ECE568 Lecture 13: The SSL Protocol

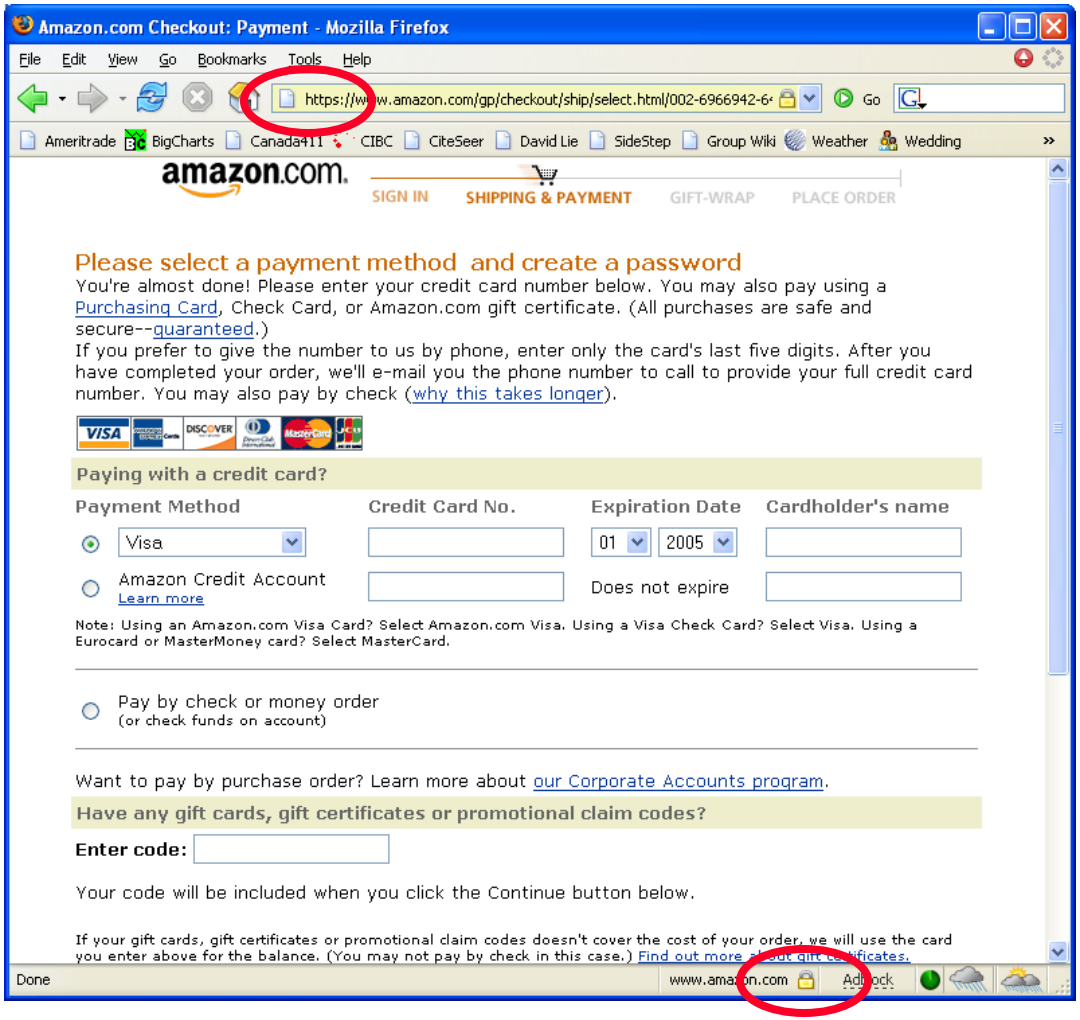
David Lie
Department of Electrical and Computer Engineering
University of Toronto

Lecture Outline

- The SSL Protocol
  - SSL Handshake
  - SSL Communication
- Security of the SSL Protocol
- Performance implications of SSL

The SSL Protocol

- The window is requesting credit card information that will be transmitted across the web.
- Since packets pass through untrusted routers between the customer's machine and Amazon, the credit card information is encrypted so that routers transferring the information cannot read it.
- The protocol used to encrypt and transfer this information is called SSL. The "https" in the URL and the icon at the bottom indicate that SSL is being used to transfer the information.

