Intro to IT Security

CS306C—Fall 2022

Prof. Antonio R. Nicolosi

Antonio.Nicolosi@stevens.edu

1870

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

SSL/TLS + SSH

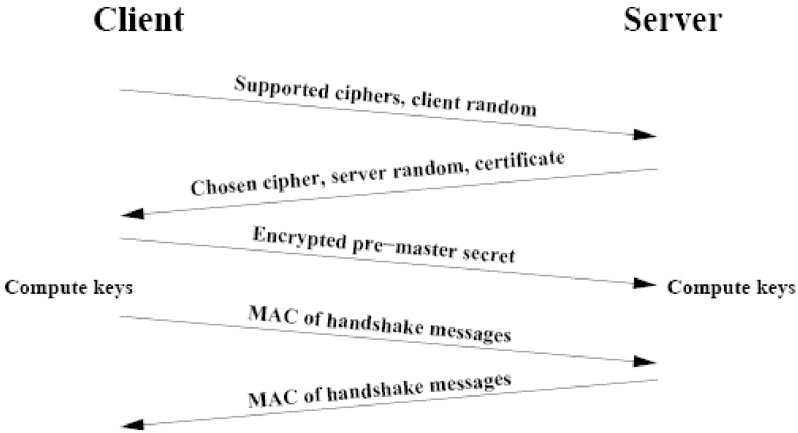
SSL/TLS Overview

- Secure Sockets Layer (SSL) and Transport Layer Security (TLS)
- SSL offers security for application-level protocols (*e.g.* HTTP)
- Authentication of server to client
- Optional authentication of client to server
- SSL protects secrecy of communication
- SSL protects integrity of communication

Purpose in more detail

- Authentication based on certification authorities (CAs)
  - Trusted third party with well-known public key
  - Certifies who owns a given public key (domain name and real name of company)
  - Example: DigiCert (owner of former Verisign brand)
- What SSL Does Not Address
  - Privacy
  - Traffic analysis
  - Trust management

Overview of SSL Handshake


Client
Server
Supported ciphers, client random
Chosen cipher, server random, certificate
Encrypted pre−master secret
Compute keys
Compute keys
MAC of handshake messages
MAC of handshake messages

From "SSL and TLS" by Eric Rescorla

Ciphersuites: Negotiating ciphers

- Server authentication algorithm (RSA, DSS)
- Key exchange algorithm (RSA, DHE)
- Symmetric cipher for secrecy (3DES, AES)
- MAC (HMAC-SHA1, HMAC-SHA256)

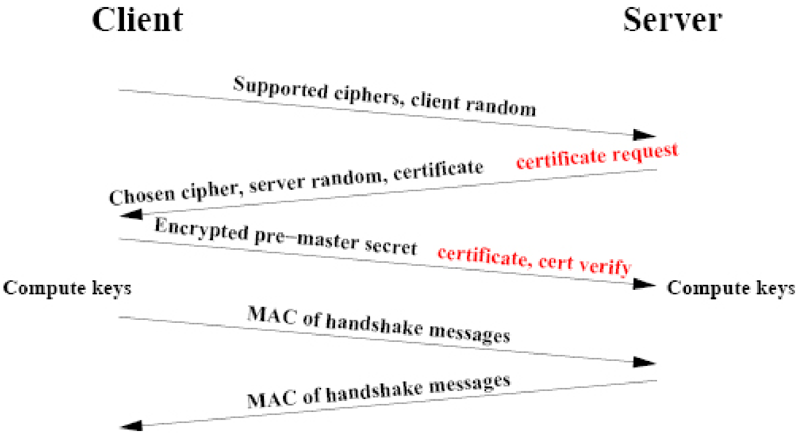
Simplified View of SSL Handshake

- Client and server negotiate on cipher selection.
- Cooperatively establish session keys.
- Use session keys for secure communication.

Client Authentication Handshake

- Server requests that client send its certificate.
- Client signs a signed digest of the handshake messages.

SSL Client Certificate

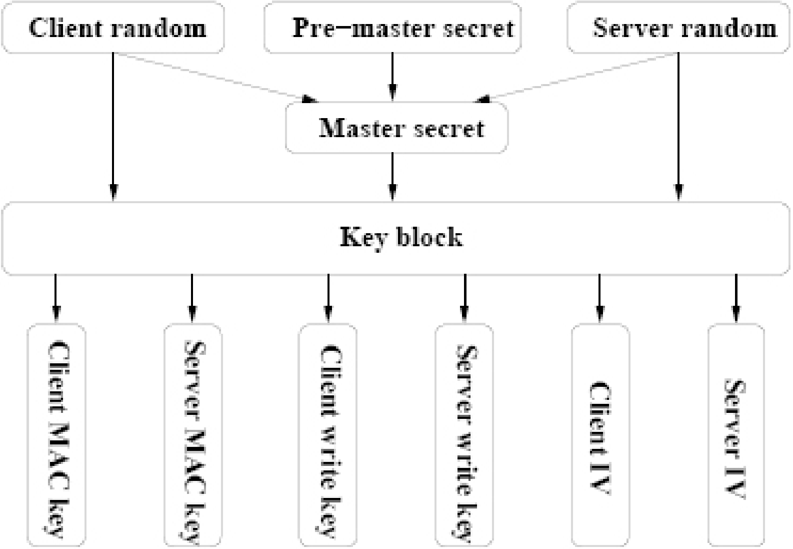

Client
Server
Supported ciphers, client random
certificate request
Chosen cipher, server random, certificate
Encrypted pre−master secret
certificate, cert verify
Compute keys
Compute keys
MAC of handshake messages
MAC of handshake messages

From “SSL and TLS” by Eric Rescorla

Establishing a Session Key

- Server and client both contribute randomness.
- Client sends server a “pre-master secret” encrypted with server’s public key.
- Use randomness and pre-master secret to create session keys:
  - Client MAC
  - Server MAC
  - ClientWrite
  - ServerWrite
  - Client IV
  - Server IV

Establishing a Session Key



From "SSL and TLS" by Eric Rescorla

What does a CA-issued Certificate Mean?

