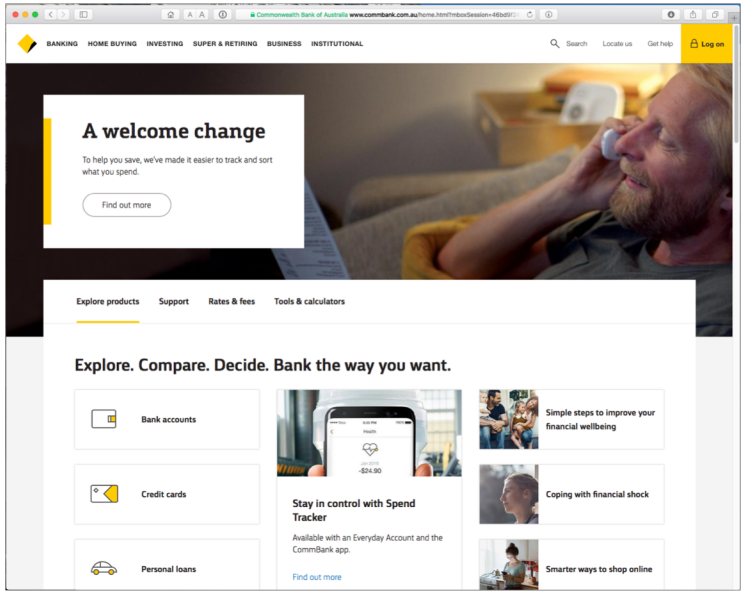
# Who am I talking to?

# Who am I talking to?

What's the Problem?

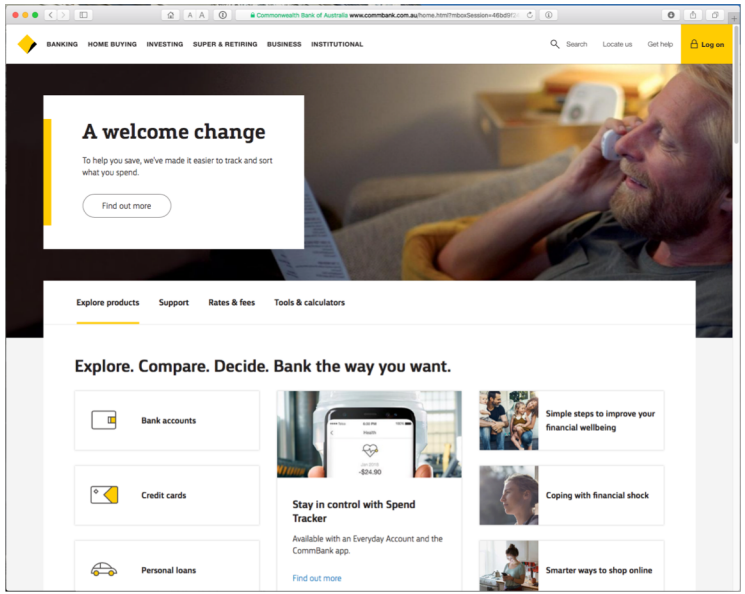
# Which Bank? My Bank!




it looks like my bank
But is it my bank?

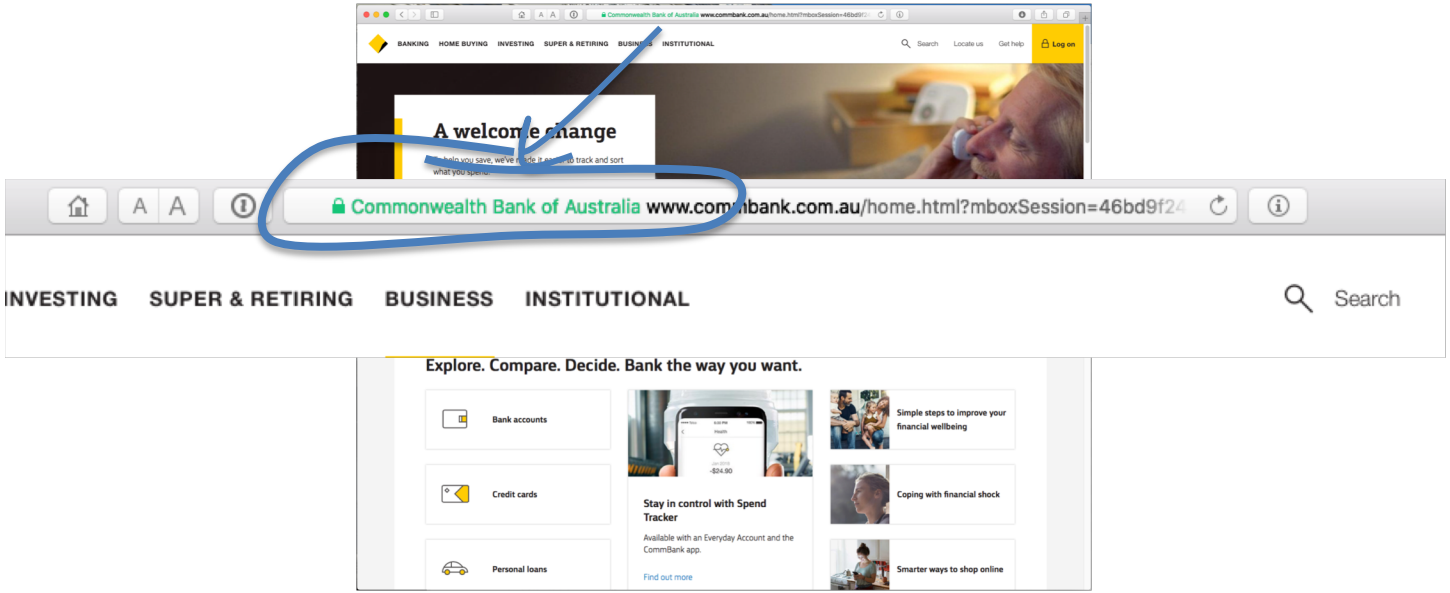
# The Question:

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

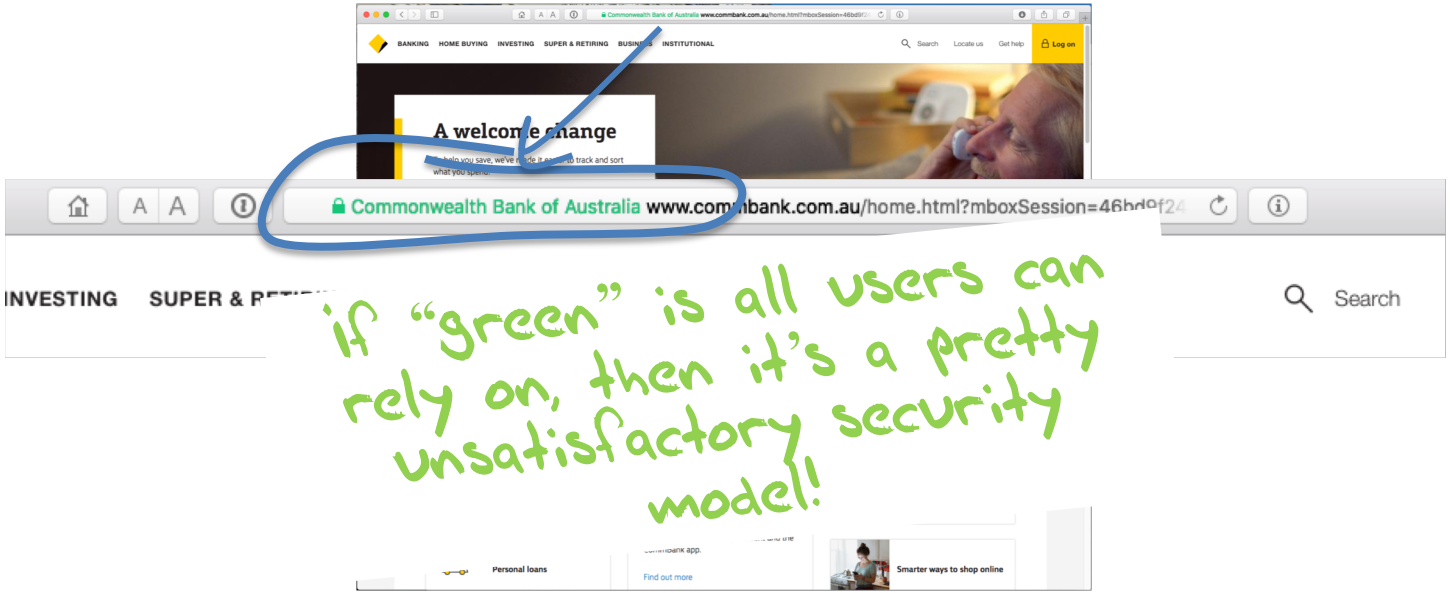



it looks like my bank
But is it my bank?

