

Defending Computer Networks

Lecture 23: Transport Layer Security

Stuart Staniford

Adjunct Professor of Computer Science

Logistics


- Guest lecture on Thursday
 - Also HW 5
- Regular lecture next Tuesday
- Thanksgiving
- Guest lecture following Tuesday
- Quiz 3, final class that Thursday

TOP STORIES IN MARKETS

1 of 12

 [Taking Care of China Inc.](#)

2 of 12

 [Pour a Coke
Down Under](#) [No Oasis for Telecom
Investors in the E...](#)

MARKETS

Banks to Take Part in New York Cybersecurity Test

About 200 I By SAABIRA CHAUDHURI

Race to the Top

Nov. 18, 2013 12:47 p.m. ET

About 200 banks next month will be required to participate in what amounts to a competition on which is best prepared to handle a cyberattack.

The New York State Department of Financial Services on Monday said it is requiring the banks it regulates—mainly community banks based in New York and branches of large foreign banks—to answer questions in real time on Dec. 12 to assess their cybersecurity policies and processes.

All the banks will be asked the questions simultaneously via a webcast. Banks will answer yes or no, and later will be able to see how they stack up against their peers. A person familiar with the matter described the competition as "kind of cybersecurity academic decathlon," the aim of which is to "foster a race to the top on these measures."

The results could scare or shame banks into beefing up their cybersecurity measures—although responses will be anonymous to encourage full disclosure from banks that might be wary of looking worse than competitors.



Jobs



Real Estate



Cars

HOME

NEWS

COMMUNITY

OPINIONS

SPORTS

OBITUARIES

LIFE

BUSINESS

HN EXTRAS

BLOGS

YOUR PHOTOS

SPE

POLICE & FIRE

ELECTION 2013

MASS & R.I.

EDUCATION

SOUTHCOAST HOMES

WHEELS

POLICE & FIRE SCANNER

100 DAYS, 100

Hot Links

@ The Herald News | Facebook | Twitter | Newsletter |



Stop paying rent!
Now is the time to buy
YOUR DREAM HOME!

CLICK HERE
FOR DETAILS

Swansea police pay \$750 "ransom" after computer virus strikes



Harvard Vanguard
Medical Associates
Atrius Health

Visit
HarvardVanguard.org

Most Insurance Accepted

Welcoming New Patients

By Brian Fraga

Herald News Staff Reporter

Posted Nov 15, 2013 @ 12:47 PM

Last update Nov 15, 2013 @ 07:17 PM

Recommend

52 people recommend this.



SWANSEA — A computer virus that encrypts files and then demands that victims pay a “ransom” to decrypt those items recently hit the Swansea Police Department.

The department paid \$750 for two **Bitcoins** — an online currency — to decrypt several images and word documents in its computer system, Swansea Police Lt. Gregory Ryan said.

“It was an education for (those who) had to deal with it,” Ryan said, adding that the virus did not affect the software program that the police department uses for police reports and booking photos.

