# Fundamentals of Information & Network Security
# ECE 471/571

Lecture #36: SSL/TLS

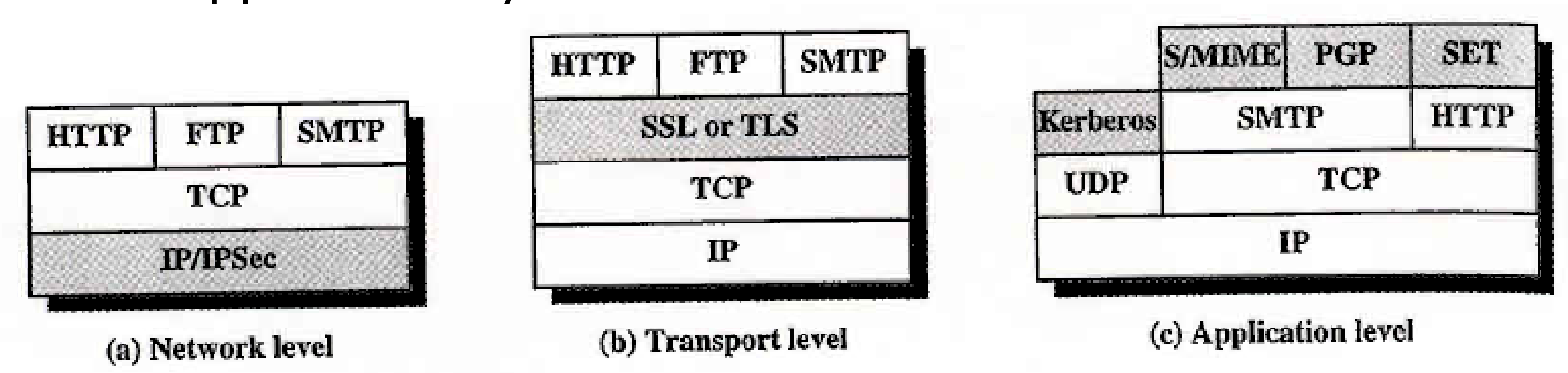
Instructor: Ming Li

Dept of Electrical and Computer Engineering

University of Arizona

# Web Security

- The emerging of E-Commerce, on-line banking, on-line purchasing, etc. requires web security.
- Approaches
    - IP layer: IPsec
    - Transport layer : SSL/TLS
        - Transparent to applications
        - Embedded in specific applications, e.g., Netscape and IE
    - Application layer



