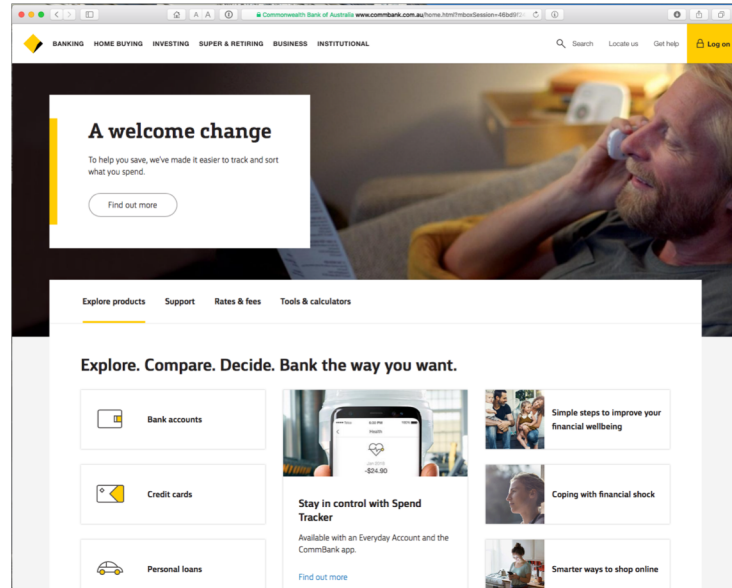


Who am I talking to?

Who am I talking to?

What's the Problem?

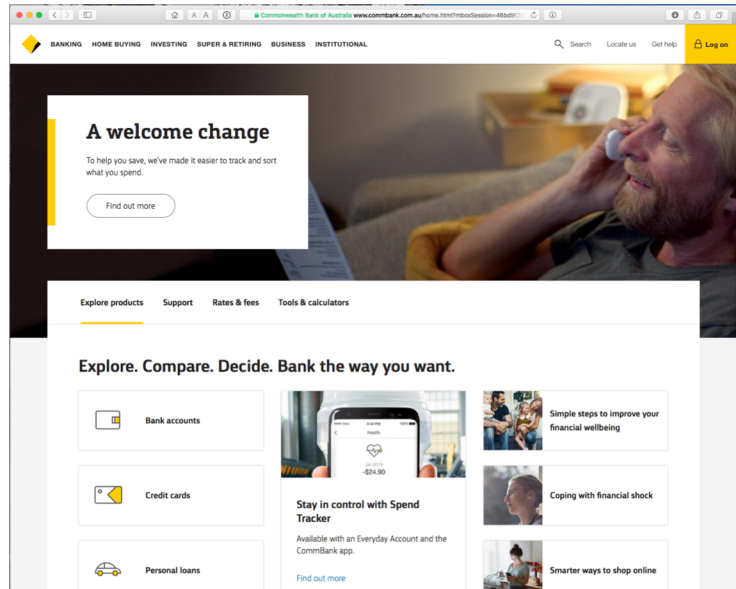
# Which Bank? My Bank!



*it looks like my bank  
But is it my bank?*

# The Question:

How do you know that you are really going to where you thought you were going to?



*it looks like my bank  
But is it my bank?*

# A Clue!

The image shows a screenshot of the Commonwealth Bank of Australia website. A blue circle highlights the address bar, which contains the URL: [www.commbank.com.au/home.html?mboxSession=46bd9f24](http://www.commbank.com.au/home.html?mboxSession=46bd9f24). Two blue arrows point from the URL to the main banner area of the website, which features the text "A welcome change" and a photograph of a woman. Below the banner, the navigation menu includes "INVESTING", "SUPER & RETIRING", "BUSINESS", and "INSTITUTIONAL". A search bar is located on the right side of the navigation menu. The main content area below the navigation menu features a heading "Explore. Compare. Decide. Bank the way you want." and several promotional tiles for "Bank accounts", "Credit cards", "Personal loans", "Stay in control with Spend Tracker", "Simple steps to improve your financial wellbeing", "Coping with financial shock", and "Smarter ways to shop online".

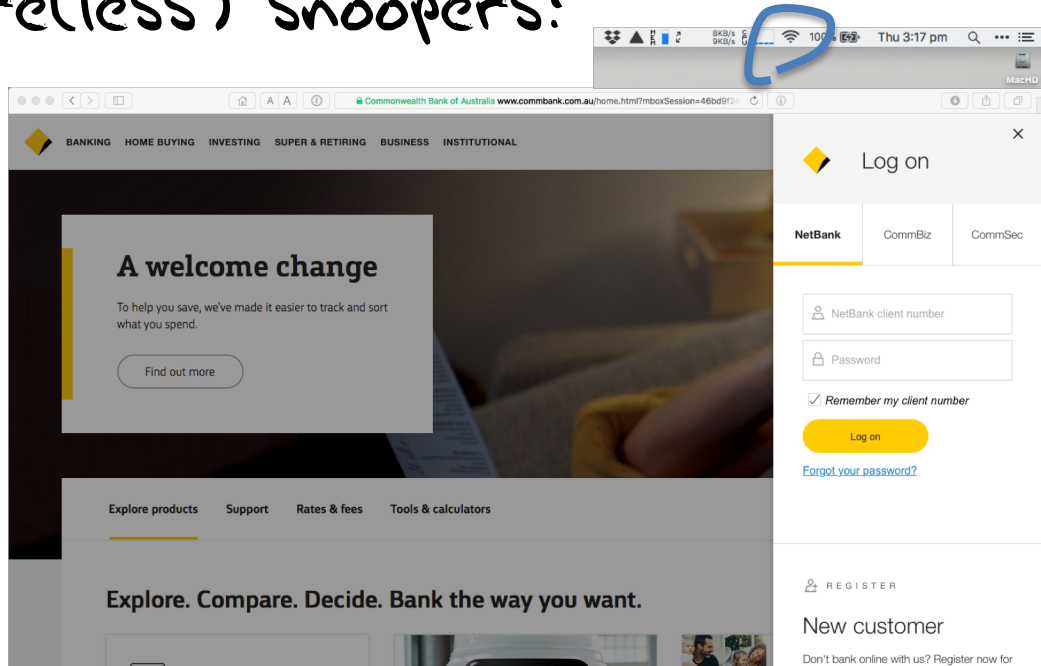
# A Clue!

The image shows a screenshot of the Commonwealth Bank of Australia website. The browser's address bar is highlighted with a blue circle and contains the URL: [www.commbank.com.au/home.html?mboxSession=46bd9f24](https://www.commbank.com.au/home.html?mboxSession=46bd9f24). A blue arrow points from the address bar to a banner on the website that says "A welcome change". A large, handwritten green note is overlaid on the page, stating: "if 'green' is all users can rely on, then it's a pretty unsatisfactory security model!".

