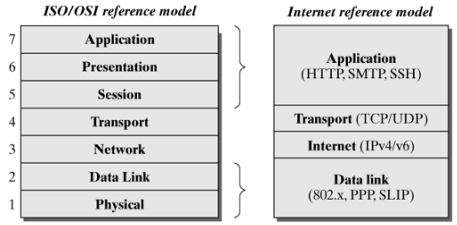
# Cybersecurity: Secure Sockets Layer

*Ozalp Babaoglu*

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

## Internet layered architecture

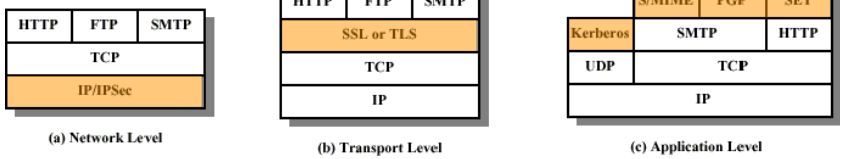
- Internet protocols are layered
- Each layer provides services to the layer above hiding details of layers below
  - Logical separation
  - Easier to develop and maintain
  - Interoperability
- Two reference models: ISO/OSI and TCP/IP






## Introduction

- Security in the Internet:
  - at which level?






## Introduction

- Security at the application level
  - Pros: designed for specific application requirements
  - Cons: requires multiple security mechanisms
- Security at the transport level
  - Pros: provides common interface to security services
  - Cons: requires (minor) modification to applications
- Security at the network level
  - Pros: works with security-ignorant applications
  - Cons: may require modifications at the OS level

## Introduction

- Security at the application level
  - S/MIME
  - PGP
  - Kerberos
  - SET — Secure Electronic Transfer
- Security at the transport level
  - SSL (Secure Sockets Layer)
- Security at the network level
  - IPSec



## SSL: Secure Sockets Layer

- Probably the most widely-used security service on the Internet
- A general purpose service implemented as a set of protocols that rely on TCP
- Proposed by *Netscape Communications Corporation* in 1994 as part of their *Navigator* browser
- Adopted as a standard by the IETF under the name *Transport Layer Security* (TLS)
- Guarantees *confidentiality*, *integrity* and *authentication* for Internet communications

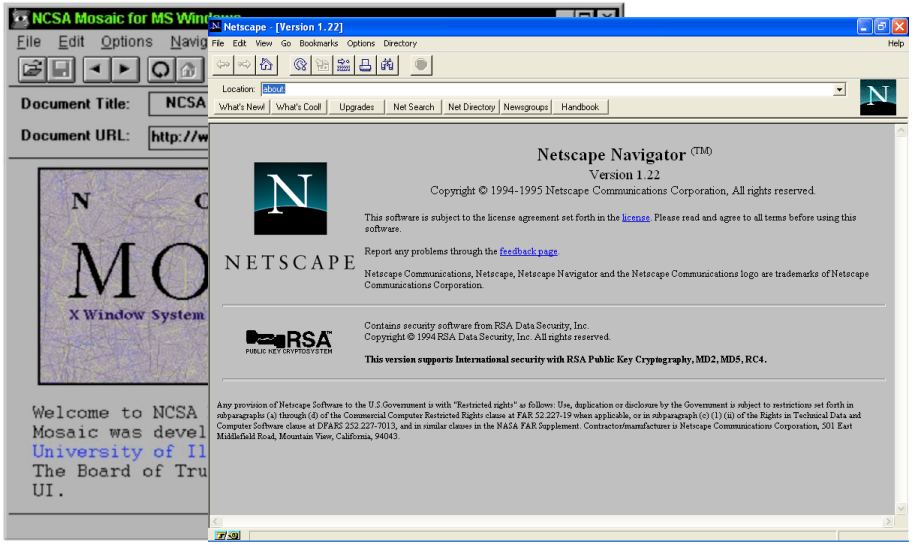


## Internet Archaeology

- Pre 1993 Internet was essentially text only: Archie, Gopher, WAIS
- 1993 — *National Center for Supercomputing Applications* at the *University of Illinois at Urbana–Champaign* releases ***Mosaic***, the first graphic web browser
- 1994 — *Mosaic Communications Corporation* founded
- October 1994 — ***Mosaic Netscape 0.9*** released
- November 1994 — Company renamed *Netscape Communications Corporation* and its product renamed ***Netscape Navigator***
- Netscape dominated the browser market until around 2000 until it lost to ***Microsoft Internet Explorer***