(a) Network level

(b) Transport level

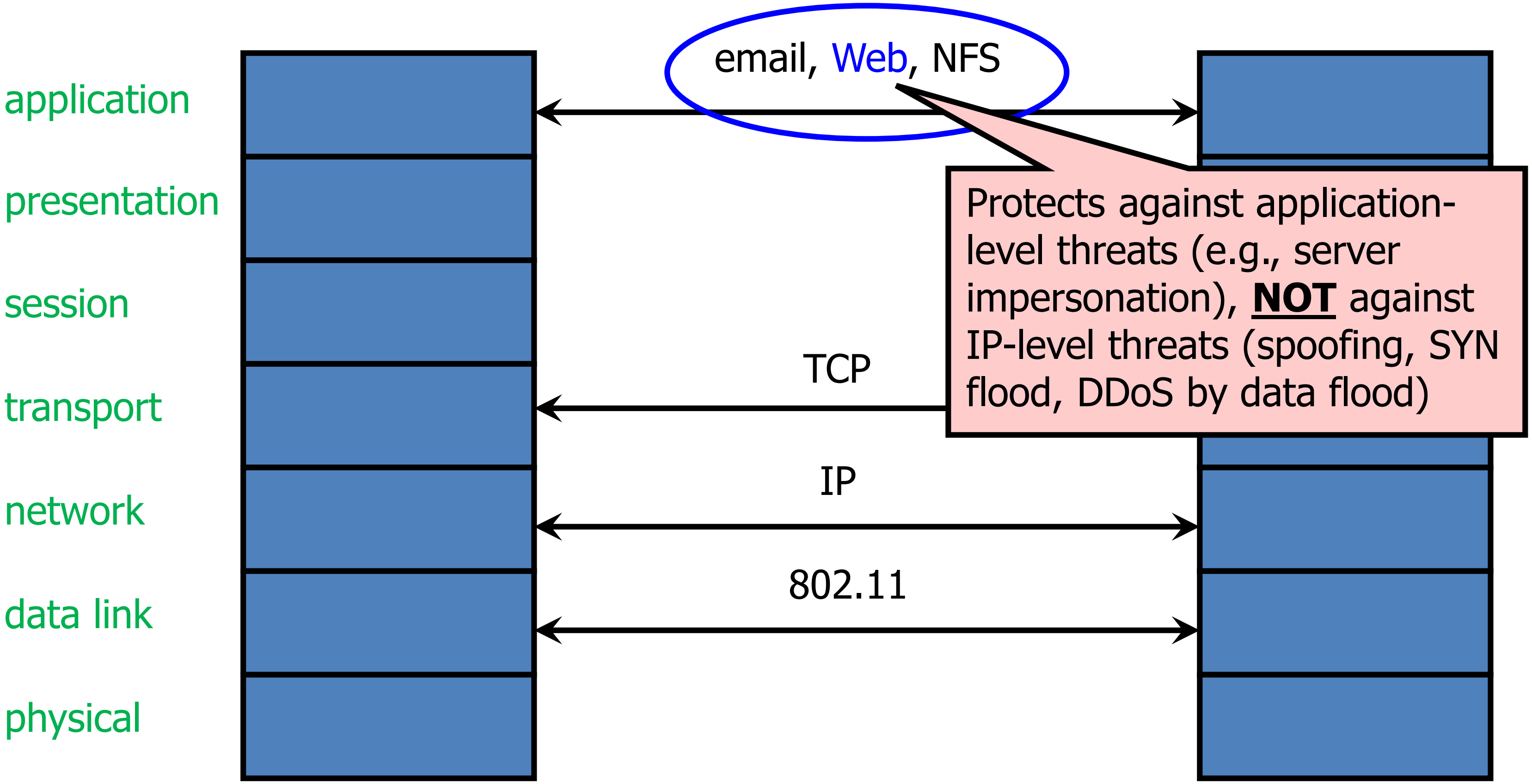
(c) Application level

**Figure 17.1** Relative Location of Security Facilities in the TCP/IP Protocol Stack

# What is SSL/TLS?

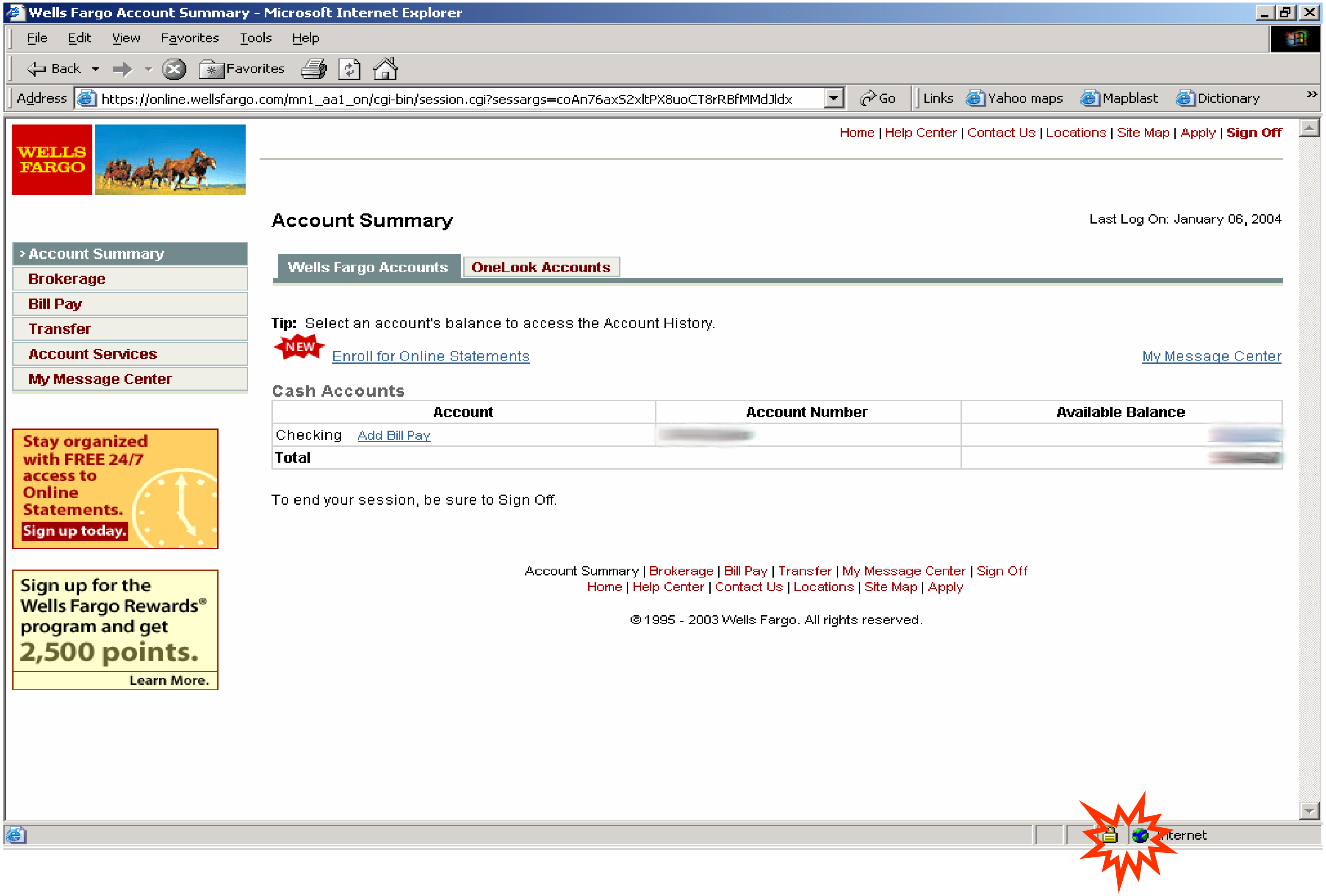
- Transport Layer Security protocol, version 1.0
  - De facto standard for Internet security
  - “The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications”
  - In practice, used to protect information transmitted between browsers and Web servers
- Based on Secure Sockets Layers protocol, ver 3.0
  - Same protocol design, different algorithms
- Deployed in nearly every Web browser
- Allow two parties to authenticate and establish a session key that is used to cryptographically protect the remainder of the session

# Application-level Protection



Protects against application-level threats (e.g., server impersonation), **NOT** against IP-level threats (spoofing, SYN flood, DDoS by data flood)

# SSL/TLS in the Real World



Wells Fargo Account Summary - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Favorites
Address https://online.wellsfargo.com/mn1_aa1_on/cgi-bin/session.cgi?sessargs=coAn76axS2xltPX8uoCT8rRBfMMdJldx
Go Links Yahoo maps Mapblast Dictionary
WELLS FARGO
Internet

Home | Help Center | Contact Us | Locations | Site Map | Apply | **Sign Off**

› Account Summary
Brokerage
Bill Pay
Transfer
Account Services
My Message Center

**Account Summary**

Last Log On: January 06, 2004

Wells Fargo Accounts | OneLook Accounts

**Tip:** Select an account's balance to access the Account History.

NEW Enroll for Online Statements

My Message Center

**Cash Accounts**

| Account | Account Number | Available Balance |
|---|---|---|
| Checking Add Bill Pay | [redacted] | [redacted] |
| **Total** | | [redacted] |

To end your session, be sure to Sign Off.