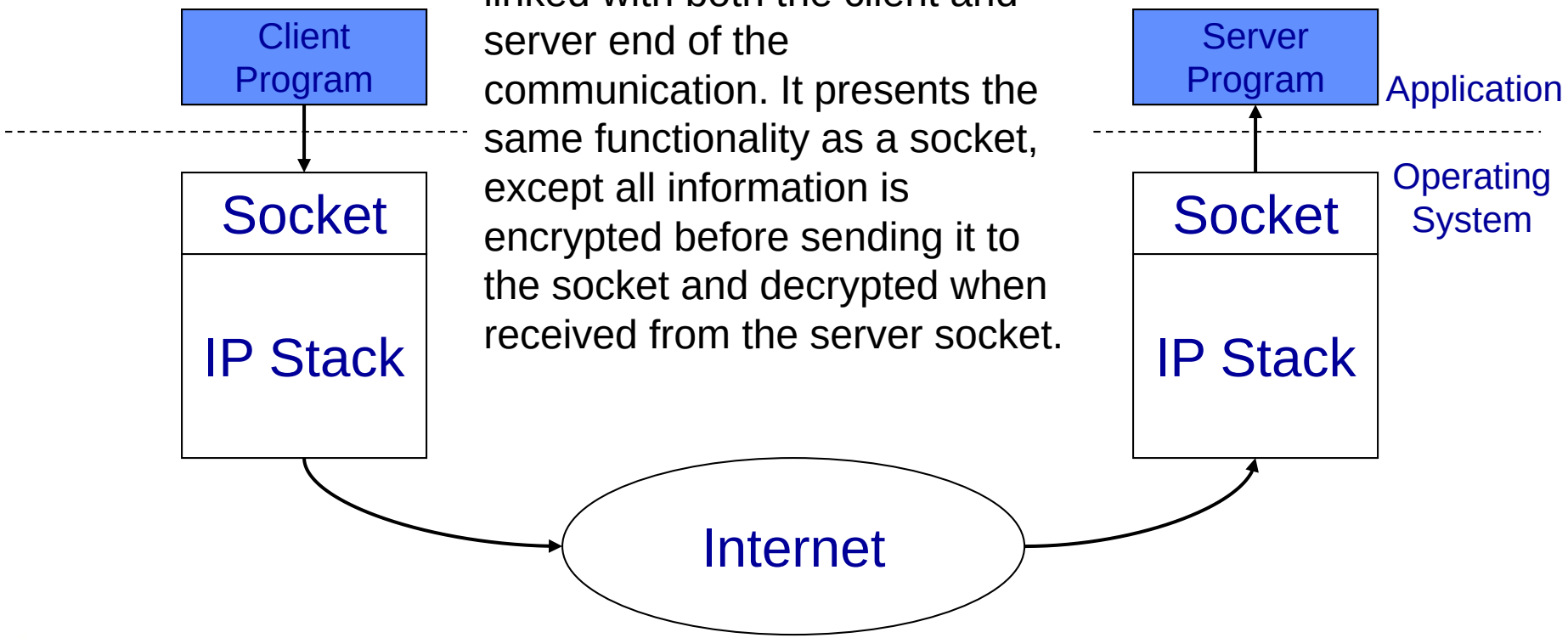


The SSL Protocol

- The **Secure Sockets Layer (SSL)** Protocol was first designed by Netscape in 1996.
  - It is also sometimes referred to as **Transport Layer Security (TLS)**
  - It has undergone several revisions, the latest is version 3.0
  - It is most commonly used to secure web sessions, but can be used to secure any application (since it replaces regular sockets).
  - To work, both ends of the communication have to support SSL. On the other hand, there is no support required from the network in between. This is often called **end-to-end** security.

The SSL Protocol

SSL is implemented as a application level library that is linked with both the client and server end of the communication. It presents the same functionality as a socket, except all information is encrypted before sending it to the socket and decrypted when received from the server socket.



