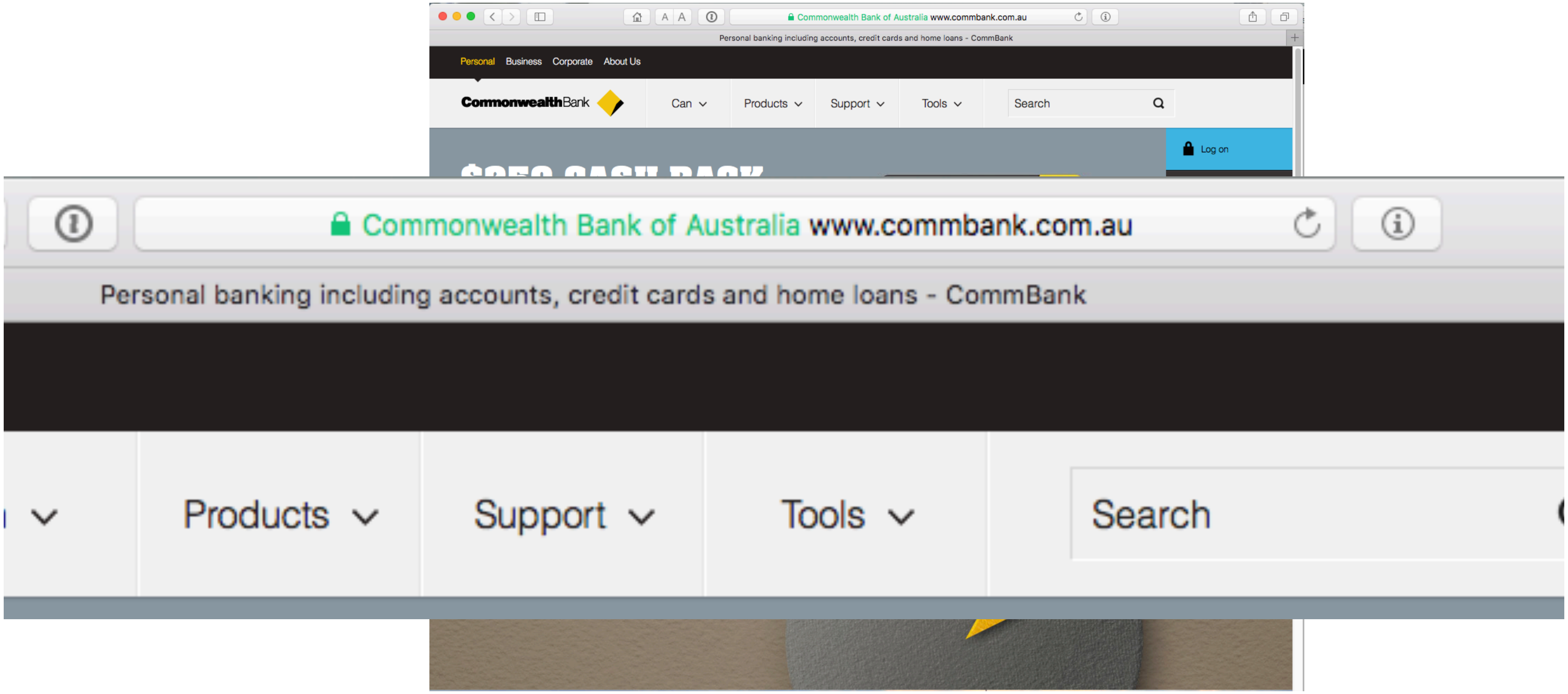
Why Dane?

