Secure Sockets Layer (SSL) and Transport Layer Security (TLS): Part 1

Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

NPTEL

References

- J. Kurose, K. Ross, "*Computer Networking: A Top Down Approach*", Sixth Edition, Pearson Education, 2013
- A. Tanenbaum, D. Wetherall, "*Computer Networks*", Fifth Edition, Pearson Education, 2012.
- L. Peterson, B. Davie, "*Computer Networks: A Systems Approach*", Fifth Edition, Morgan Kaufmann, 2012.
- W. Stallings, "*Cryptography and Network Security: Principles and Practice*", Pearson Education, 7th edition, 2016
- E. Rescorla, "*SSL and TLS: Designing and Building Secure Systems*", Addison-Wesley, 2001

SSL

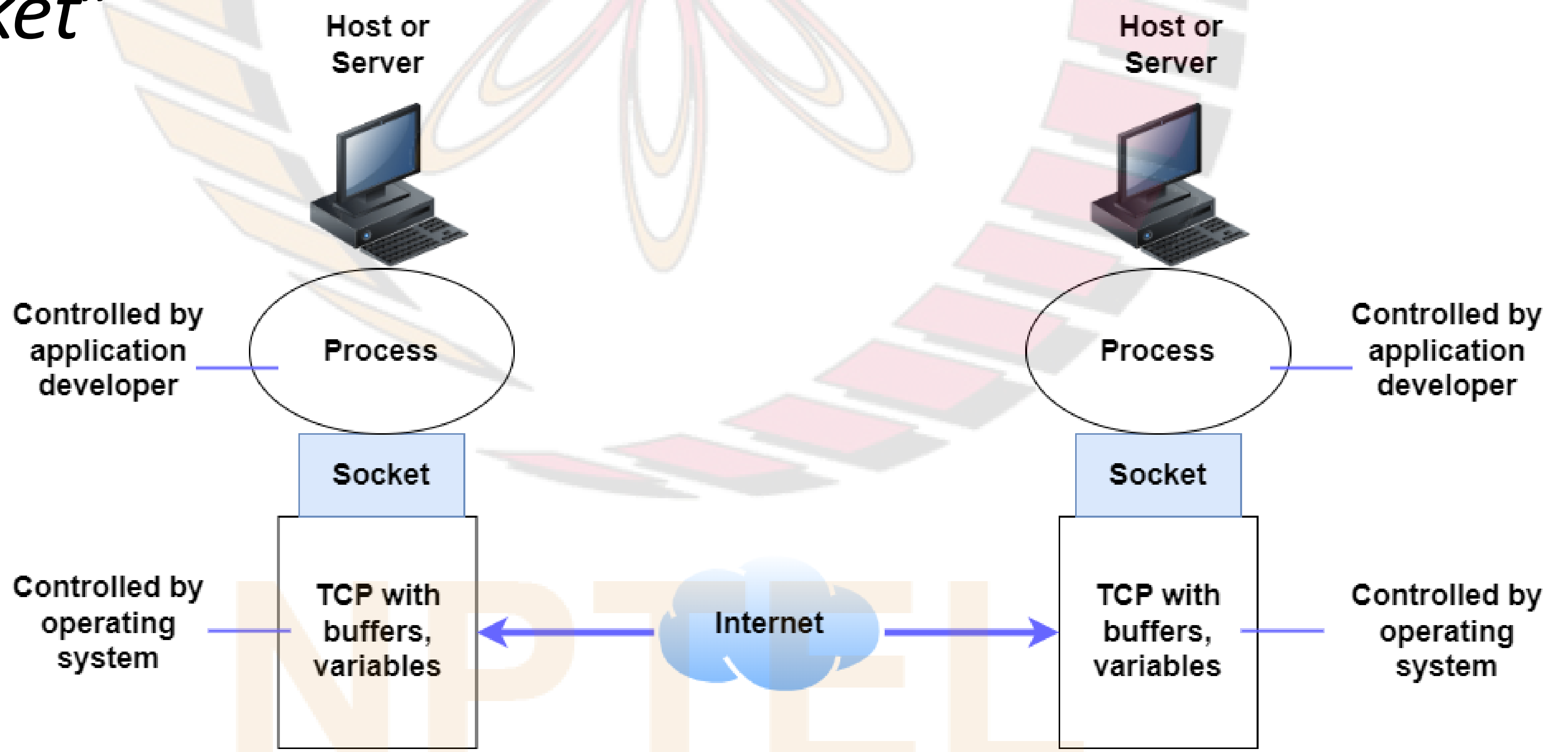
- Adds following security services to TCP:
  - ❑ Confidentiality
  - ❑ Message Integrity
  - ❑ Server Authentication
  - ❑ Client Authentication
- A slightly modified version of SSL, called "Transport Layer Security (TLS)", has been standardized
- Extensively used in Internet to secure web (HTTP) traffic, *e.g.*, by Google, Amazon, eBay, etc.
  - ❑ especially useful when sensitive information needs to be transferred, *e.g.*, credit card details in e-commerce transactions, passwords while logging in to Gmail, Yahoo! Mail, etc.
  - ❑ when SSL used by browser, URL begins with https: instead of http:

Example

- Suppose Bob visits “Alice Incorporated” website and wants to order perfume
- The website sends a form to Bob, in which he enters type of perfume and quantity required, address, credit card information, etc., and submits form
- Examples of attacks by an intruder Trudy in above scenario (assuming no security mechanisms used):
  - ❑ may sniff Bob’s messages, obtain credit card information and use it to make purchases (since no *confidentiality* used)
  - ❑ may modify Bob’s messages, *e.g.*, having him order ten times more perfume than required (since no *message integrity* used)
  - ❑ may intercept all messages (*e.g.*, if Trudy controls a compromised intermediate router) and send fake webpage with “Alice Incorporated” name and erroneous details (since no *server authentication* used)
- Such attacks prevented by SSL since it adds confidentiality, message integrity and server authentication to TCP
- **Note**: Since SSL secures TCP, apart from HTTP, it can be used to secure any application that uses TCP:
  - ❑ e.g., FTPS: file transfer over SSL

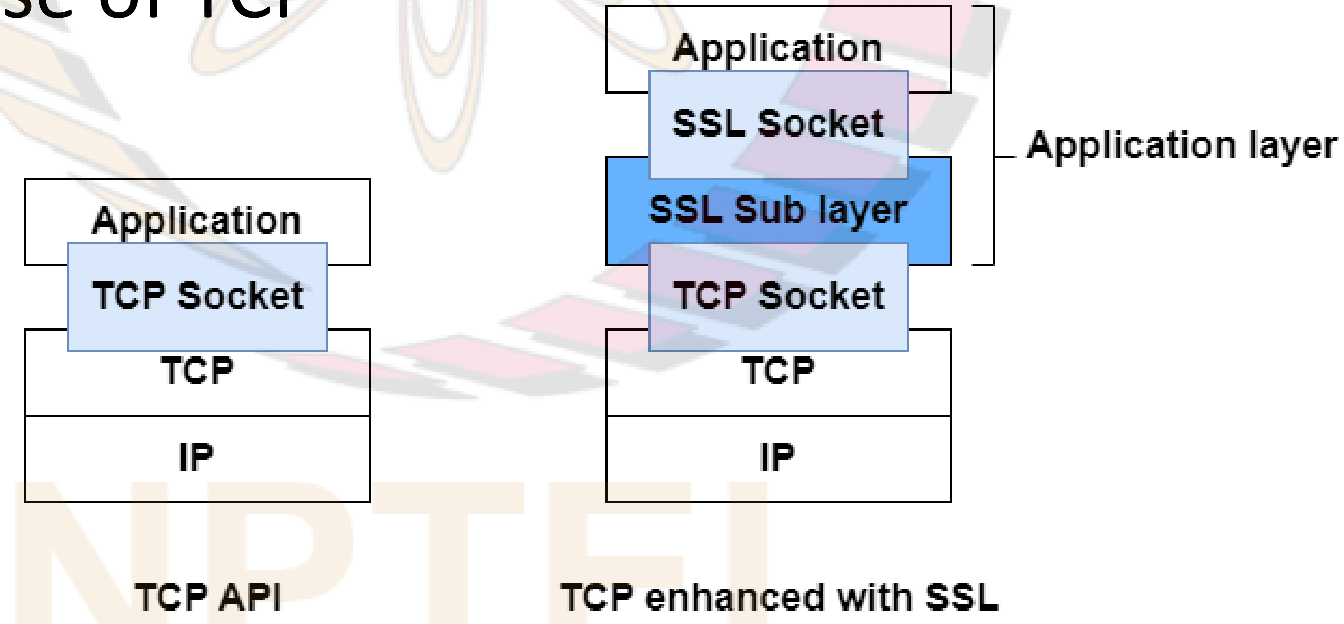
TCP Socket

- Consider an application that uses TCP (without SSL)
- Application process (in application layer) at each end sends messages into and receives messages from network through a software interface called *"socket"*



SSL (contd.)

- SSL provides an API similar to TCP's API
  - ❑commands to establish and close connections, send and receive messages included in API
- SSL technically resides in application layer; however, an application-developer can use it similar to use of TCP



Simplified Version of SSL

- First, we study a simplified version of SSL
  - ❑later: actual SSL
- SSL (and the simplified version) have three phases:
  - ❑Handshake
  - ❑Key derivation
  - ❑Data transfer
- Consider a communication session between a client (Bob) and a server (Alice)
  - ❑server has a certificate for its public key from a CA

Simplified Version of SSL: Handshake

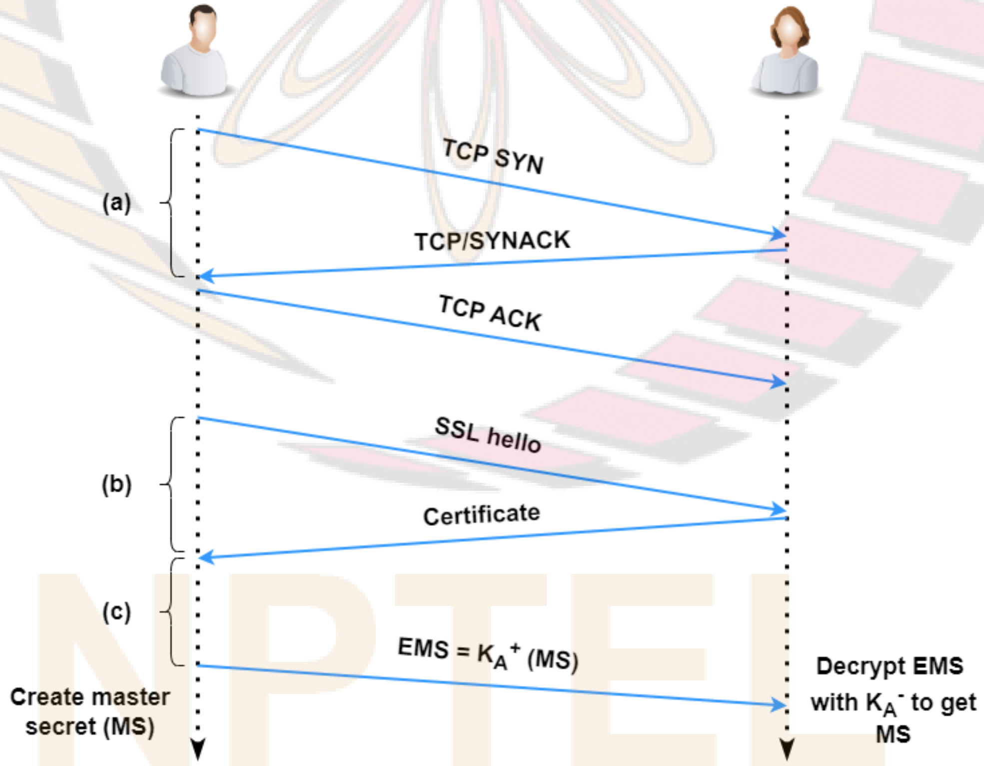
a) Bob and Alice establish a TCP connection

