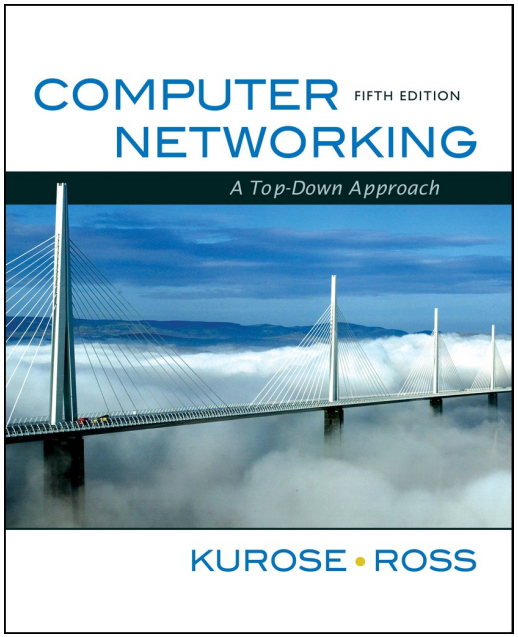
Capítulo 8
Seguridad en Redes
Conexiones TCP Seguras: SSL



Basado en:
Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

Capítulo 8 contenidos

8.1 ¿Qué es la seguridad en la red?

8.2 Principios de criptografía

8.3 Integridad de mensajes

8.4 Dando seguridad a e-mail

8.5 Conexiones TCP seguras: SSL

8.6 Seguridad en capa de Red: IPsec

8.7 Seguridad en redes locales inalámbricas

8.8 Cortafuegos y Sistemas de detección de intrusión (IDS)

SSL: Secure Sockets Layer
Sockets seguros (Capa 4)

- Protocolo de seguridad ampliamente difundido
  - Usado en la mayoría de los navegadores y servidores web
  - https
  - Usado en transferencias de comercio electrónico.
- Diseñado originalmente por Netscape en 1993
- Existen variantes:
  - TLS: transport layer security, RFC 2246
- Provee
  - Confidencialidad
  - Integridad
  - Autenticación
- Objetivos originales:
  - Permitir el comercio electrónico en la Web
  - Encriptación (especialmente de números de tarjetas de créditos)
  - Autenticación de servidores Web
  - Opcionalmente autenticación de clientes
  - Minimizar riesgos al hacer negocios con nuevos clientes
- Disponible para toda conexión TCP
  - Interfaz de socket segura