# A Clue!

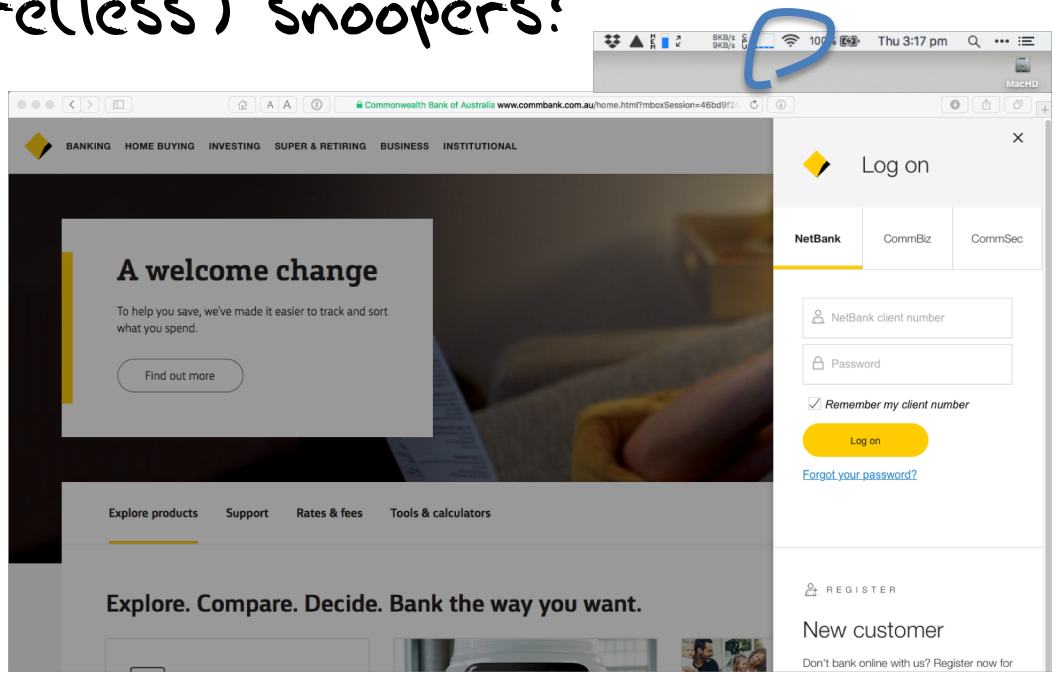

BANKING
HOME BUYING
INVESTING
SUPER & RETIRING
BUSINESS
INSTITUTIONAL
Search
Locate us
Get help
Log on
A welcome change
Commonwealth Bank of Australia www.commbank.com.au/home.html?mboxSession=46bd9f24
INVESTING
SUPER & RETIRING
BUSINESS
INSTITUTIONAL
Search
Explore. Compare. Decide. Bank the way you want.
Bank accounts
Credit cards
Personal loans
Stay in control with Spend Tracker
Available with an Everyday Account and the CommBank app.
Find out more
Simple steps to improve your financial wellbeing
Coping with financial shock
Smarter ways to shop online

# A Clue!


BANKING
HOME BUYING
INVESTING
SUPER & RETIRING
INSTITUTIONAL
Search
Locate us
Get help
Log on
A welcome change
Commonwealth Bank of Australia
INVESTING
Search
if "green" is all users can rely on, then it's a pretty unsatisfactory security model!
Personal loans
Find out more
Smarter ways to shop online

# 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…

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE FORU

BORDER GATEWAY PROTOCOL ATTACK —

## Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency

Almost 1,300 addresses for Amazon Route 53 rerouted for two hours.

DAN GOODIN - 4/25/2018, 5:00 AM

amazon.com®

Amazon

123

Amazon lost control of a small number of its cloud services IP addresses for two hours on Tuesday morning when hackers exploited a known Internet-protocol weakness that let them to redirect traffic to rogue destinations. By subverting Amazon's domain-resolution service, the attackers masqueraded as cryptocurrency website MyEtherWallet.com and stole about $150,000 in digital coins from unwitting end users. They may have targeted other Amazon customers as well.

The incident, which started around 6 AM California time, hijacked 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

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

Just 2 hours was enough!

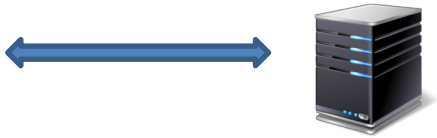
And numerous other incidents

# 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

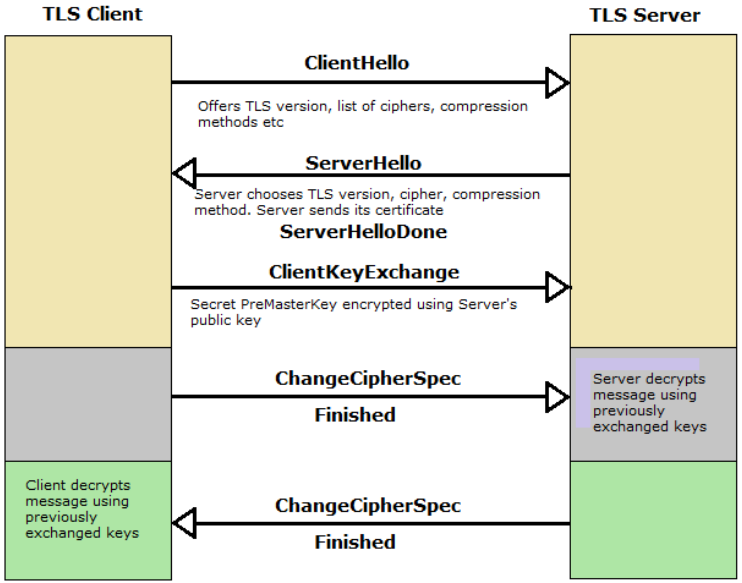
And i am NOT a customer of the internet Bank of Akamai!

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 tricker question…

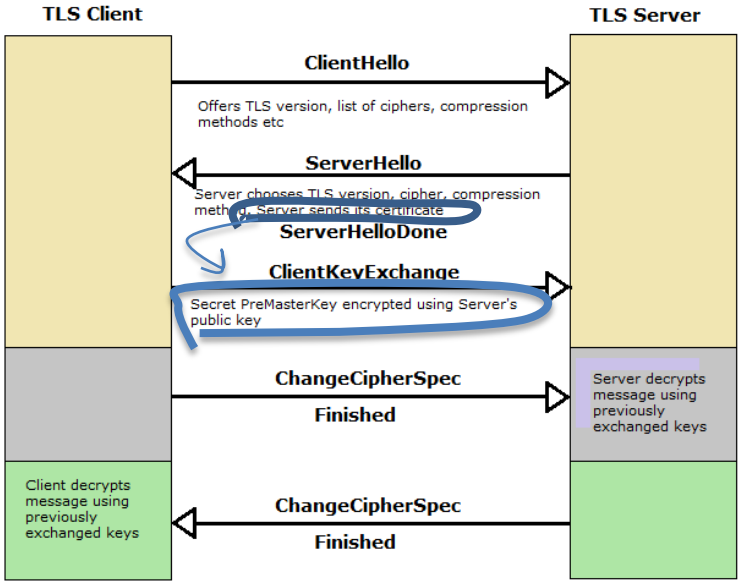
How can my browser tell the difference between an intended truth and a lie?