*if "green" is all users can rely on, then it's a pretty unsatisfactory security model!*

# Leakage

Also, how can you keep your session a secret from wire(less) snoopers?



# Why is this important?

Because it may not be your bank that you are providing your credentials to

The connection may not be as secure as you might like it to be



# Because sometimes...

The image is a screenshot of a news article from Ars Technica. The article title is "Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency". The sub-headline reads "Almost 1,300 addresses for Amazon Route 53 rerouted for two hours." The author is "DAN GOODIN" and the date is "4/25/2018, 5:00 AM". The main image is the Amazon.com logo. Below the logo, there is a social media share count of 123 and icons for Facebook and Twitter. The article text describes a security incident where Amazon lost control of a small number of its cloud services IP addresses for two hours on Tuesday morning. Hackers exploited a known Internet-protocol weakness to redirect traffic to rogue destinations, masquerading as the cryptocurrency website MyEtherWallet.com and stealing about \$150,000 in digital coins. The incident started around 6 AM California time, hijacking roughly 1,300 IP addresses, Oracle-owned Internet Intelligence said on Twitter. The malicious redirection was caused by fraudulent routes that were announced by Columbus, Ohio-based eNet, a large Internet service provider that is referred to as autonomous system 10297. Once in place, the eNet announcement caused Hurricane Electric and possibly Hurricane Electric customers and other eNet peers to send traffic over the same unauthorized routes. The 1,300 addresses belonged to Route 53, Amazon's domain name system service.

Just 2 hours was enough!

The attackers managed to steal about \$150,000 of currency from MyEtherWallet users,

And numerous other incidents

# Opening the Connection: First Steps



Client:

*DNS Query:*

www.commbank.com.au?



*DNS Response:*

23.77.138.30

Here's what happens when  
I connect to my bank

The first step is a DNS  
transaction to get an IP  
address

*TCP Session:*

TCP Connect 23.77.138.30, port 443



Then a TCP session is  
started

# Hang on...

```
$ dig -x 23.77.138.30 +short  
a23-77-138-30.deploy.static.akamaitechnologies.com.
```

That's not an IP addresses that was allocated to the Commonwealth Bank!

The Commonwealth Bank of Australia has been assigned the address blocks:

140.168.0.0 - 140.168.255.255 and

203.17.185.0 - 203.17.185.255

# Hang on...

```
$ dig -x 23.77.138.30 +short  
a23-77-138-30.deploy.static.akamaitechnologies.com.
```

That's an Akamai address block

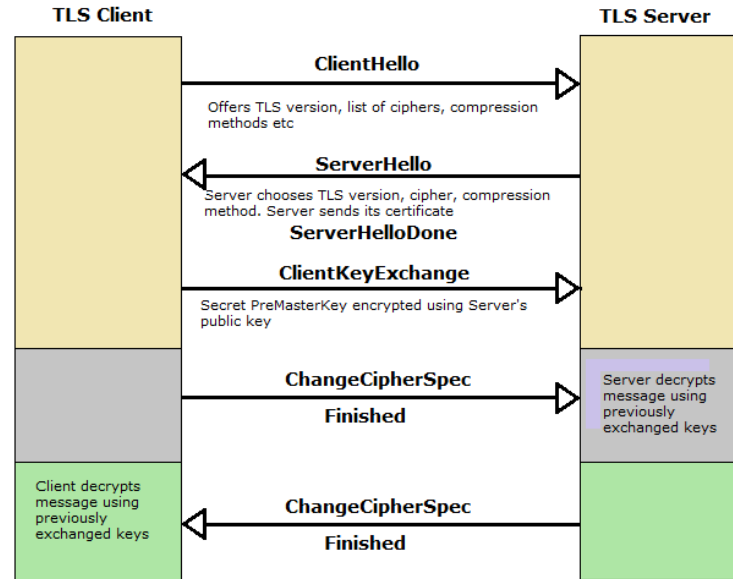
And i am **NOT** a customer of the internet Bank of Akamai!

Why should my browser trust that 23.77.138.30 is really the “proper” web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

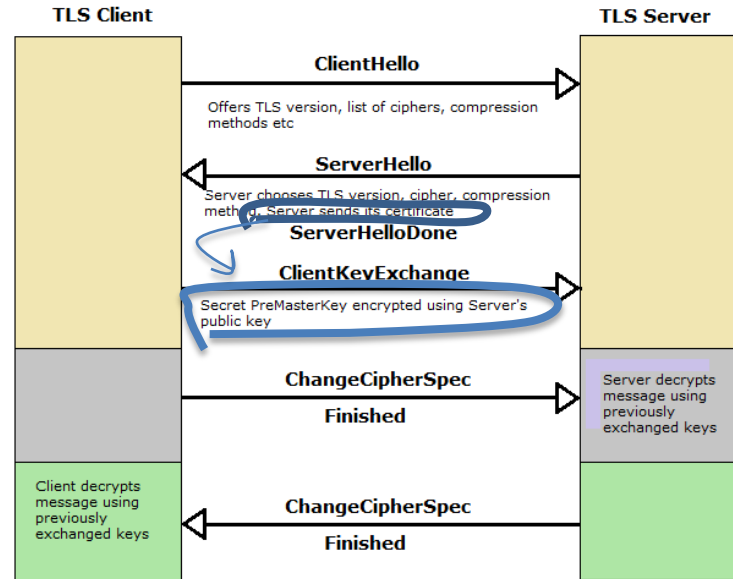
# A trickier question...

How can my browser tell the difference between an intended truth and a lie?

