Secure Socket Layer (SSL)
Https uses SSL: E-Commerce, banking, etc. ("There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS."

SSL facts:
- Runs in a user level process on top of TCP
- Why over TCP?
  - In order delivery
  - Reliability
- SSL v2 developed by Netscape in 1995
- Microsoft developed PCT
- SSL v3 is commonly deployed today
- IETF proposed Transport Layer Security (TLS) – widely deployed

Two phases: Handshake and Data transfer

SSL Handshake Phase:

```
<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>------------------------</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;---------------Chosen cipher,R_{Bob},Certificate-----------------------</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------------------K_{Bob}{s}--------------------------</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------------MAC (handshake msgs sent/received, SRVR)-----</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------------MAC (handshake msgs sent/received, CLNT) --&gt;</td>
</tr>
</tbody>
</table>
```

Both sides find master secret $K=f(s,R_A,R_B)$ to generate 2 keys for encrypting (one for each side), 2 keys for integrity protection (one for each side), and two IV's for cipher block chaining (one for each side).

This only provides server authentication. Client authentication is optional in SSL. It can be done through client certificates. However, in most cases this is done by having clients login (username,password).

To remove the RSA component of the handshake in multiple connections closely spaced in time (e.g., in the case of http 1.0), a session ID can be returned in message 2. The session ID is then used in messages 1 and 2 in further connections removing the need for message 3. Checkout the figure from the text book.

Messages 4 and 5 are for preventing Downward attacks (changing Supported ciphers and/or Chosen cipher).
[Question asked in class: Is it sufficient to only compute the hash of the messages sent by Bob and Alice in 4 and 5, respectively? Without including received messages, Alice, would not know if someone tampered with her messages or Bob would not know if someone tampered with his messages. Including both sent and received messages allows Alice and Bob to know if anyone tampered with any of the messages.]

$R_{Alice}$ and $R_{Bob}$ are 32 octets, with the first 4 bytes being the Unix time (since 1970).

IPSec lets you pick and choose among these choices but SSL creates cipher suites that you can choose from such as:

SSL_RSA_EXPORT_WITH_DES40_CBC_SHA

What if a server wishes to authenticate a client using the client’s certificate? It specifies the CAs it trusts.

The above protocol is does not provide PFS. However, one could incorporate D-H key exchange in the above protocol to obtain the pre-master secret to obtain PFS.

Exportability (now not used):

SSLv2: Domestic connections allowed 128 bit encryptions

Exportable allowed 40 bit, with 88 bits sent in the clear

Exportable public key length limited to 512 bits

e.g. $K_{Bob}^s,88$ other bits (where $s$ is only 40 bits)

with 512 RSA keys, we have $512\{40\}$

So what about domestic servers with international clients?

SSLv3:

Client

Server (using 1024 bit keys)

Generates an ephemeral 512 bit private/public key pair

$\leftarrow$Chosen cipher,$R_{Bob}$,Certificate,$K_B$ $\leftarrow$($K_{B,temp}^s$)------(msg2)

--------$K_{B,temp}^s$[40 bits of s],88-------------------------->(msg3)

This ends up providing a "weak" Perfect Forward Secrecy since the ephemeral keys (including the ephemeral private key) are forgotten afterwards.

**SSL Data Transfer Phase:**

The figure below is only a rough depiction of the data transfer phase. Please look at the figure from the book in the section on the data transfer phase for a much better depiction.

- Structure of a block of data for the SSL Data Transfer Phase:
With,

**SEQ**: sequence number of transferring block.

**RH** (Recode Header): includes record type (handshaking/data phase); SSL version (2/3); and the length of the RH - 3 fields.

**Data**: actual data being transferred.

**HMAC**: digest of (SEQ,RH,Data) for integrity protection.

**Padding**: to assure the length of (Data,HMAC,padding) satisfies the chosen encryption method.

Notice: (SEQ,RH,Data) is integrity protected, while (Data,HMAC,padding) is encrypted.

The SEQ field is added to prevent block replay attack (adversary resends the same block to break the application logic). However, this SEQ field is implicitly used, it means SEQ field is not sent or received, but the sender/receiver will keep track of this field and add the SEQ before forwarding the block to higher layers.