Events Calendar

Other News

- <http://kstp.com/news/stories/s3222085.shtml>

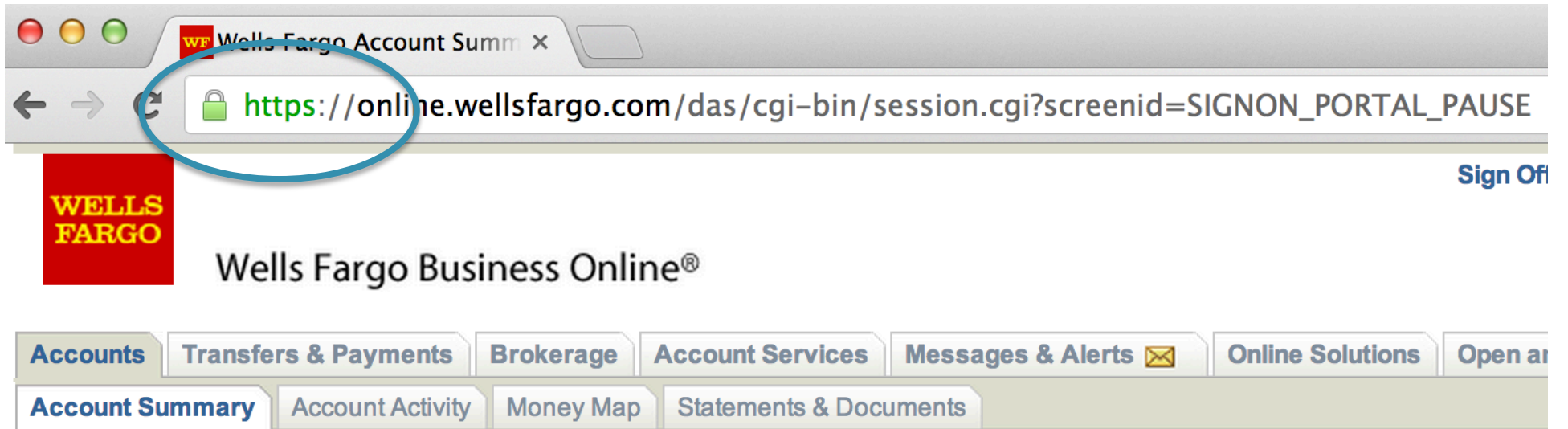
Main Focus of Today

- Start on TLS (formerly known as SSL)
 - And various attendant technologies

Main Goals of TLS/SSL

- Transport Layer Security/Secure Sockets Layer
- Original goal was secure HTTP: HTTPS
- Now heavily used as basis for VPNs
 - Virtual Private Network
 - Way to provide secure network connections to remote users
- Used for email (POP/SMTP/IMAP over SSL)
- Used for SIP (VOIP protocol)

HTTPS



Last Sign On: November 18, 2013

Account Summary

TLS History

- 1995: SSL 2.0 (Netscape)
- 1996: SSL 3.0 (Netscape, later RFC 6101)
- 1999: TLS 1.0 (RFC 2246)
- 2006: TLS 1.1 (RFC 4346)
- 2008: TLS 1.2 (RFC 5246, 6176)
 - Supported in latest versions of all major browsers

TLS: Step 1

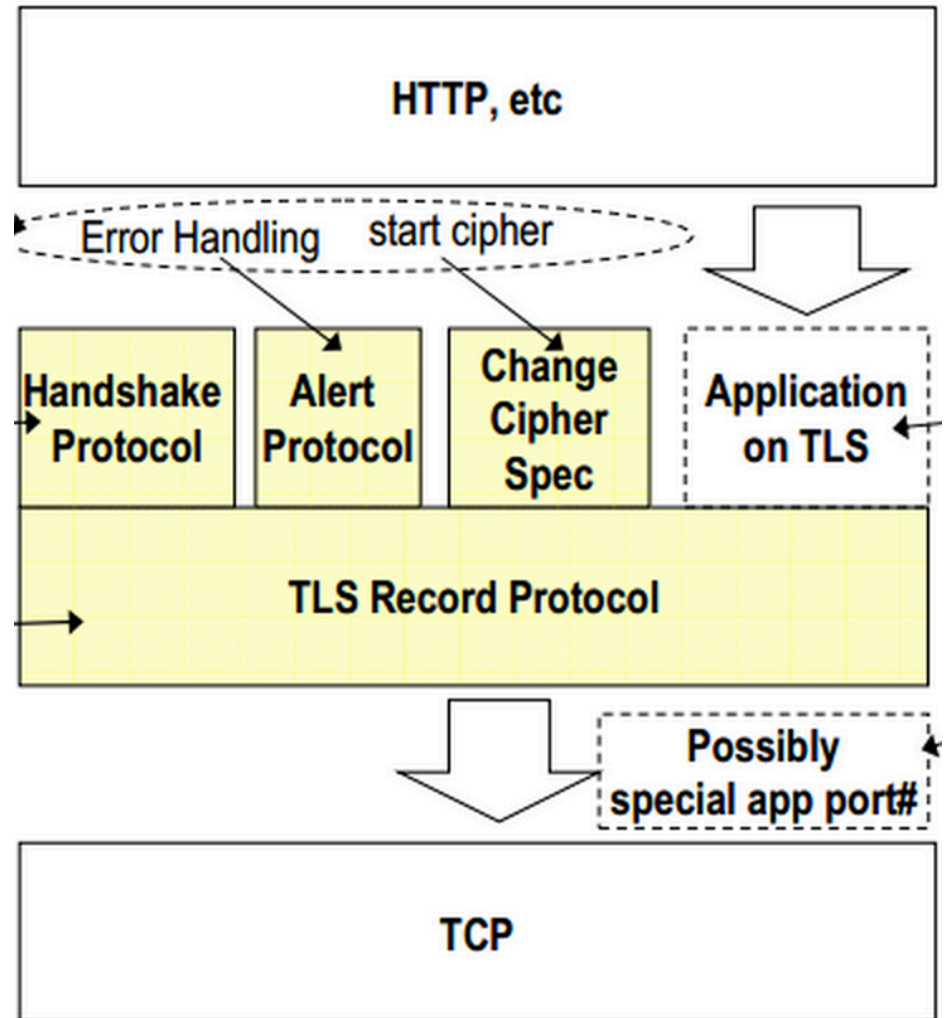
- Establish unencrypted connection
 - Typically over TCP
 - Eg HTTPS over port 443
 - Has also been implemented over UDP
 - And...