# Secure Connections using TLS 1.2



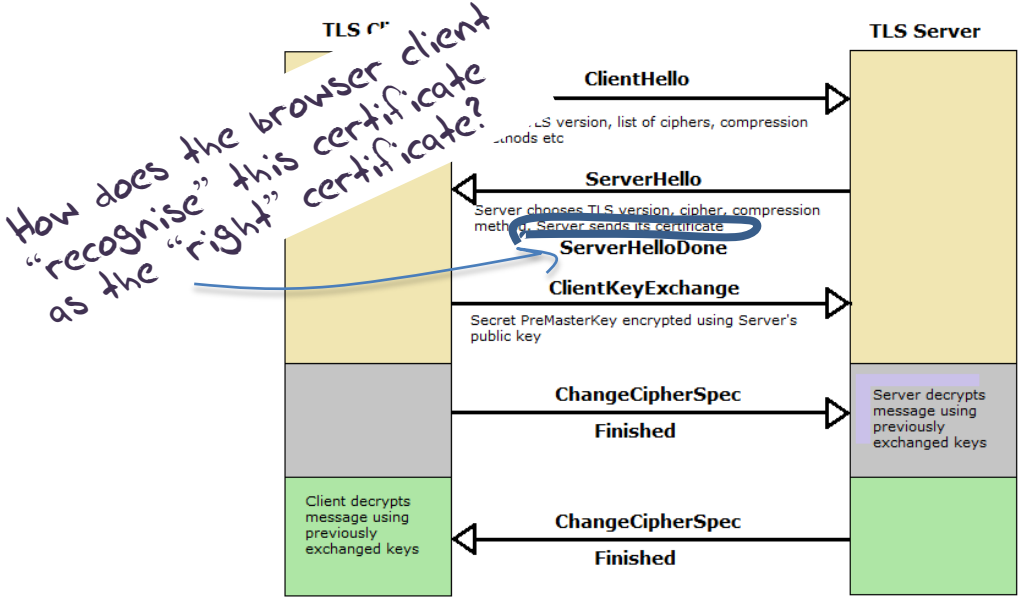

https://rhsecurity.wordpress.com/tag/tls/

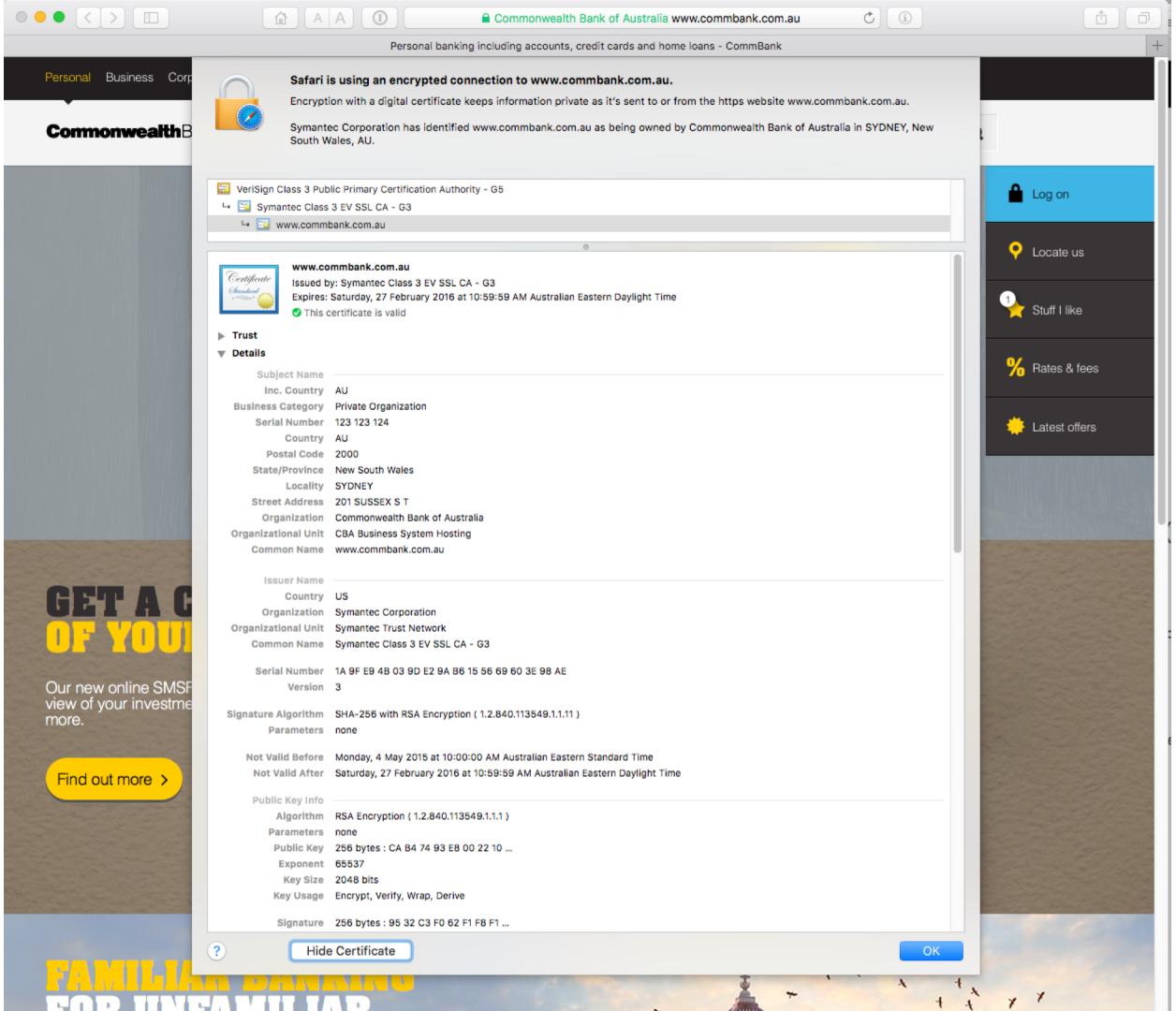
# Secure Connections using TLS 1.2

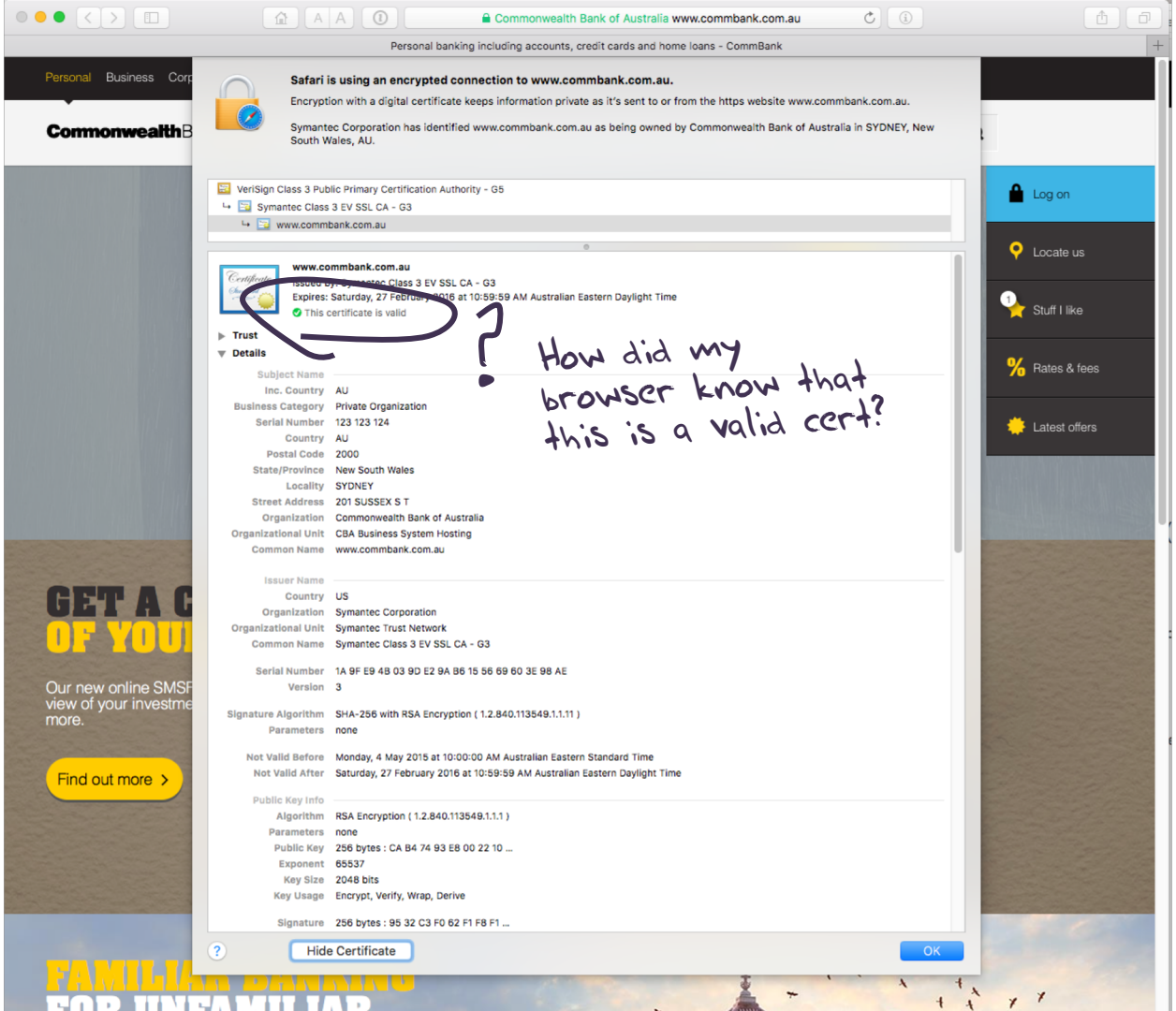



https://rhsecurity.wordpress.com/tag/tls/

# Secure Connections using TLS 1.2




Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Personal
Business
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 ( 1.2.840.113549.1.1.11 )
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 & fees
Latest offers
Find out more

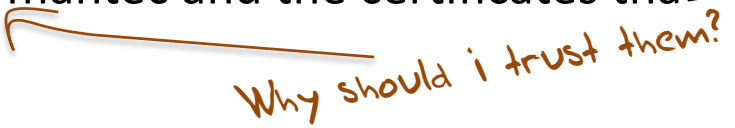

Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Personal
Business
Safari is using an encrypted connection to www.commbank.com.au.
Encryption with a digital certificate keeps information private as it's sent to or from the 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
