



Web Security

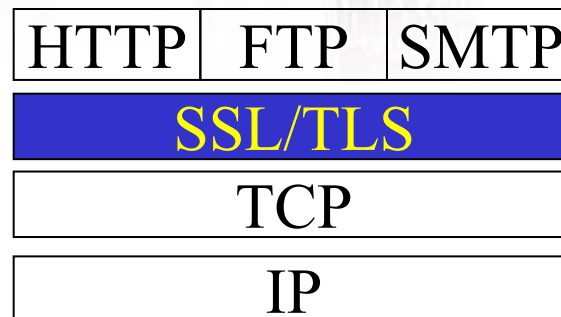
SSL/TLS and Certificates

Chun-Jen (James) Chung

Arizona State University

What is SSL/TLS?

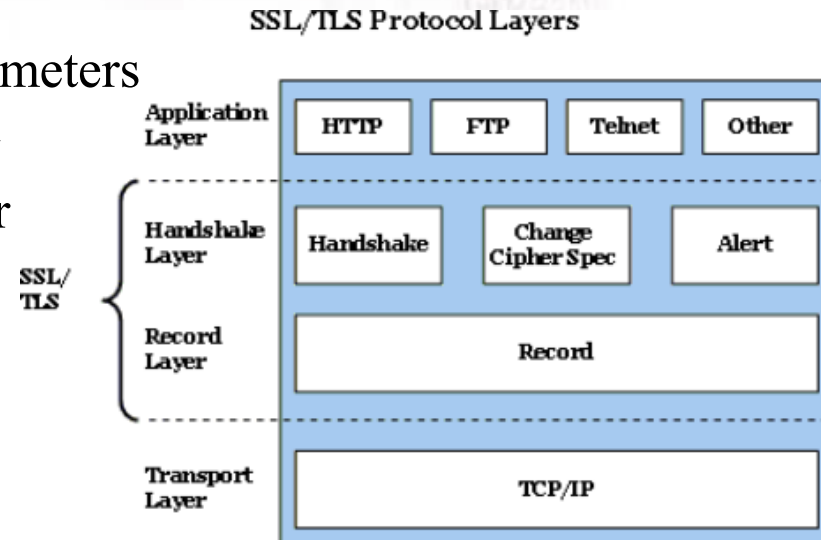
- SSL – Secure Socket Layer
- TLS – Transport Layer Security
- Both provide a secure transport connection (data encryption and authentication) between applications and servers.
- SSL version 3.0 has been implemented in many web browsers and widely used on the Internet.
- TLS can be viewed as SSL v3.1
 - Same protocol design, different algorithm



At session (or above) layer

SSL/TLS Components

- Handshake Protocol
 - negotiation of security algorithms and parameters
 - Use public-key cryptography to *establish a shared secret key* between client and server
 - server authentication and optionally client authentication
- Record Protocol
 - Fragmentation/compression/encryption
 - Using secret key to provide message authentication and integrity protection
- Alert Protocol
 - error messages (fatal alerts and warnings)
- Change Cipher Spec Protocol
 - a single message that indicates the end of the SSL handshake



Sessions and Connections

- **Connection:**
 - A peer-to-peer relationships in the transport layer. Every connection is associated with one session.
- **Session:**
 - An association between a client and a server created by the handshake protocol.
 - Define a set of cryptographic security parameters, which can be shared among multiple connections.
 - Avoid the expensive negotiation of new security parameters for each connection.

SSL Statefulness

- Multiple secure connections in a session
- Connections of the same session share the session state
- Current **operating** state for read and write (receive and send)
- **Pending** read and write states created during Handshake Protocol

Session State

- session identifier
 - arbitrary byte sequence chosen by the server to identify the session
- peer certificate
 - X509 certificate of the peer; may be null
- compression method
- cipher spec
 - encryption (null, DES, 3DES) and MAC (MD5, SHA-1) algorithm used, and cryptographic attributes (e.g., hash size, IV size, ...)
- master secret
 - 48-byte secret shared between the client and the server
- is resumable
 - a flag indicating whether the session can be used to initiate new connections
- connection states