## Internet Archaeology

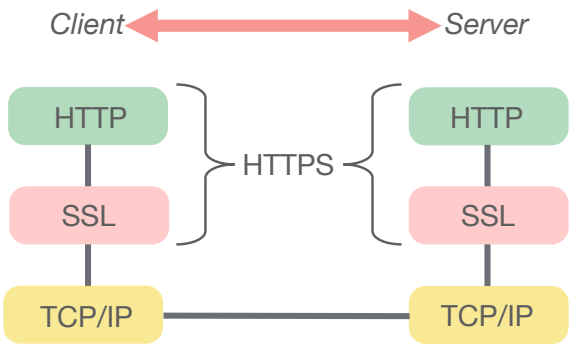


## SSL: Secure Sockets Layer

- E-commerce
- On-line trading
- Internet banking
- Any time confidential data (password, credit card number) needs to be sent to a remote host



## SSL: Secure Sockets Layer






## SSL: Secure Sockets Layer

- Based on
  - *Symmetric* ciphers
  - *Asymmetric* ciphers
  - Certificates
  - Message Authentication Code (MAC)

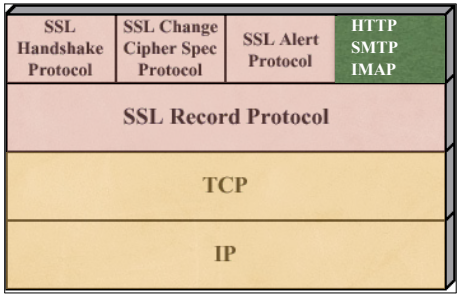


## Hybrid solution for secret key management

1. $A$ generates $(K_A[pub], K_A[priv])$
2. $A$ announces its public key to $B$: $\{K_A[pub], A\}$
3. $B$ generates session key $K_S$
4. $B$ sends session key to $A$: $C(K_A[pub], K_S)$
5. $A$ decrypts to obtain $K_S = D(K_A[priv], C(K_A[pub], K_S))$
6. $A$ can delete $(K_A[pub], K_A[priv])$
7. $A$ and $B$ switch to symmetric cryptography using the session key $K_S$

# SSL: Implementation

- SSL ***Handshake protocol*** creates a channel that is secure, reliable and authenticated between client and server
- SSL ***Record protocol*** transports messages in encapsulated blocks that are encrypted and authenticated



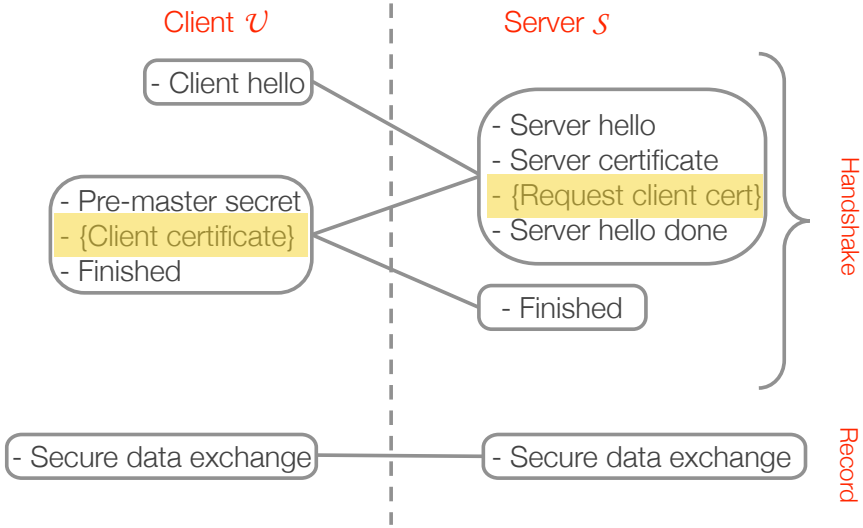



# SSL: Handshake and Record

- ***Handshake***: uses *public-key cryptography* to establish a secure channel between client and server such that
  - there is mutual *authentication*
  - client and server agree on *encryption/decryption algorithms*
  - client and server agree on a *secret key*
- ***Record***: uses *private-key cryptography* with the agreed upon algorithms and secret key to confidentially exchange data