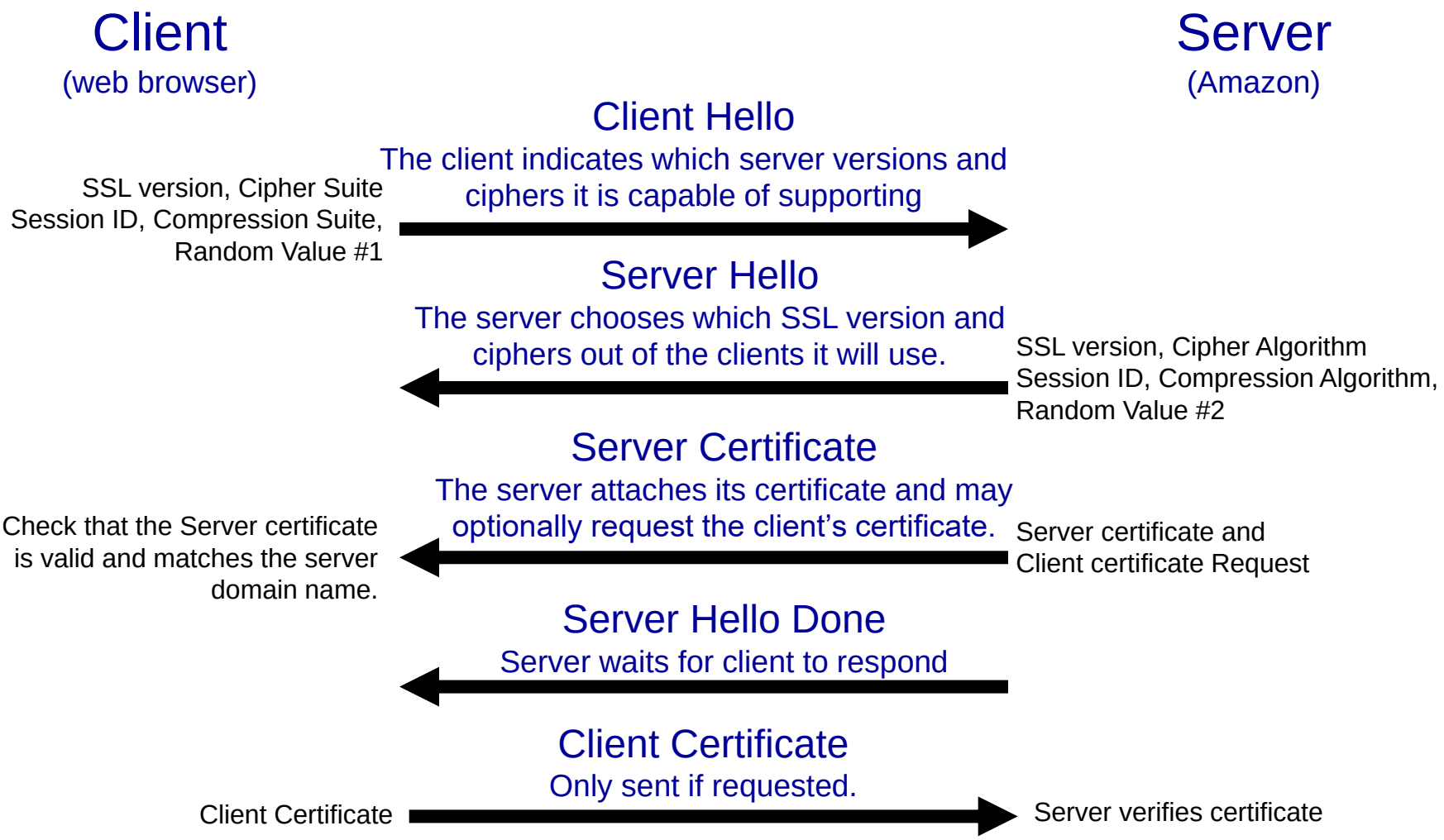
SSL Mechanics

- SSL has 2 phases:
  1. **Key Exchange or Handshake**: The initial phase establishes a shared secret key between the sender and the receiver. Any authentication is also performed. Compatibility between different versions is also handled here. Since this happens only once for any exchange, it can be relatively slow
  2. **Communication**: Once the keys are setup, an arbitrary number of messages can be exchanged between the two parties in both directions. This could involve a large amount of data, so this phase is pretty efficient.

