
 WILLIAM & MARY


CSCI 454/554 Computer and Network Security

Topic 8.3 SSL/TLS

 **Outline** WILLIAM & MARY

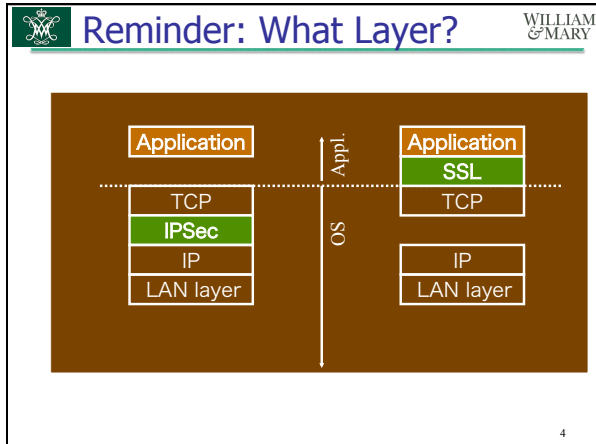
1. Overview
2. The SSL Record Protocol
3. The SSL Handshake and Other Protocols

2

 WILLIAM & MARY

Overview of SSL

3



- Protocols** WILLIAM & MARY
- Goal: application independent security
 - Originally for HTTP, but now used for many applications
 - Each application has an assigned TCP port, e.g., https (HTTP over SSL) uses port 443
 - Secure Sockets Layer (SSL)
 - the de facto standard for web-based security
 - v3 was developed with public review
 - Transport Layer Security (TLS)
 - TLS v1.0 very close to SSL v3.1
- 5

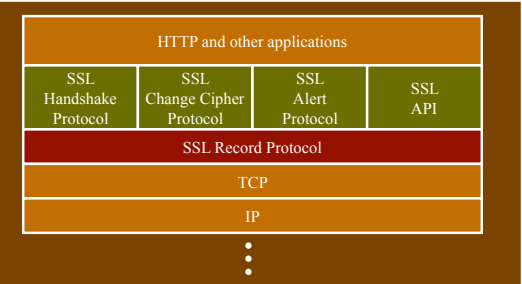
Protocol over SSL/TLS WILLIAM & MARY

Keyword	Decimal	Description
nsiiops	261/tcp	IIOP Name Service over TLS/SSL
https	443/tcp	http protocol over TLS/SSL
ddm-ssl	448/tcp	DDM-SSL
smtps	465/tcp	smtp protocol over TLS/SSL
nntp	563/tcp	nntp protocol over TLS/SSL
ssh	614/tcp	SSH
ldaps	636/tcp	ldap protocol over TLS/SSL
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
ftps	990/tcp	ftp, control, over TLS/SSL
telnet	992/tcp	telnet protocol over TLS/SSL
imaps	993/tcp	imap4 protocol over TLS/SSL
ircs	994/tcp	irc protocol over TLS/SSL
pop3s	995/tcp	pop3 protocol over TLS/SSL

6

WILLIAM & MARY

SSL Architecture



The diagram illustrates the SSL architecture stack. At the top is 'HTTP and other applications'. Below this are four protocols: 'SSL Handshake Protocol', 'SSL Change Cipher Protocol', 'SSL Alert Protocol', and 'SSL API'. These four protocols are grouped under the 'SSL Record Protocol'. Below the SSL Record Protocol is 'TCP', and at the bottom is 'IP'. Vertical dots below 'IP' indicate that other lower-level protocols may exist.

