SSL and TLS

Dr. Mana Khatua
Assistant Professor
Dept. of CSE, IIT Guwahati
Email: manaskhatua@iitg.ac.in

Web Security

- Literally all businesses, most govt. agencies, and many individuals now have Web sites.
- Those are accessed through graphical Web browsers over Internet.
- However, in reality, the Internet and the Web are extremely vulnerable to attack !

Source: https://cwatch.comodo.com/blog/website-security/what-is-website-security/

Web Security Considerations

- The World Wide Web (WWW) is
  - ✓ a **client-server application** running over the Internet and TCP/IP intranets

Few characteristics of Web usage suggest the need for security tools:

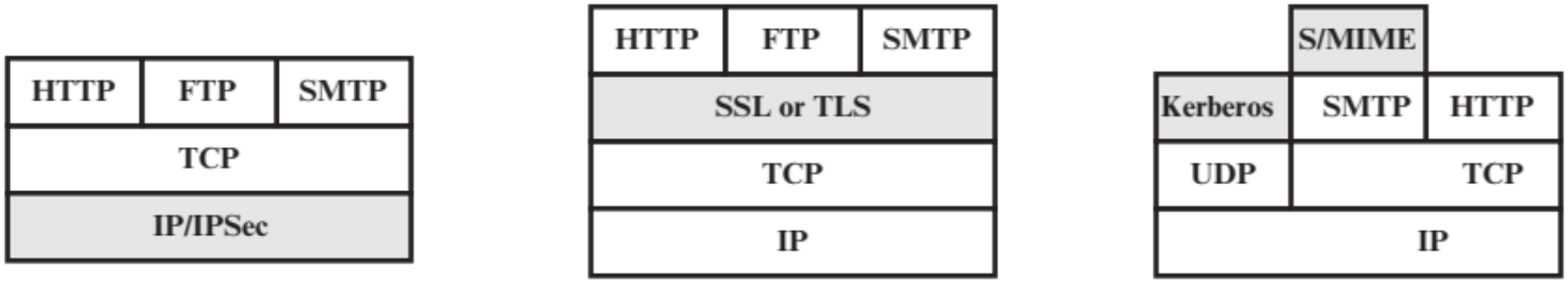
- Web servers are relatively easy to configure and manage
- Web content is increasingly easy to develop
- The underlying software is complex
  - May hide many potential security flaws
- A Web server can be exploited as a launching pad
  - into the corporation's or agency's entire computer complex
- Casual and untrained users (in security matters) are common clients for Web-based services
  - Such users are not necessarily aware of the security risks that exist,
  - and, do not have the tools or knowledge to take effective countermeasures

Threats on Web

	Threats	Consequences	Countermeasures
Integrity	•Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit	•Loss of information •Compromise of machine •Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	•Eavesdropping on the net •Theft of info from server •Theft of data from client •Info about network configuration •Info about which client talks to server	•Loss of information •Loss of privacy	Encryption, Web proxies
Denial of Service	•Killing of user threads •Flooding machine with bogus requests •Filling up disk or memory •Isolating machine by DNS attacks	•Disruptive •Annoying •Prevent user from getting work done	Difficult to prevent
Authentication	•Impersonation of legitimate users •Data forgery	•Misrepresentation of user •Belief that false information is valid	Cryptographic techniques

Web Traffic Security Approaches

A number of approaches for Web Security is possible.



(a) Network level

(b) Transport level

(c) Application level

Network Level: One way to provide Web security is to use IP security (IPSec).

- Advantages: (1) transparent to end users and applications;
  (2) provides a general-purpose solution;
  (3) includes a traffic filtering capability

Transport Level: more general-purpose solution is to implement security just above TCP

- E.g. Secure Sockets Layer (SSL), Transport Layer Security (TLS)

Application Level: Application-specific security services are embedded within the particular application.

- E.g. Kerberos

Source: Cryptography and Network Security – Principles and Practice, by William Stallings, 7th Edition, Pearson India, 2017

Secure Sockets Layer (SSL)

- SSL is an encryption-based Internet security protocol.
  - purpose of ensuring **privacy**, **authentication**, and **data integrity** in Internet communications.
  - developed by Netscape Communications Corporation

- **Authentication** between client and server – using handshake protocol
- **High privacy** – by encryption-decryption
- **Data integrity** – by digitally signed data