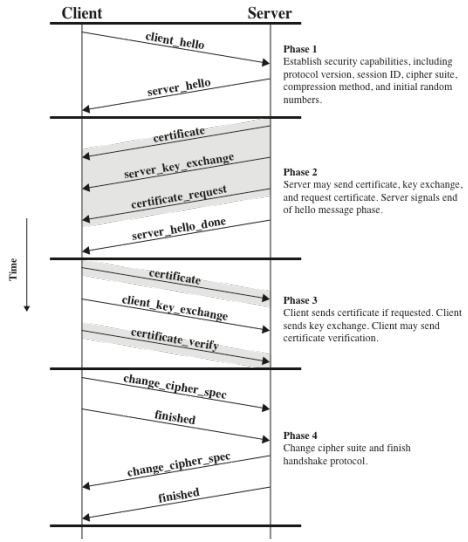


# SSL: Handshake e Record






# SSL: Handshake Protocol



## SSL – Sessions and Connections

- SSL Session
  - A long-lasting association between a client and a server
  - Created by the *Handshake Protocol*
  - Associated to a set of security parameters
  - Used to avoid the expensive negotiation of new security parameters
- SSL Connection
  - A transport connection between a client and a server
  - Connections are transient
  - Every connection is associated with one session
- Between any pair of parties
  - There may be multiple connections
  - Normally there is a single session



## SSL – Sessions and Connections

- Session state
  - ***Session identifier***: arbitrary byte sequence to identify an active session
  - ***Peer certificate***: an X509.v3 certificate of the peer; may be null
  - ***Compression method***: used to compress data prior to encryption
  - ***Cipher spec***: specifies the data encryption algorithm
  - ***Master secret***: 48 byte secret shared between client and server



## SSL – Sessions and Connections

- Connection State
  - ***Client/Server random***: Random byte sequences used as identifier chosen by the client and the server at each connection
  - ***Client/Server write MAC secret key***: Secret key used in *Message Authentication Code* (MAC) operations on data sent by the client/server
  - ***Client/Server write secret key***: Encryption key for data encrypted by the client/server and decrypted by the server/client
  - ***Sequence Numbers***



## SSL Authentication

- Authentication of the server to the client through a certificate is mandatory
- Authentication of the client to the server is optional
- If requested by the server, the client usually authenticates itself through a mechanism that does *not* require certificates such as login/password
- Because certificates for SSL are usually expensive and clients should not be expected to incur their cost for accessing a secure server

# SSL Authentication

- TLS/SSL Certificate prices from Thawte

| | SSL Web Server with EV | SSL Web Server | SSL123 |
|---|---|---|---|
| Issuance Time | Most certificates issued in **1-3 days** | Most certificates issued in **one day** | Most certificates issued in **minutes** |
| Price: (•) 1 Year ( ) 2 Years | Best for: Credit Card Transacting Websites Banks and Financial Institutions<br>$344<br>BUY NOW RENEW | Best for: Enterprise Applications Business Websites<br>$218<br>[ ] add wildcard + $470<br>BUY NOW RENEW | Best for: Securing Internal Servers Private Websites<br>$149<br>[ ] add wildcard + $596<br>BUY NOW RENEW |
| Identity validation and customer assurance | Prominent visible assurance to increase trust and boost customer confidence | Visible assurance to customers that your website and domain are tied to your organization. | SSL encryption with padlock icon |



# SSL – Record Protocol

- SSL Record Protocol provides
  - ***Confidentiality***: The *Handshake Protocol* defines a shared secret key that is used to encrypt SSL payloads
  - ***Integrity***: The *Handshake Protocol* defines a shared secret key that is used to generate *Message Authentication Codes* attached to payloads