- Relies on TCP for reliable communication

7

WILLIAM & MARY

Architecture (Cont'd)

- **Handshake protocol:** establishment of a session key
- **Change Cipher protocol:** start using the previously-negotiated encryption / message authentication
- **Alert protocol:** notification (warnings or fatal exceptions)
- **Record protocol:** protected (encrypted, authenticated) communication between client and server


8

WILLIAM & MARY

SSL Services


- Peer authentication
- Negotiation of security parameters
- Generation / distribution of session keys
- Data confidentiality
- Data integrity

9

 **Connections and Sessions** WILLIAM & MARY


- **SSL Session**
 - an association between peers
 - created through a handshake, negotiates security parameters, can be **long-lasting**
- **SSL Connection**
 - a type of service (i.e., an application) between a client and a server
 - **transient**
- Multiple connections can be part of a single session

10

 **Session Parameters** WILLIAM & MARY



- Session ID
- X.509 public-key **certificate** of peer
- **Compression** algorithm to use
- **Cipher** specification: encryption algorithm, message digest, etc.
- **Master** (session) **secret: 48-byte** (384 bits) secret negotiated between peers

11

 **Connection Parameters** WILLIAM & MARY



- Server and client **nonces**
- Server and client **authentication keys**
- Server and client **encryption keys**
- Server and client **initialization vectors**
- Current message **sequence number**

12

 **Ciphers Supported by SSL** 



- DES+HMAC/SHA-1
- 3DES+HMAC/SHA-1
- RC4+MD5
- RC2+MD5
- +others

13

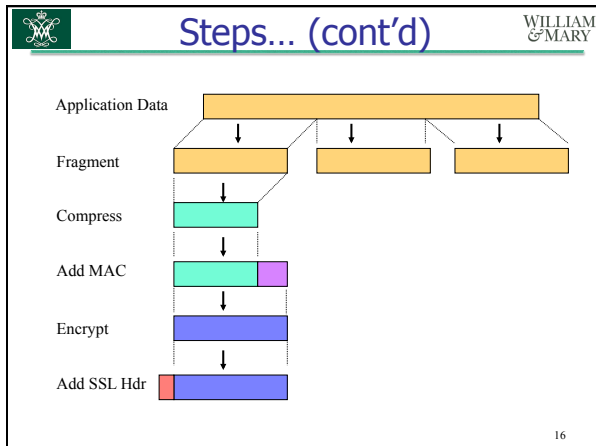
The SSL Record Protocol

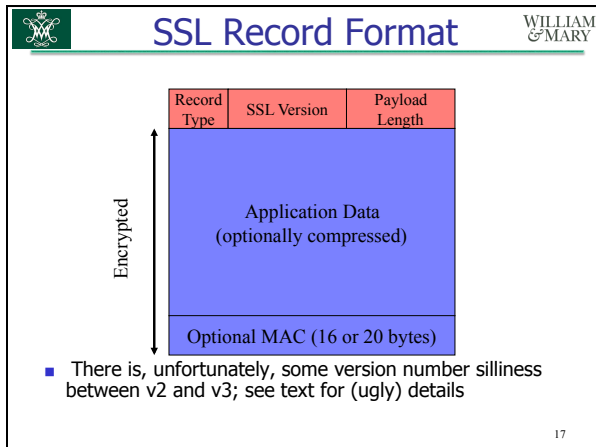
14

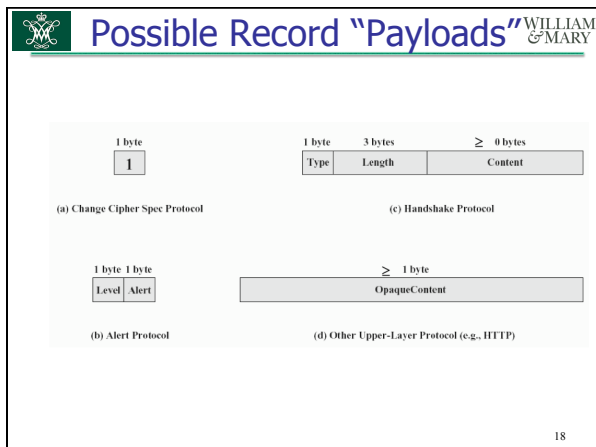
 **Protocol Steps** 

1. Fragment data stream into **records**
 - each with a maximum length of 2^{14} (=16K) bytes
2. **Compress** each record
3. Create **message authentication code** for each record
4. **Encrypt** each record