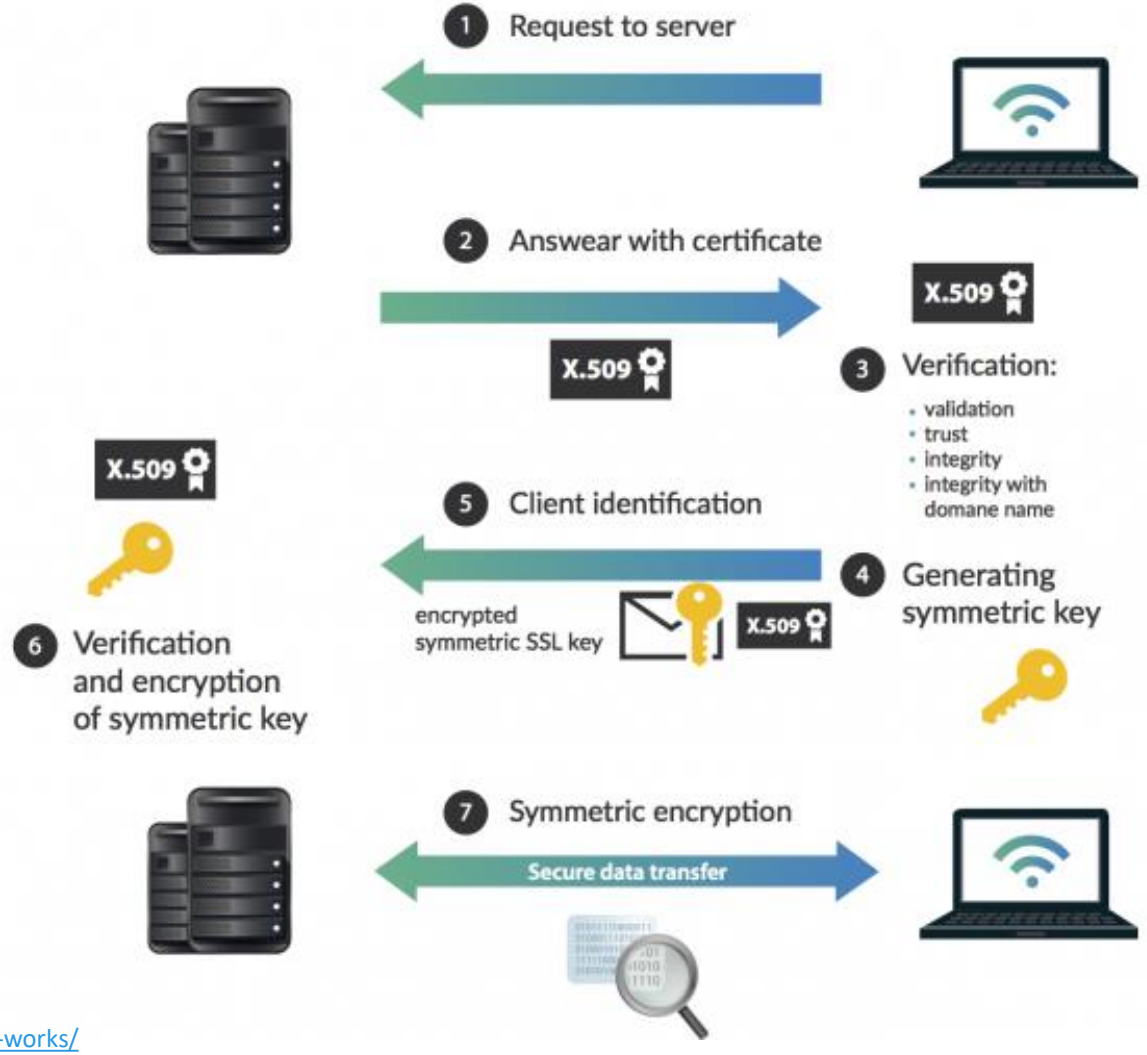
- At present it has been deprecated by IETF, and released the successor TLS in 1999.

➢ SSL-enabled browsers can communicate securely with server having digital certificate, using SSL.
  - So, a browser that does not support HTTP over SSL cannot request URLs using HTTPS.

✓ HTTPS represents a unique protocol that combines SSL and HTTP.
  - As HTTPS (HTTP + SSL) and HTTP are different protocols and use different ports (443 and 80, respectively), we can run both SSL and non-SSL requests simultaneously.

SSL Certificates – How it works?

- An **SSL certificate** is a digital document
  - that mainly binds the identity of a website to a cryptographic key pair (public key & private key)
- An SSL certificate is a type of X.509 public key certificate, but it is a Server Certificate.
- SSL certificates work by establishing an encrypted connection between a web browser and a server using shared secret key