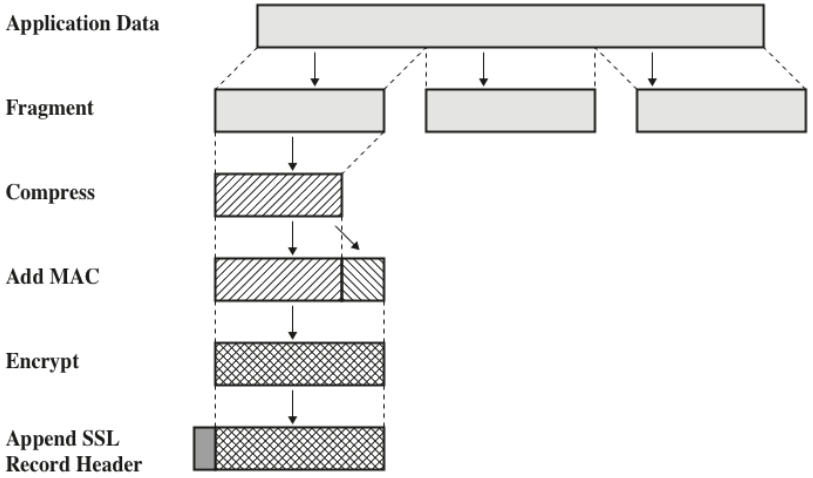


# SSL – Record Protocol

- The original message is fragmented into $2^{14}$ byte blocks
- Each fragment is numbered, (optionally) compressed, extended with MAC, encrypted with the master secret key and transmitted using TCP
- The receiver reverses the operations and reconstructs the original message which is passed to the upper application layer (HTTP, SMTP, IMAP, etc.)



# SSL – Record Protocol



# MAC in SSL Record Protocol

- Each fragment is numbered and extended with MAC
- MAC is computed as hash (MD5 or SHA-1) of the block
  (*fragment* | *seq_no* | master secret | *padding*)
- Where *seq_no* is 64 bits and therefore will not repeat within a single session
- Sequence numbers allow replay attacks within a single session to be detected
- They also serve to detect lost blocks (that need to be regenerated and resent)
- MAC are encrypted together with the data using symmetric cryptography with the master secret



# Importance of random bytes

- The “client hello”, “server hello” and pre-master secret messages of the handshake protocol contain sequences of *random bytes*
- The secrecy of the session key, and thus the security of the communication channel created by SSL, depends heavily on the randomness (unpredictability) of these bytes
- Thus, it is crucial that the SSL implementation be based on high-quality pseudo-random generators



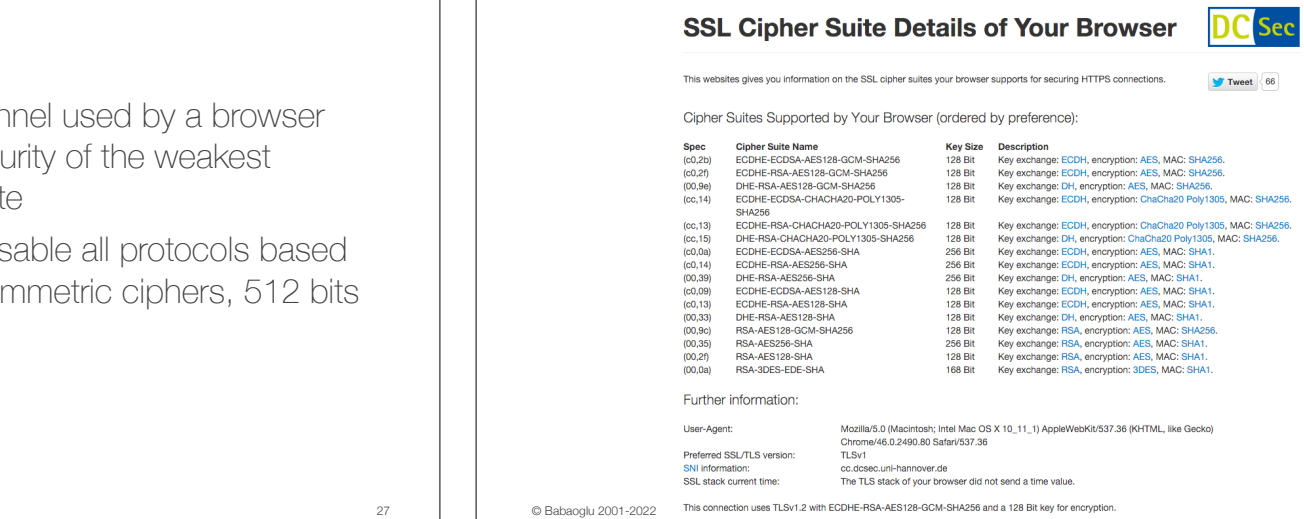
# SSL and browsers

- The overall security of the SSL channel used by a browser cannot be any greater than the security of the weakest protocol in the browser’s cipher suite
- In your browser, it is advisable to disable all protocols based on short keys (64 bits or less for symmetric ciphers, 512 bits or less for asymmetric ciphers)



# SSL and browsers

- https://cc.dcsec.uni-hannover.de/



**SSL Cipher Suite Details of Your Browser** DC Sec

This websites gives you information on the SSL cipher suites your browser supports for securing HTTPS connections. Tweet 66