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 ( 1.2.840.113549.1.1.11 )
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
How did my browser know that this is a valid cert?
Log on
Locate us
Stuff I like
Rates & fees
Latest offers
Find out more

# Domain Name Certification

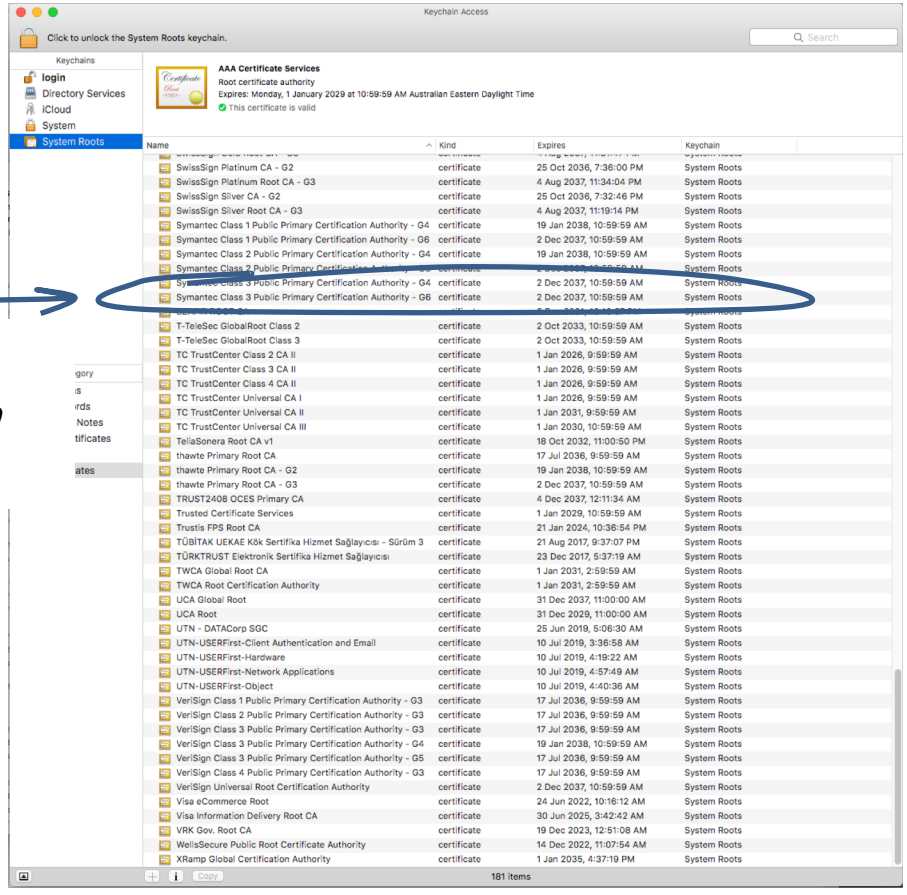
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called "Symantec"
- Who was willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value
- So if I can 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, 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 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, as long as I am prepared to trust Symantec and the certificates that they issue
- Symantec NEVER lie!



# Local Trust

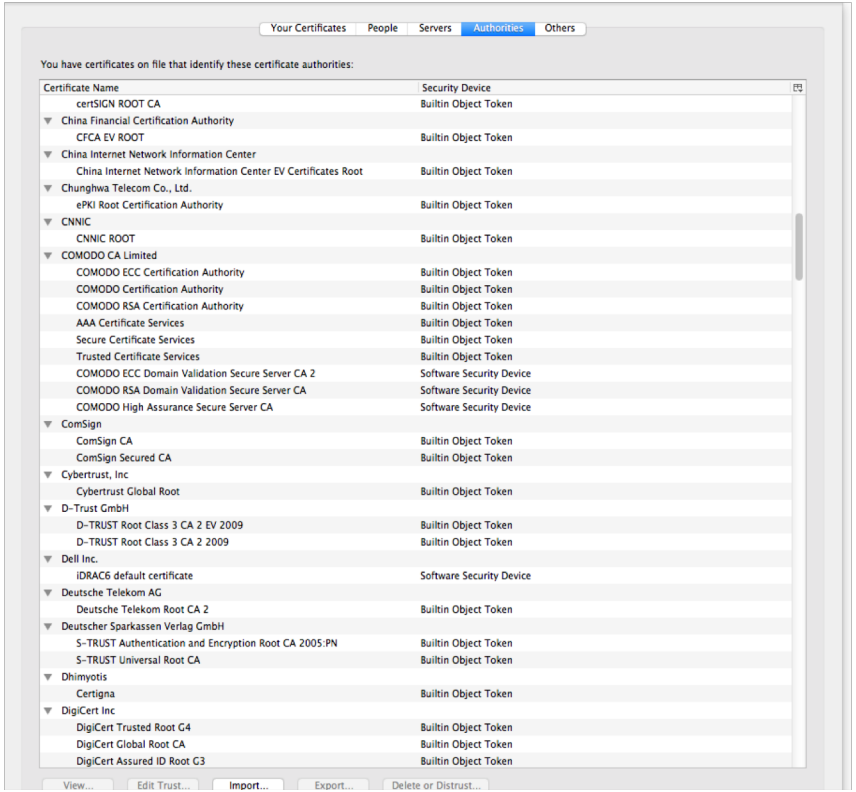



The cert I'm being asked to trust was issued by a certification authority that my browser already trusts – so I trust that cert!

# Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?



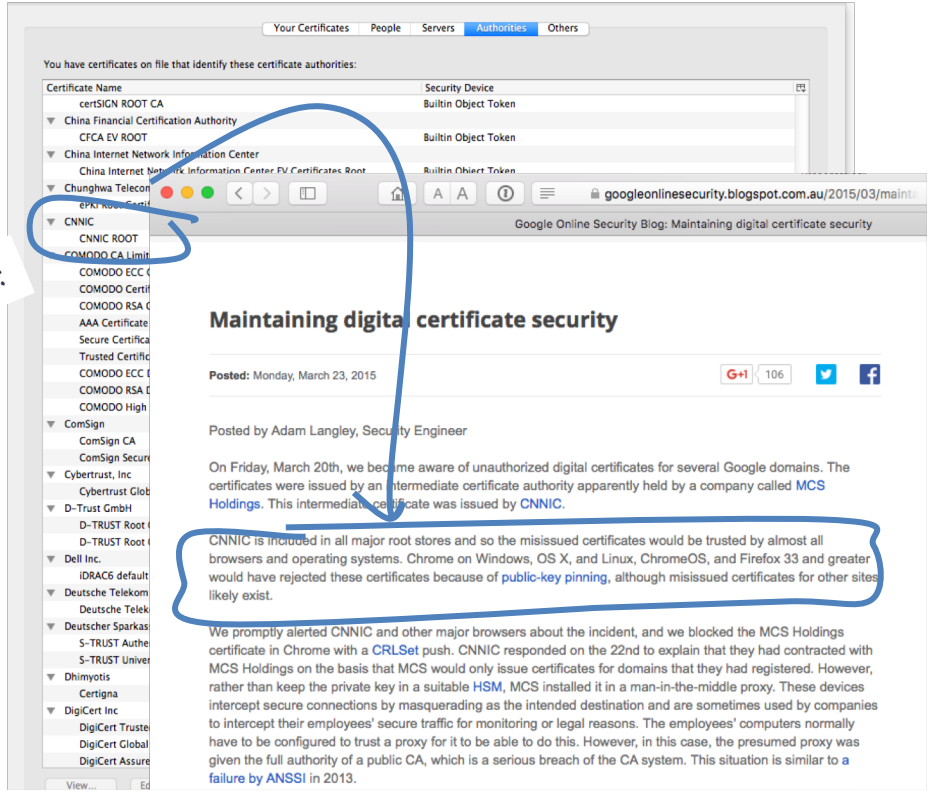

* cre·du·li·ty
/krəˈd(y)o͞olədē/
*noun*
a tendency to be too ready to believe that something is real or true.

# Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?

Evidently Not!



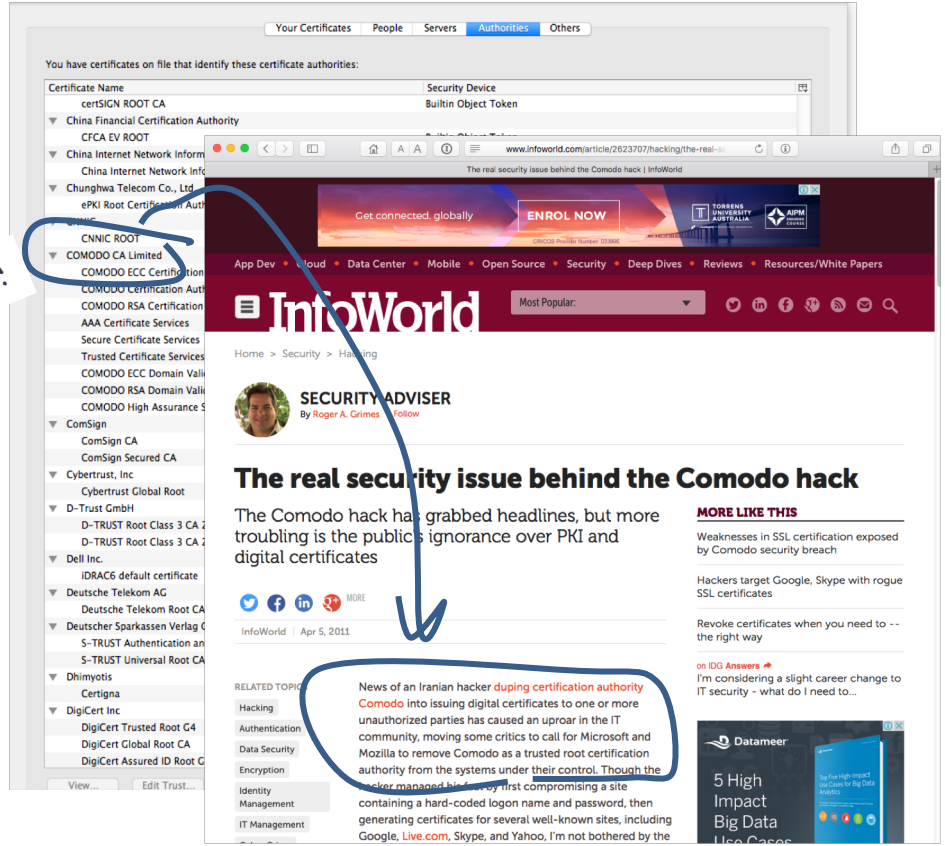

* cre·du·li·ty
/krəˈd(y)o͞olədē/
*noun*
a tendency to be too ready to believe that something is real or true.

# Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?

Evidently Not!



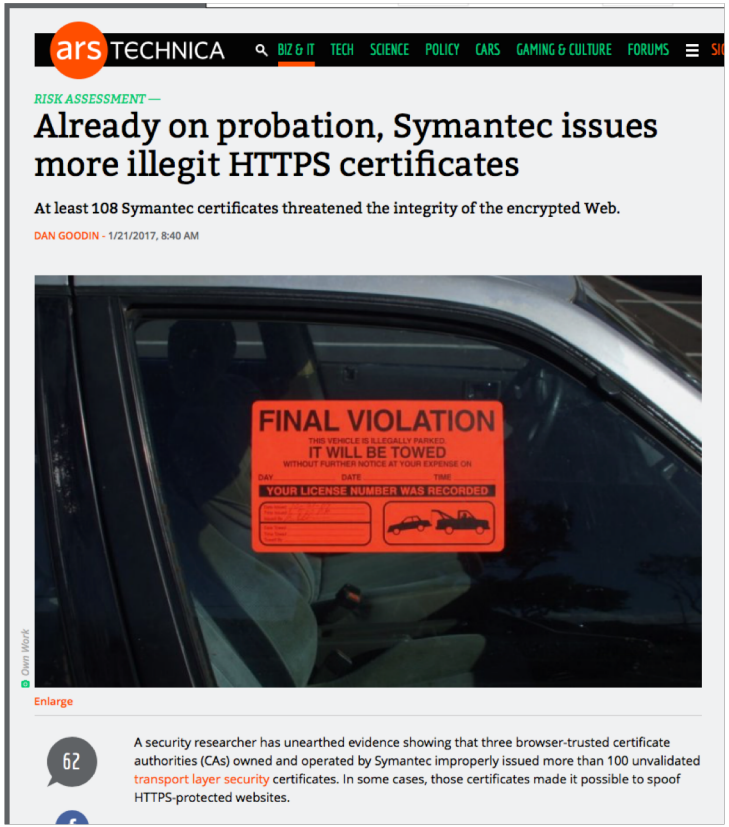

* **cre·du·li·ty**
/krəˈd(y)o͞olədē/
*noun*
a tendency to be too ready to believe that something is real or true.