Source https://www.internetum.com/en/what-is-ssl-certificate-and-how-it-works/

Types of SSL Certificates

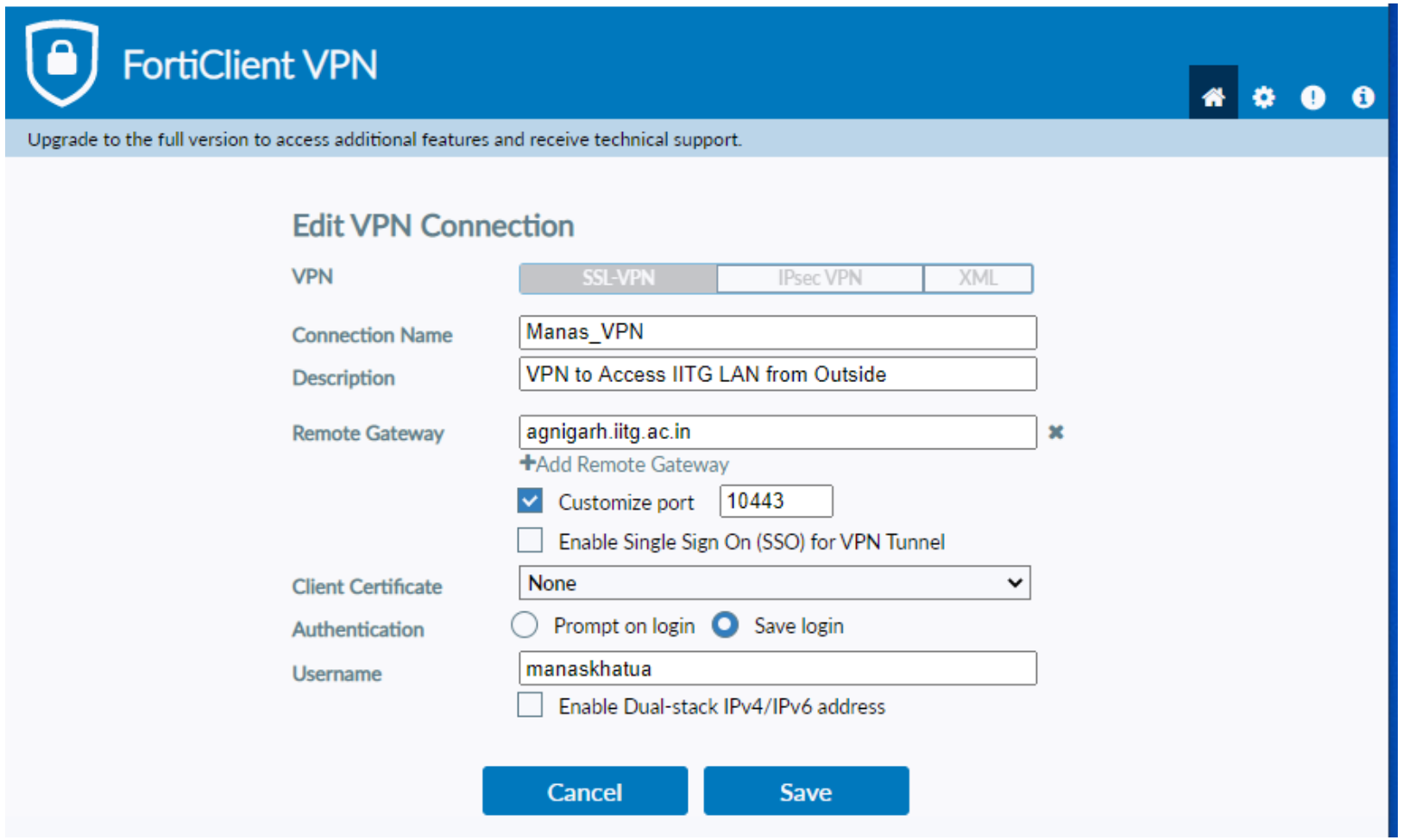
Domain Validation (DV) Certificates

Organization Validated (OV) Certificates

Extended Validation (EV) Certificates

- **Single-domain:** A single-domain SSL certificate applies to only one domain
  - e.g. www.iitg.ac.in
- **Wildcard:** A wildcard SSL certificate also applies to only one domain. However, it also includes that domain's subdomains.
  - e.g, a wildcard certificate could cover www.iitg.ac.in , www.cse.iitg.ac.in , www.eee.iitg.ac.in , etc.
- **Multi-domain:** As the name indicates, multi-domain SSL certificates can apply to multiple unrelated domains.

Example Application



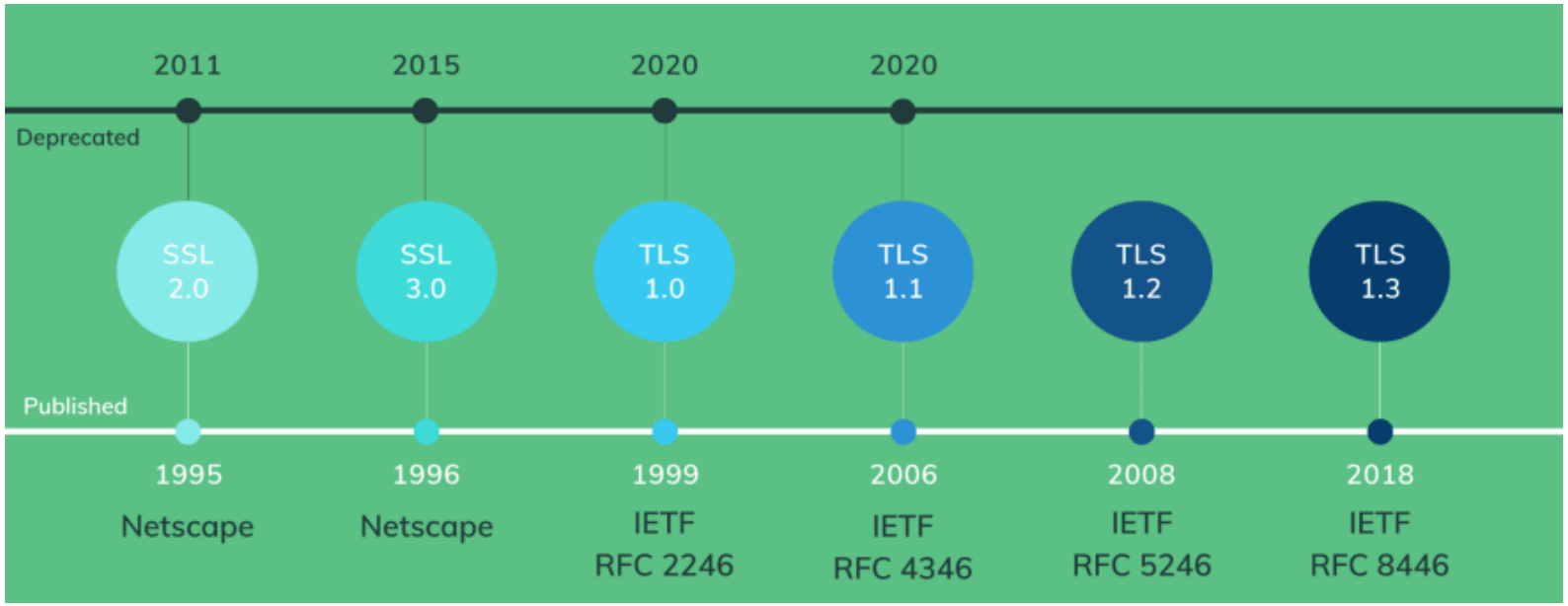
Transport Layer Security

Transport Layer Security (TLS) [RFC 5246]**:**

TLS is a Internet standard that evolved from a commercial protocol SSL

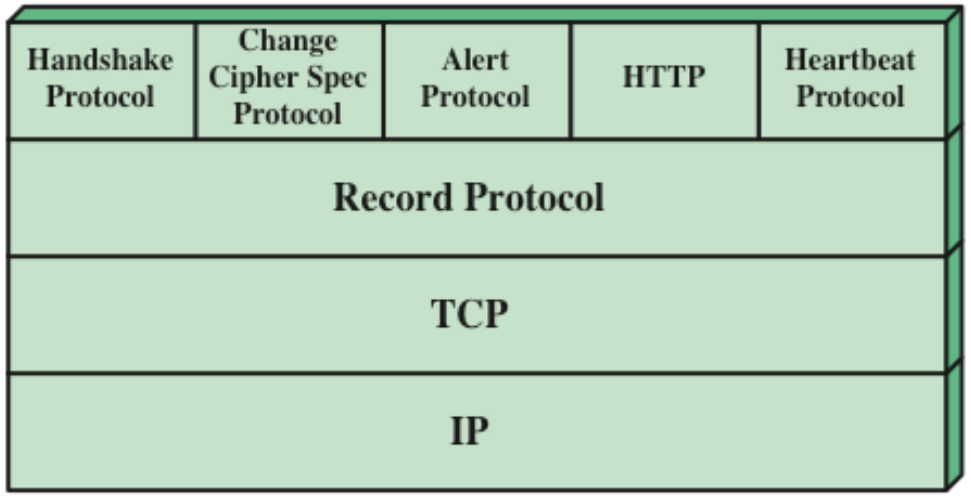
TLS make use of TCP to provide a reliable end-to-end secure service