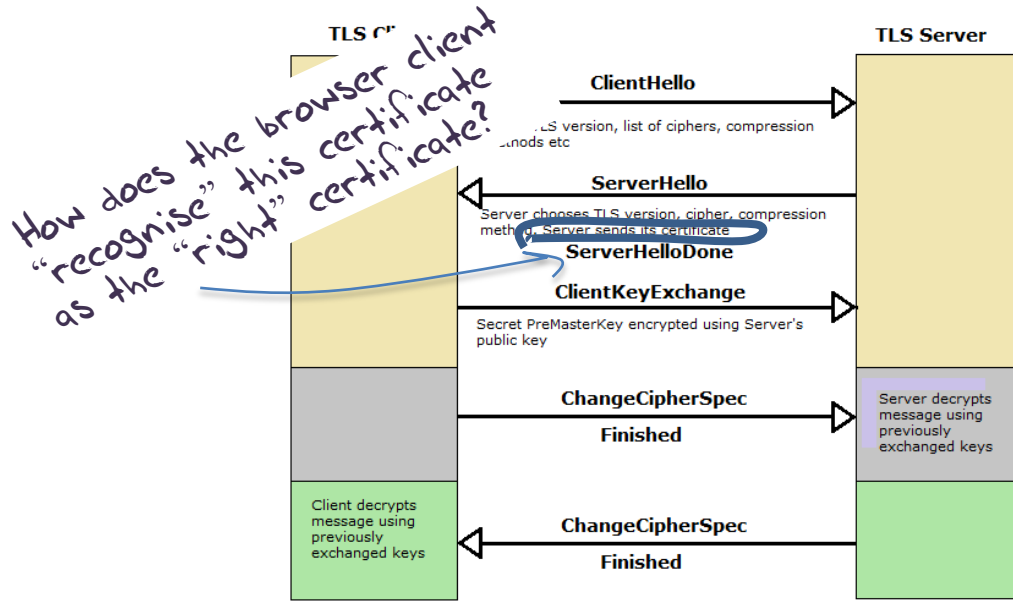
# Secure Connections using TLS 1.2



# Secure Connections using TLS 1.2



# Secure Connections using TLS 1.2





**Safari is using an encrypted connection to www.commbank.com.au.**

Encryption with a digital certificate keeps information private as it's sent to or from the https website www.commbank.com.au.

Symantec Corporation has identified www.commbank.com.au as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

VeriSign Class 3 Public Primary Certification Authority - G5  
 Symantec Class 3 EV SSL CA - G3  
 www.commbank.com.au

**www.commbank.com.au**

Issued by: Symantec Class 3 EV SSL CA - G3  
 Expires: Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time  
 This certificate is valid

► **Trust**▼ **Details**

Subject Name	_____
Inc. Country	AU
Business Category	Private Organization
Serial Number	123 123 124
Country	AU
Postal Code	2000
State/Province	New South Wales
Locality	SYDNEY
Street Address	201 SUSSEX S T
Organization	Commonwealth Bank of Australia
Organizational Unit	CBA Business System Hosting
Common Name	www.commbank.com.au
Issuer Name	_____
Country	US
Organization	Symantec Corporation
Organizational Unit	Symantec Trust Network
Common Name	Symantec Class 3 EV SSL CA - G3
Serial Number	1A 9F E9 4B 03 9D E2 9A B6 15 56 69 60 3E 98 AE
Version	3
Signature Algorithm	SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.1 )
Parameters	none
Not Valid Before	Monday, 4 May 2015 at 10:00:00 AM Australian Eastern Standard Time
Not Valid After	Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
Public Key Info	_____
Algorithm	RSA Encryption ( 1.2.840.113549.1.1.1 )
Parameters	none
Public Key	256 bytes : CA B4 74 93 E8 00 22 10 ...
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 95 32 C3 F0 62 F1 F8 F1 ...



Hide Certificate

OK

Log on

Locate us

Stuff I like

Rates &amp; fees

Latest offers

**GET A C  
OF YOU**

Our new online SMSF  
view of your investme  
more.

Find out more &gt;

**FAMILIAR BANKING  
FOR UNFAMILIAR**




Safari is using an encrypted connection to www.commbank.com.au.

Encryption with a digital certificate keeps information private as it's sent to or from the https website www.commbank.com.au.

Symantec Corporation has identified www.commbank.com.au as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

VeriSign Class 3 Public Primary Certification Authority - G5  
Symantec Class 3 EV SSL CA - G3  
www.commbank.com.au

 **www.commbank.com.au**  
Issued by VeriSign Class 3 EV SSL CA - G3  
Expires: Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time  
This certificate is valid

Trust  
Details

Subject Name	
Inc. Country	AU
Business Category	Private Organization
Serial Number	123 123 124
Country	AU
Postal Code	2000
State/Province	New South Wales
Locality	SYDNEY
Street Address	201 SUSSEX S T
Organization	Commonwealth Bank of Australia
Organizational Unit	CBA Business System Hosting
Common Name	www.commbank.com.au
Issuer Name	
Country	US
Organization	Symantec Corporation
Organizational Unit	Symantec Trust Network
Common Name	Symantec Class 3 EV SSL CA - G3
Serial Number	1A 9F E9 4B 03 9D E2 9A B6 15 56 69 60 3E 98 AE
Version	3
Signature Algorithm	SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.1 )
Parameters	none
Not Valid Before	Monday, 4 May 2015 at 10:00:00 AM Australian Eastern Standard Time
Not Valid After	Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
Public Key Info	
Algorithm	RSA Encryption ( 1.2.840.113549.1.1.1 )
Parameters	none
Public Key	256 bytes : CA B4 74 93 E8 00 22 10 ...
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 95 32 C3 F0 62 F1 F8 F1 ...

? How did my browser know that this is a valid cert?



Hide Certificate

OK

- Log on
- Locate us
- Stuff I like
- Rates & fees
- Latest offers

GET A G  
OF YOU

Our new online SMSF view of your investments more.