# 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

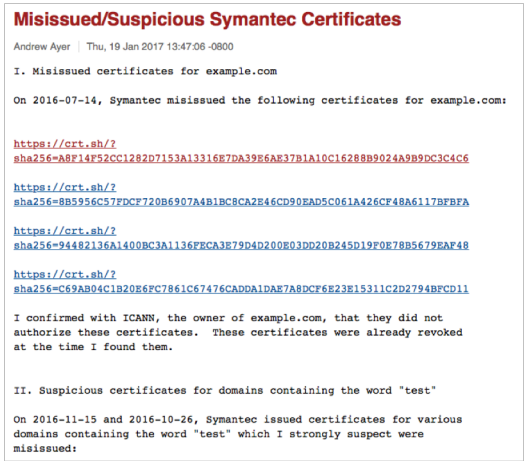


Enlarge

62

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=8B5956C57FDCF720B6907A4B1BC8CA2E46CD90EAD5C061A426CF48A6117BFBFA

https://crt.sh/?sha256=94482136A1400BC3A1136FECA3E79D4D200E03DD20B245D19F0E78B5679EAF48

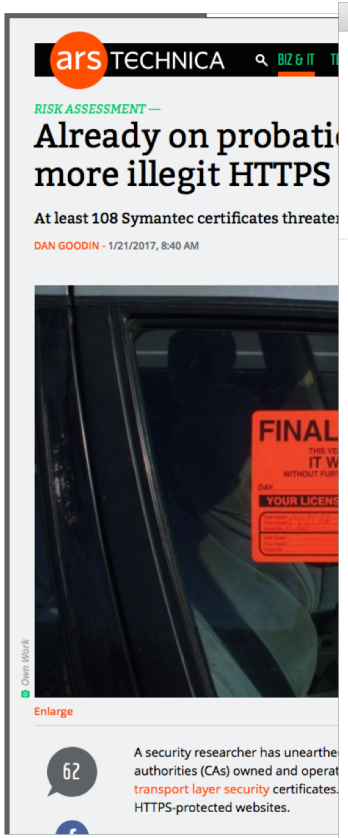
https://crt.sh/?sha256=C69AB04C1B20E6FC7861C67476CADDA1DAE7A8DCF6E23E15311C2D2794BFCD11

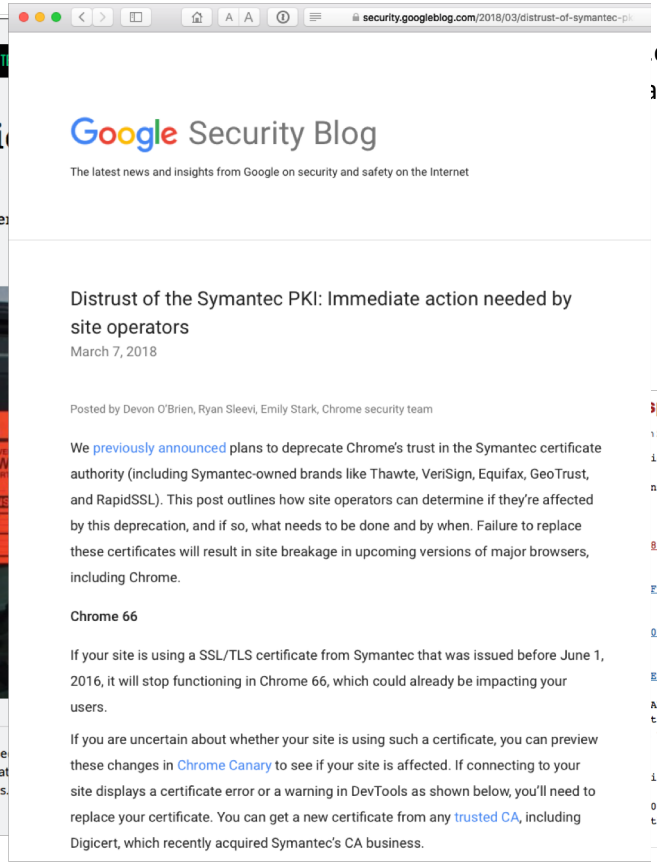
I confirmed with ICANN, the owner of example.com, that they did not authorize these certificates. These certificates were already revoked at the time I found them.

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

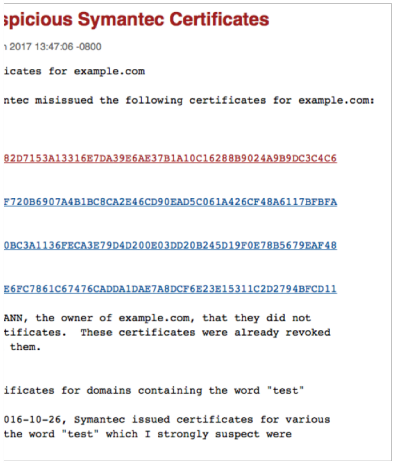
On 2016-11-15 and 2016-10-26, Symantec issued certificates for various domains containing the word "test" which I strongly suspect were misissued:

# Well, hardly ever




.com/security/2017/01/already-
ntec-issues-more-illegit-https-



# 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

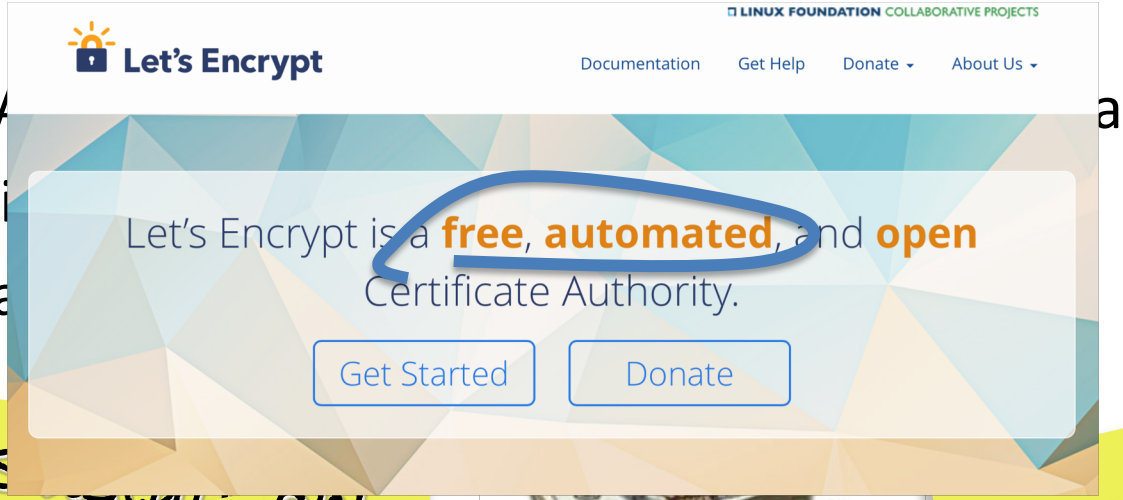
Than what 'wins' in the market?