Few Applications: Web (HTTP+TLS); Email (SMTP+TLS); File Transfer (FTP+TLS)



Source https://dev.to/techschoolguru/a-complete-overview-of-ssl-tls-and-its-cryptographic-system-36pd

TLS Architecture



- The main purpose of handshake is for authentication and key exchange.
- We will achieve both confidentiality and integrity by using record protocol.

TLS is not a single protocol, but rather two layers of protocols.

- Record Protocol provides **basic security services** to upper layer protocols such as HTTP.
- Three higher-layer protocols are defined as part of TLS:
  - ✓ Handshake Protocol
  - ✓ Change Cipher Spec Protocol
  - ✓ Alert Protocol

  } used to manage TLS exchanges

  - ✓ Heartbeat Protocol is also used as fourth one.

TLS Connection & TLS Session

TLS connection

- It is a transport that provides a suitable type of service
- For TLS, such connections are peer-to-peer relationships
- Connections are transient i.e. temporary
- Every connection is associated with one session

TLS session

- It is 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
- Used to avoid the expensive negotiation of new security parameters for each connection

Session State Parameters

Session identifier	• An arbitrary byte sequence chosen by the server to identify an active or resumable session state
Peer certificate	• An X509.v3 certificate of the peer; this element of the state may be null
Compression method	• The algorithm used to compress data prior to encryption
Cipher spec	• Specifies the bulk data encryption algorithm and a hash algorithm used for MAC calculation
Master secret	• 48-byte secret shared between the client and the server
Is resumable	• A flag indicating whether the session can be used to initiate new connections

Connection State Parameters

Server and 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 encryption key for data encrypted by the server and decrypted by the client

Client write key

- The 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 TLS 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 $2^{64} - 1$

TLS Record Protocol

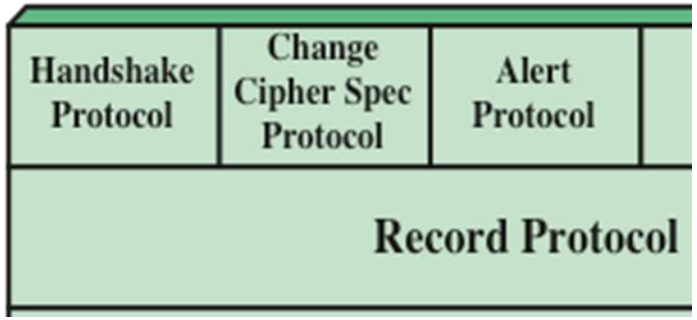
It provides two basic security services for TLS connections

Confidentiality

The Handshake Protocol defines a shared secret key (SK) that is used for conventional Encryption of TLS payloads

Message integrity

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



Record Protocol Operation

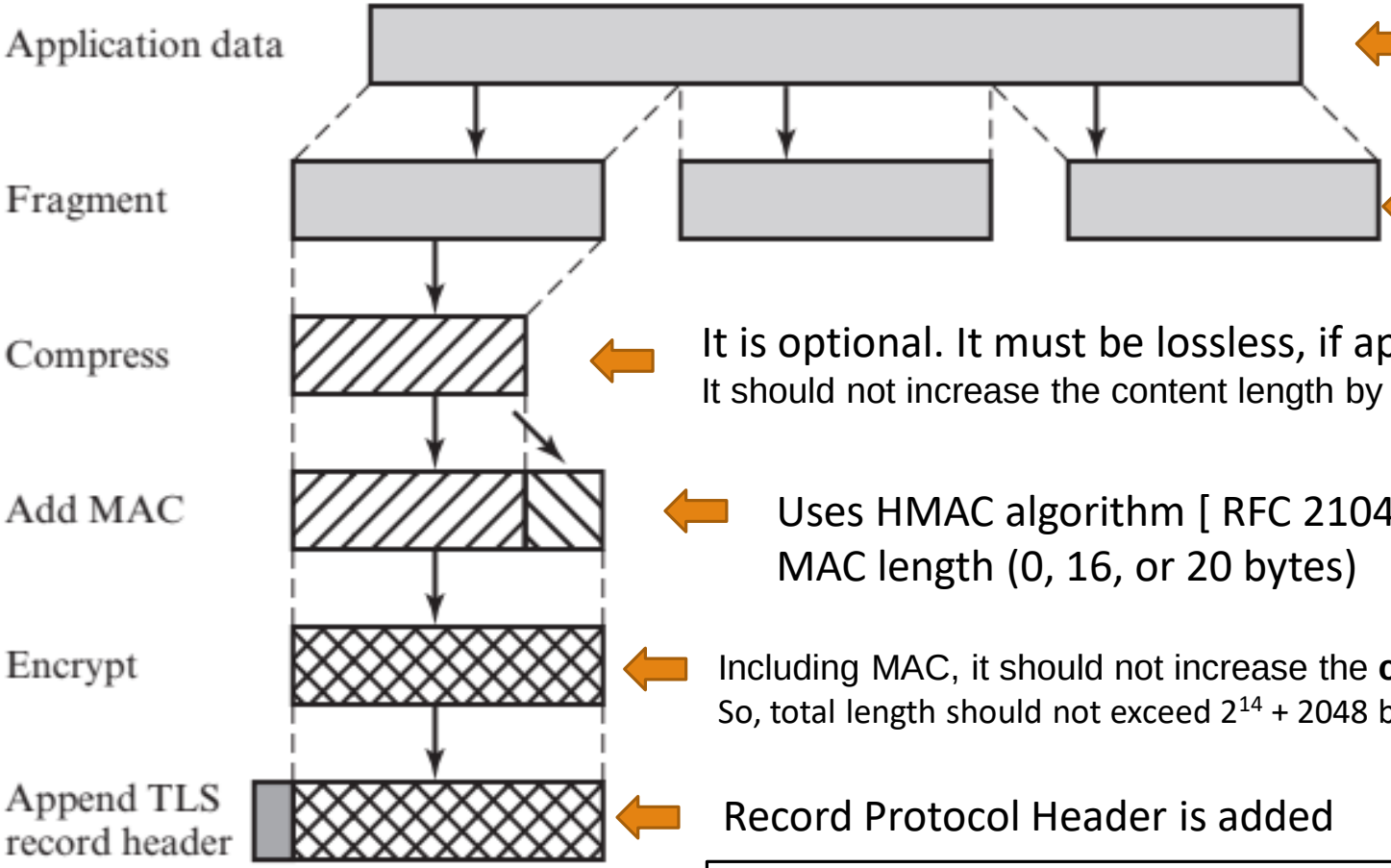
Operation steps before transmission:

1) Takes input an application message
2) Fragments the data into manageable blocks
3) Optionally compress
4) Applies MAC
5) Encrypts
6) Add Header
7) Then transmits into a TCP segment

Following are performed on received data:

1) Remove header
2) Decrypted
3) Verified
4) Decompressed
5) Re-assembled
6) Then deliver to higher-level users

Record Protocol Operation



Accept this data

Block = 2^{14} bytes
= 16384 bytes

It is optional. It must be lossless, if applied.
It should not increase the content length by more than 1024 bytes

Uses HMAC algorithm [RFC 2104]
MAC length (0, 16, or 20 bytes)

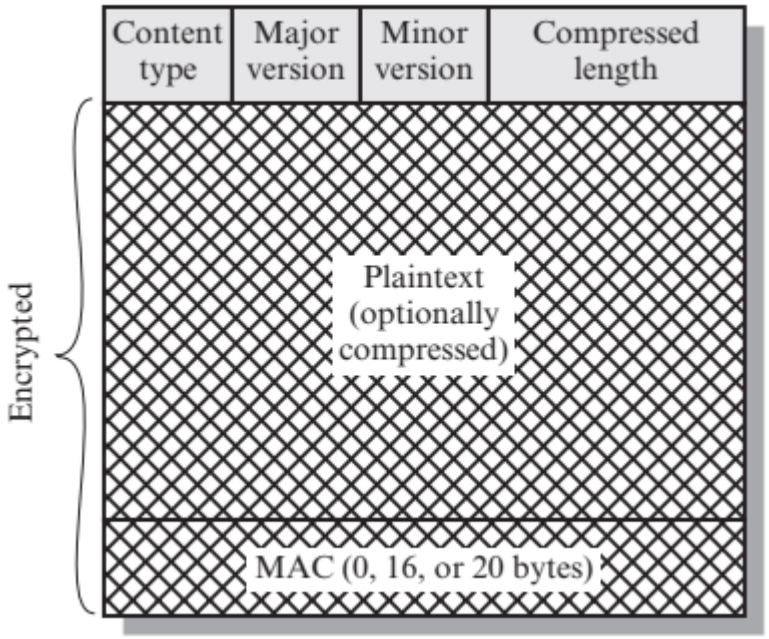
Including MAC, it should not increase the **content length** by more than 1024 bytes.
So, total length should not exceed 2^{14} + 2048 bytes.

Record Protocol Header is added

Content Type (8 bits)	Major Version (8 bits)
Minor Version (8 bits)	Compressed Length (16 bits)

Block Cipher		Stream Cipher	
Algorithm	**Key Size**	**Algorithm**	**Key Size**
AES 3DES	128, 256 168	RC4-128	128

TLS Record Format



Content Types:

- change_cipher_spec
- alert
- handshake
- application_data

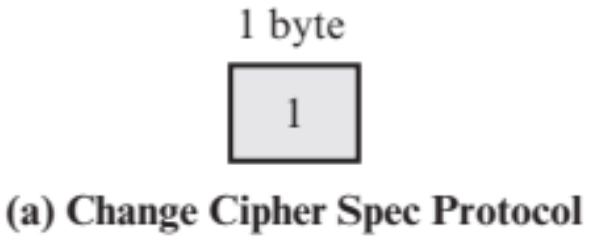
Major Version: Major TLS version.
For TLSv2, it is 3.

Minor Version: Minor TLS version.
For TLSv2, it is 1.

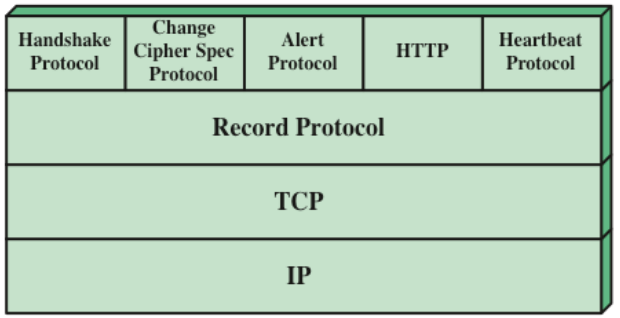
Compressed Length:

- Final length of the fragment (max 2^{14}+2048 bytes)

Change Cipher Spec Protocol



(a) Change Cipher Spec Protocol



Change Cipher Spec Protocol consists of a single message , which consists of a single byte with the value 1.

The sole purpose of this message is to cause **the pending state to be copied into the current state**, which updates the cipher suite to be used on this connection.

Alert Protocol

It is used to convey TLS-related alerts to the peer entity.