SSL y TCP/IP

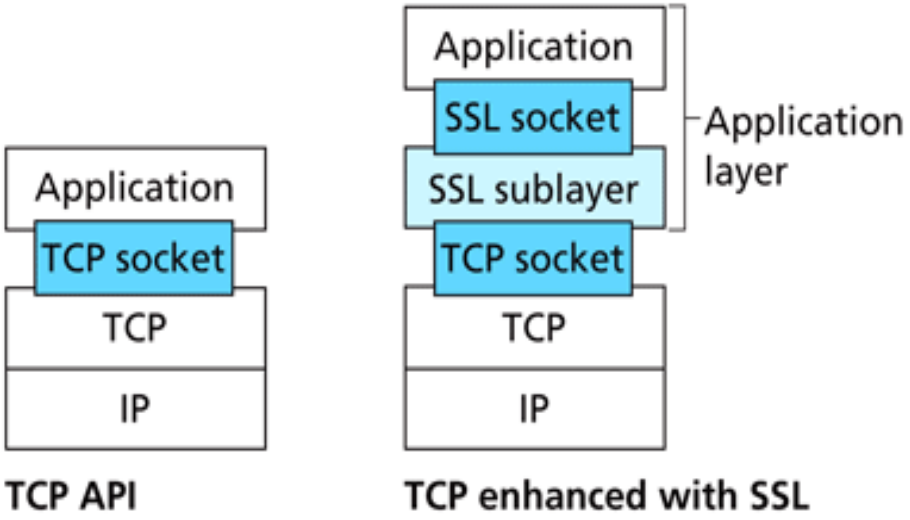
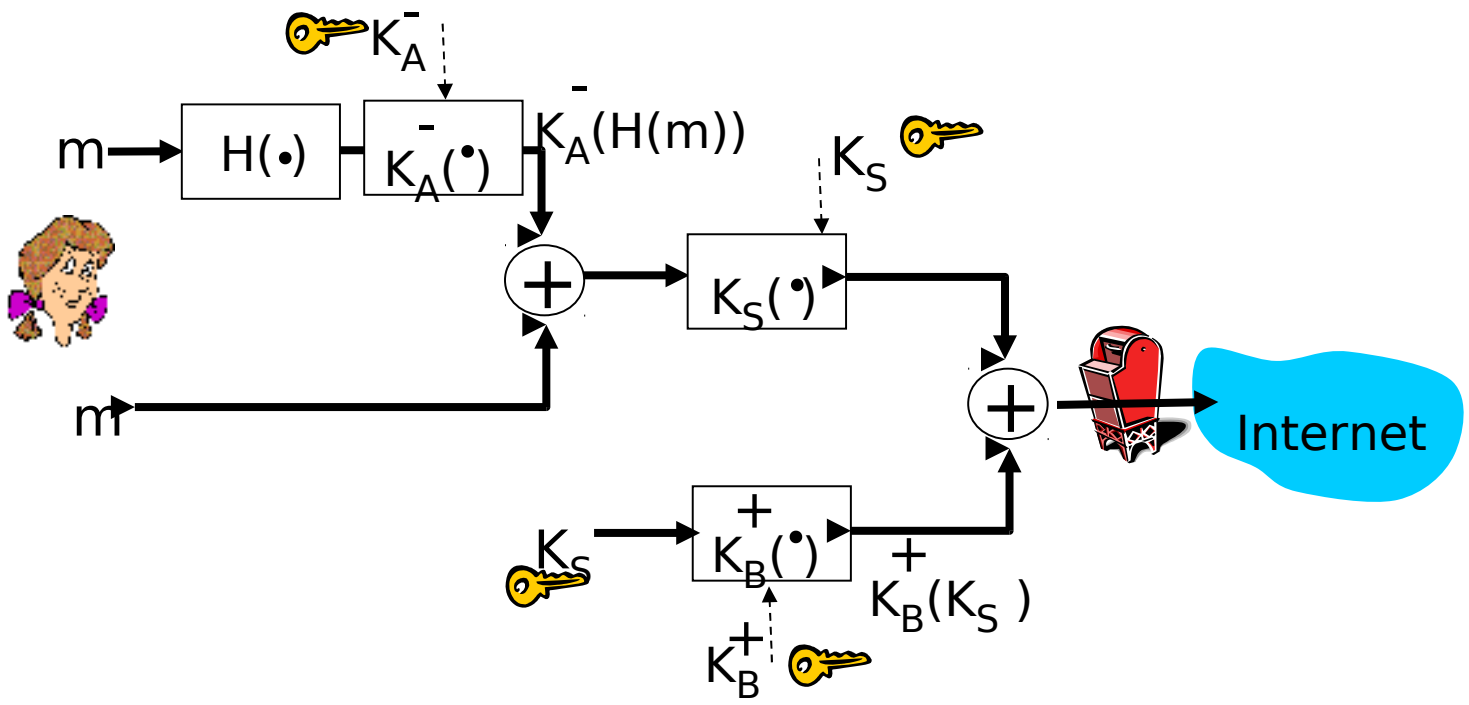


Figure 8.27 ♦ Although SSL technically resides in the application layer, from the developer's perspective it is a transport-layer protocol.