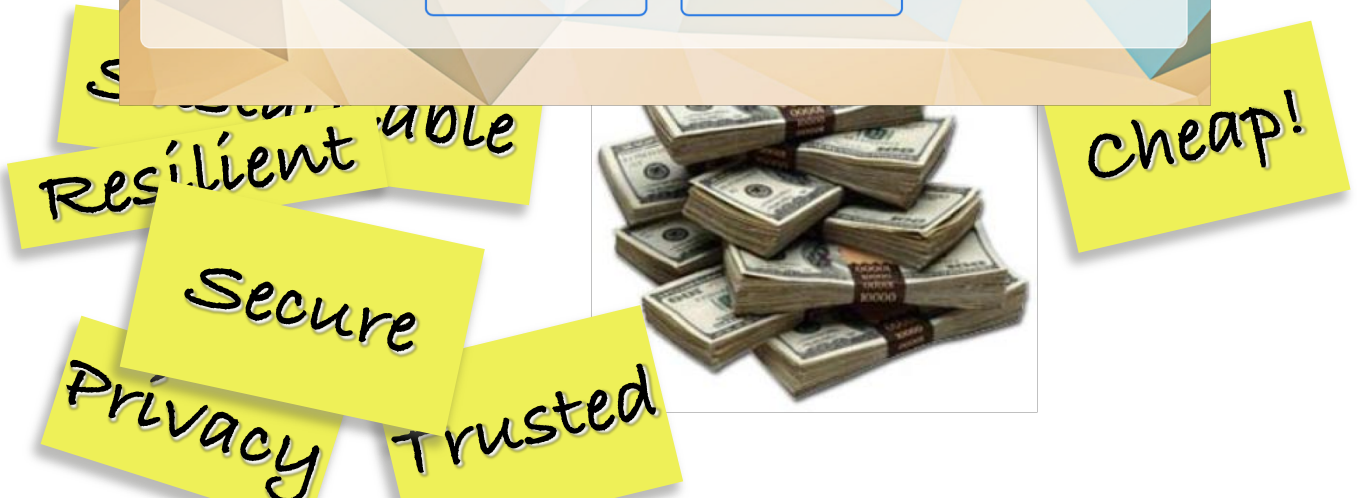
# In a market for security

Where CA … are

And quali…

Than wha…



# 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

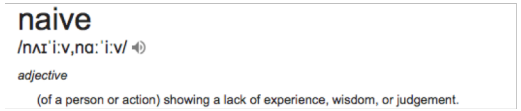
- analogous to blockchain for each CA, but with a centralised authority model

# 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



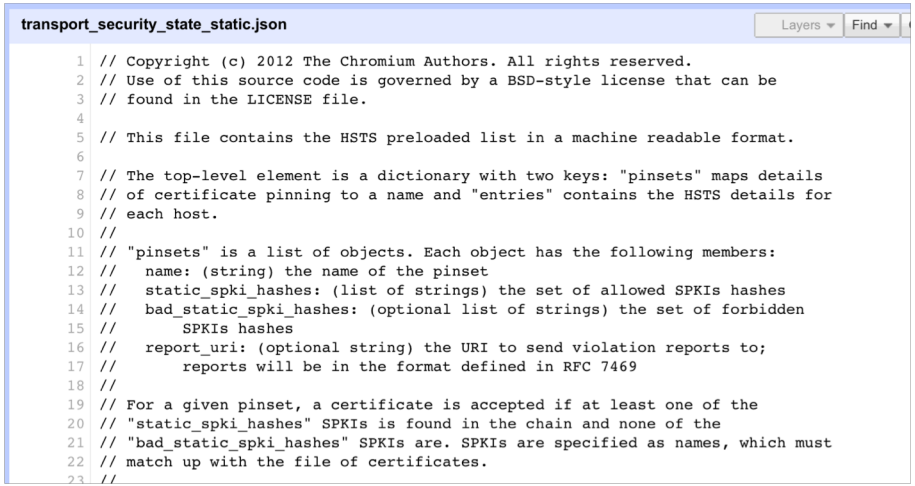
# Pinning: Narrowing the Trust Space

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



# Coded Browser Pinning

https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport_security_state_static.json