Possible Future Implementations

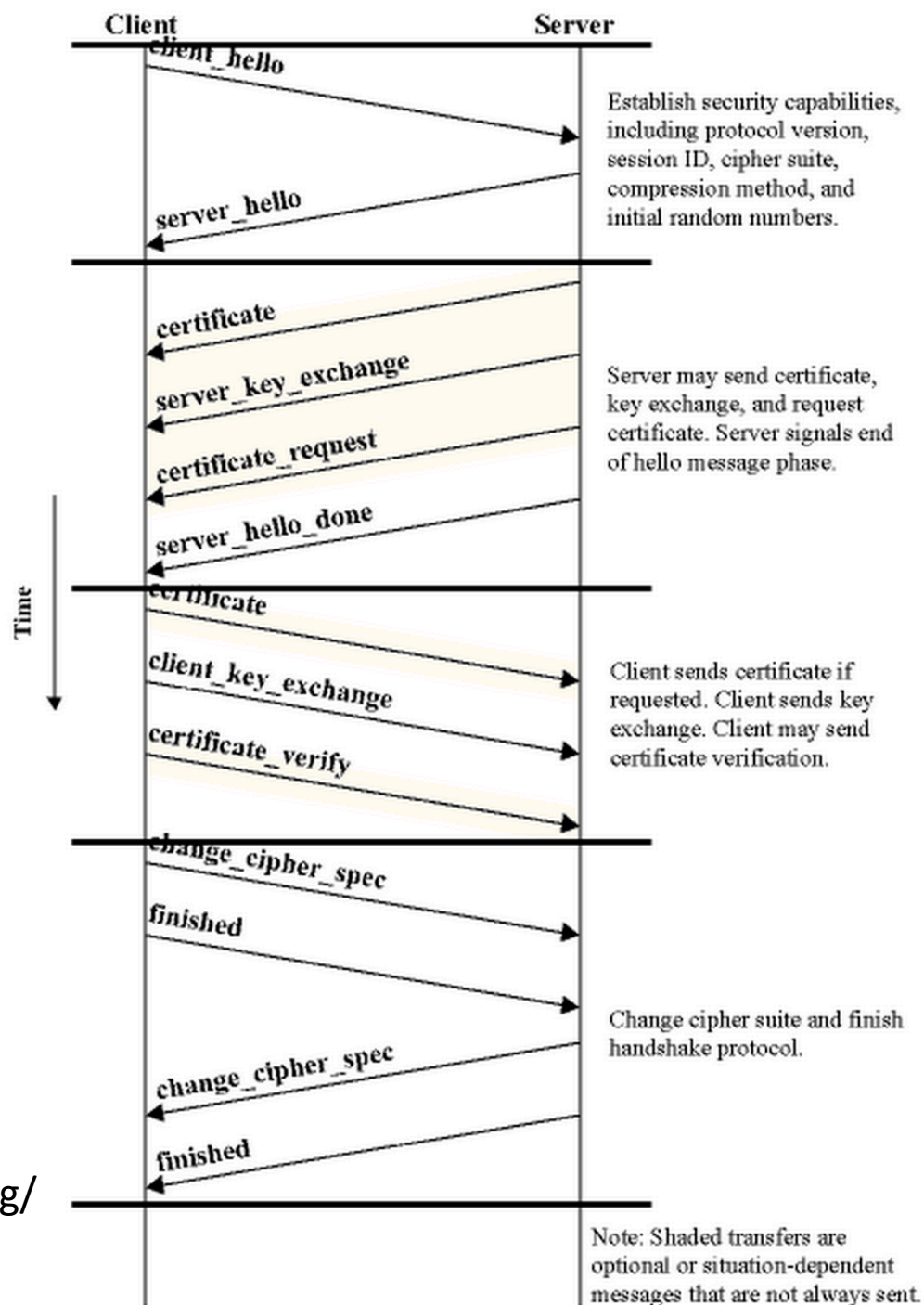


TLS Handshake

- To establish parameters of remaining conversation
- Works over TLS Record layer
- Can also change operation of record layer