Stay organized with FREE 24/7 access to Online Statements. Sign up today.

Sign up for the Wells Fargo Rewards® program and get 2,500 points. Learn More.

Account Summary | Brokerage | Bill Pay | Transfer | My Message Center | Sign Off
Home | Help Center | Contact Us | Locations | Site Map | Apply

© 1995 - 2003 Wells Fargo. All rights reserved.

# History of the Protocol

- SSL 1.0
  - Internal Netscape design, early 1994?
  - Lost in the mists of time
- SSL 2.0
  - Published by Netscape, November 1994
  - Several weaknesses
- SSL 3.0
  - Published as an Internet draft document
  - Designed by Netscape and Paul Kocher, November 1996
- TLS 1.0
  - Internet standard based on SSL 3.0, January 1999, by IETF
  - <u>Not</u> interoperable with SSL 3.0
  - TLS uses HMAC instead of MAC; can run on any port

# TLS Basics

- TLS consists of four protocols
  - Familiar pattern for key exchange protocols
- Handshake protocol
  - Use public-key cryptography to establish a shared secret key between the client and the server

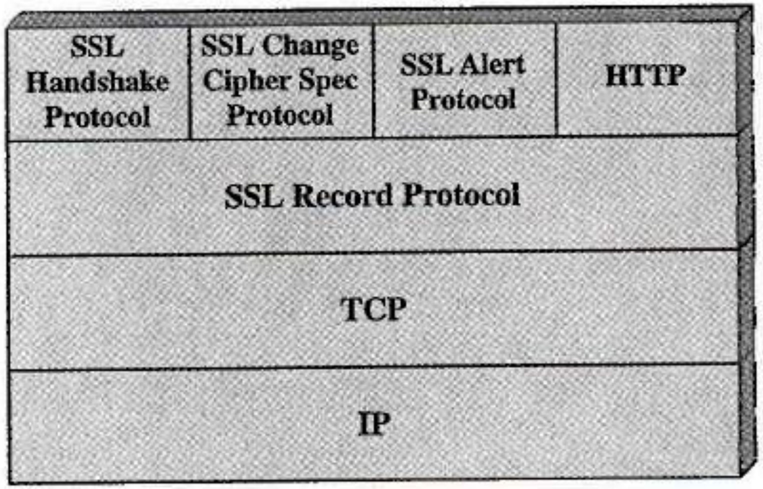
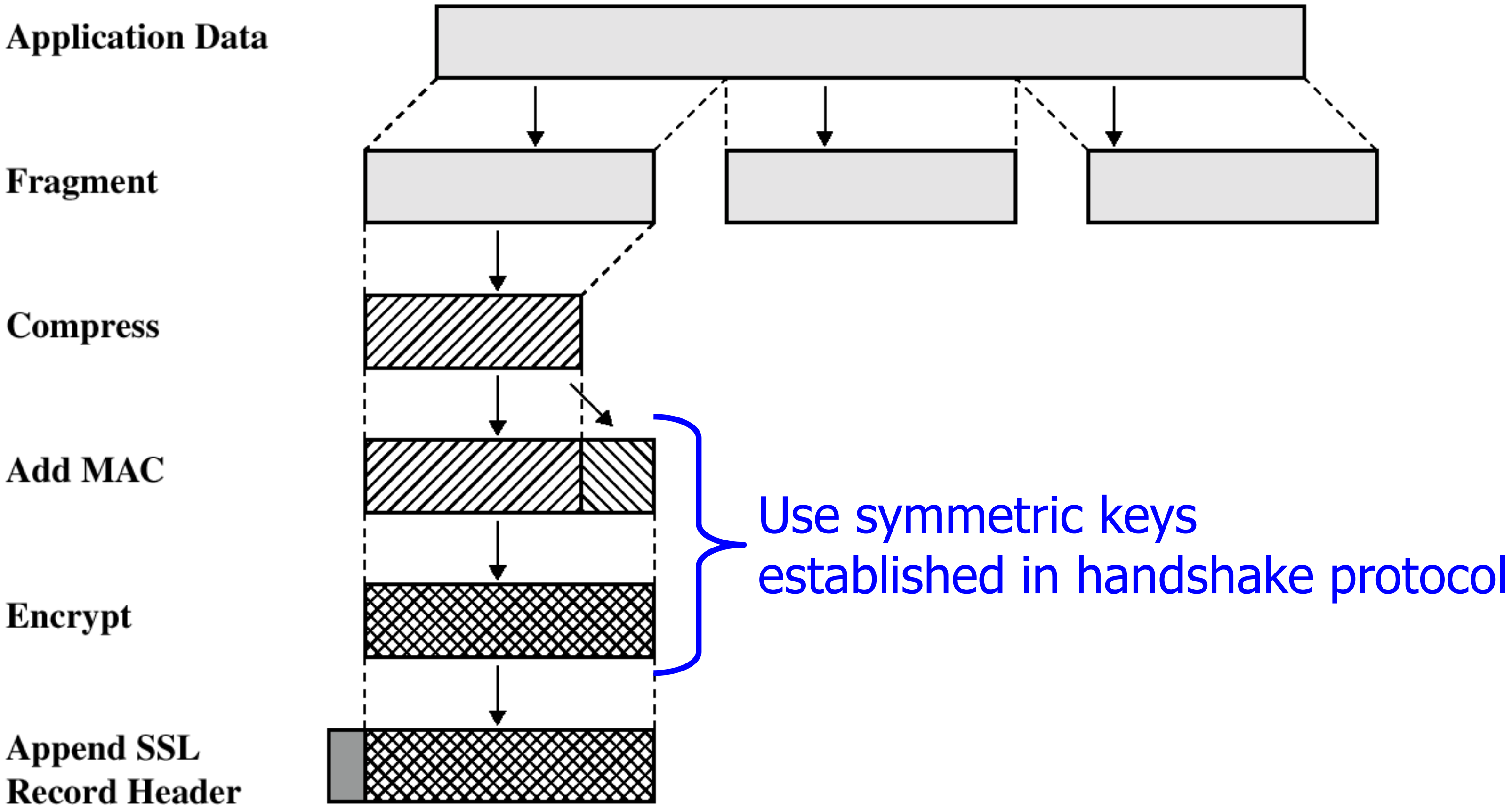


