•Compromise of machine</li><li>•<u>Vulnerability</u> to all other threats</li></ul>	Cryptographic checksums
<b>Confidentiality</b>	<ul style="list-style-type: none"><li>•Eavesdropping on the net</li><li>•Theft of info from server</li><li>•Theft of data from client</li><li>•Info about network configuration</li><li>•Info about which client talks to server</li></ul>	<ul style="list-style-type: none"><li>•Loss of information</li><li>•Loss of privacy</li></ul>	Encryption, Web proxies
<b>Denial of Service</b>	<ul style="list-style-type: none"><li>•Killing of user threads</li><li>•Flooding machine with bogus requests</li><li>•Filling up disk or memory</li><li>•Isolating machine by DNS attacks</li></ul>	<ul style="list-style-type: none"><li>•Disruptive</li><li>•Annoying</li><li>•Prevent user from getting work done</li></ul>	Difficult to prevent
<b>Authentication</b>	<ul style="list-style-type: none"><li>•Impersonation of legitimate users</li><li>•Data forgery</li></ul>	<ul style="list-style-type: none"><li>•Misrepresentation of user</li><li>•Belief that false information is valid</li></ul>	Cryptographic techniques

Figure: A Comparison of Threats on the Web[1]

# Security facilities in the TCP/IP protocol stack

## Web Security Considerations

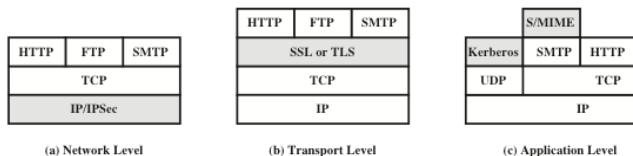


Figure: Relative location of security facilities in the TCP/IP protocol stack [1]

# Secure Socket Layer

## SSL – Secure Socket Layer

- A widely used security service
- General purpose service implemented as a set of protocols relying on TCP.
- Part of underlying Protocol Suit
  - ▶ Transparent to applications.
- Embedded in specific packages, such as web browsers.

# SSLv3

## SSL – Secure Socket Layer

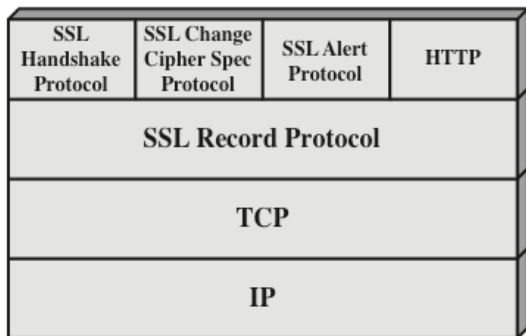


Figure: SSL protocol stack[1]



# SSL Session

## SSL – Secure Socket Layer

### Session

- An association between a client and a server.
- Created by the handshake protocol.
- Defines a set of cryptographic security parameters.
- Shared amongst several connections.
- Avoids expensive negotiations.

# SSL Connection

SSL – Secure Socket Layer

## Connection

- A transport that provides a suitable type of service.
- Transient.
- Every connection is associated with one session.

# SSL Record Protocol

SSL – Secure Socket Layer

## Services

- Confidentiality
- Message Integrity

# SSL Record Protocol II

## SSL – Secure Socket Layer

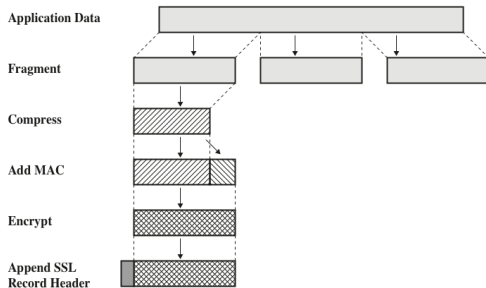


Figure: SSL Record protocol[1]

# SSLv3 MAC

## SSL – Secure Socket Layer

```
hash(MAC_write_secret || pad_2 ||  
hash(MAC_write_secret || pad1 || seq_num ||  
SSL_Compressed.type || SSLCompressed.length ||  
SSLCompressed.fragment))
```

# Supported algorithms

## SSL – Secure Socket Layer

Block Cipher		Stream Cipher	
Algorithm	Key size	Algorithm	Key Size
AES	128,256	RC-40	40
IDEA	128	RC-128	128
RC2-40	40		
DES-40	40		
DES	56		
3DES	168		
Fortezza	80		

# Change Cipher Spec Protocol

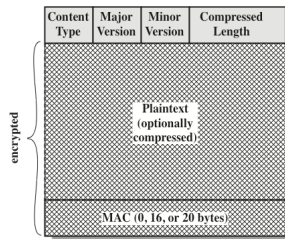
## SSL – Secure Socket Layer

- Used for changing the pending state to current state.
- Contains a single byte with the value 1.