Each message in this protocol consists of two bytes:

- **Level (1 byte)**
  - Warning (value 1) – for sending warning
  - Fatal (value 2) -- TLS immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established.
- **Alert (1 byte)**
  - contains a code that indicates the specific alert

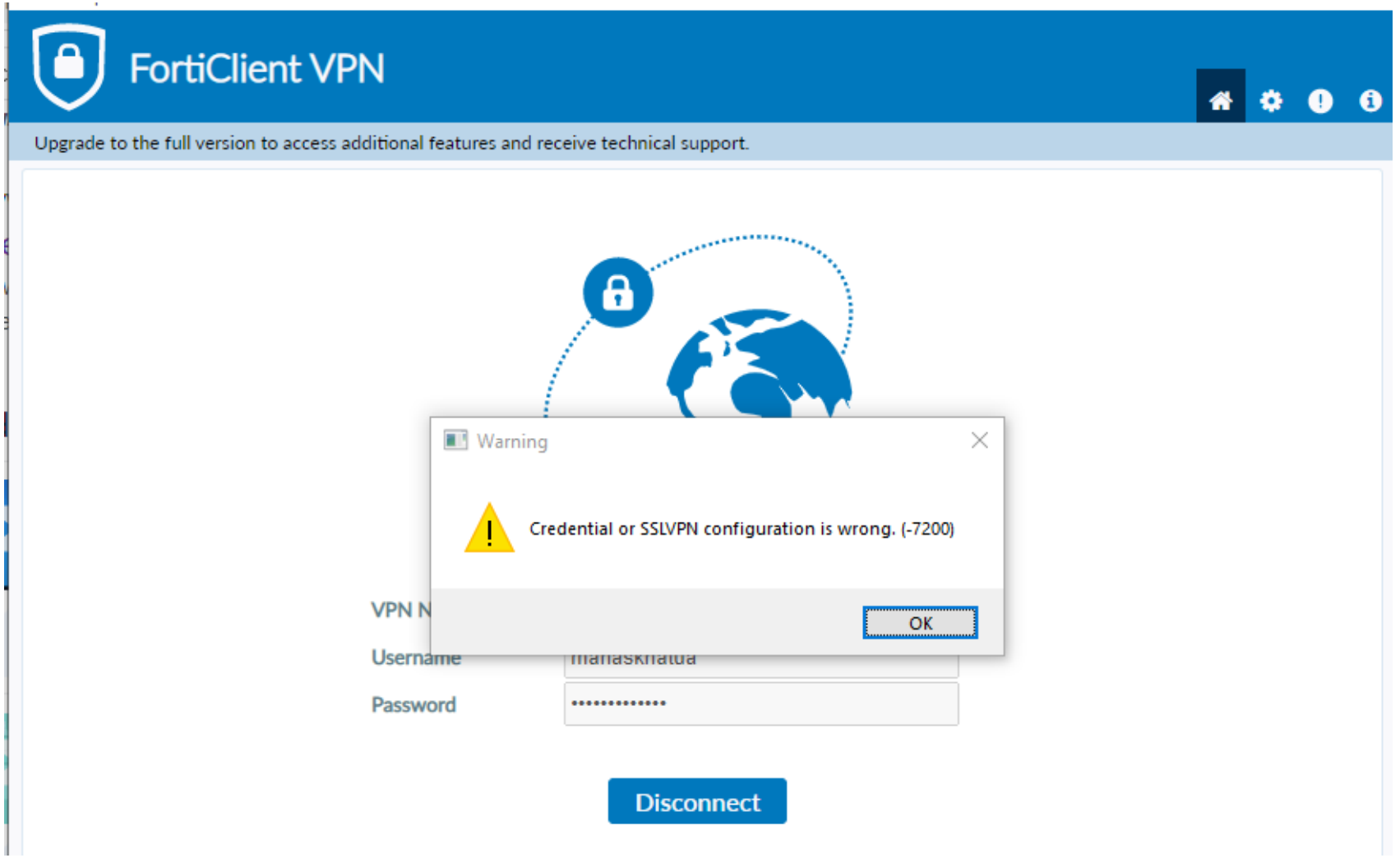
Few Fatal alerts:

- unexpected_message
- bad_record_mac
- decompression_failure
- handshake_failure
- decryption_failed
- record_overflow
- access_denied

Few Warning alerts:

- bad_certificate
- certificate_revoked
- certificate_expired
- certificate_unknown
- user_canceled
- no_renegotiation

Example of an Alert



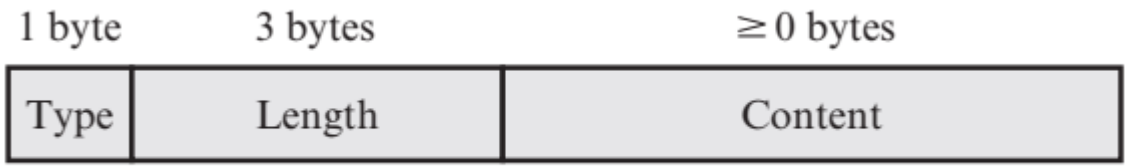
Handshake Protocol

This protocol allows the server and client

- to authenticate each other
- to negotiate an Encryption and MAC algorithms
- to negotiate cryptographic keys to be used to protect data sent in a TLS record.

The Handshake Protocol is used before any application data is transmitted.