**Figure 17.2** SSL Protocol Stack

- Record protocol
  - Use the secret key established in the handshake protocol to protect communication between the client and the server
- Change cipher spec protocol
- Alert protocol

# Record Protocol

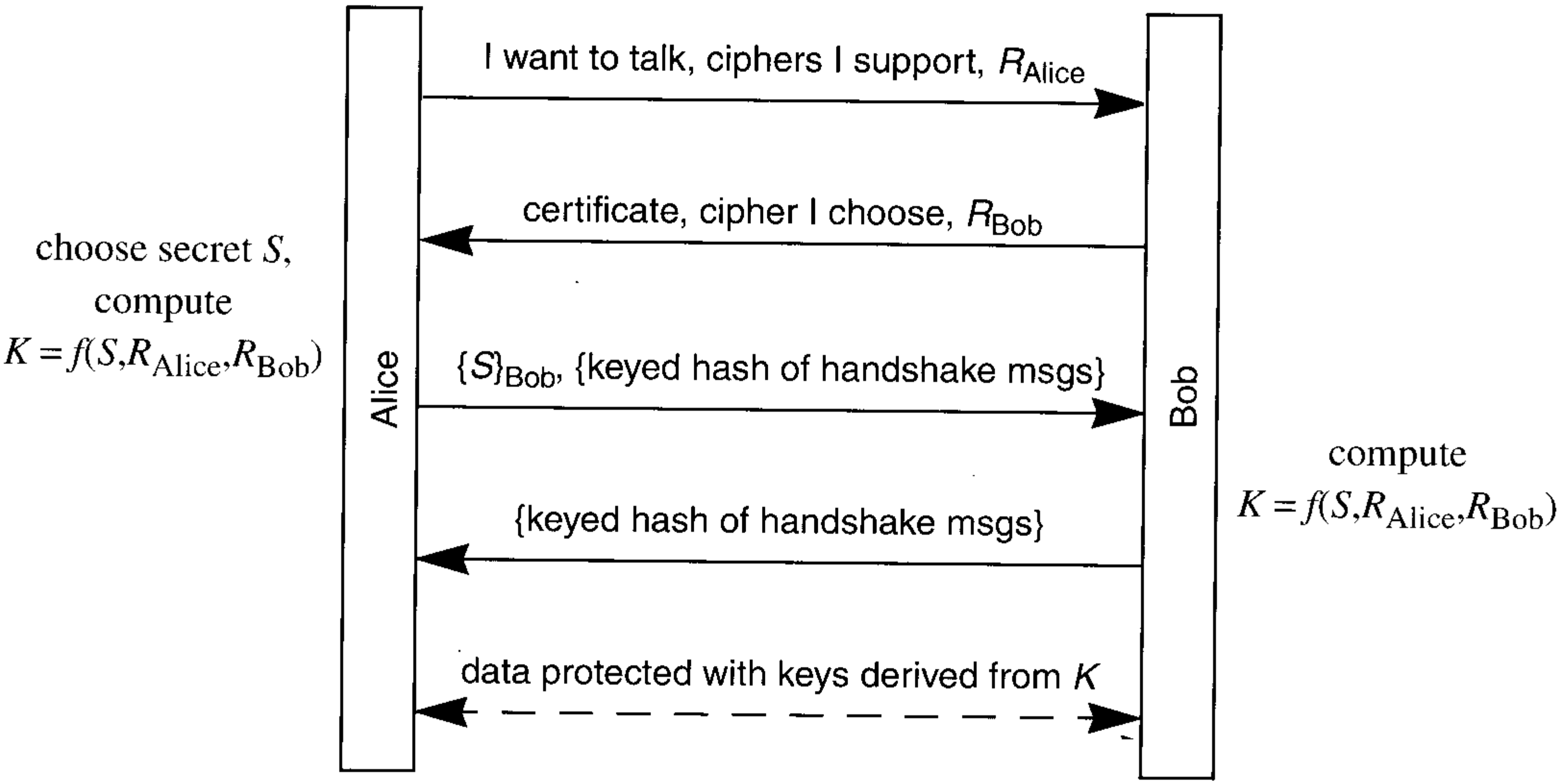
- SSL Record protocol provides two services for SSL connections
  - Confidentiality & Message integrity



# Handshake Protocol

- Two parties: client and server
- Negotiate version of the protocol and the set of cryptographic algorithms to be used
  - Interoperability between different implementations of the protocol
- Authenticate client and server (optional)
  - Use digital certificates to learn each other’s public keys and verify each other’s identity
- Use public keys to establish a shared secret
- Used before any application data transmitted