15









SSL Handshake Protocol

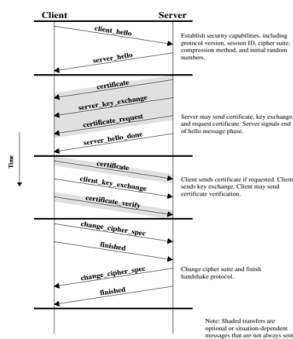


Phases of Protocol

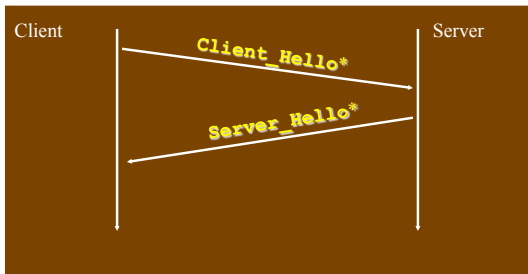
- I. Establish security capabilities
 - version of SSL to use
 - cipher + parameters to use
- II. Authenticate server (optional), and perform key exchange
- III. Authenticate client (optional), and perform key exchange
- IV. Finish up



All the Messages



WILLIAM & MARY
I. Establish Security Capabilities



■ Messages marked with * are mandatory

22

WILLIAM & MARY
Client Hello Message

- Transmitted in plaintext
- Contents
 - highest SSL version understood by client
 - R_C : a 4-byte timestamp + 28-byte random number
 - session ID: 0 for a new session, non-zero for a previous session
 - list of supported cryptographic algorithms
 - list of supported compression methods

23

WILLIAM & MARY
Server Hello Message

- Also transmitted in plaintext
- Contents
 - minimum of (highest version supported by server, highest version supported by client)
 - R_S : 4-byte timestamp and 28-byte random number
 - session ID
 - a cryptographic choice selected from the client's list
 - a compression method selected from the client's list

24

II. Server Auth. / Key Exchange



- The Server_Certificate message is optional, but **almost always used** in practice

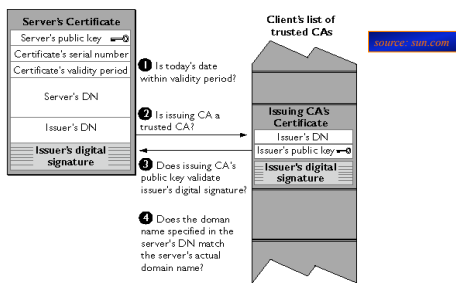
25

Server Certificate Message

- Contains a certificate with server's public key, in X.509 format
 - or, a chain of certificates if required
- The server certificate is **necessary** for any key exchange method except for anonymous Diffie-Hellman

26

Authenticating the Server



- Step #4: Domain name in certificate **must** match domain name of server (not part of SSL protocol, but clients should check this)

27

Key Exchange Methods Supported WILLIAM & MARY

- **RSA** (server must have a certificate)
- **Ephemeral Public Key**
 - public keys are exchanged, signed using long-term RSA keys
- **Fixed Diffie-Hellman**
 - server provides the D-H public parameters in a certificate
 - client responds with D-H public key either in a certificate, or in a key exchange message
- **Anonymous Diffie-Hellman**
 - Diffie-Hellman without authentication
 - Susceptible to Man-in-the-middle attack

28

Server Key Exchange Message WILLIAM & MARY

- Needed for...
 - anonymous D-H
 - ephemeral public key

29

Server Key Exchange WILLIAM & MARY


Handshake	Server Key Exchange (Diffie-Hellman)	Server Key Exchange (RSA)
Type		
Length		
Data	p (modulus, prime) g (generator) $g^y \text{ mod } p$ Signature	m (modulus = $p \cdot q$) e (pub. exp.) Signature
Diffie-Hellman		RSA
Client Computes: $\text{PreMasterSecret} = (g^y)^x \text{ mod } p$		Client Computes: $y = \text{PreMasterSecret}^e \text{ mod } p$
Client Sends : g^x to server		Client sends : y to server
Server Computes: $\text{PreMasterSecret} = (g^x)^y \text{ mod } p$		Server Computes: $\text{PreMasterSecret} = y^d \text{ mod } p$