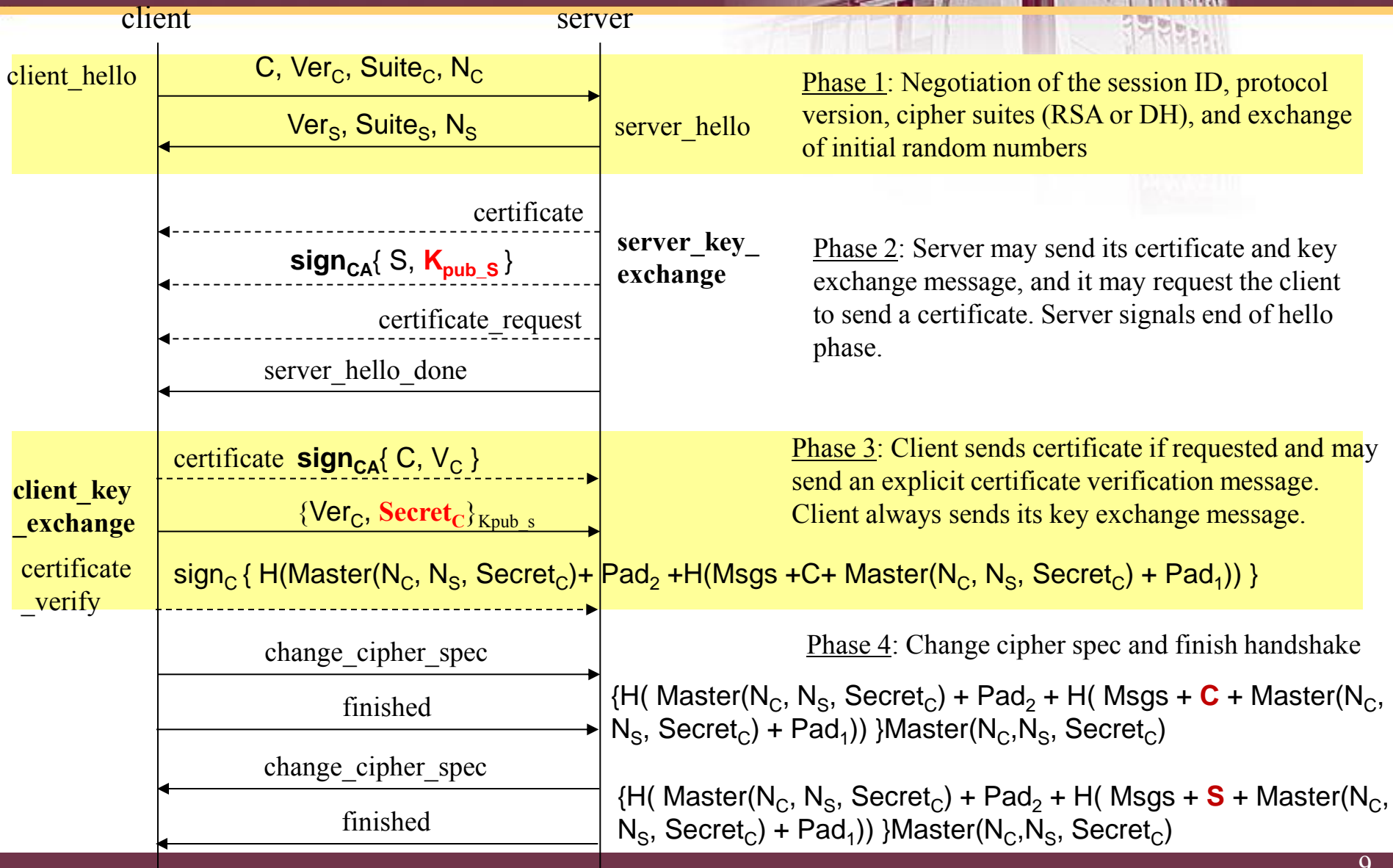
Connection State

- server and client random
 - random byte sequences chosen by the server and the client for every connection
- server/client write MAC secret
 - secret key used in MAC operations on data sent by the server/client
- server/client write key
 - secret encryption key for data encrypted by the server/client
- initialization vectors
 - an IV is maintained for each encryption key if DES CBC mode is used
- sending and receiving sequence numbers
 - sequence numbers are 64 bits long
 - reset to zero after each Change Cipher Spec message

