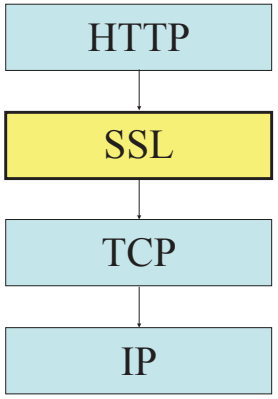
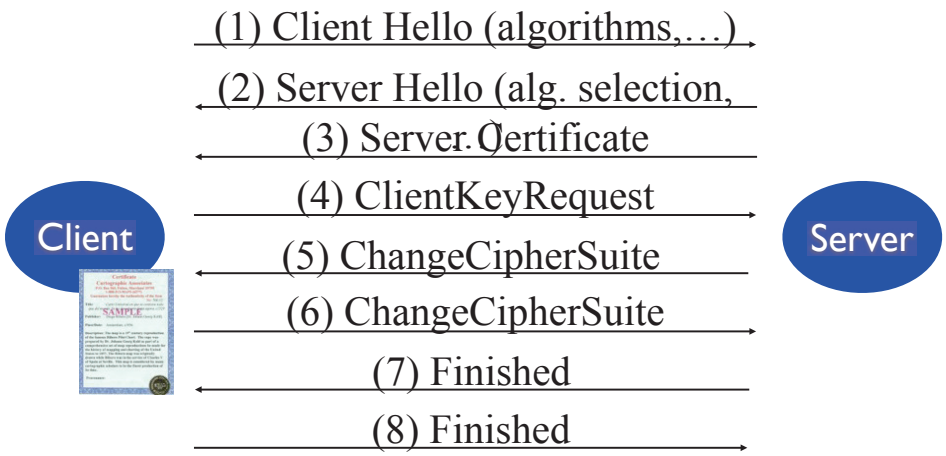
Web Transport Security: SSL

- Secure socket Layer (SSL/TLS)
- Used to authenticate servers
  - Uses certificates, "root" CAs
- Can authenticate clients
- Inclusive security protocol
- Security at the socket layer
  - Transport Layer Security (TLS)
  - Provides
    - authentication
    - confidentiality
    - integrity



SSL Handshake



Simplified Protocol Detail

Participants: Alice/A (client) and Bob/B (server)
Crypto Elements : Random R, Certificate C, k_i^+ Public Key (of i)
Crypto Functions : Hash function $H(x)$, Encryption $E(k, d)$, Decryption $D(k, d)$, Keyed MAC $HMAC(k, d)$

1. Alice → Bob R_A
2. Bob → Alice R_B, C_B
   Alice pick pre-master secret S
   Alice calculate master secret $K = H(S, R_A, R_B)$
3. Alice → Bob $E(k_B^+, S), HMAC(K,'CLNT' + [\#1, \#2])$
   Bob recover pre-master secret $S = D(k_B^-, E(k_B^+, S))$
   Bob calculate master secret $K = H(S, R_A, R_B)$
4. Bob → Alice $HMAC(K,'SRVR' + [\#1, \#2])$

Note: Alice and Bob : IV Keys, Encryption Keys, and Integrity Keys 6 keys,where each key $k_i = g_i(K, R_A, R_B)$, and g_i is key generator function.

SSL Tradeoffs

- Pros
  - Server authentication*
  - GUI clues for users
  - Built into every browser
  - Easy to configure on the server
  - Protocol has been analyzed like crazy
- Cons
  - Users don't check certificates
  - Too easy to obtain certificates
  - Too many roots in the browsers
  - Some settings are terrible