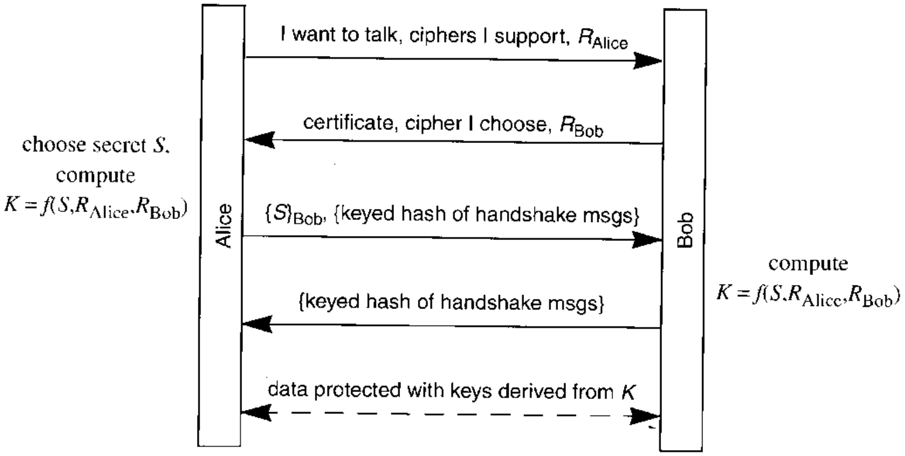
# Handshaking


I want to talk, ciphers I support, $R_{\text{Alice}}$
certificate, cipher I choose, $R_{\text{Bob}}$
choose secret $S$,
compute
$K = f(S, R_{\text{Alice}}, R_{\text{Bob}})$
Alice
$\{S\}_{\text{Bob}}$, {keyed hash of handshake msgs}
Bob
compute
$K = f(S, R_{\text{Alice}}, R_{\text{Bob}})$
{keyed hash of handshake msgs}
data protected with keys derived from $K$

**Protocol 19-1.** (simplified) SSLv3/TLS

# Computing Keys

- Pre-master key S
- Master key $K=f(S,R_{Alice},R_{bob})$


choose secret $S$,
compute
$K = f(S,R_{Alice},R_{Bob})$
Alice
I want to talk, ciphers I support, $R_{Alice}$
certificate, cipher I choose, $R_{Bob}$
$\{S\}_{Bob}$, {keyed hash of handshake msgs}
{keyed hash of handshake msgs}
data protected with keys derived from $K$
Bob
compute
$K = f(S,R_{Alice},R_{Bob})$

**Protocol 19-1.** (simplified) SSLv3/TLS

- 6 session keys (for each direction)
  - encryption key
  - integrity-protection key
  - IV
  - hash results of K, $R_{Alice}$, and $R_{Bob.}$