# Alert Protocol

## SSL – Secure Socket Layer

- Used to send alerts to the peer.
- Message is encrypted and optionally compressed.
- Two bytes – Severity and Alert

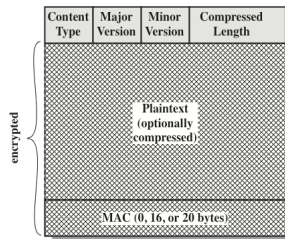




# Alert Protocol II

## SSL – Secure Socket Layer

- Severity
  - ▶ Warning – Session is still running.
  - ▶ Fatal – Session is closed.
- Alert
  - ▶ Type of alert



# SSL Handshake protocol

## SSL – Secure Socket Layer

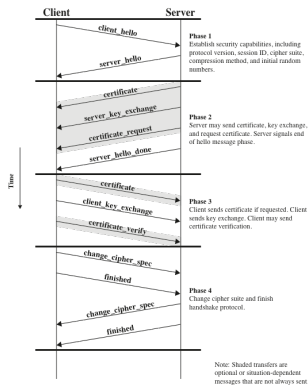


Figure: Handshake Protocol Action[1]

# Generating Cryptographic Parameters

## SSL – Secure Socket Layer

```
master_secret = MD5(pre_master_secret || SHA('A' ||
pre_master_secret || ClientHello.random ||
ServerHello.random)) ||
MD5(pre_master_secret || SHA('BB' ||
pre_master_secret || ClientHello.random ||
ServerHello.random)) ||
MD5(pre_master_secret || SHA('CCC' ||
pre_master_secret || ClientHello.random ||
ServerHello.random))
```

# Generating key block

## SSL – Secure Socket Layer

```
Key_block =  
MD5(master_secret || SHA('A' || master_secret ||  
ServerHello.random || ClientHello.random)) ||  
MD5(master_secret || SHA('BB' || master_secret ||  
ServerHello.random || ClientHello.random)) ||  
MD5(master_secret || SHA('CCC' || master_secret ||  
ServerHello.random || ClientHello.random))
```

# Transport Layer Security

## TLS – Transport Layer Security

- An IETF standardization of SSL.
- Defined in RFC 5246[2]

# TLS MAC

## TLS – Transport Layer Security

```
hash(MAC_write_secret  $\oplus$  pad2 ||  
hash((MAC_write_secret  $\oplus$  pad1) || seq_num ||  
TLSCompressed.type || TLSCompressed.version ||  
TLSCompressed.length || TLSCompressed.fragment)  
)
```

# TLS Key generation

## TLS – Transport Layer Security

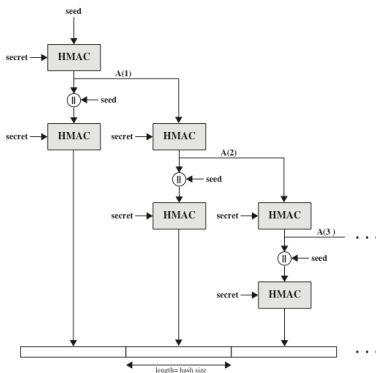


Figure: TLS key generation[1]

# HTTP over SSL/TLS

## HTTPS

- HTTPS uses either SSL or TLS
- Following elements are encrypted.
  - ▶ URL of the requested document
  - ▶ Contents of the document
  - ▶ Contents of browser forms
  - ▶ Cookies
  - ▶ Content of HTTP header



# SSH

- A protocol used for secure remote communications.
- Designed to be simple and inexpensive.
- Initially meant to replace TELNET and other remote login schemes.
- Provides file transfers, tunneling et cetera.
- Server / Client
- Consists of three protocols