- SSL provee una interfaz de programación de aplicaciones (API) para desarrollar aplicaciones
- Existen Bibliotecas SSL en C y clases SSL para Java y C++

Se podría hacer algo similar a PGP:



- Pero queremos enviar flujos de byte y datos interactivos
- Queremos un conjunto de claves por toda la conexión.
- Queremos intercambio de certificados como aparte del protocolo en fase de establecimiento de conexión (handshake)

Cifrado simétrico más común en SSL

- DES – Data Encryption Standard: bloques
- 3DES – Triple strength: bloques
- RC2 – Rivest Cipher 2: bloques
- RC4 – Rivest Cipher 4: flujo (stream)

Cifrado de clave pública

- RSA

Cifrado SSL

- Herramientas de Cifrado
  - Algoritmos de clave pública
  - Algoritmos de encriptación simetrica
  - Algoritmos MAC (Message Authentication Code)
- SSL permite varios mecanismos de cifrado
- Negociación: Cliente y servidor deben acordar mecanismos de cifrado
- Cliente ofrece opciones; el servidor toma una.

SSL: Handshake (1)

Propósito

1. Autenticar el servidor
2. Negociación: acordar algoritmos de cifrado.
3. Establecer claves
4. Autenticatión del cliente (opcional)

SSL: Handshake (2)

1. El cliente envía una lista de algoritmos que soporta, junto con un número de unicidad del cliente (para evitar replicación de mensajes).
2. Servidor elije algoritmo desde lista; envía: su elección + certificado + número de unicidad del servidor
3. Cliente verifica certificado, extrae clave pública del servidor, genera “pre_master_secret”, lo encripta con clave pública de servidor, lo envía al servidor
4. Cliente y servidor calculan independientemente las claves de encriptación y MAC a partir de pre_master_secret y números de unicidad (ambos comparten estas cuatro claves)
5. Cliente envía un MAC de todos los mensajes de handshake
6. Servidor envía un a MAC de todos los mensajes de handshake

SSL: Handshaking (3)

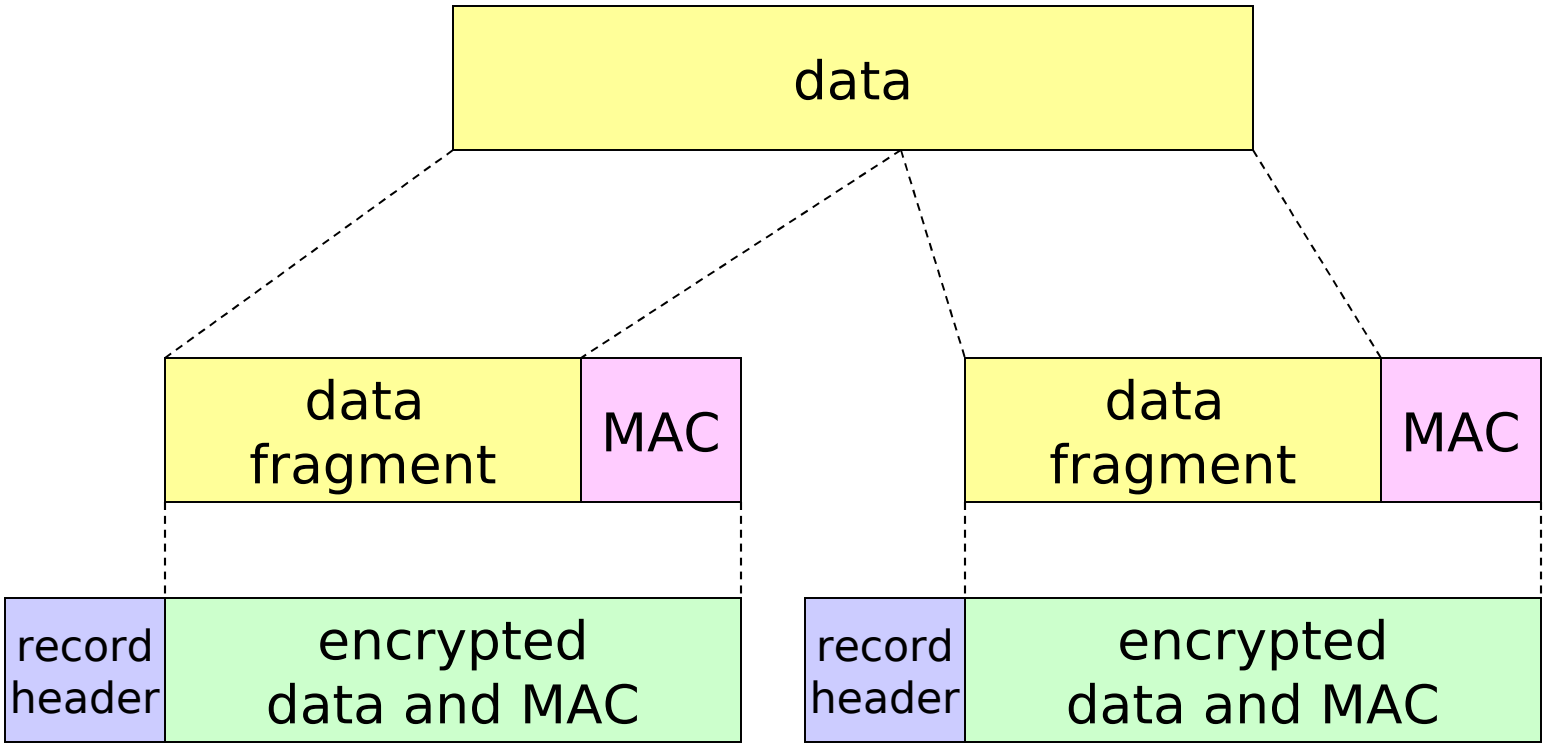
Los últimos 2 pasos protejen el handshake de ser observados