Find out more >

FAMILIAR DRIVING FOR UNFAMILIAR

# Domain Name Certification

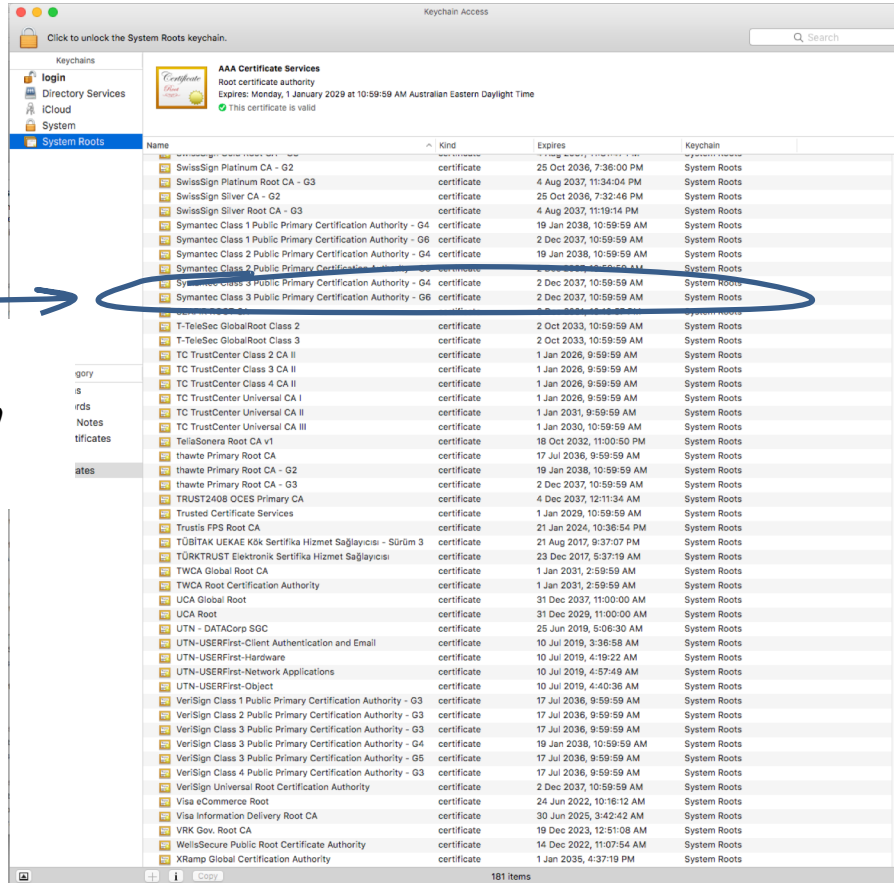
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “Symantec”
- Who was willing to vouch (in a certificate) that the entity who goes by the domain name of [www.commbank.com.au](http://www.commbank.com.au) also has a certain public key value
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to an entity that is able to demonstrate knowledge of the private key for [www.commbank.com.au](http://www.commbank.com.au), as long as I am prepared to trust Symantec and the certificates that they issue
- Symantec NEVER lie!

# Domain Name Certification

- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “Symantec”
- Who was willing to vouch (in a certificate) that the entity who goes by the domain name of [www.commbank.com.au](http://www.commbank.com.au) also has a certain public key value
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to an entity that is able to demonstrate knowledge of the private key for [www.commbank.com.au](http://www.commbank.com.au), as long as I am prepared to trust Symantec and the certificates that they issue
- Symantec NEVER lie!

Why should i trust them?

# Local Trust



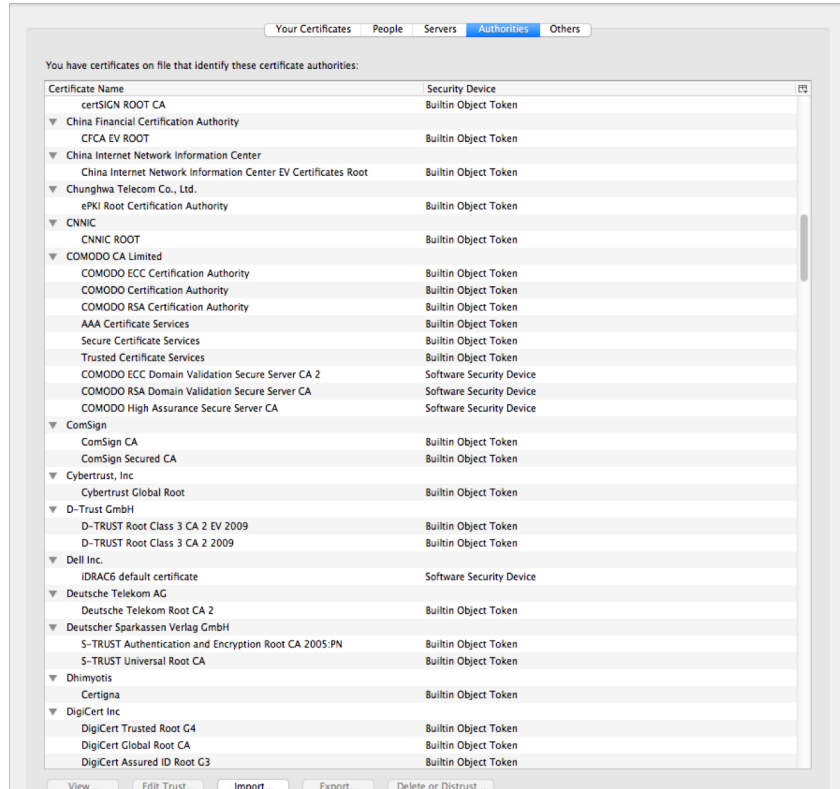
The cert I'm being asked to trust was issued by a certification authority that my browser already trusts - so I trust that cert!

# Local Trust or Local Credulity\*?

That's a big list of people to Trust

Are they all trustable?

\* cre·du·li·ty  
/kriˈd(y)ooledē/  
noun  
a tendency to be too ready to believe that something is real or true.



# Local Trust or Local Credulity\*?

That's a big list of people to Trust

Are they all trustable?

Evidently Not!

\* **cre·du·li·ty**  
/kriˈd(y)ooledē/  
noun  
a tendency to be too ready to believe that something is real or true.

certSIGN ROOT CA  
China Financial Certification Authority  
CFCA EV ROOT  
China Internet Network Information Center  
China Internet Network Information Center EV Certificates Root  
Chunghwa Telecom  
CNNIC  
CNNIC ROOT  
COMODO CA Limited  
COMODO ECC  
COMODO Certificate Authority  
COMODO RSA  
AAA Certificate Authority  
Secure Certificate Authority  
Trusted Certificate Authority  
COMODO ECC  
COMODO RSA  
COMODO High Assurance  
ComSign  
ComSign CA  
ComSign Secure  
Cybertrust, Inc  
Cybertrust Global  
D-Trust GmbH  
D-TRUST Root  
D-TRUST Root  
Dell Inc.  
IDRAC6 default  
Deutsche Telekom  
Deutsche Telekom  
Deutscher Sparkassen  
S-TRUST Authentication  
S-TRUST Universal  
Dhimyotis  
Certigna  
DigiCert Inc  
DigiCert Trust  
DigiCert Global  
DigiCert Assurance

### Maintaining digital certificate security

Posted: Monday, March 23, 2015

Posted by Adam Langley, Security Engineer

On Friday, March 20th, we became aware of unauthorized digital certificates for several Google domains. The certificates were issued by an intermediate certificate authority apparently held by a company called MCS Holdings. This intermediate certificate was issued by CNNIC.

CNNIC is included in all major root stores and so the misissued certificates would be trusted by almost all browsers and operating systems. Chrome on Windows, OS X, and Linux, ChromeOS, and Firefox 33 and greater would have rejected these certificates because of [public-key pinning](#), although misissued certificates for other sites likely exist.