Handshake Overview



<http://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/tls/tls.html>

TLS Client Hello

Client Hello. The client initiates a session by sending a Client Hello message to the server. The Client Hello message contains:

- **Version Number.** The client sends the version number corresponding to the highest version it supports. Version 2 is used for SSL 2.0, version 3 for SSL 3.0, and version 3.1 for TLS. Although the IETF RFC for TLS is TLS version 1.0, the protocol uses 3.1 in the version field to indicate that it is a higher level (newer and with more functionality) than SSL 3.0.
- **Randomly Generated Data.** ClientRandom[32], the random value, is a 4-byte number that consists of the client's date and time plus a 28-byte randomly generated number that will ultimately be used with the server random value to generate a master secret from which the encryption keys will be derived.
- **Session Identification (if any).** The sessionID is included to enable the client to resume a previous session. Resuming a previous session can be useful, because creating a new session requires processor-intensive public key operations that can be avoided by resuming an existing session with its established session keys. Previous session information, identified by the sessionID, is stored in the respective client and server session caches.
- **Cipher Suite.** The A list of cipher suites available on the client. An example of a cipher suite is TLS_RSA_WITH_DES_CBC_SHA, where TLS is the protocol version, RSA is the algorithm that will be used for the key exchange, DES_CBC is the encryption algorithm (using a 56-bit key in CBC mode), and SHA is the hash function.
- **Compression Algorithm.** The requested compression algorithm (none currently supported).

[http://technet.microsoft.com/en-us/library/cc785811\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc785811(v=ws.10).aspx)

TLS Server Hello

Server Hello. The server responds with a Server Hello message. The Server Hello message includes:

- **Version Number.** The server sends the highest version number supported by both sides. This is the lower of: the highest version number the server supports and the version sent in the Client Hello message.
- **Randomly Generated Data.** ServerRandom[32], the Random Value, is a 4-byte number of the server's date and time plus a 28-byte randomly generated number that will be ultimately used with the client random value to generate a master secret from which the encryption keys will be derived
- **Session Identification (if any).** This can be one of three choices.
 - New session ID – The client did not indicate a session to resume so a new ID is generated. A new session ID is also generated when the client indicates a session to resume but the server can't or won't resume that session. This latter case also results in a new session ID.
 - Resumed Session ID– The id is the same as indicated in the client hello. The client indicated a session ID to resume and the server is willing to resume that session.
 - Null – this is a new session, but the server is not willing to resume it at a later time so no ID is returned.
- **Cipher Suite.** The server will choose the strongest cipher that both the client and server support. If there are no cipher suites that both parties support, the session is ended with a “handshake failure” alert.
- **Compression Algorithm.** Specifies the compression algorithm to use (none currently supported).

TLS Certificate

- Next the server sends it's certificate.
- So what is a certificate?
- That requires something of a detour...
 - Cryptographic hash/message digest
 - Digital signature
 - Certificate

Cryptographic Hash

- Take an input plain text m
- Outputs a message digest of fixed size $h(m)$
- Such that
 - Any change in input will change output significantly (usually totally)
 - Computationally infeasible, given h , to find m
 - Computationally infeasible to find m_1 and m_2 such that $h(m_1) = h(m_2)$

Example Cryptographic Hashes

- ~~MD5~~ (long gone bad)
- ~~SHA-1~~ (starting to go bad)
- SHA-256 (currently recommended)
- SHA-3 (on horizon)
- Try it
 - openssl md5 sp.c
 - openssl sha1 sp.c
 - openssl sha256 sp.c
 - openssl no-sha256 or openssl help
 - openssl version

How SHA-1 Works

- Examine the pseudo-code at
- <http://en.wikipedia.org/wiki/SHA-1>

Digital Signature

- Scheme for guaranteeing that a message is what it appears to be
 - Authentication
 - was not created by an imposter instead of sender
 - Non-repudiation
 - Sender cannot deny having sent it
 - Integrity
 - Message not altered in transit

Digital Signature Implementation

- Sender
 - Hash message with cryptographic hash fn
 - Encrypt hash with *private* key (eg RSA)
 - Attach signature to message
- Receiver
 - decrypt signature with *public* key
 - Recompute hash
 - Make sure signature matches message hash
- Essentially proves that sender was in possession of private key associated with given public key