# Protocol Stack

## SSH

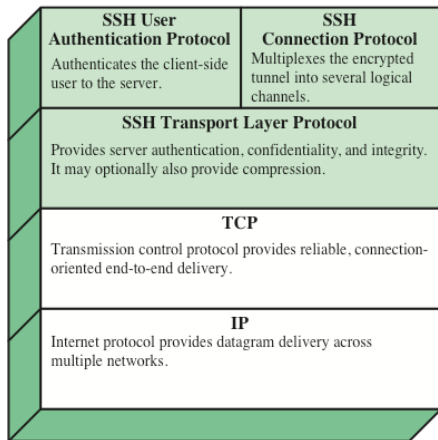


Figure: SSH Protocol Stack [1]

# Transport Layer Protocol

## SSH

- Directly above transport layer.
- Based on the server possessing a public/private key pair.
- Can have multiple host keys using different asymmetric encryption schemes.
- Multiple hosts may share the same key.
- The server host key is used during key exchange to authenticate the host.
- Client must possess the servers public key before a secure connection can be set up.

# Servers public key trust level

SSH

## Local database

- Client stores the public key in a local database.
- Associates the public key based on host name.
- Problems?

# Servers public key trust level II

SSH

## Using a CA

- Use a trusted CA.
- host name to key association.
- Only need to store and trust the CA public key.
- Problems?

# Packet Exchange

## SSH

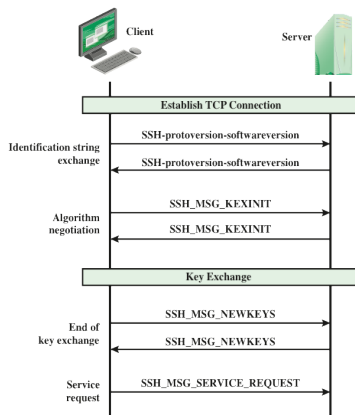


Figure: SSH Transport Layer Packet Exchange[1]

# User Authentication Protocol

## SSH

- Used to authenticate the client to the server.
- Three authentication methods
  - ▶ Public key
  - ▶ Password
  - ▶ Hostbased

# Connection Protocol

## SSH

- Runs on top of the SSH Transport Layer Protocol.
- Assumes that a secure authentication connection (tunnel) is in use.
- Allows multiplexing a number of logical channels.



### Channel mechanism

- Communication using SSH are supported using separate channels.
- Server or Client can open a channel.
- For each channel, each side associates a unique channel number.
- Channels are flow controlled using a window mechanism
- A channel progresses through the steps: Opening, data transfer, closing.

# SSH Connection Protocol

## SSH

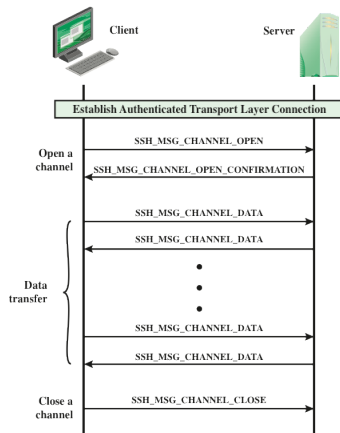


Figure: SSH Connection Protocol[1]

# Channel Types

## SSH

### Four channel types used in SSH Connection Protocols

- Session – Remote execution of a program, such as shell, file transfer, et cetera.
- X11 – X window system
- forwarded-tcpip – Remote port forwarding
- direct-tcpip – Local port forwarding

# Port Forwarding

## SSH

Converts any insecure TCP connection into a secure SSH connection.

### Local forwarding

- Allows the client to redirect traffic through an SSH connection.
- Listen on a local port and forwards all traffic through an SSH tunnel

### Remote forwarding

Client acts on the servers behalf

### Dynamic forwarding

SSH acts as a SOCKS-proxy.

# References

- [1] William Stallings. *Network security essentials : applications and standards*. 5th ed. International Edition. Pearson Education, 2013. ISBN: 978-0-273-79336-6.
- [2] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). Updated by RFCs 5746, 5878, 6176. Internet Engineering Task Force, Aug. 2008. URL: <http://www.ietf.org/rfc/rfc5246.txt>.