Geoff Huston
Chief Scientist, APNIC

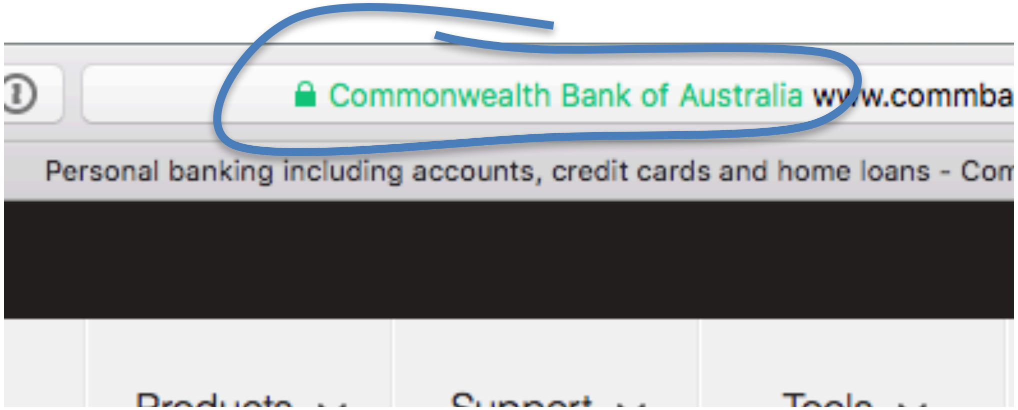
Security on the Internet

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


Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Personal
Business
Corporate
About Us
CommonwealthBank
Can
Products
Support
Tools
Search
Log on
Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Products
Support
Tools
Search

Security on the Internet

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



Commonwealth Bank of Australia www.commba

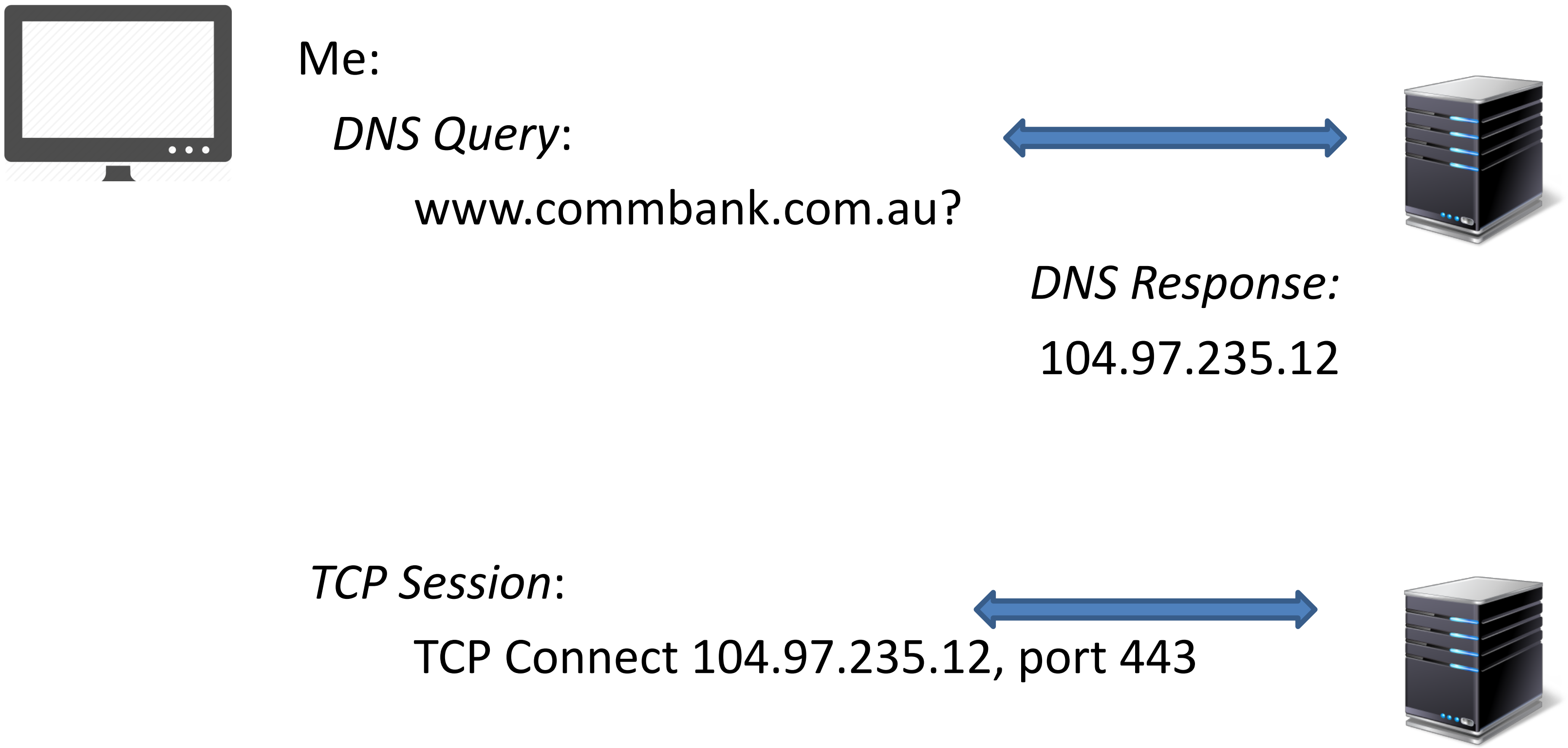
Personal banking including accounts, credit cards and home loans - Com