X.509 Certificates

- Public Key Infrastructure
 - Dates back to 1988
 - RFC 5280 (IETF version) for practical purposes
- Tries to solve the problem
 - How do I find/validate the public key for X?
- Relies on the idea of chain of trust

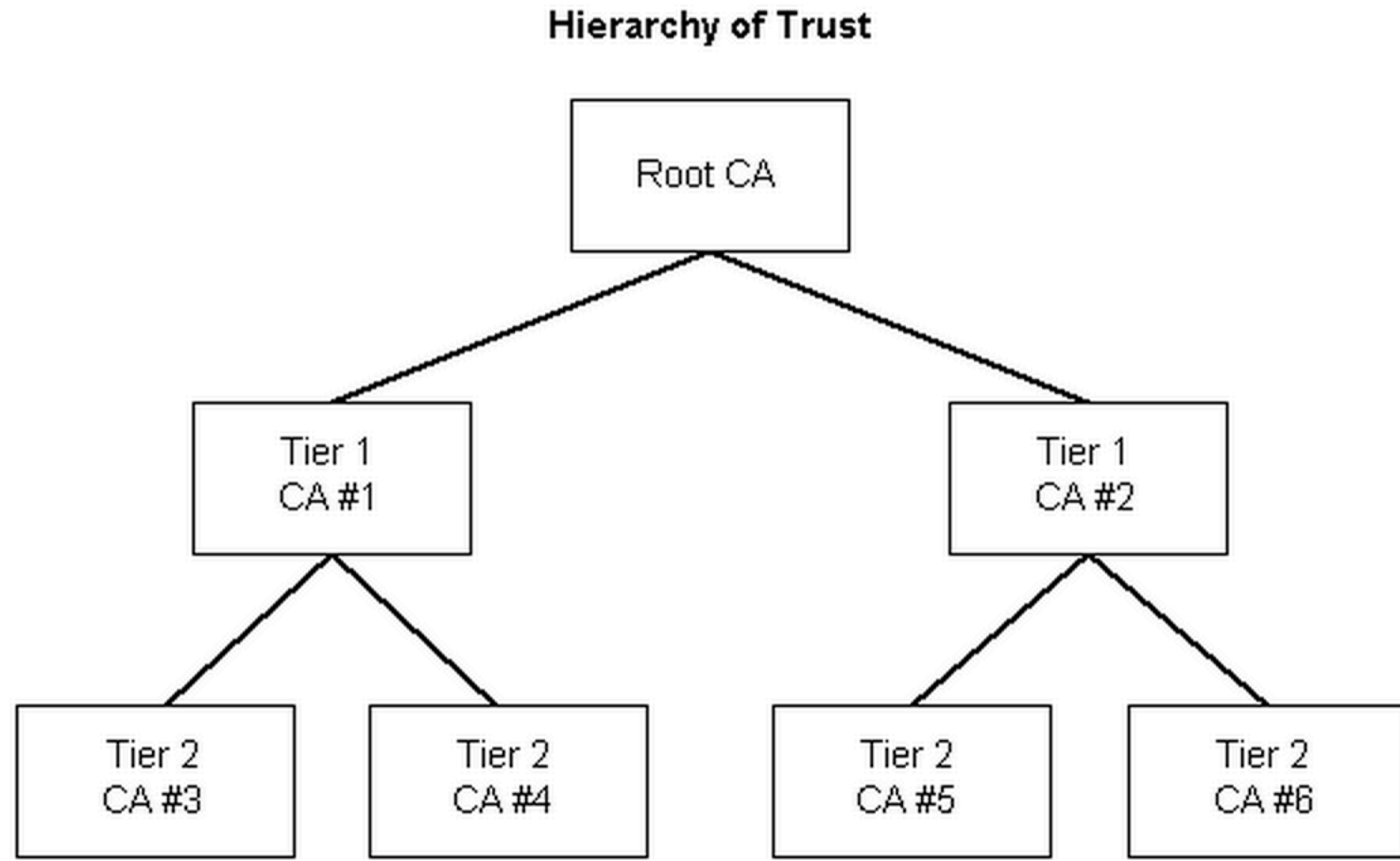
What is in X.509 Cert?