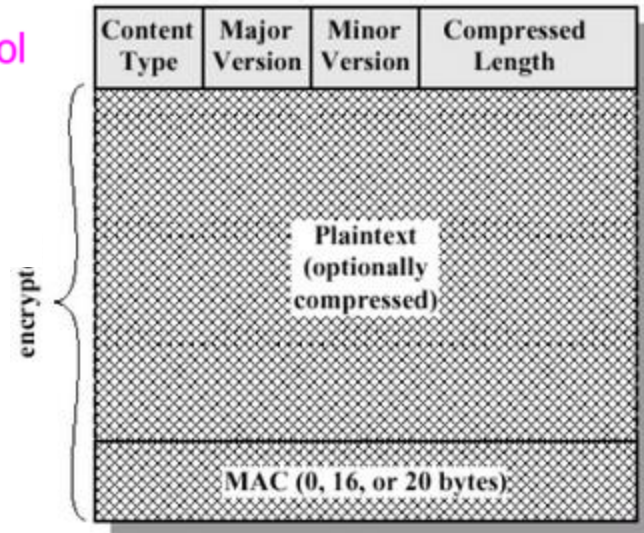
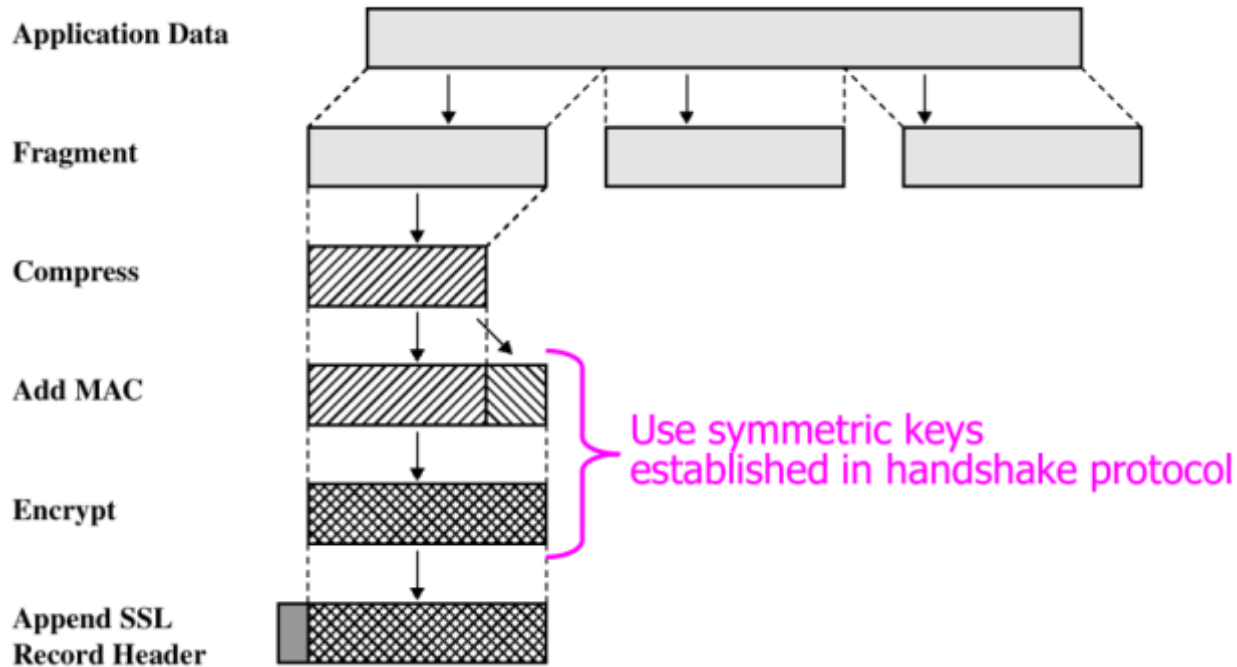
SSL/TLS Handshake Protocol