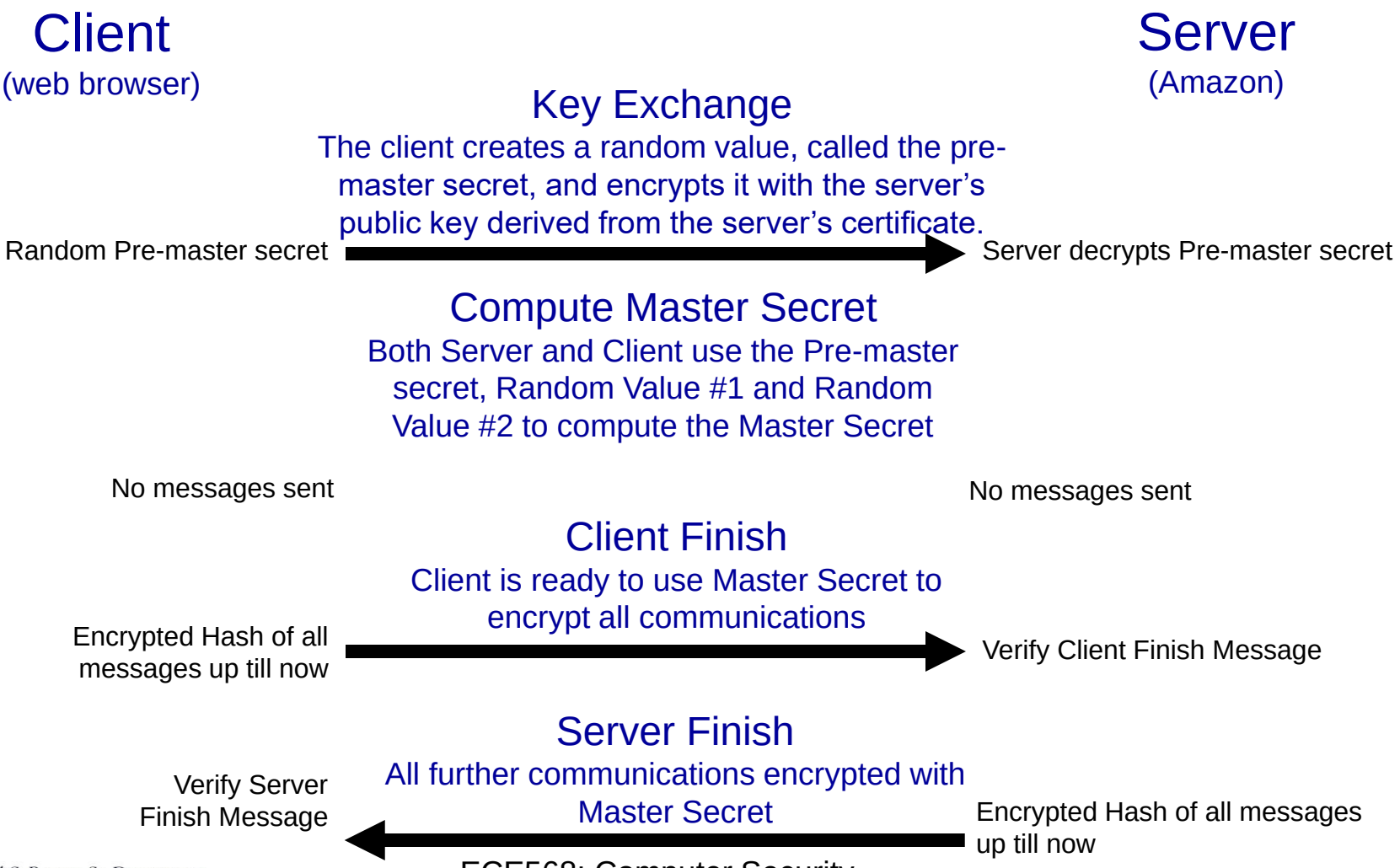
SSL Handshake

- The SSL Handshake has 3 purposes:
  1. Establish the suite of ciphers each side supports and what version of the protocol is being used.
  2. Securely establish a shared secret that can be used as a session key (for symmetric encryption).
  3. Authenticate each other's identities via certificates. Note this authenticates the identities of the machines, not the users (i.e. people) making the requests.
     - User authentication (i.e. a website asking you for a username/password) is not done by the SSL protocol, but by scripts or servlets on the web server.
     - Note that client machine authentication is optional and usually not done (since web servers will connect to any client).