Each message in Handshake Protocol has following three fields:



(c) Handshake Protocol

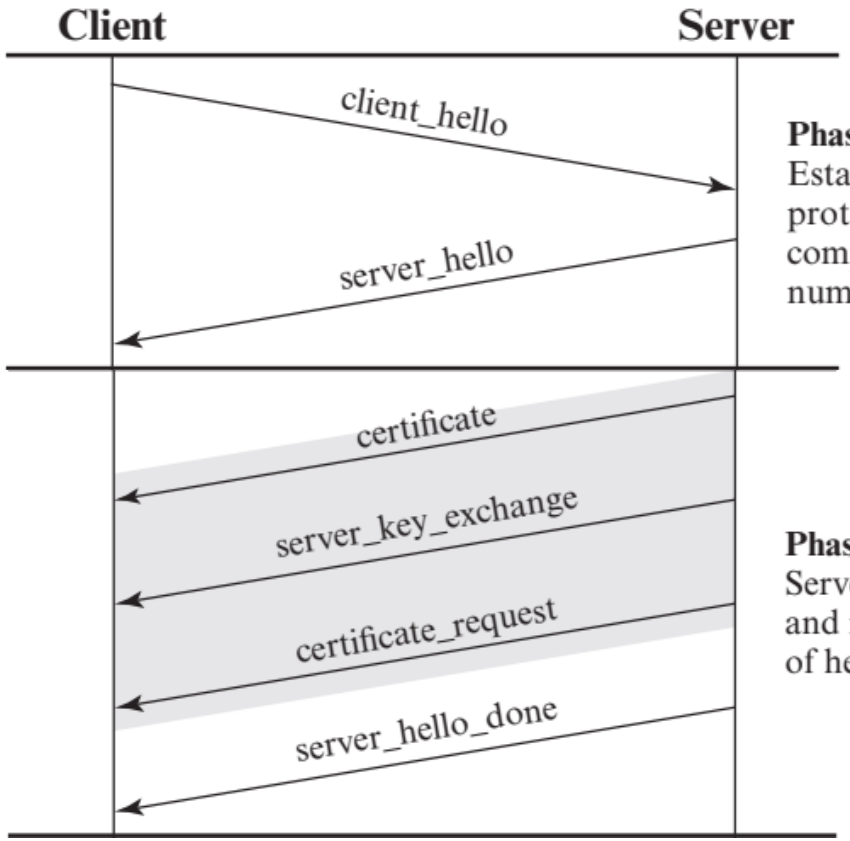
- **Type (1 byte)**: Indicates one of 10 messages
- **Length (3 bytes)**: The length of the message in bytes.
- **Content (# 0 bytes)**: The parameters associated with this message

Cont…

Initial exchange between client and server is needed

- To establish a logical connection, and
- To establish security capabilities

It consists of 4 Phases



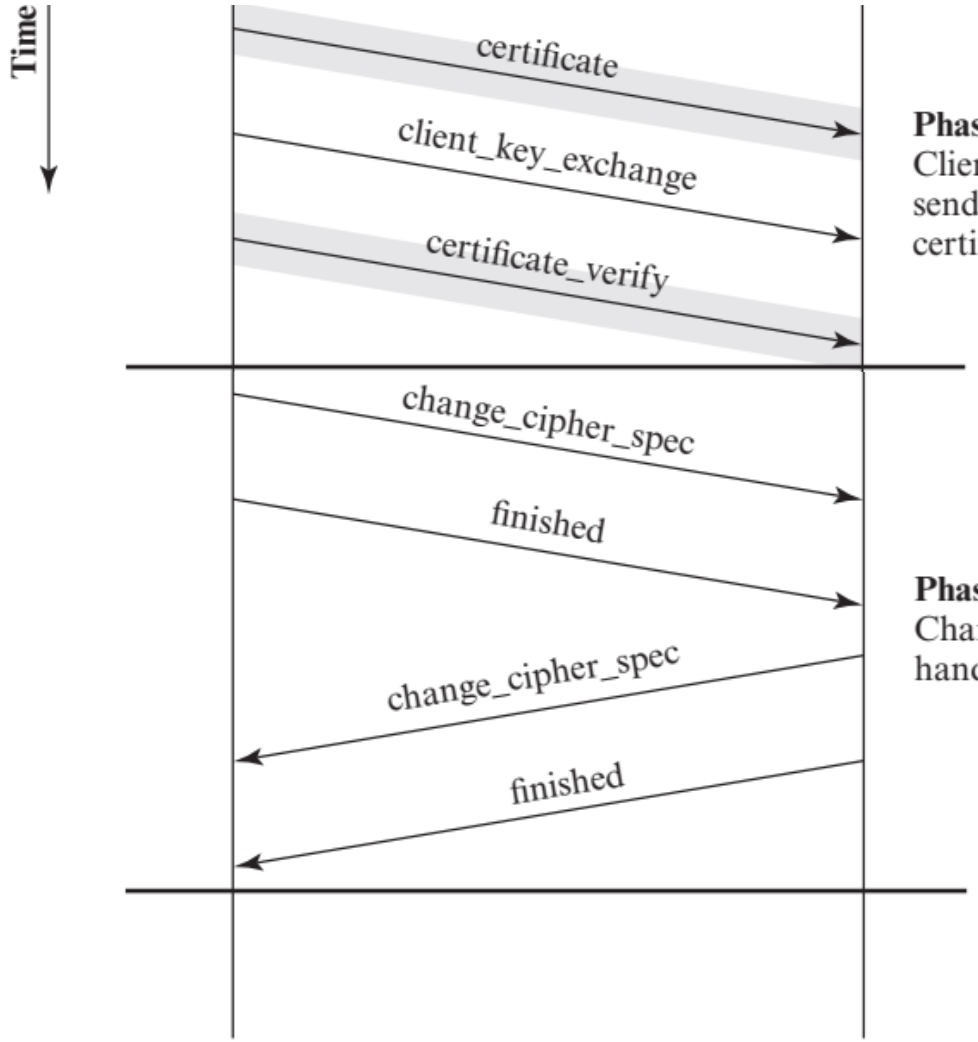
Establish Security Capabilities

Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Server Authentication and Key Exchange

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Cont…



Client Authentication and Key Exchange

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Finish

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Heartbeat Protocol (RFC 6250)

- A heartbeat is a periodic signal generated by hardware / software
  - to indicate/notify normal operation or
  - to synchronize other parts of a system.

- A heartbeat protocol is typically used to monitor the availability of a protocol entity.

