

# SSL/TLS

---

Slides by Vitaly Shmatikov  
University of Texas  
updated slightly by smo IHK

slide 1

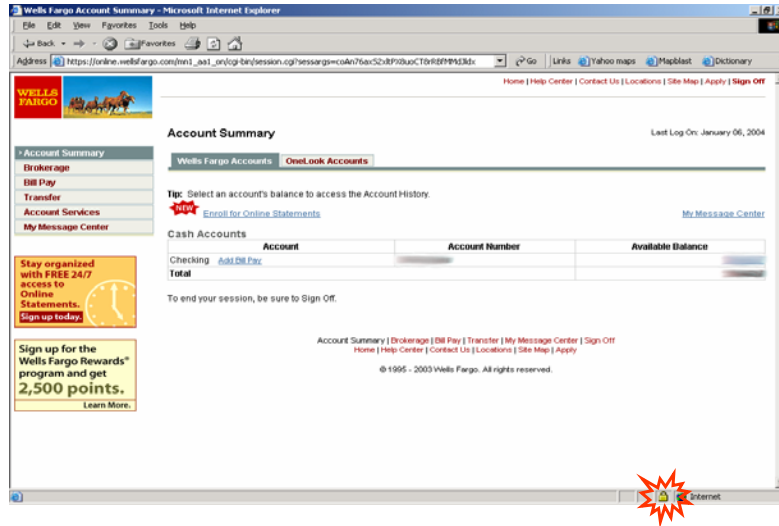
## What is SSL / TLS?

---

- ◆ 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

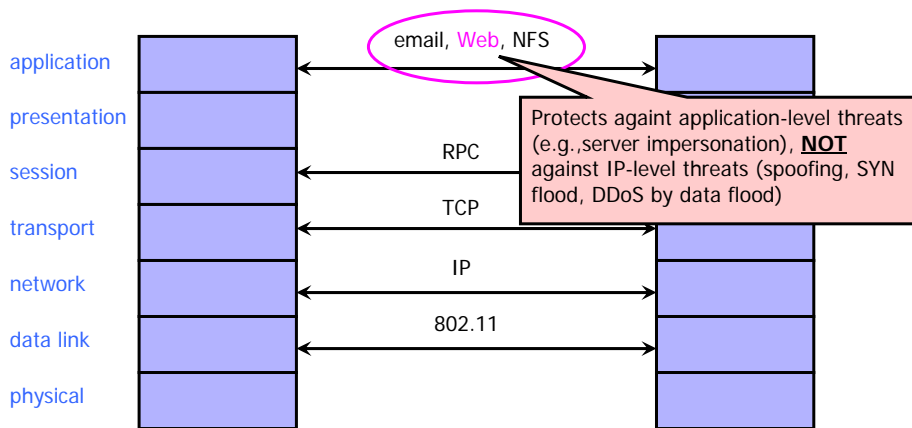
slide 2

# SSL / TLS in the Real World



slide 3

# Application-Level Protection



slide 4

## 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
  - Designed by Netscape and Paul Kocher, November 1996
- ◆ TLS 1.0
  - Internet standard based on SSL 3.0, January 1999
  - Not interoperable with SSL 3.0
    - TLS uses HMAC instead of MAC; can run on any port
  - TLS 1.2 RFC 5246 august 2008

slide 5

## “Request for Comments”

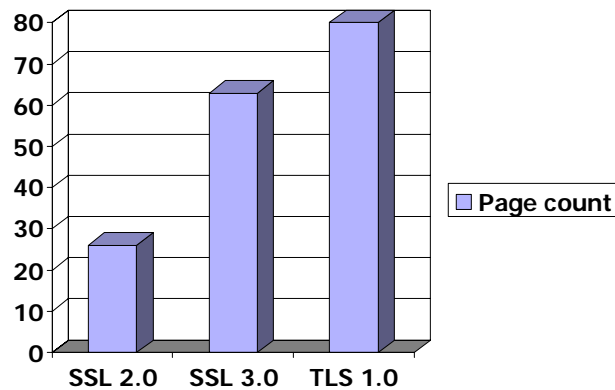
---

- ◆ Network protocols are usually disseminated in the form of an RFC
- ◆ TLS version 1.0 is described in RFC 2246
  - version 1.1 RFC 4346 v1.2 RFC 5346
- ◆ Intended to be a self-contained definition of the protocol
  - Describes the protocol in sufficient detail for readers who will be implementing it and those who will be doing protocol analysis
  - Mixture of informal prose and pseudo-code

slide 6

## Evolution of the SSL/TLS RFC

---



slide 7

## TLS Basics

---

- ◆ TLS consists of **two** 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
- ◆ Record protocol
  - Use the secret key established in the handshake protocol to protect communication between the client and the server
- ◆ We will focus on the handshake protocol