- Cliente típicamente ofrece un rango de algoritmos de cifrado, algunos robustos y otros débiles.
- “Man-in-the middle” podría borrar los robustos de la lista
- Los últimos 2 pasos lo evitan
  - Los últimos dos mensajes son encriptados.

SSL: Handshaking (4)

- Por qué usar dos números de unicidad aleatorios?
- Supongamos el intruso observa todos los mensajes entre Alicia y Bob.
- Más tarde, intruso establece una conexión TCP con Bob y envía exactamente la misma secuencia.
  - Bob (Amazon) piensa que Alicia hace dos compras separadas de lo mismo.
  - Solución: Bob envía diferentes números aleatorios cada vez en cada conexión. Así las claves de cifrado serán distintas ambas veces.
  - Mensajes del intruso fallarán los chequeos de integridad de Bob.

SSL: Registro del Protocolo

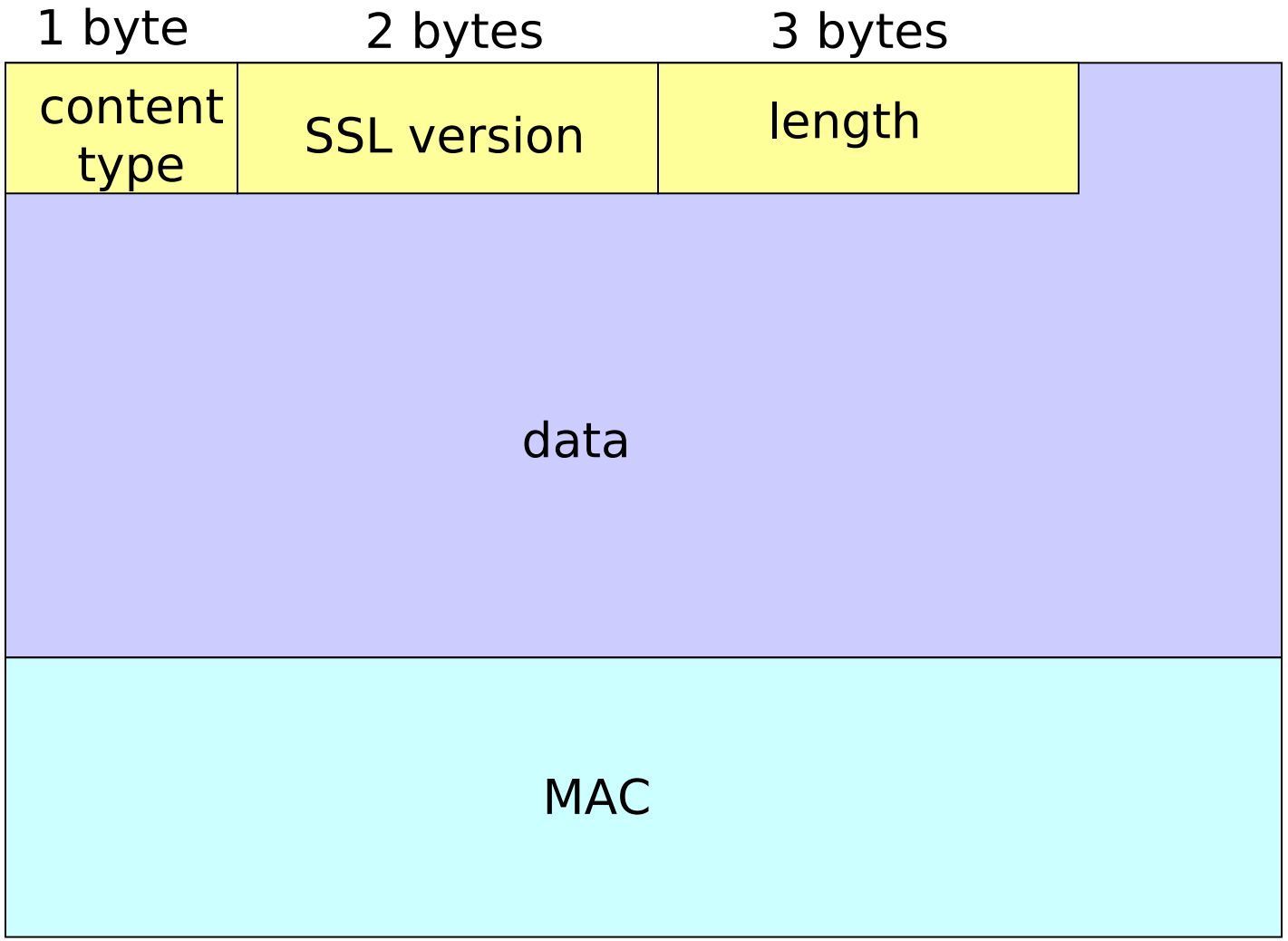


Record header: contiene: tipo, versión, largo

MAC: incluye número de secuencia, clave MAC M_x

Fragment: cada fragmento SSL máx 2^{14} bytes (~16 Kbytes)

SSL: Formato del registro



Data y MAC van cifradas (algoritmo simétrico)

Conexión Real

handshake: ClientHello

handshake: ServerHello

handshake: Certificate