30

- Normally not used, because in **most** applications
 - **only the server** is authenticated
 - client is authenticated at the application layer, if needed
- Two parameters
 - certificate type accepted, e.g., RSA/signature only, DSS/signature only, ...
 - list of certificate authorities recognized (i.e., trusted third parties)




- Contains a certificate, or chain of certificates if needed

 Client Key Exchange Message WILLIAM & MARY


- If using RSA, the **pre-master secret S**, **encrypted** with the server's public key
- If using D-H, the client's public key

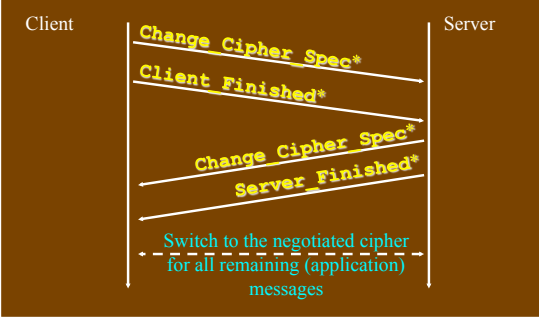
34

 Client Certificate Verify Msg WILLIAM & MARY

- Proves the client is the valid owner of a certificate (i.e., knows the corresponding private key)
- Only sent following any client certificate that has signing capability

35

 IV. Finish Up WILLIAM & MARY



```

sequenceDiagram
    participant Client
    participant Server
    Client->>Server: Change Cipher Spec*
    Client->>Server: Client Finished*
    Server-->>Client: Change Cipher Spec*
    Server-->>Client: Server Finished*
    Note over Client,Server: Switch to the negotiated cipher for all remaining (application) messages
  
```

36



Change Cipher Spec Msg

WILLIAM & MARY

- Confirms the change of the current state of the session to a newly-negotiated set of cryptographic parameters
- **Finished** Messages
 - keyed hash of the previous handshake messages to prevent man-in-the-middle-attacks from succeeding

37



"Abbreviated" Protocol Possible

WILLIAM & MARY

- Allows **resumption** of a previously-established session
 - does not require authentication of server or client
 - does not exchange keys
- Details omitted

38



Creating the "Master" Secret

WILLIAM & MARY


- The master secret is a one-time (per session) **48-byte** (= 16+16+16) value
- Parameters
 - the **pre-master secret S** has previously been communicated using RSA or D-H
 - the client nonce R_c
 - the server nonce R_s
- Computation: $K =$

$$\text{MD5}(S \parallel \text{SHA-1}(\text{"A"} \parallel S \parallel R_c \parallel R_s)) \parallel$$

$$\text{MD5}(S \parallel \text{SHA-1}(\text{"BB"} \parallel S \parallel R_c \parallel R_s)) \parallel$$


$$\text{MD5}(S \parallel \text{SHA-1}(\text{"CCC"} \parallel S \parallel R_c \parallel R_s))$$

39

 **Cryptographic Parameters** WILLIAM & MARY


- Generated from
 - the master secret K
 - Rc
 - Rs
- Values to be generated
 - client authentication and encryption keys
 - server authentication and encryption keys
 - client encryption IV
 - server encryption IV

40

 **Alert Protocol Examples** WILLIAM & MARY

- Type 1: **Fatal_Alert**
 - ex.: **Unexpected_Message, Bad_MAC,** etc.
 - connection is immediately terminated
- Type 2: **Warning**
 - ex.: **No_Certificate, Close_Notify**

41

 **Summary** WILLIAM & MARY

1. SSL is the de facto authentication/ encryption protocol standard for HTTP
 - becoming popular for many other protocols as well
2. Allows negotiation of cryptographic methods and parameters

42
