

TLS and HTTPS

CSE 461 Section 3

...But First: Helpful tools

- `ifconfig` – See your host's network interfaces
- `dig`, `whois` – Lookup ip by host
- `ipcalc -h`, `nslookup` – Lookup host by ip

HTTP Wireshark Example

TLS Fundamentals


- “Transport Layer Security” protocol
- Standard protocol for encrypting Internet traffic
- Previously known as SSL (Secure Sockets Layer), which has been around since 1994
- TLS is a slightly modified version of SSL version 3
- Used for HTTPS (HTTP Secure) traffic
- Supported by nearly every web browser


<https://mail.google.com/mail/u/0/#inbox>

mail.google.com ✕

Identity verified


Permissions **Connection**

 The identity of this website has been verified by Google Internet Authority G2 but does not have public audit records. [Certificate information](#)

 Your connection to mail.google.com is encrypted with 256-bit encryption.

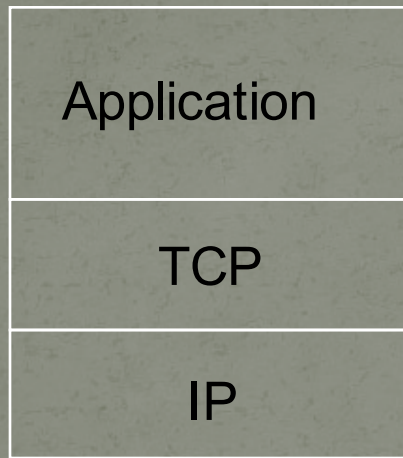
The connection uses TLS 1.2.

The connection is encrypted and authenticated using CHACHA20_POLY1305 and uses ECDHE_ECDSA as the key exchange mechanism.

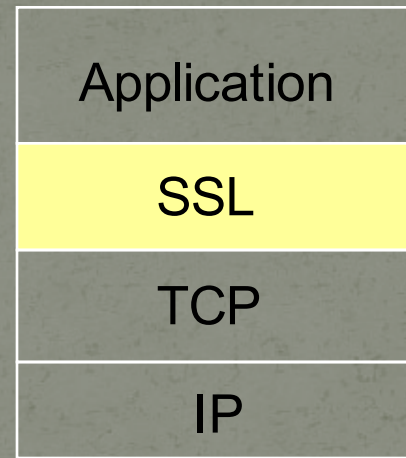
 **Site information**
You first visited this site on Nov 14, 2014.

[What do these mean?](#)

SSL and TCP/IP



normal application



application with SSL

- ❖ SSL provides application programming interface (API) to applications
- ❖ C and Java SSL libraries/classes readily available

Purposes for TLS

- Provides encrypted TCP connection
- Data integrity
- End-point authentication

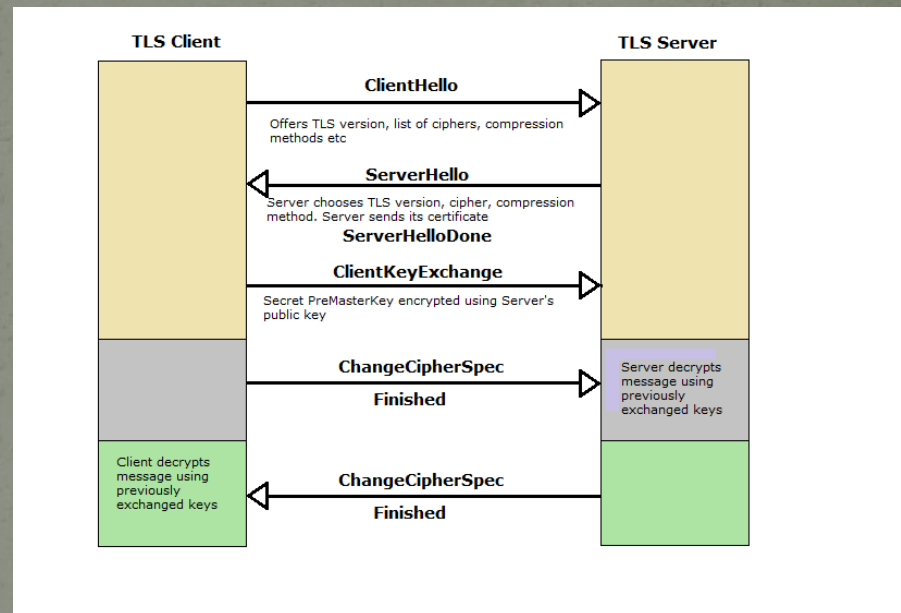
TLS and CONNECT

- HTTP CONNECT is used to establish a two-way connection “tunnel” between two parties
- After this, a “triple handshake” is performed over the tunnel
- After the handshake, the two parties can communicate securely
- We’ll take a closer look at this handshake