- No one knows exactly.
- That a public key belongs to someone authorized to represent a hostname?
- That a public key belongs to someone who is associated in some way with a hostname?
- That a public key belongs to someone who has lots of paper trails associated to a company related to a hostname?
- That the CA has no liability?

How to Get a Certificate

- Pay DigiCert ($300+)
- Get “Doing Business As” (DBA) license from city call ($20)
  - No on-line check for name conflicts . . . can I do business as Microsoft?
- Letterhead from company ($0)
- Notarized document (need driver’s license) ($0)
- Conclusions:
  - Easy to get a fraudulent certificate (many CAs)
  - Maybe not so easy to avoid prosecution afterwards

Man in the Middle Attacks

- Man in the middle attack foils user:
  - Attacker emulates server when talking to client
  - Attacker emulates client when talking to server
  - Attacker passes most messages through unmodified
  - Attacker substitutes own public key for client’s & server’s
  - Attacker records secret data, or tampers to cause damage

SSH Overview

- Widely-used secure remote login program
- MACs/encrypts all data sent over the network
  - Version 2 of protocol basically gets this right
  - Open to man in the middle attack on first server access
- Often sends password at start of session
  - Gets sent encrypted in a single TCP packet
- Assuming crypto secure (& no MiM), how to attack?

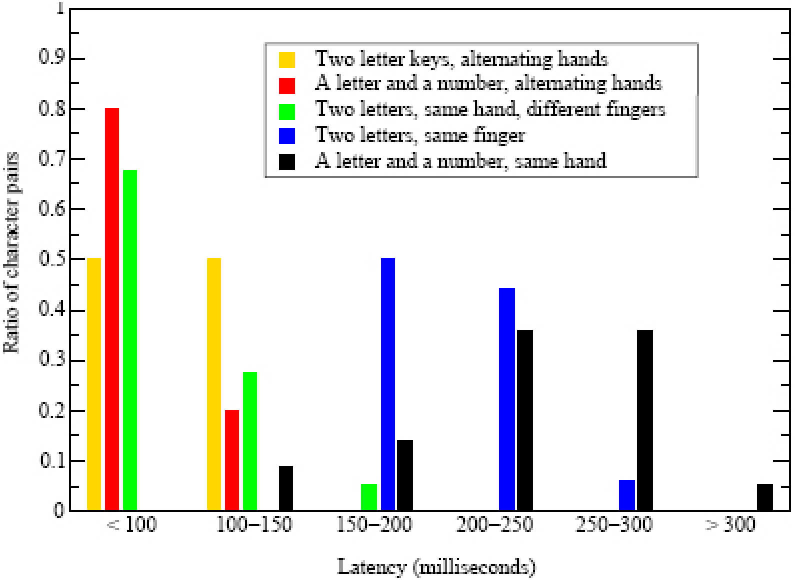
Packet Size

- Transmitted packets rounded to multiple of 8 bytes
  - Version 1 even had exact packet-size in the clear
- Can tell if user’s password is less than 7 chars
  - Password sent in one packet of initial exchange
- Why do we care?
  - Might tell you which account to try to crack

Inter-Keystroke Timings

- Each character typed causes a packet to be sent
  - Typical inter-character times 10–300 msec
  - Typical network round-trip time 10 of msec
  - Can get very accurate timing information by eavesdropping
- What can you learn from this?
  - Some character sequences harder to type than others
  - E.g., v–b is much slower to type than v–o
  - In general, characters with different hands faster
  - Two characters typed with same finger are much slower
  - Digits, special chars also slower
- Idea: Use timing to learn about passwords

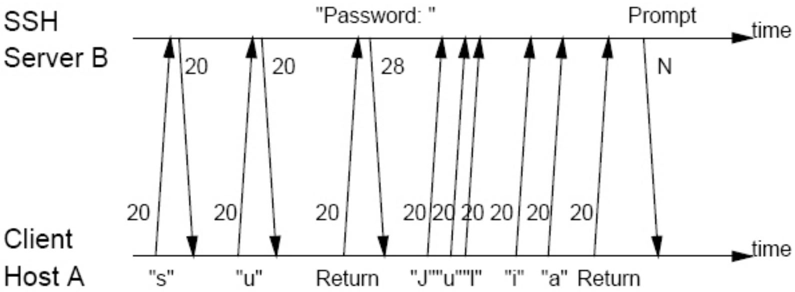
Character Latency



How to Know Password is being Typed

- Traffic signature
  - E.g., echo turned off when password typed
- Multi-user attack
  - E.g., run ps on machine to see when victim runs pgp
- Nested ssh attack
  - See remote host open SSH connection to another host

Example: su Command



- “Password” prompt — 28 char packet
- Echo turned off for password, no return packets

How to Work around the Problem

- Send dummy packets when in echo mode
  - Foils traffic signature detection of passwords
- Adding random delays to packets?
  - Latencies in 100s of msec, so need big random delays
  - Can still get info by averaging many sessions
  - Delay might get seriously annoying
- Constant bit-rate traffic
  - Practical for one session over a modem