- 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

SSL Handshake Protocol



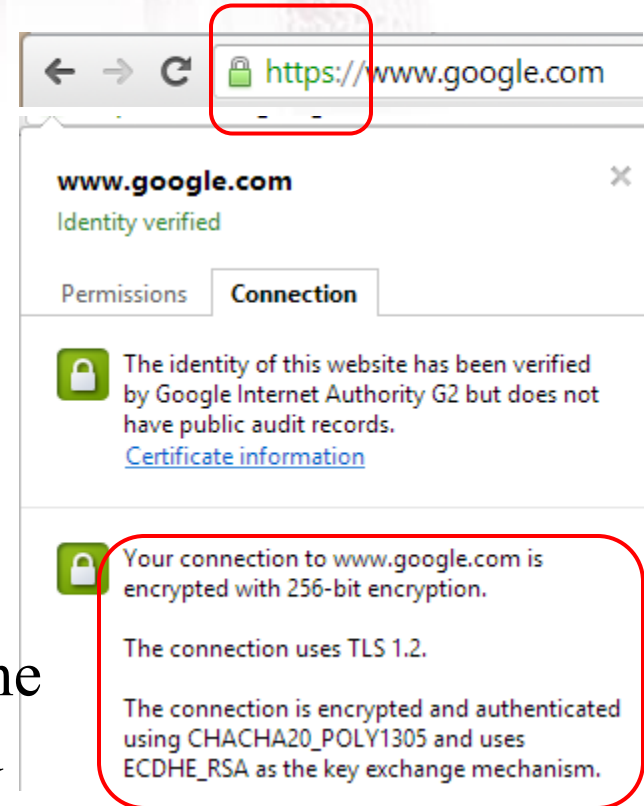
SSL/TLS Record Protocol



TLS uses HMAC

HTTPS

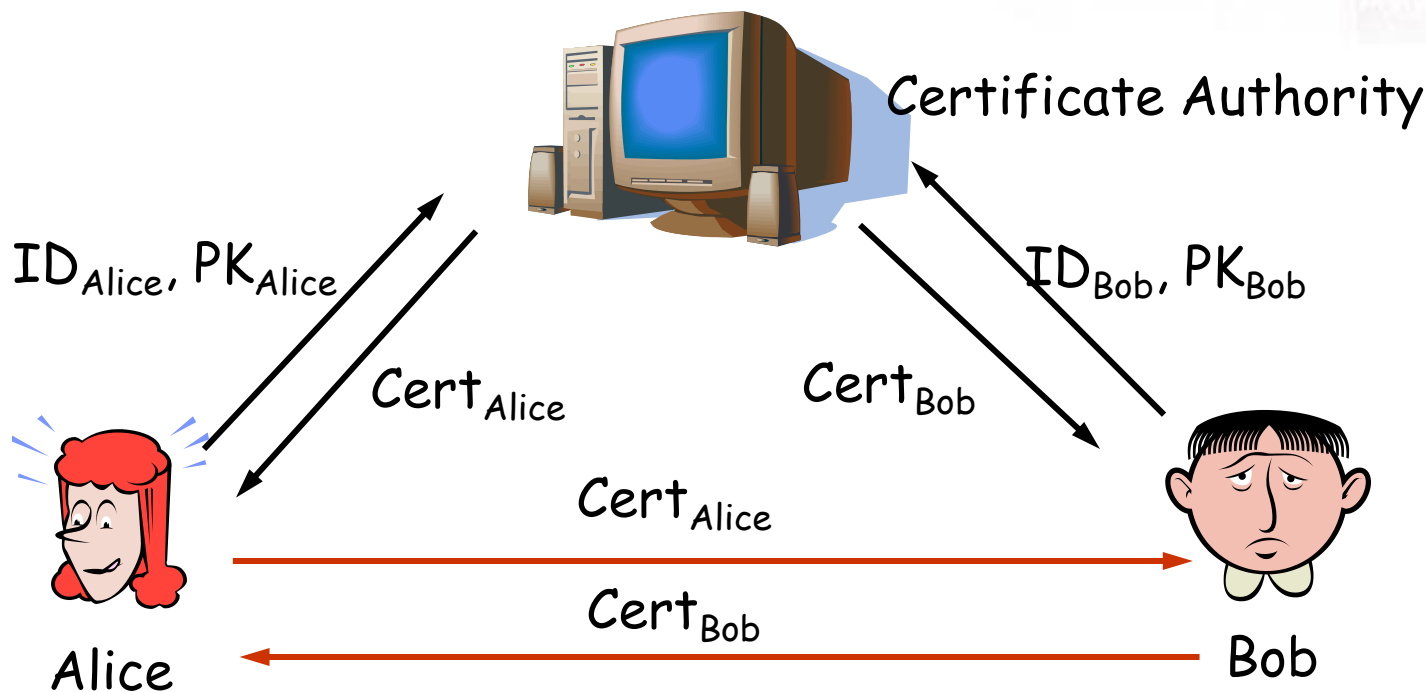
- HTTP Secure, HTTP over SSL, HTTP over TLS
- HTTPS connections need their own port - port 443
- Require X.509 certificates to check the identity of the peer
- Require Certificate Authority (CA) and Public-key Infrastructure (PKI) to verify the relation between owner of a certificate and the certificate, as well as to generate, sign, and administer the validity of certificates