b) Alice sends the certificate (provided by a CA) for her public key, K_A^+, to Bob

- ❑ used by Bob to verify that the public key actually belongs to Alice

c) Bob generates a Master Secret (MS) (only for this SSL session), encrypts it using K_A^+ and sends the Encrypted MS (EMS) to Alice

- ❑ EMS decrypted by Alice using K_A^-



Simplified Version of SSL: Key Derivation

- The MS is used by Alice and Bob to generate four symmetric keys:
  - ❑ E_B: key used for encryption of data sent from Bob to Alice
  - ❑ E_A: key used for encryption of data sent from Alice to Bob
  - ❑ M_B: key used for message integrity of data sent from Bob to Alice (used as the authentication key to generate MAC)
  - ❑ M_A: key used for message integrity of data sent from Alice to Bob
- Reasons for generating four different keys from MS instead of using MS itself as the key for all four tasks:
  - ❑ in a given direction, if same key used for encryption of a block and for generating MAC for that block, attacker may be able to use this fact and knowledge of the two algorithms to get some information about the plaintext data
  - ❑ if same keys used in both directions, attacker may be able to take an encrypted block from client and send it back to client as if it was sent from server

Simplified Version of SSL: Data Transfer

- Alice and Bob start sending data, with encryption and MAC generation done using above keys, to each other
- One way: generate a MAC for entire data and send it at the end
- Shortcoming:
  - ❑ large delay until data can be used by applications
- Better approach:
  - ❑ data stream in each direction broken into “records’, and a MAC appended to each record
- To generate MAC, Bob finds hash of (record data, M_B)
- Also, Bob encrypts (record data, MAC) using E_B and passes it to TCP for transport to Alice
- Can message integrity of above scheme be breached by Trudy, an intruder who controls a compromised intermediate router?

Simplified Version of SSL: Data Transfer (contd.)

- Message integrity of above scheme can be breached by Trudy by deleting, replaying or reordering records, *e.g.*:
  - Trudy may capture two packets sent by Bob, reverse their order, adjust the TCP sequence numbers, which are not encrypted, and send them to Alice
- Scheme to defend against such attacks:
  - assign an SSL sequence number to each record
- Sequence number not included in SSL record itself, but following scheme used
- Bob maintains a sequence number counter, which begins at 0 and increments for each SSL record he sends; Alice maintains a counter that increments each time an SSL record is received
- Sequence number included in calculation of MAC for a record:
  - MAC is hash of (record data, M_B, *current sequence number*)
- Alice checks whether MAC is correct using her own sequence number counter
- Advantage of above scheme over that in which sequence number is included in SSL record:
  - sequence numbers included by TCP; discrepancies can be checked using MAC
  - hence, not necessary to include sequence number in SSL record; bandwidth saved by omitting it
  - intruder can maintain a counter that increments each time an SSL record is sent; in general, a cipher weakened if an intruder has access to some known plaintext and corresponding ciphertext

NPTEL

SSL Record Format

- Type field indicates whether the record is:
  - ❑handshake message,
  - ❑message that contains application data or
  - ❑message used to close the connection
- Length field is used by SSL at receiving end to extract SSL record from TCP byte stream
  - ❑note that one SSL record need not correspond to one TCP packet
- Note that type, version and length are not encrypted; however, they are included in MAC calculation
  - ❑hence, modification in them by intruder can be detected