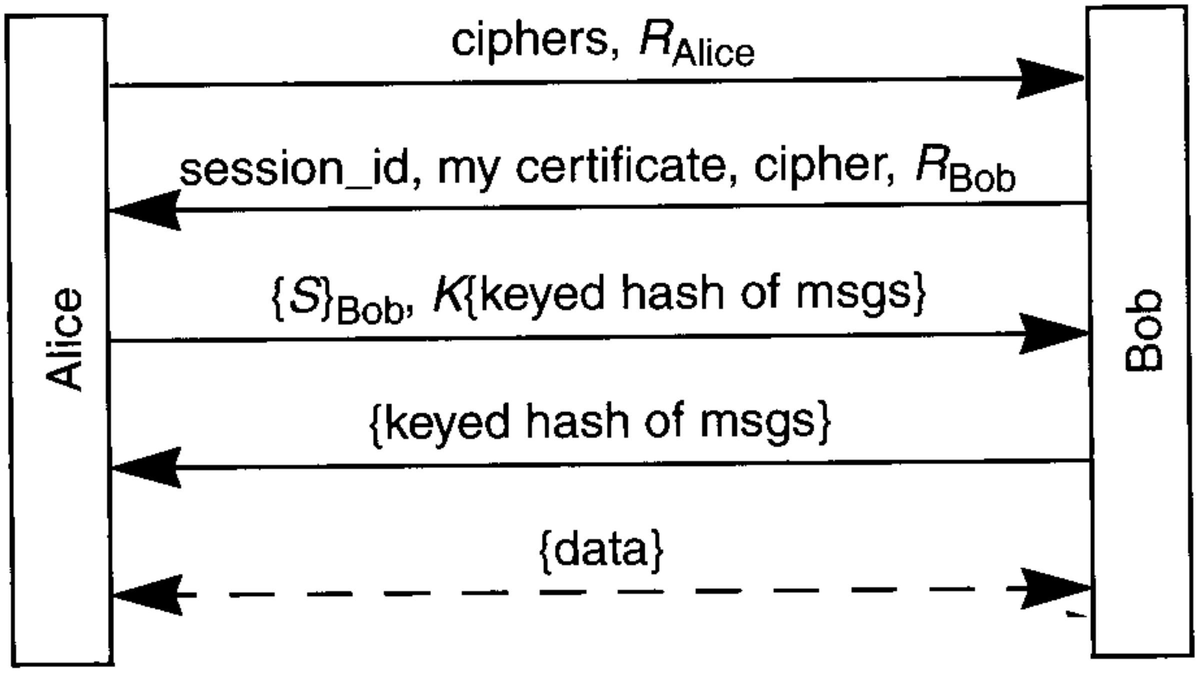
# Connection and Session

- Connection: A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationship. The connections are transient. Every connection is associated with one session.

- Session: A SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Session defines a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

# Session Initiation

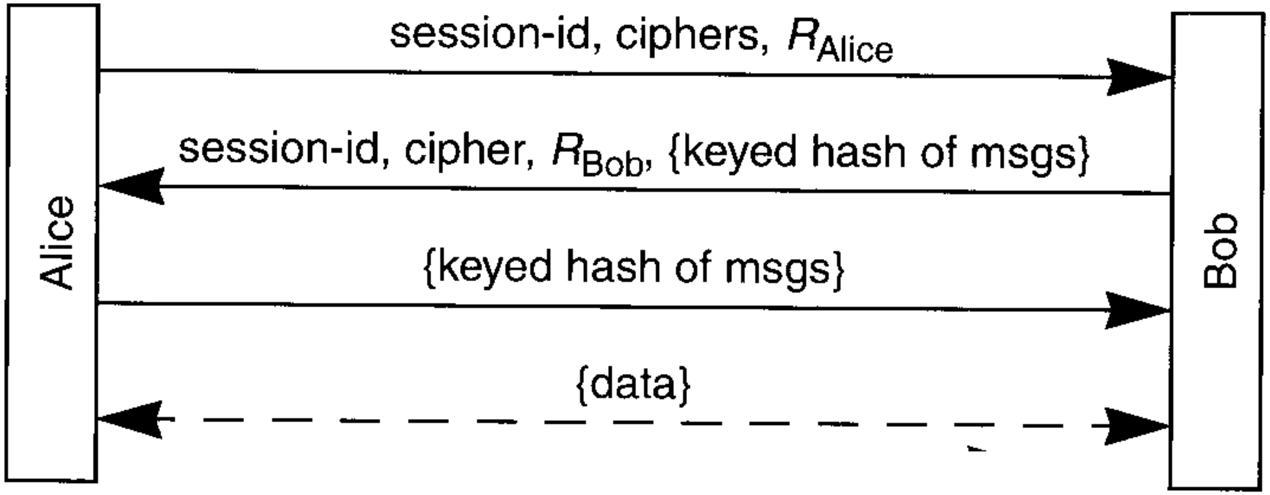
- If session resumption is allowed, the server sends the client a *session_id* in the 2nd message and stores (*session_id , master key*).


ciphers, $R_{\text{Alice}}$
session_id, my certificate, cipher, $R_{\text{Bob}}$
$\{S\}_{\text{Bob}}$, $K$\{keyed hash of msgs\}
\{keyed hash of msgs\}
\{data\}
Alice
Bob

**Protocol 19-2.** Session initiation if no previous state

# Session Resumption

- When resuming a session, the client present the *session_id* in the first message so they can use the same master secret and skip the public key portion of the handshake.


Alice
Bob
session-id, ciphers, $R_{\text{Alice}}$
session-id, cipher, $R_{\text{Bob}}$, {keyed hash of msgs}
{keyed hash of msgs}
{data}

**Protocol 19-3.** Session resumption if both sides remember session-id

# Client Authentication

- Normally the clients send name/password to the server as application data
- The server has the option to send a “certificate” request in message 2 of the handshaking.

# Reading Assignment

- Preview
  - [Kaufman] Chapters 23 (firewalls)