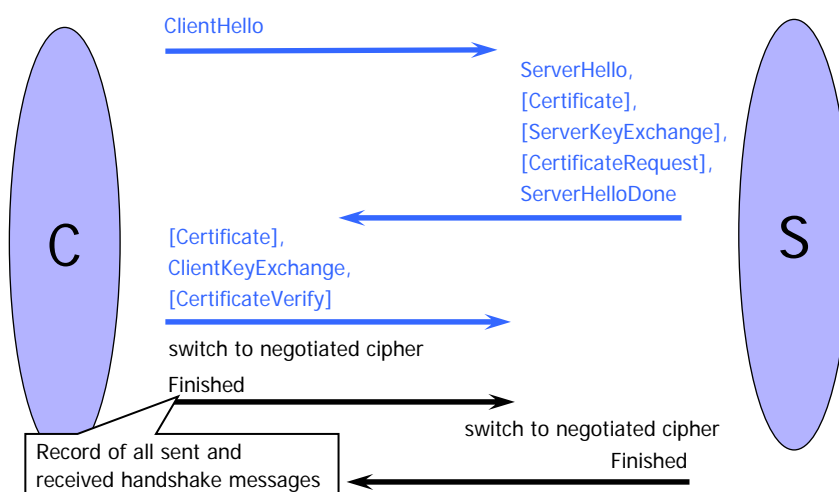
slide 8

## TLS 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

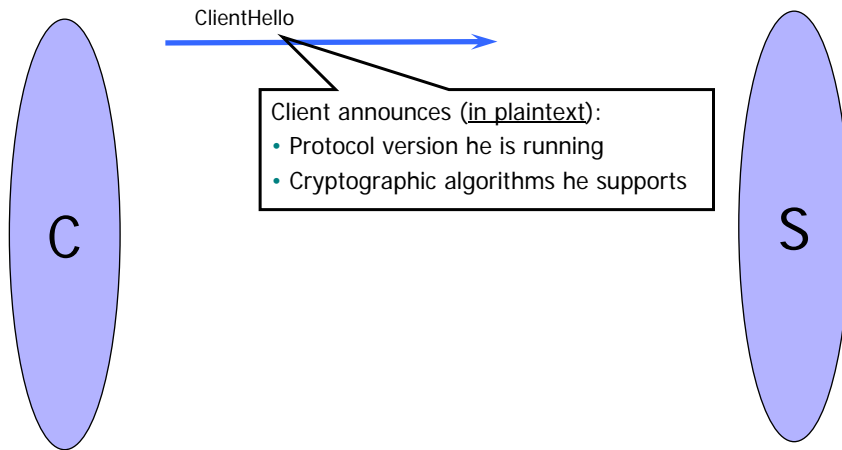
slide 9

## Handshake Protocol Structure



slide 10

## ClientHello



slide 11

## ClientHello (RFC)

```
struct {  
    ProtocolVersion client_version;  
    Random random;  
    SessionID session_id;  
    CipherSuite cipher_suites;  
    CompressionMethod compression_methods;  
} ClientHello
```

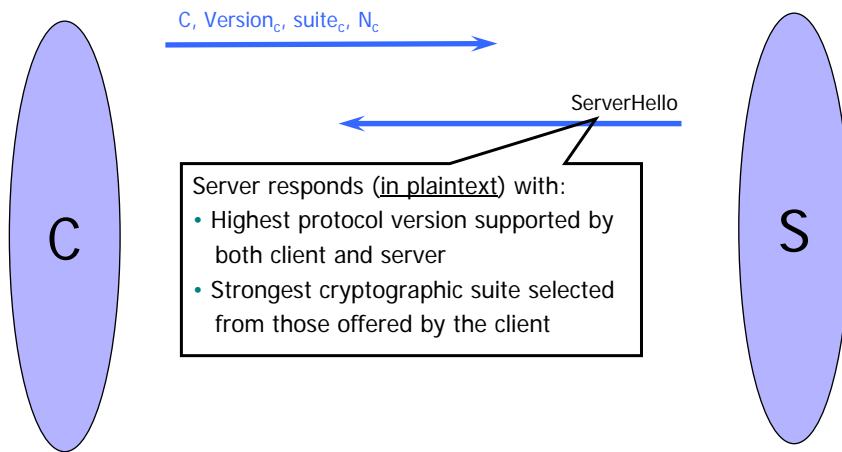
Highest version of the protocol supported by the client

Session id (if the client wants to resume an old session)

Set of cryptographic algorithms supported by the client (e.g., RSA or Diffie-Hellman)