handshake: ServerHelloDone

handshake: ClientKeyExchange

ChangeCipherSpec

handshake: Finished

ChangeCipherSpec

handshake: Finished

application_data

application_data

Alert: warning, close_notify

Desde aquí todo va Encriptado con Claves de sesión

Sigue TCP Fin intercambio

Derivación de Claves

- Con los números de unicidad del cliente y del servidor, y una “pre-master secret” se ingresan a un generador de número pseudo aleatorios.
  - Se genera así el “Master Secret”
- El “Master secret” y nuevos números de unicidad ingresados a otro generador de números aleatorios crea varias claves
- Las claves son:
  - Clave MAC del cliente
  - Clave MAC del servidor
  - Clave de encriptación del cliente
  - Clave de encriptación del servidor
  - Vector de inicialización del cliente (IV)
  - Vector de inicialización del servidor (IV)

Capítulo 8 contenidos

8.1 ¿Qué es la seguridad en la red?

8.2 Principios de criptografía

8.3 Integridad de mensajes

8.4 Dando seguridad a e-mail

8.5 Conexiones TCP seguras: SSL

8.6 Seguridad en capa de Red: IPsec

8.7 Seguridad en redes locales inalámbricas

8.8 Cortafuegos y Sistemas de detección de intrusión (IDS)