Distribution of Public Keys

- Public announcement or public directory
 - Risks: forgery and tampering
- **Public-key certificate**
 - Signed statement specifying the key and identity
 - $SIG_{Alice}(\text{“Bob”}, PK_B)$
- Common approach: **certificate authority (CA)**
 - An agency responsible for certifying public keys
 - Browsers are pre-configured with 100+ of trusted CAs
 - A public key for any website in the world will be accepted by the browser if certified by one of these CAs

Public-Key Certificates



$$Cert_{Alice} = \langle ID_{Alice}, SN, Expiry, PK_{Alice}, Sig_{CA}(ID_{Alice}, SN, Expiry, PK_{Alice}) \rangle$$

Pre-installed Trusted CAs

The image shows two windows from the Windows Certificate Manager. The left window, titled 'Certificates', displays a list of trusted root certificates. The right window, titled 'Certificate', shows the details for a selected VeriSign Trust Network certificate.

Certificates Window:

Issued To	Issued By	Expiration	Friendly Name
thawte Primary Ro...	thawte Primary Root CA	7/16/2036	thawte
Thawte Timestampi...	Thawte Timestamping...	12/31/2020	Thawte T...
UTN - DATACorp SGC	UTN - DATACorp SGC	6/24/2019	USERTrus
UTN-USERFirst-Obj...	UTN-USERFirst-Object	7/9/2019	USERTrus
VeriSign Class 3 Pu...	VeriSign Class 3 Public...	7/16/2036	VeriSign
VeriSign Class 3 Pu...	VeriSign Class 3 Public...	7/16/2036	VeriSign
VeriSign Trust Netw...	VeriSign Trust Network	5/18/2018	VeriSign
VeriSign Trust Netw...	VeriSign Trust Network	8/1/2028	VeriSign

Certificate Window (Details tab):

Field	Value
Version	V1
Serial number	46 a4 33 bd 76 1f 6a 49 e6 a8...
Signature algorithm	sha1RSA
Signature hash algorithm	sha1
Issuer	VeriSign Trust Network, (c) 19...
Valid from	Sunday, May 17, 1998 5:00:0...
Valid to	Friday, May 18, 2018 4:59:59 ...
Subject	VeriSign Trust Network (c) 19...

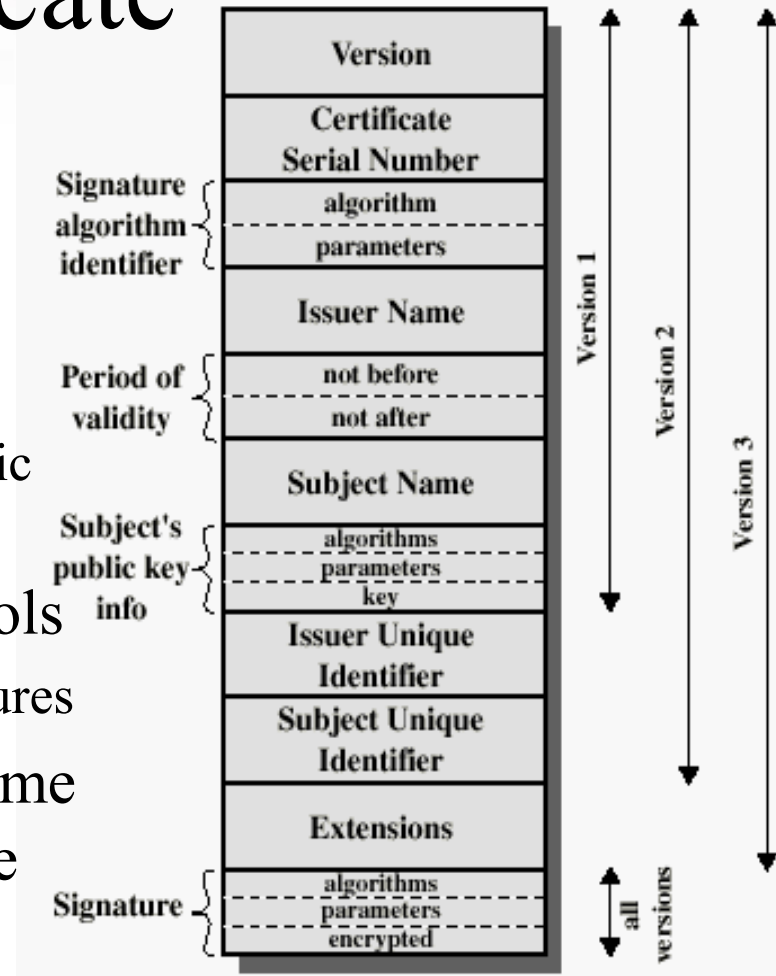
The 'Subject' field contains a hexadecimal string representing the public key:

```
30 81 89 02 81 81 00 cc 5e d1 11 5d 5c 69
d0 ab d3 b9 6a 4c 99 1f 59 98 30 8e 16 85
20 46 6d 47 3f d4 85 20 84 e1 6d b3 f8 a4
ed 0c f1 17 0f 3b f9 a7 f9 25 d7 c1 cf 84
63 f2 7c 63 cf a2 47 f2 c6 5b 33 8e 64 40
04 68 c1 80 b9 64 1c 45 77 c7 d8 6e f5 95
29 3c 50 e8 34 d7 78 1f a8 ba 6d 43 91 95
8f 45 57 5e 7e c5 fb ca a4 04 eb ea 97 37
54 30 6f bb 01 47 32 33 cd dc 57 9b 64 69
```