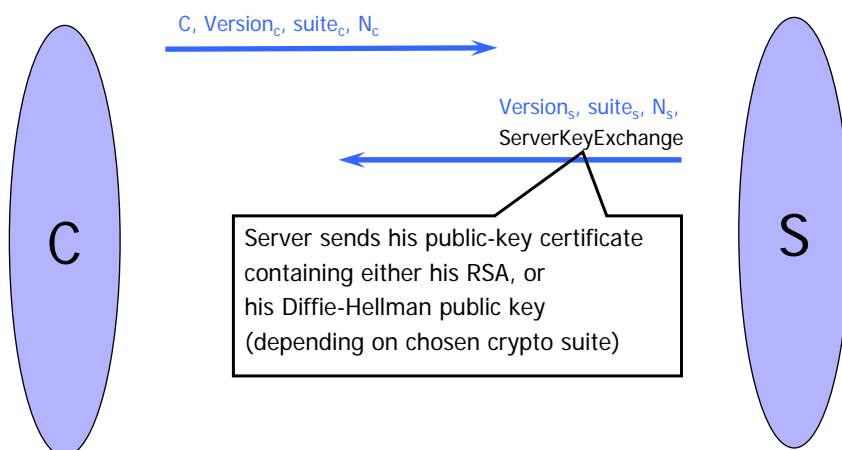
slide 12

## ServerHello



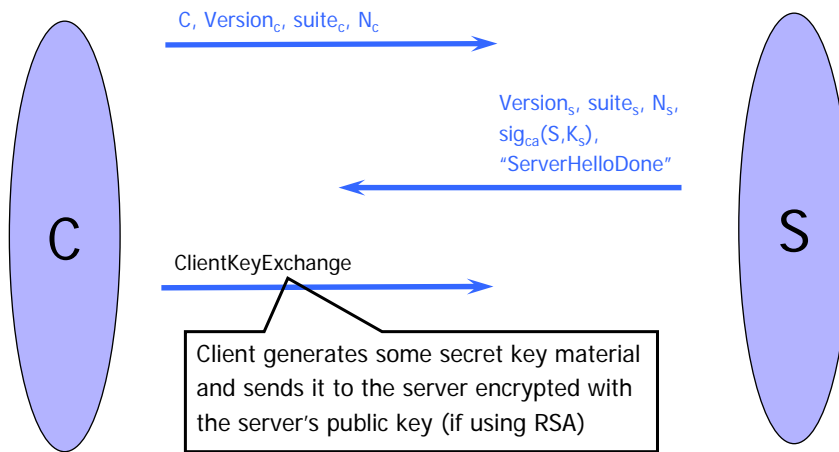
slide 13

## ServerKeyExchange



slide 14

## ClientKeyExchange



slide 15

## ClientKeyExchange (RFC)

```
struct {  
    select (KeyExchangeAlgorithm) {  
        case rsa: EncryptedPreMasterSecret;  
        case diffie_hellman: ClientDiffieHellmanPublic;  
    } exchange_keys  
} ClientKeyExchange
```

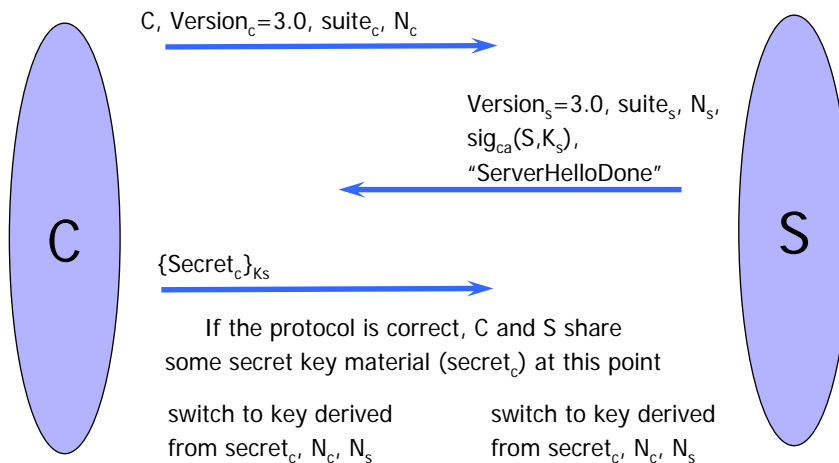
```
struct {  
    ProtocolVersion client_version;  
    opaque random[46];  
} PreMasterSecret
```