- It runs on top of the TLS Record Protocol
- It consists of two message types:
  - *heartbeat_request*
  - *heartbeat_response*.
- The use of it is established during Phase 1 of the Handshake protocol

- The heartbeat serves two purposes.
  - First, it assures the sender that the recipient is still alive, even though there may not have been any activity over the underlying TCP connection for a while.
  - Second, the heartbeat generates activity across the connection during idle periods, which avoids closure by a firewall that does not tolerate idle connections.

Cryptographic Computations

Two further items are of interest:

1) 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
   - The creation is in two stages
     - First, a pre_master_secret is **exchanged**
     - Second, the master_secret is **calculated by both parties**

For pre_master_secret exchange, there are two possibilities.

- **RSA**
  - A 48-byte pre_master_secret is generated by the client, encrypted with the server's public RSA key, and sent to the server.
  - The server decrypts the ciphertext using its private key to recover the pre_master_secret.
- **Diffie-Hellman**
  - Both client and server generate a Diffie–Hellman public key. After these are exchanged, each side performs the Diffie–Hellman calculation to create the shared pre_master_secret.

Both sides now compute the master_secret as follows:

master_secret = PRF(pre_master_secret, "master secret", ClientHello.random || ServerHello.random)

where, ClientHello.random and ServerHello.random are the two nonce values exchanged in the initial hello messages.
PRF: Pseudo Random Function

Cont..

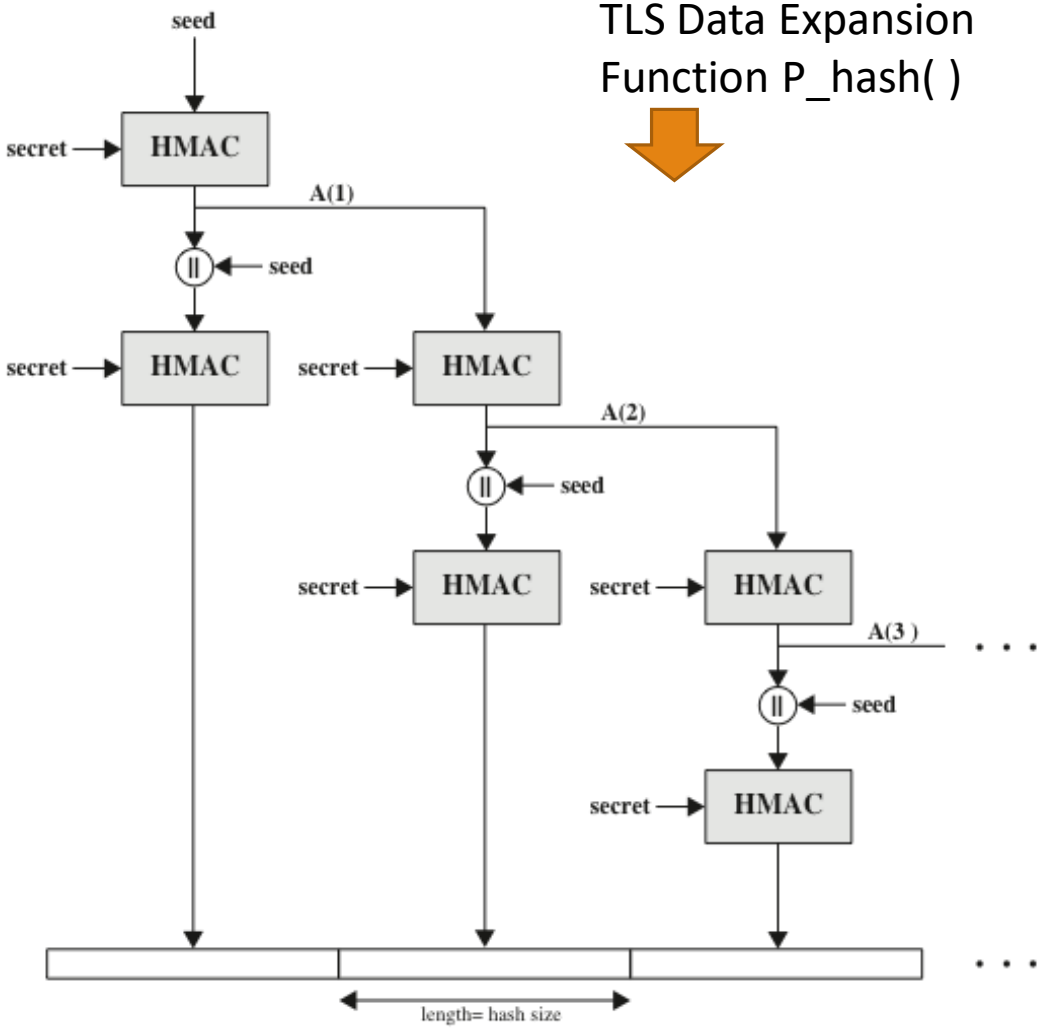
Two further items are of interest:

1) 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
   - The creation is in two stages
     - First, a pre_master_secret is **exchanged**
     - Second, the master_secret is **calculated by both parties**

2) The generation of cryptographic parameters from the master secret
   - CipherSpecs require many parameters
     - A client write MAC secret ; A server write MAC secret
     - A client write key ; A server write key
     - A client write IV; A server write IV
   - These are generated from the master secret by hashing the master secret into a sequence of secure bytes of sufficient length

key_block =
MD5(master_secret ‘ SHA (=A> ‘ master_secret ‘ ServerHello.random ‘ ClientHello.random)) ‘
MD5(master_secret ‘ SHA(=BB> ‘ master_secret ‘ ServerHello.random ‘ ClientHello.random)) ‘
. . . . . .

PseudoRandom Function



TLS Data Expansion Function P_hash( )

To make PRF as secure as possible, it uses two hash algorithms.

PRF is defined as,

PRF (secret, label, seed) = P_<hash>(secret, label || seed)

PRF takes as input a secret value, an identifying label, and a seed value and produces an output of arbitrary length.


Thank
You