We promptly alerted CNNIC and other major browsers about the incident, and we blocked the MCS Holdings certificate in Chrome with a [CRLSet](#) push. CNNIC responded on the 22nd to explain that they had contracted with MCS Holdings on the basis that MCS would only issue certificates for domains that they had registered. However, rather than keep the private key in a suitable [HSM](#), MCS installed it in a man-in-the-middle proxy. These devices intercept secure connections by masquerading as the intended destination and are sometimes used by companies to intercept their employees' secure traffic for monitoring or legal reasons. The employees' computers normally have to be configured to trust a proxy for it to be able to do this. However, in this case, the presumed proxy was given the full authority of a public CA, which is a serious breach of the CA system. This situation is similar to a [failure by ANSSI](#) in 2013.

# Local Trust or Local Credulity\*?

That's a big list of people to Trust

Are they all trustable?

Evidently Not!

\* cre·du·li·ty  
/k're'd(y)oolədē/  
noun

a tendency to be too ready to believe that something is real or true.

The image shows a Windows Certificate Authority list on the left and an InfoWorld article on the right. A blue arrow points from the 'COMODO CA Limited' entry in the list to the article. A blue circle highlights a paragraph in the article.

**Certificate Name** | Security Device

certSIGN ROOT CA | Built-in Object Token

- China Financial Certification Authority
  - CPCA EV ROOT
- China Internet Network Information Center
  - China Internet Network Information Center
- Chunghwa Telecom Co., Ltd.
  - ePKI Root Certification Authority
- CNNIC ROOT
  - COMODO CA Limited
  - COMODO ECC Certification Authority
  - COMODO Certification Authority
  - COMODO RSA Certification Authority
  - AAA Certificate Services
  - Secure Certificate Services
  - Trusted Certificate Services
  - COMODO ECC Domain Validation Services
  - COMODO RSA Domain Validation Services
  - COMODO High Assurance Services
- ComSign
  - ComSign CA
  - ComSign Secured CA
- Cybertrust, Inc.
  - Cybertrust Global Root
- D-Trust GmbH
  - D-TRUST Root Class 3 CA
  - D-TRUST Root Class 3 CA
- Dell Inc.
  - IDRAC6 default certificate
- Deutsche Telekom AG
  - Deutsche Telekom Root CA
- Deutscher Sparkassen Verlag G...
  - S-TRUST Authentication an...
  - S-TRUST Universal Root CA
- Dhimyotis
  - Certigna
- DigiCert Inc.
  - DigiCert Trusted Root G4
  - DigiCert Global Root CA
  - DigiCert Assured ID Root C...

**InfoWorld** | Most Popular: [dropdown]

Home > Security > Hacking

**SECURITY ADVISER**  
By Roger A. Grimes | Follow

## The real security issue behind the Comodo hack

The Comodo hack has grabbed headlines, but more troubling is the public's ignorance over PKI and digital certificates

InfoWorld | Apr 5, 2011

**MORE LIKE THIS**

- Weaknesses in SSL certification exposed by Comodo security breach
- Hackers target Google, Skype with rogue SSL certificates
- Revoke certificates when you need to -- the right way
- on IDG Answers → I'm considering a slight career change to IT security - what do I need to...

**RELATED TOPICS**

- Hacking
- Authentication
- Data Security
- Encryption
- Identity Management
- IT Management

News of an Iranian hacker duping certification authority Comodo into issuing digital certificates to one or more unauthorized parties has caused an uproar in the IT community, moving some critics to call for Microsoft and Mozilla to remove Comodo as a trusted root certification authority from the systems under their control. Though the hacker managed to do so by first compromising a site containing a hard-coded logon name and password, then generating certificates for several well-known sites, including Google, Live.com, Skype, and Yahoo, I'm not bothered by the

**Datameer**

5 High Impact Big Data Use Cases



But my bank used Symantec

And Symantec NEVER lies in the certificates they issue

**Never?**

# Well, hardly ever


ars TECHNICA **BIZ & IT** TECH SCIENCE POLICY CARS GAMING & CULTURE FORUMS

RISK ASSESSMENT —

## Already on probation, Symantec issues more illegit HTTPS certificates

At least 108 Symantec certificates threatened the integrity of the encrypted Web.

DAN GOODIN · 1/21/2017, 8:40 AM



Enlarge

A security researcher has unearthed evidence showing that three browser-trusted certificate authorities (CAs) owned and operated by Symantec improperly issued more than 100 unvalidated [transport layer security](#) certificates. In some cases, those certificates made it possible to spoof HTTPS-protected websites.

<http://arstechnica.com/security/2017/01/already-on-probation-symantec-issues-more-illegit-https-certificates/>

### Misissued/Suspicious Symantec Certificates

Andrew Ayer | Thu, 19 Jan 2017 13:47:06 -0800

#### I. Misissued certificates for example.com

On 2016-07-14, Symantec misissued the following certificates for example.com:

<https://crt.sh/?sha256=A8F14F52CC1282D7153A13316E7DA39E6AE37B1A10C16288B9024A9B9DC3C4C6>

<https://crt.sh/?sha256=8B5956C57FDC720B6907A4B1BC8CA2E46CD90EAD5C061A426CF48A6117BFBFA>

<https://crt.sh/?sha256=94482136A1400BC3A1136FCA3E79D4D200E03DD20B245D19F0E78B5679EAF48>

<https://crt.sh/?sha256=C69A8D4C1B20B6FC7861C67476CADD1DAE7A8DCF6E23E15311C2D2794BFCDD11>

I confirmed with ICANN, the owner of example.com, that they did not authorize these certificates. These certificates were already revoked at the time I found them.

#### II. Suspicious certificates for domains containing the word "test"

On 2016-11-15 and 2016-10-26, Symantec issued certificates for various domains containing the word "test" which I strongly suspect were misissued:

# Well, hardly ever

ars TECHNICA

RISK ASSESSMENT —

## Already on probation: more illegit HTTPS

At least 108 Symantec certificates threaten

DAN GOODIN · 1/21/2017, 8:40 AM

Enlarge

62

A security researcher has unearthed the authorities (CAs) owned and operated by Symantec. These certificates are used to protect HTTPS-protected websites.

security.googleblog.com/2018/03/distrust-of-symantec-pki

## Google Security Blog

The latest news and insights from Google on security and safety on the Internet

### Distrust of the Symantec PKI: Immediate action needed by site operators

March 7, 2018

Posted by Devon O'Brien, Ryan Sleevi, Emily Stark, Chrome security team

We [previously announced](#) plans to deprecate Chrome's trust in the Symantec certificate authority (including Symantec-owned brands like Thawte, VeriSign, Equifax, GeoTrust, and RapidSSL). This post outlines how site operators can determine if they're affected by this deprecation, and if so, what needs to be done and by when. Failure to replace these certificates will result in site breakage in upcoming versions of major browsers, including Chrome.