transport_security_state_static.json

Layers ▾ Find ▾

```
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

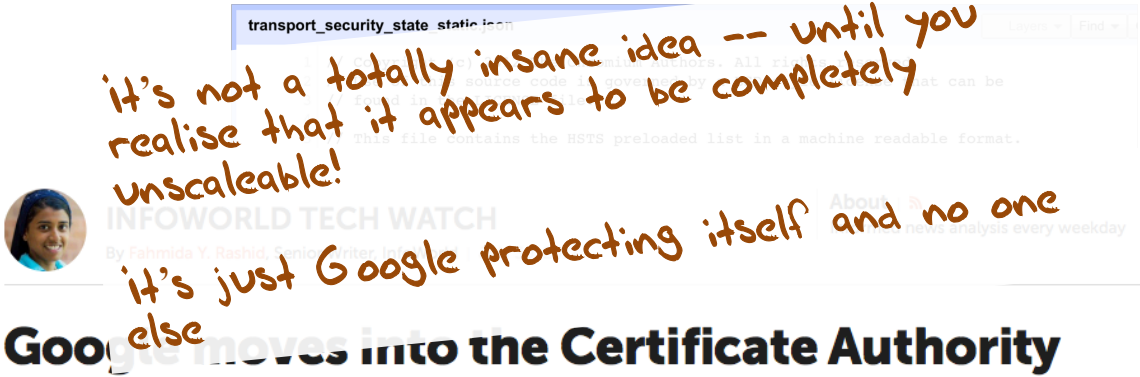
Informed news analysis every weekday

## 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

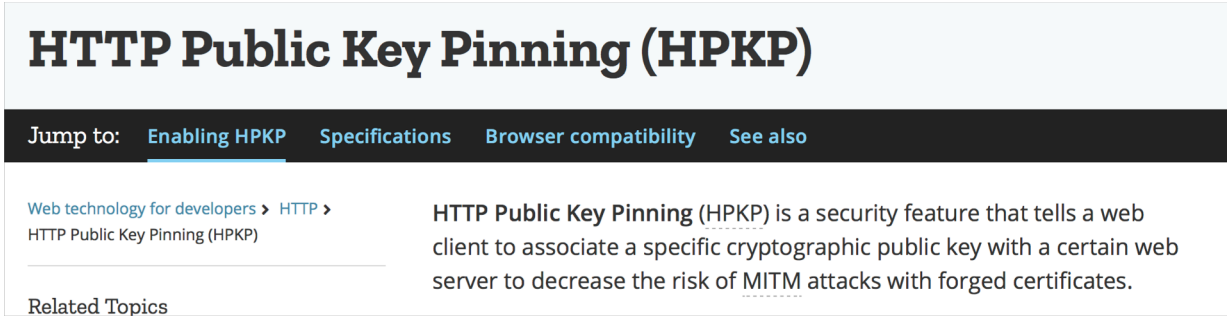



it's not a totally insane idea -- until you realise that it appears to be completely unscaleable!

it's just Google protecting itself and no one else

# 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)

Related Topics

**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.

# 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 public key certificate should be presented to the user when connecting to a service

  or

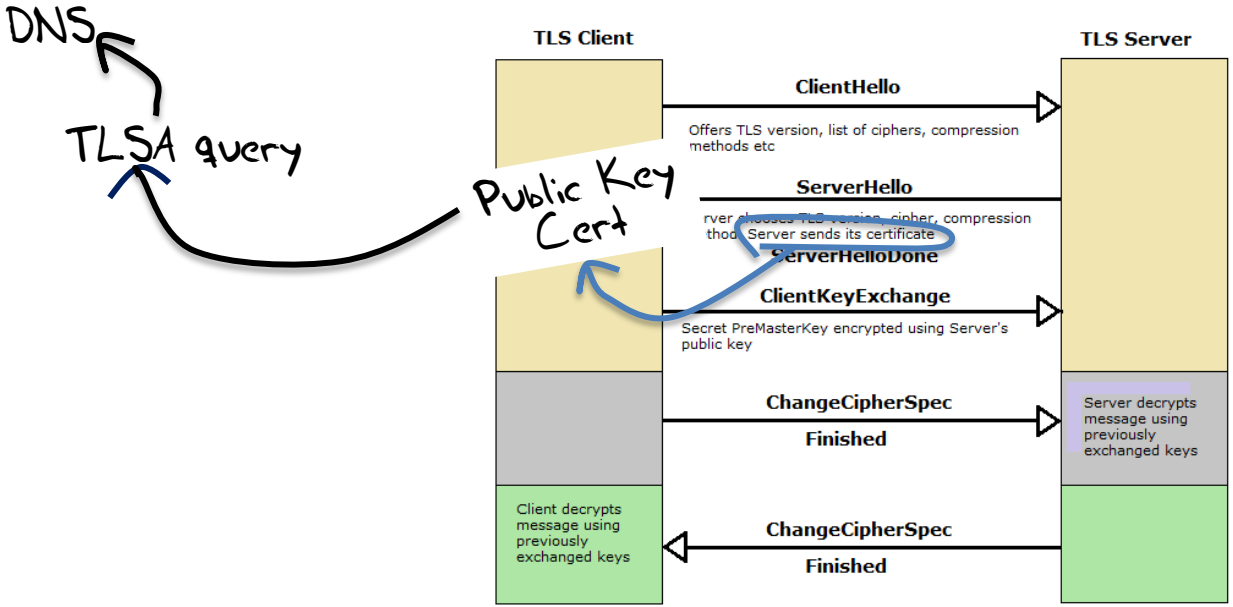
  - specify which public key will be used when connecting to a service

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

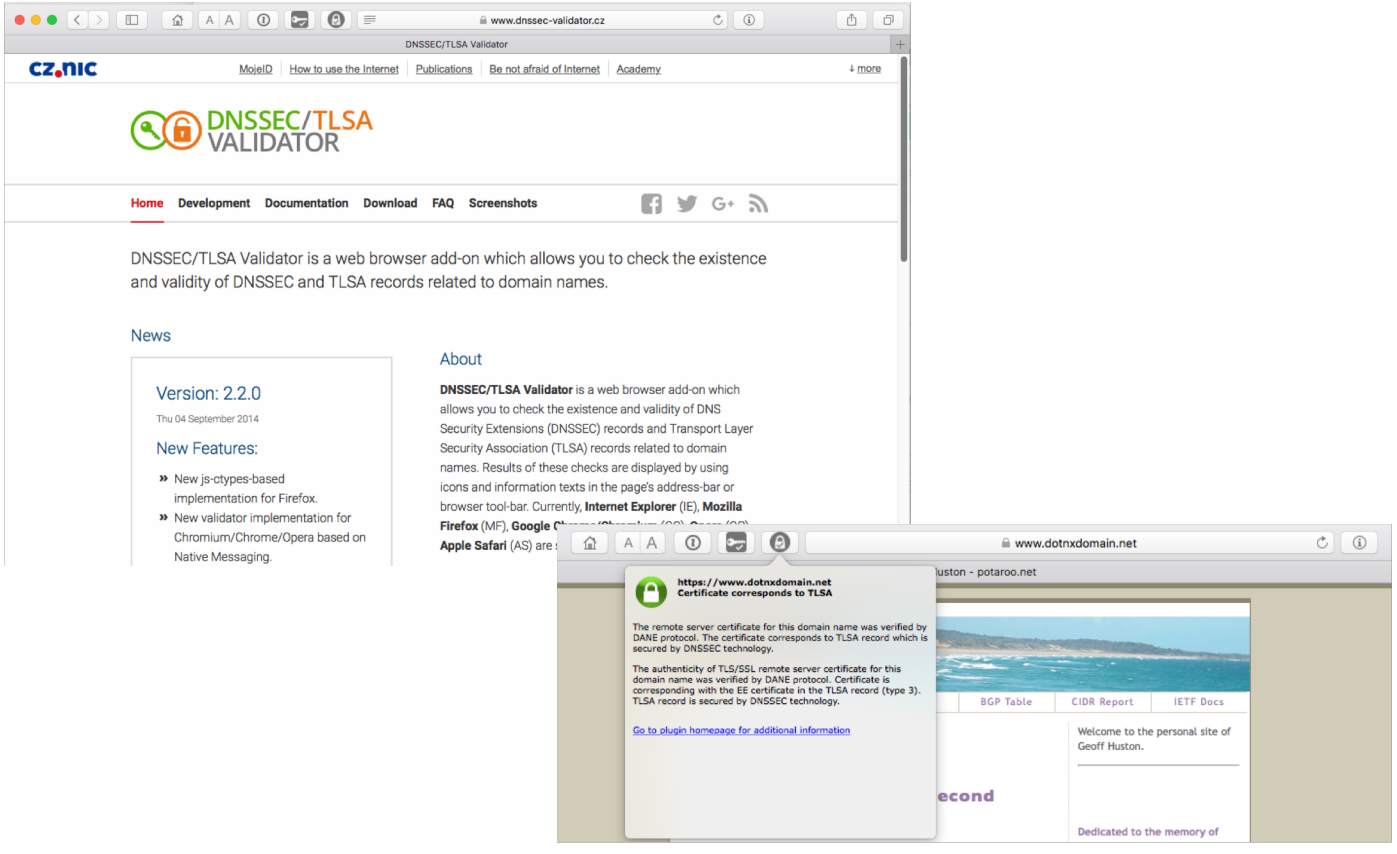
# TLS with DANE

- 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 vendoras

# 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?

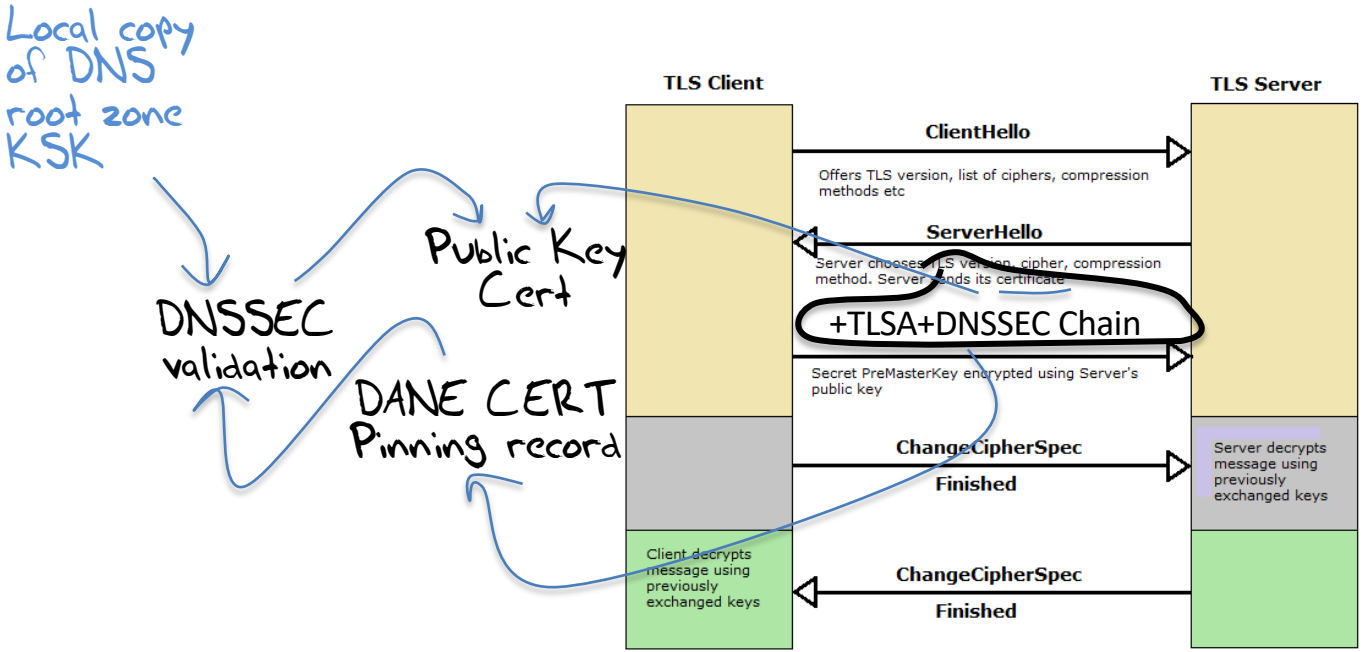
## draft-ietf-tls-dnssec-chain-extension-07

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



# 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 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