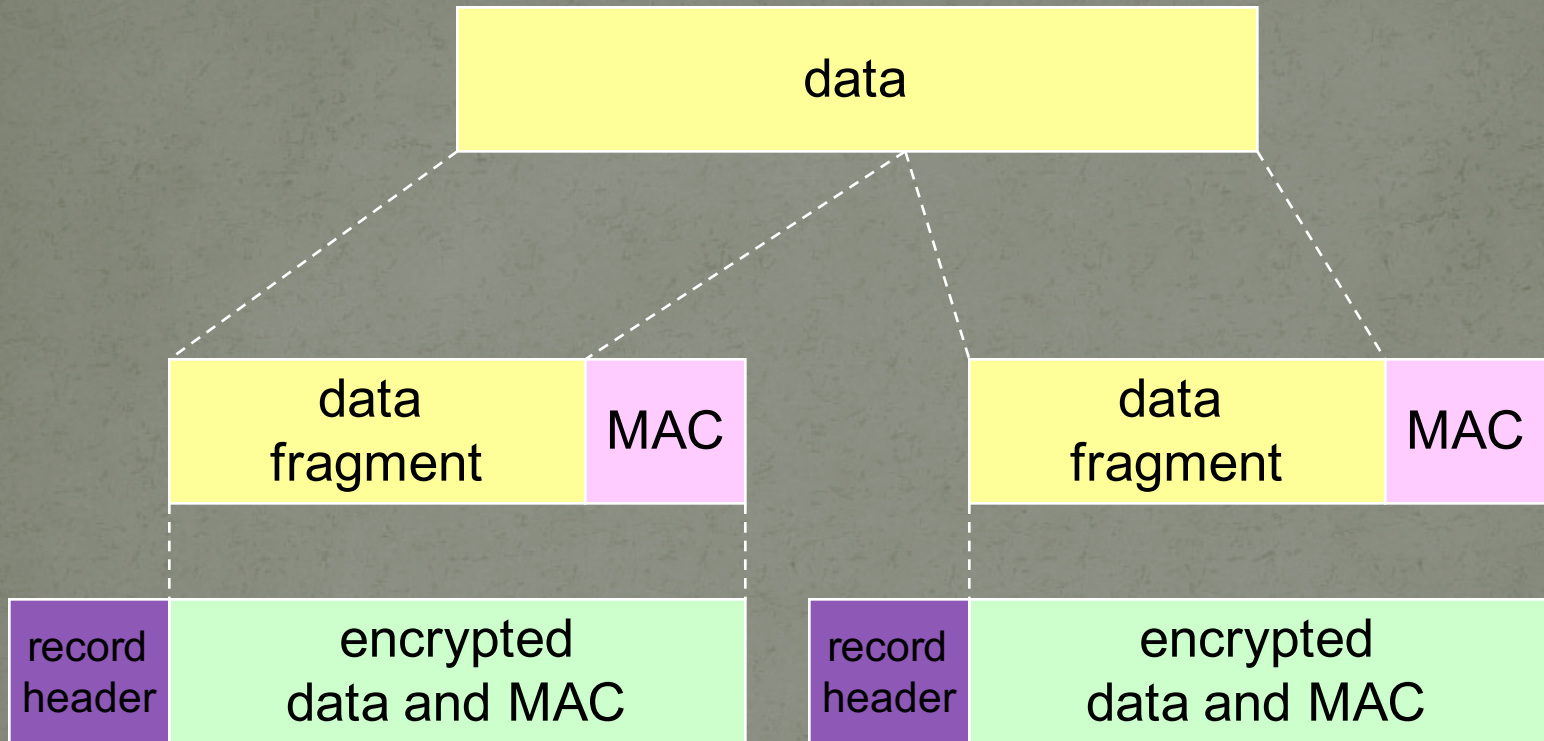
TLS Handshake Protocol (Rough Details)

Purpose

1. server authentication
2. negotiation: agree on crypto algorithms
3. establish keys
4. client authentication (optional)



SSL record protocol



record header: content type; version; length

MAC: includes sequence number, MAC key M_x

fragment: each SSL fragment 2^{14} bytes (~16 Kbytes)

HTTPS Wireshark Example