An arrow points from the text 'Public Key (RSA 1024 bit)' to the start of this hexadecimal string.

X.509 Certificate

- Internet standard (1988-2000)
- Specifies certificate format
 - used in IPsec and SSL/TLS
- Specifies certificate directory service
 - For retrieving other users' CA-certified public keys
- Specifies a set of authentication protocols
 - For proving identity using public-key signatures
- Can use with any digital signature scheme and hash function, but must hash before signing



Certificate Example

User Name
Certificate Version
Validity Period
Serial No
User's Public Key
Other user attributes
CA's name
CA's signature (of all the above)

User Name: www.google.com

Certificate Version: [V3](#)

Validity Period: [Feb 12, 14 - June 11, 14](#)

Serial No: [4d cc 87 66 51 3f 02 14](#)

User's Public Key: [RSA \(2048 bits\)](#)

Other attributes: [e.g. signing algorithm: sha1RSA](#)

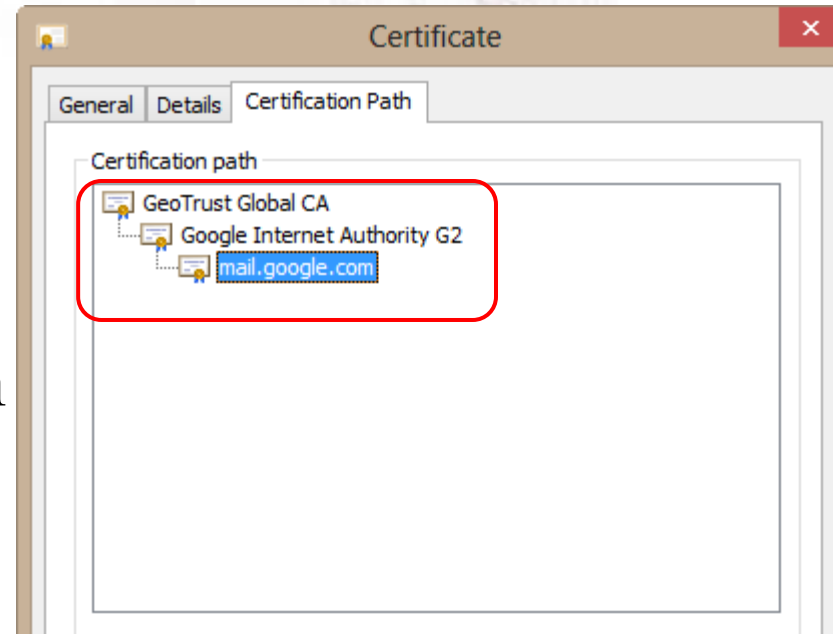
CA's name: [Google Internet Authority G2](#)

CA's signature: [1024-bit data](#)

$Cert_A = \langle ID_A, PK_A, \text{Validity Period}, \dots \text{Sign}_{CA}(ID_A, PK_A, \text{Validity Period}, \dots) \rangle$

CA Hierarchy

- Browsers have several trusted **root certificate authorities**
- A Root CA signs certificates for intermediate CAs, they sign certificates for lower-level CAs, etc.
 - Certificate “**chain of trust**”
 - GeoTrust (root) → Google Internet Authority → mail.google.com
- Client (browser) authenticates this chain of certificates beginning from the root CA.



Alternative: Web of Trust

- Used in PGP (Pretty Good Privacy)
- Instead of a single root certificate authority, each person has *a set of keys they “trust”*
 - If public-key certificate is signed by one of the “trusted” keys, the public key contained in it will be deemed valid
- Trust can be *transitive*
 - Can use certified keys for further certification