#### Chrome 66

If your site is using a SSL/TLS certificate from Symantec that was issued before June 1, 2016, it will stop functioning in Chrome 66, which could already be impacting your users.

If you are uncertain about whether your site is using such a certificate, you can preview these changes in [Chrome Canary](#) to see if your site is affected. If connecting to your site displays a certificate error or a warning in DevTools as shown below, you'll need to replace your certificate. You can get a new certificate from any [trusted CA](#), including Digicert, which recently acquired Symantec's CA business.

com/security/2017/01/already-symantec-issues-more-illegit-https-

### Suspicious Symantec Certificates

12017 13:47:06 -0800

ificates for example.com

ntec misissued the following certificates for example.com:

[82D7153A13316E7DA39E6AE37B1A10C1628BB9024A9B9DC3C4C6](#)

[F720B6907A4B1BC8CA2E46CD90EAD5C061A426CF48A6117BFBFA](#)

[0BC3A1136FCA3E79D4D200E03DD20B245D19F0E78B5679EAF48](#)

[E6FC7861C67476CADD1DAE7A8DCF6E623E15311C2D2794BFCDD1](#)

ANN, the owner of example.com, that they did not trust. These certificates were already revoked.

ificates for domains containing the word "test"

016-10-26, Symantec issued certificates for various domains containing the word "test" which I strongly suspect were

# What's going wrong here?

- The TLS handshake cannot specify **which** CA should be used by the client to validate the digital certificate that describes the server's public key
- The result is that your browser will allow any CA to be used to validate a certificate!
- Which is an exploited weakness in the CA model

# What's going wrong here?

- There is no incentive for quality in the CA marketplace
- Why pay more for any certificate when the entire CA structure is only as strong as the weakest CA?
- And you browser trusts a LOT of CAs!
  - About 60 – 100 CA's
  - About 1,500 Subordinate RA's
  - Operated by 650 different organisations

# In a market for security

Where CA's compete with each other for market share  
And quality offers no protection  
Than what 'wins' in the market?

Sustainable  
Resilient

Secure

Privacy

Trusted

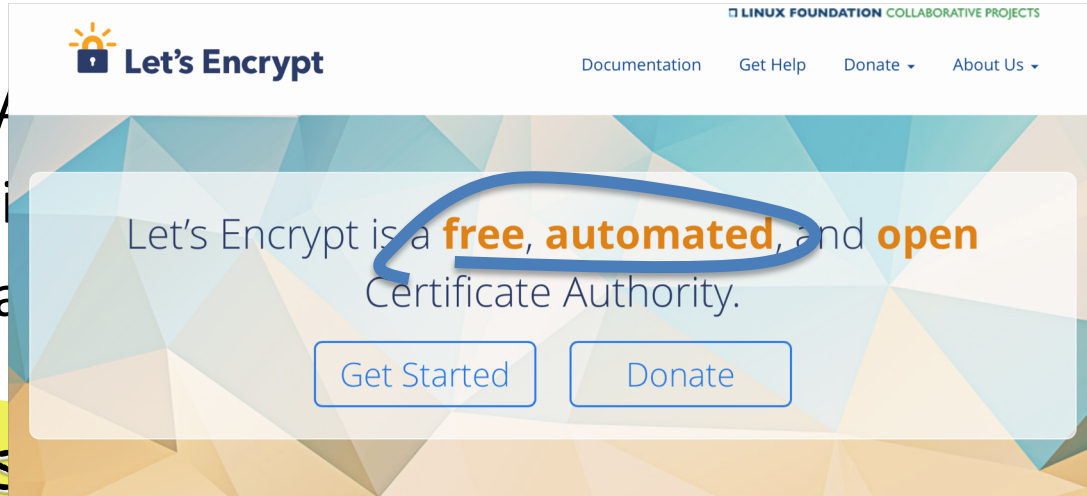


cheap!

# In a market for security

Where CA  
And quality  
Than what

are



LINUX FOUNDATION COLLABORATIVE PROJECTS

Let's Encrypt

Documentation Get Help Donate About Us

Let's Encrypt is a **free, automated, and open** Certificate Authority.

Get Started Donate

Resilient

Secure

Privacy

Trusted



cheap!



Who am I talking to?

What can we do about it?

# What can we do about it?

- The problem with “who am I talking to?” lies in the situation of widely distributed trust in the WebPKI CA environment
- How can we improve this situation?

# Is this your Certificate?

How can a user be assured that the certificate that they are being presented with, signed and published by a CA that their browser / platform is prepared to trust, is the genuine certificate?

# Certificate Transparency

Certificate Transparency is the current response from the CAB Forum

CT is an effort to make the problem **everyone's** problem by requiring all trusted CAs to publish immutable logs of all the certificates they issue

- analogous to blockchain for each CA, but with a centralised authority model

# Certificate Transparency

- Make the problem everyone's problem by requiring all trusted CAs to publish all the certificates they issue
- Leave it to the service publisher to figure out if a fake cert has been issued and logged in the CT logs
  - But what then?
  - How does the user figure out whether the service point they are accessing has been attacked with a fake cert?

# Certificate Transparency is Naïve!

- CT attempts to set a universal threshold that all CAs must pass in order to be trusted by a browser
- But won't really protect my browsing
  - Inspection of CT logs by third parties is not fast, thorough, timely nor effective
  - And revocation of certs requires browsers to perform revocation checks every time (which they don't)
  - Brief (and even long-held) windows of opportunity for exploits still exist

naive

/nɑːˈiːv, nɑːˈiːv/ ⓘ

adjective

(of a person or action) showing a lack of experience, wisdom, or judgement.