- Certificate version (eg 3)
- Serial Number (eg 480025)
- Algorithm ID (eg 'sha1WithRSAEncryption')
- Issuer (CA = Certificate Authority, eg Geotrust)
- Validity Time Interval
 - Cert not to be used outside of this

More in X.509 Cert

- Subject: (eg domain of website)
- Subject Public Key info
 - Algorithm
 - Public Key data
- X509 extensions
- Signature

Certificate Authorities



[http://msdn.microsoft.com/en-us/windows/aa382479\(v=vs.90\)](http://msdn.microsoft.com/en-us/windows/aa382479(v=vs.90))

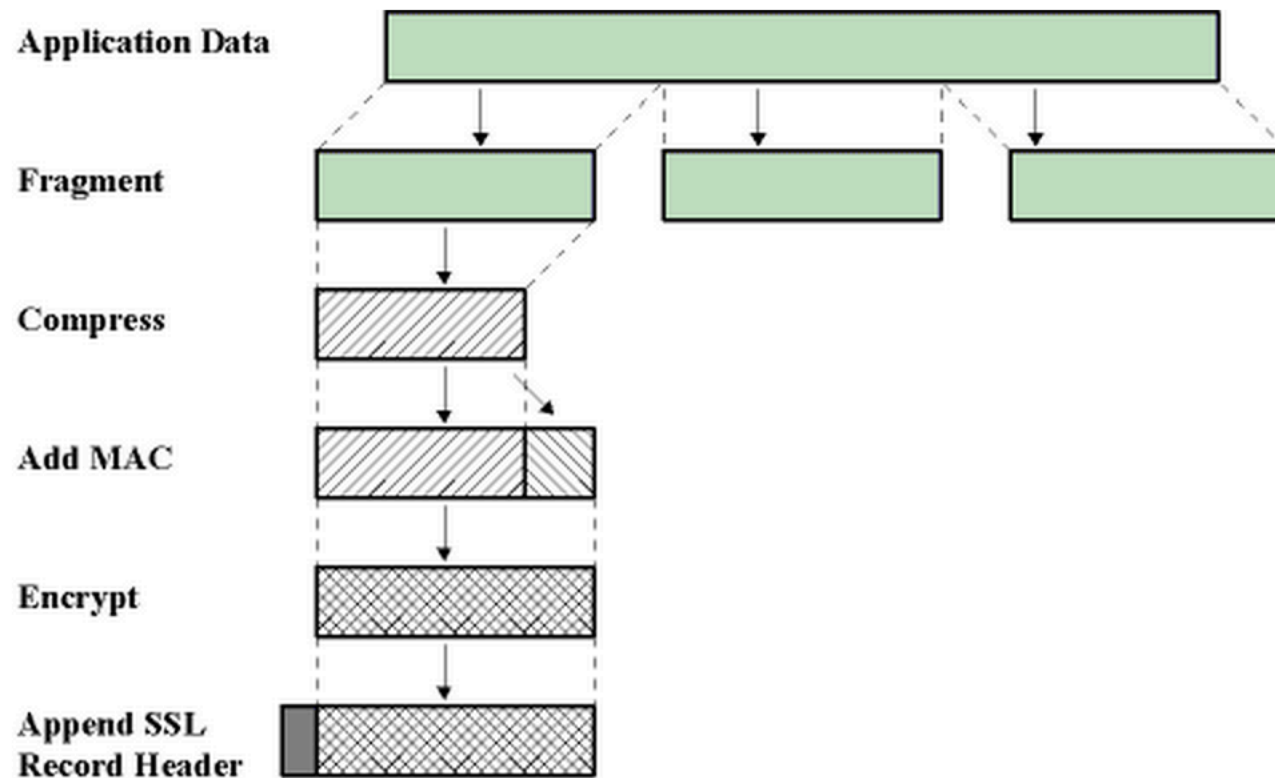
Hundreds of Root CAs

- Eg let's examine the list for Mozilla:
- <http://www.mozilla.org/projects/security/certs/included/>

Obtaining Certificate to Play

- <https://www.networking4all.com/en/support/tools/site+check/>
- Eg try with www.nytimes.com
- `openssl x509 -in nytimes-cert.txt -text |more`

TLS Record Layer



<http://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/tls/tls.html>

TLS Record Format

Byte +0	Byte +1	Byte +2	Byte +3	Major Version	Minor Version	Version Type
Content type				3	0	SSL 3.0
Version		Length		3	1	TLS 1.0
(Major)	(Minor)	(bits 15..8)	(bits 7..0)	3	2	TLS 1.1
Protocol message(s)				3	3	TLS 1.2
MAC (optional)				Hex	Dec	Type
Padding (block ciphers only)				0x14	20	ChangeCipherSpec
				0x15	21	Alert
				0x16	22	Handshake
				0x17	23	Application