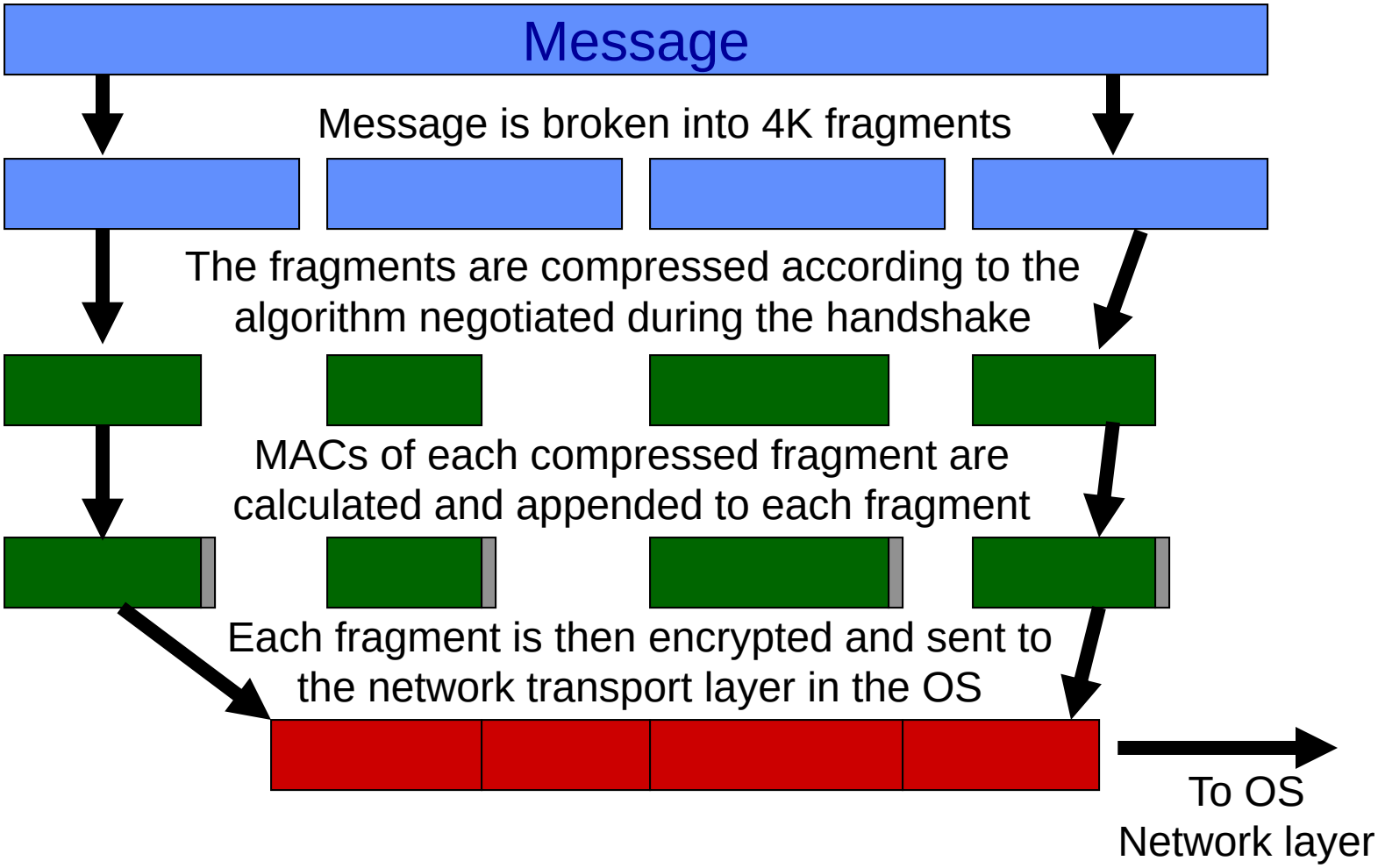
SSL Handshake Detail



SSL Handshake Detail



SSL Communication



SSL Security Features

- Protection against spoofing (forgery):
  - All encrypted packets are accompanied by a MAC (hash)
- Protection against splicing (forgery):
  - Fragments are numbered using sequence numbers in the MACs. The numbers are 64-bits long so they are unlikely to wrap (they act as a nonce as well)
  - During the handshake, hash at the end contains information for all previous messages, packets cannot be substituted between session handshakes.
- Protection against replay (forgery):
  - The handshake cannot be replayed since random numbers are selected by both sides.
- Protection against man-in-the-middle attacks (spoofing authentication):
  - Certificates are used to authenticate public keys used for encryption.

Performance Issues with SSL

- For the most part, SSL does not impose much of a performance penalty:
  - Most of the data is encrypted with a symmetric block cipher, which is fast.
  - During the handshake, the server must perform a asymmetric decryption with it's private key which is very expensive (on the order of 1000 times worse than a symmetric cipher)
  - For servers that do a lot of SSL communication, this becomes the bottleneck.
  - To help, the protocol specifies a "Resume" command so if client and server have previously agreed on a pre-master secret, they can reuse it.

Hardware SSL acceleration

- There are hardware acceleration solutions to this:
  - Specialized silicon to do public key operations (>10K SSL transactions/second). Delegating the SSL operations to dedicated hardware is called “SSL offloading”
  - For sites that also transfer a lot of traffic, symmetric accelerators are a good solution for cost and power.

Silicom cryptoaccelerator card