# Pinning: Narrowing the Trust Space

## CA / Public Key *Pinning*

- Communicate to the client which CA / which certificate / which public key to trust for a given service name
- Exactly how to undertake this communication in a way that is tamperproof is the challenge

# Coded Browser Pinning

[https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport\\_security\\_state\\_static.json](https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport_security_state_static.json)

```
transport_security_state_static.json
1 // Copyright (c) 2012 The Chromium Authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license that can be
3 // found in the LICENSE file.
4
5 // This file contains the HSTS preloaded list in a machine readable format.
6
7 // The top-level element is a dictionary with two keys: "pinsets" maps details
8 // of certificate pinning to a name and "entries" contains the HSTS details for
9 // each host.
10 //
11 // "pinsets" is a list of objects. Each object has the following members:
12 //   name: (string) the name of the pinset
13 //   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
14 //   bad_static_spki_hashes: (optional list of strings) the set of forbidden
15 //     SPKIs hashes
16 //   report_uri: (optional string) the URI to send violation reports to;
17 //     reports will be in the format defined in RFC 7469
18 //
19 // For a given pinset, a certificate is accepted if at least one of the
20 // "static_spki_hashes" SPKIs is found in the chain and none of the
21 // "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
22 // match up with the file of certificates.
23 //
```



# Coded Browser Pinning


[https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport\\_security\\_state\\_static.json](https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport_security_state_static.json)

```
transport_security_state_static.json
1 // Copyright (c) 2012 The Chromium Authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license that can be
3 // found in the LICENSE file.
4
5 // This file contains the HSTS preloaded list in a machine readable format.
```



## INFOWORLD TECH WATCH

By **Fahmida Y. Rashid**, Senior Writer, InfoWorld | JAN 30, 2017

About | 


Informed news analysis every weekday

## Google moves into the Certificate Authority business

Google doesn't seem to trust the current system, as it has launched its own security certificates

# Coded Browser Pinning

[https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport\\_security\\_state\\_static.json](https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport_security_state_static.json)



transport\_security\_state\_static.json

it's not a totally insane idea -- until you realise that it appears to be completely unscalable!



it's just Google protecting itself and no one else

## Google <sup>else</sup> moves into the Certificate Authority business

Google doesn't seem to trust the current system, as it has launched its own security certificates

# Content Pinning

## HPKP

### HTTP Public Key Pinning (HPKP)

Jump to: [Enabling HPKP](#) [Specifications](#) [Browser compatibility](#) [See also](#)

[Web technology for developers](#) > [HTTP](#) >  
[HTTP Public Key Pinning \(HPKP\)](#)

**HTTP Public Key Pinning (HPKP)** is a security feature that tells a web client to associate a specific cryptographic public key with a certain web server to decrease the risk of [MITM](#) attacks with forged certificates.

Related Topics

# Content Pinning with HPKP

The issues here include

- CA migration can become really convoluted

- There appears to be a Trust on First Use issue

- A MITM attack could withhold the HPKP record, or even substitute its own

Is the effort worth it? Low deployment numbers suggest otherwise!

The Google Chrome team recently deprecated support for HPKP in Chrome because of its perceived complexity and potential side-effects.

# DNS Pinning

Where better to find out the public key associated with a DNS-named service than to look it up in the DNS?

If you are prepared to believe the DNS to give you an IP address for the service, then why wouldn't you also trust the DNS to give you the right pinning record?

(As long as you are using DNSSEC, of course!)

# CAA Pinning

- Use a DNS record to specify which CA(s) may issue a WebPKI certificate for a domain
- Specified in RFC 6844
- It's not clear how CAA protects a user
  - If a user can subvert a CA then its likely that they would also be able to subvert the CA's CAA check
  - Unless the user is also prepared to retrieve and check the CAA record then this appears to largely a palliative measure
  - But if the user checks the CAA record, then why not just use DANE?

# DANE Pinning

- Use a DNS server record to:
  - specify which CA(s) may issue a WebPKI certificate for connections to a service
  - or
  - specify which EE public key certificate should be presented to the user when connecting to a service
  - or
  - specify which public key will be used when connecting to a service

# DANE Pinning

- Use a DNS server record to:
  - specify which CA(s) may issue a WebPKI certificate for connections to a service
  - or
  - specify which certificate should be presented to the user when connecting to a service
  - or
  - specify which public key will be used when connecting to a service

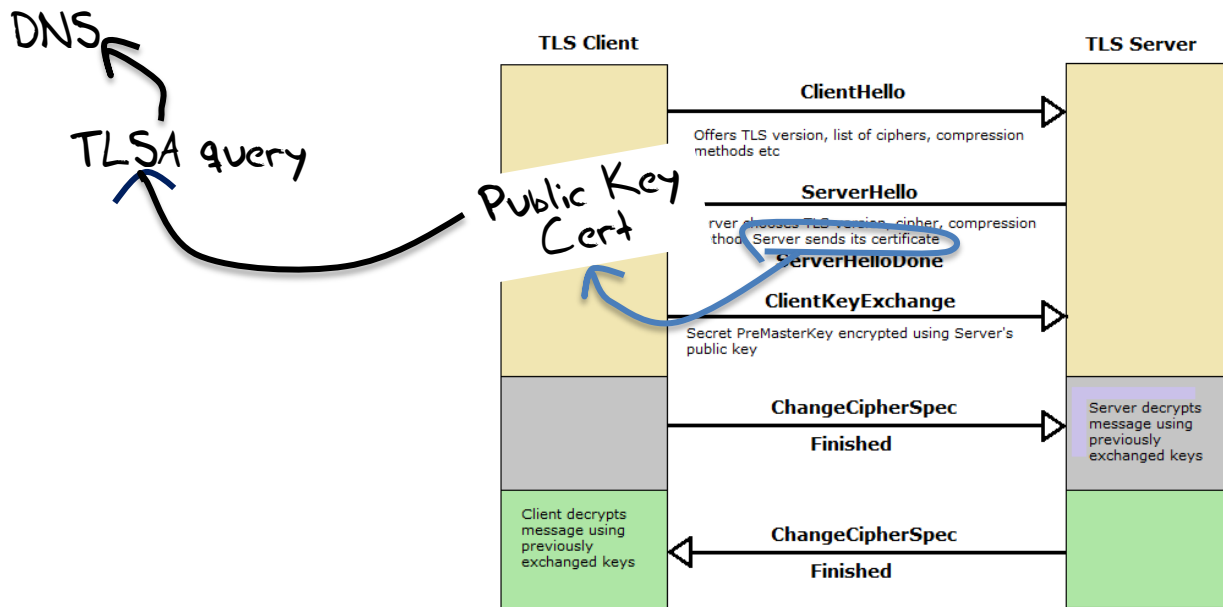
*Note that CAA is used to pin domains in the DNS while DANE is used to pin service records in the DNS*