Random bits from which symmetric keys will be derived (by hashing them with nonces)

slide 16

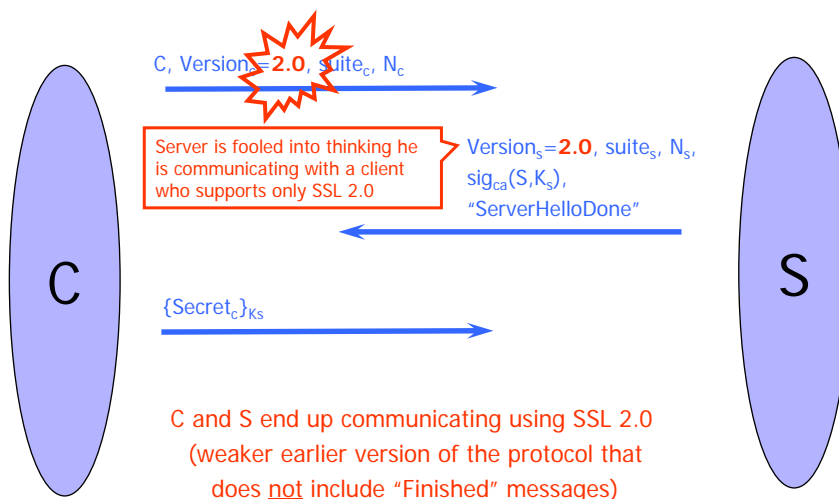


## "Core" SSL 3.0 Handshake



slide 17

## Version Rollback Attack



slide 18

## SSL 2.0 Weaknesses (Fixed in 3.0)

---

- ◆ Cipher suite preferences are not authenticated
  - “Cipher suite rollback” attack is possible
- ◆ Weak MAC construction
- ◆ SSL 2.0 uses padding when computing MAC in block cipher modes, but padding length field is not authenticated
  - Attacker can delete bytes from the end of messages
- ◆ MAC hash uses only 40 bits in export mode
- ◆ No support for certificate chains or non-RSA algorithms, no handshake while session is open

slide 19

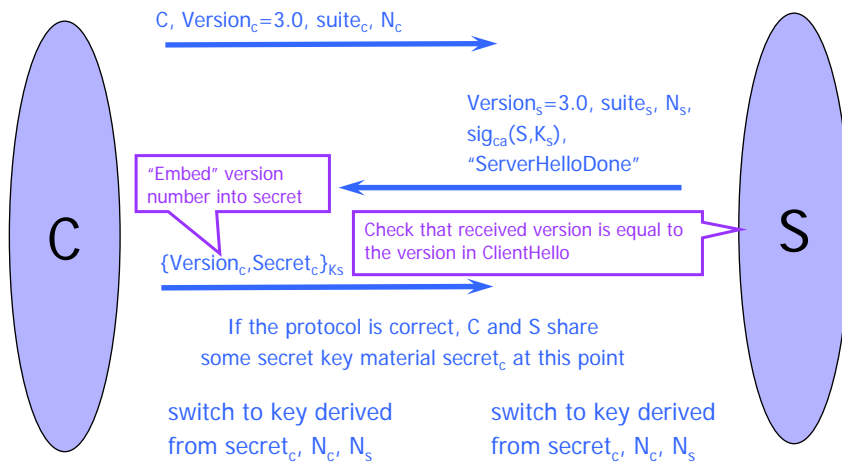
## “Chosen-Protocol” Attacks

---

- ◆ Why do people release new versions of security protocols? Because the old version got broken!
- ◆ New version must be **backward-compatible**
  - Not everybody upgrades right away
- ◆ Attacker can fool someone into using the old, broken version and exploit known vulnerability
  - Similar: fool victim into using weak crypto algorithms
- ◆ Defense is hard: must authenticate version early
- ◆ Many protocols had “version rollback” attacks
  - SSL, SSH, GSM (cell phones)

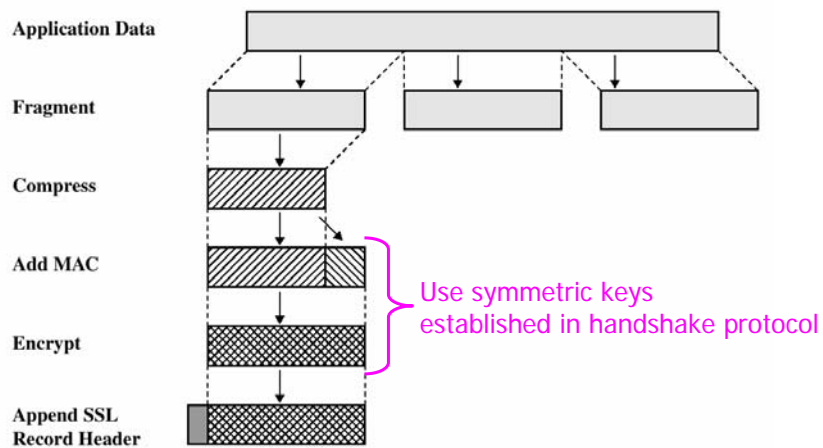
slide 20

## Version Check in SSL 3.0



slide 21

## SSL/TLS Record Protection



slide 22