Cipher Suites Supported by Your Browser (ordered by preference):

| Spec | Cipher Suite Name | Key Size | Description |
|---|---|---|---|
| (c0,2b) | ECDHE-ECDSA-AES128-GCM-SHA256 | 128 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA256. |
| (c0,2f) | ECDHE-RSA-AES128-GCM-SHA256 | 128 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA256. |
| (00,9e) | DHE-RSA-AES128-GCM-SHA256 | 128 Bit | Key exchange: DH, encryption: AES, MAC: SHA256. |
| (cc,14) | ECDHE-ECDSA-CHACHA20-POLY1305-SHA256 | 128 Bit | Key exchange: ECDH, encryption: ChaCha20 Poly1305, MAC: SHA256. |
| (cc,13) | ECDHE-RSA-CHACHA20-POLY1305-SHA256 | 128 Bit | Key exchange: ECDH, encryption: ChaCha20 Poly1305, MAC: SHA256. |
| (cc,15) | DHE-RSA-CHACHA20-POLY1305-SHA256 | 128 Bit | Key exchange: DH, encryption: ChaCha20 Poly1305, MAC: SHA256. |
| (c0,0a) | ECDHE-ECDSA-AES256-SHA | 256 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA1. |
| (c0,14) | ECDHE-RSA-AES256-SHA | 256 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA1. |
| (00,39) | DHE-RSA-AES256-SHA | 256 Bit | Key exchange: DH, encryption: AES, MAC: SHA1. |
| (c0,09) | ECDHE-ECDSA-AES128-SHA | 128 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA1. |
| (c0,13) | ECDHE-RSA-AES128-SHA | 128 Bit | Key exchange: ECDH, encryption: AES, MAC: SHA1. |
| (00,33) | DHE-RSA-AES128-SHA | 128 Bit | Key exchange: DH, encryption: AES, MAC: SHA1. |
| (00,9c) | RSA-AES128-GCM-SHA256 | 128 Bit | Key exchange: RSA, encryption: AES, MAC: SHA256. |
| (00,35) | RSA-AES256-SHA | 256 Bit | Key exchange: RSA, encryption: AES, MAC: SHA1. |
| (00,2f) | RSA-AES128-SHA | 128 Bit | Key exchange: RSA, encryption: AES, MAC: SHA1. |
| (00,0a) | RSA-3DES-EDE-SHA | 168 Bit | Key exchange: RSA, encryption: 3DES, MAC: SHA1. |

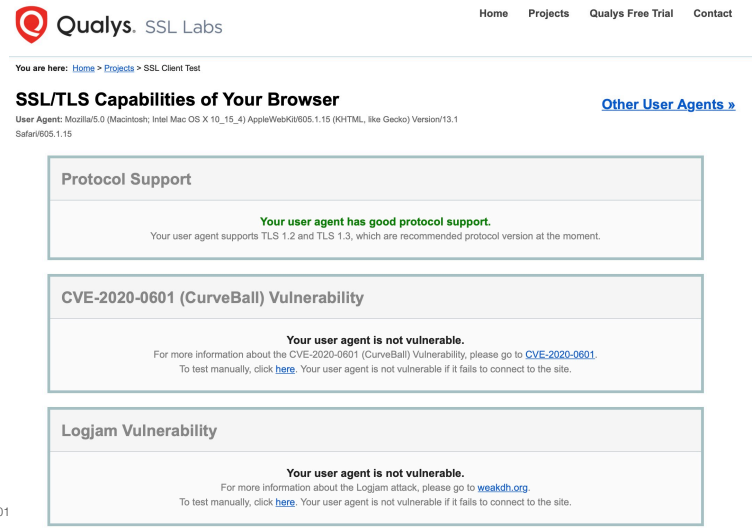
Further information:

| | |
|---|---|
| User-Agent: | Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.80 Safari/537.36 |
| Preferred SSL/TLS version: | TLSv1 |
| SNI information: | cc.dcsec.uni-hannover.de |
| SSL stack current time: | The TLS stack of your browser did not send a time value. |

This connection uses TLSv1.2 with ECDHE-RSA-AES128-GCM-SHA256 and a 128 Bit key for encryption.

# SSL and browsers

- https://www.ssllabs.com/ssltest/viewMyClient.html






# SSL Certificate Errors

- https://badssl.com