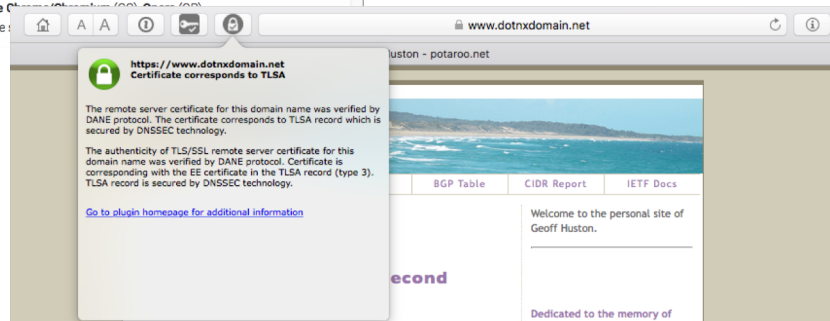
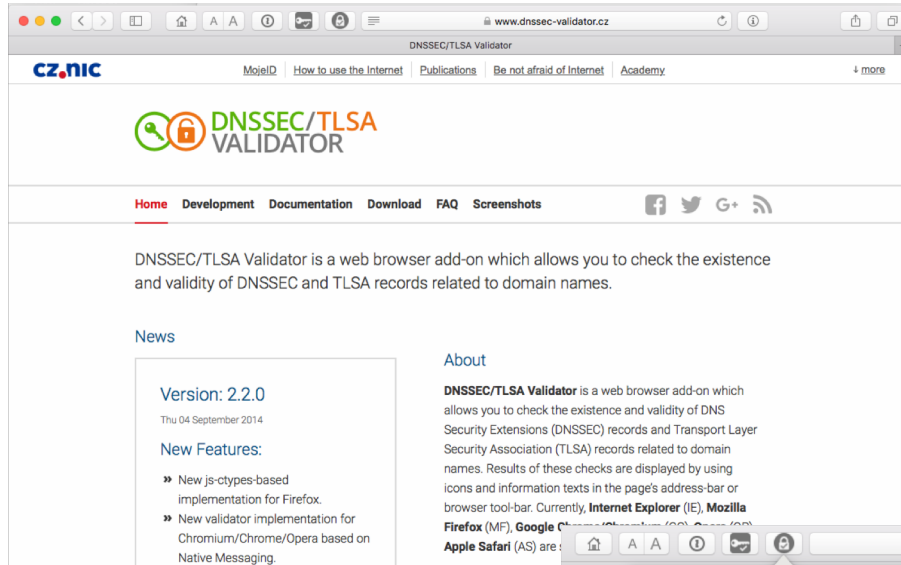
# TLS with DANE

- Client receives server cert in Server Hello
  - Client lookups the DNS for the TLSA Resource Record of the domain name
  - Client validates the presented certificate against the TLSA RR
- Client performs Client Key exchange

# TLS Connections



# DANE Does DNS via a Browser Extension



## But...

- DNSSEC as we know it today is just not good enough
- DNSSEC validation should not be outsourced to the recursive resolver - setting the AD bit in a DNS response is not good enough
- A client needs to directly validate the DNSSEC-signed DANE response
  - This requires more DNS queries
  - And this takes (too much) time
  - And we get pushback from browser vendors

# Faster DNSSEC Validation?

## RFC 7901 - CHAIN Query Requests in DNS

- Allows a client to make an “omnibus” DNS query to a recursive resolver to retrieve the set of DNSSEC RRs between the QNAME and a trust point in a single DNS transaction

# DANE as a TLS Extension?

## draft-ietf-tls-dnssec-chain-extension-07

The extension described here allows a TLS client to request that the TLS server return the DNSSEC authentication chain corresponding to its DANE record. If the server is configured for DANE authentication, then it performs the appropriate DNS queries, builds the authentication chain, and returns it to the client. The server will usually use a previously cached authentication chain, but it will need to rebuild it periodically as described in [Section 5](#). The client then authenticates the chain using a pre-configured trust anchor.

This specification is based on Adam Langley's original proposal for serializing DNSSEC authentication chains and delivering them in an X.509 certificate extension [[I-D.agl-dane-serializechain](#)]. It modifies the approach by using wire format DNS records in the serialized data (assuming that the data will be prepared and consumed by a DNS-specific library), and by using a TLS extension to deliver the data.

As described in the DANE specification [[RFC6698](#)] [[RFC7671](#)], this procedure applies to the DANE authentication of X.509 certificates or raw public keys [[RFC7250](#)].

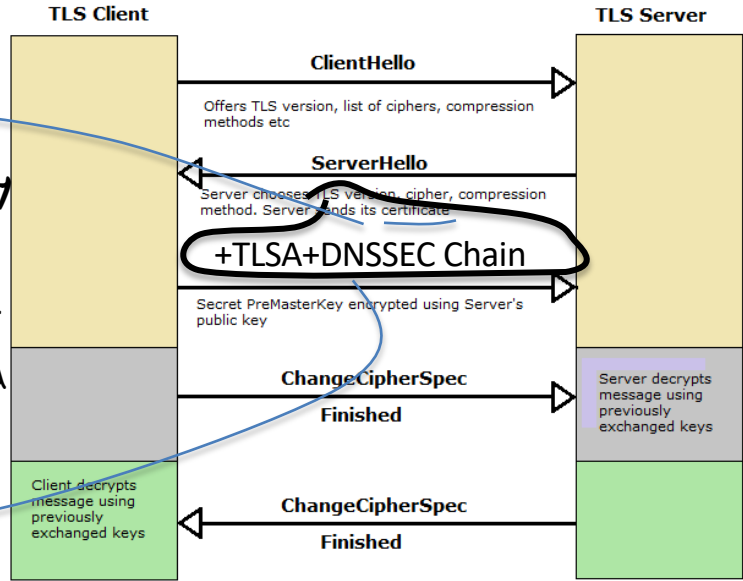
# TLS + DANE Chain Connections

Local copy of DNS root zone KSK

DNSSEC validation

Public Key Cert

DANE CERT Pinning record



# What now?

It appears that we still need WebPKI certs for the moment, but we need to make them more robust in the face of continued attack

- DANE+DNSSEC could be useful in adding assurance to the WebPKI in a role of WebPKI CA pinning
- So far we have not figured out how to reliably catch instances of withholding a DNS TLS extension without paying a DNS query time delay penalty
  - Which implies that DANE TLS extension probably represents one more thing to go wrong without a compelling case that can be made about what it actually manages to do to protect the user
  - Or we can work out a way to catch withholding efficiently



# Conclusions

Corrupting a trusted CA is a nightmare scenario for the WebPKI

- DANE appears to offer a natural and compelling alternative to the WebPKI by offering a dynamic system that provides authenticated data to the user that does not rely on expansive trust
- But there are some issues that exist in the DNS, DNSSEC and DANE
  - Registry practices to ensure that there are very robust defences against domain name hijacking are lacking today and will be lacking tomorrow
  - Centralising trust in a single model creates a single point of vulnerability for the entire system
  - The KSK model is fragile
  - Overloading the DNS with large payloads stresses the UDP-based system beyond their viability, but the case to justify shift to DNS over <X> architectures has a limited value proposition outside of DNSSEC/DANE-based use cases

Thanks