My Bank's web site

Or at least i think its my bank because it looks a bit familiar and there is a green icon of a lock

So it HAS to be my bank – hasn't it?

Connection Steps



Me:

DNS Query:

www.commbank.com.au?

DNS Response:

104.97.235.12

TCP Session:

TCP Connect 104.97.235.12, port 443

Hang on…

```
$ dig -x 104.97.235.12 +short
a104-97-235-12.deploy.static.akamaitechnologies.com.
```

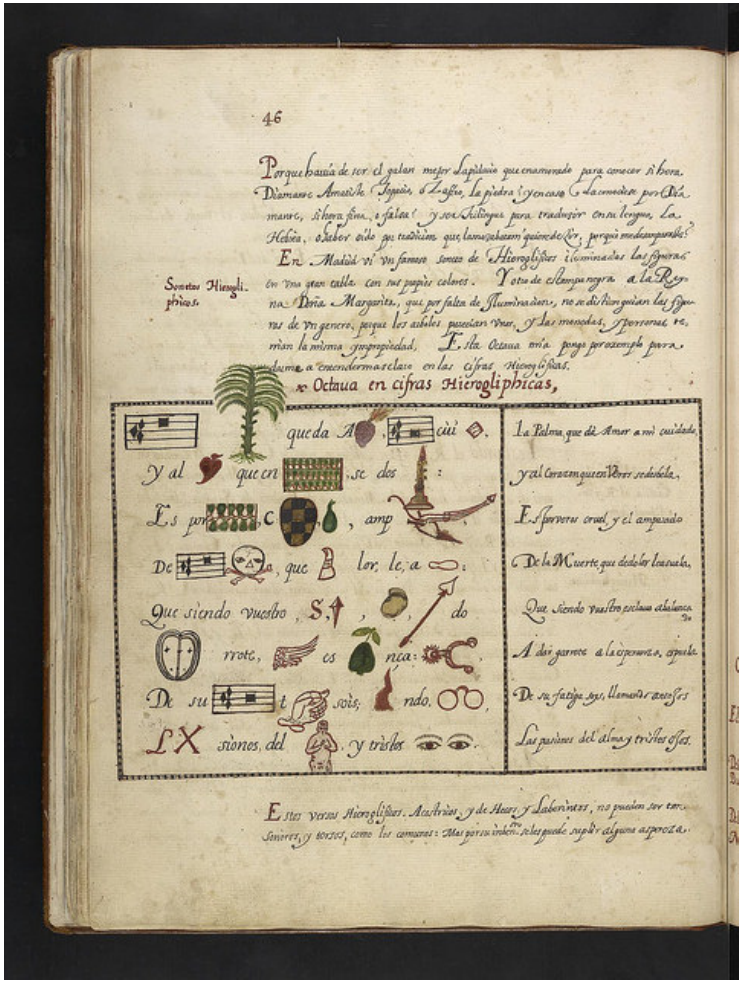
That’s not an IP addresses that was allocated to the Commonwealth Bank!

The Commonwealth Bank of Australia has 140.168.0.0 - 140.168.255.255 and 203.17.185.0 - 203.17.185.255

So why should my browser trust that 104.97.235.12 is really the “proper” web site for the Commonwealth Bank of Australia and not some dastardly evil scam?

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

It's all about cryptography


queda A , cüi .
y al que en se dos :
Es por c , amp
De , que lor, le, a :
Que siendo vuestro , S , do
rrote, es nca:
De su t sois, ndo.
LX siones, del , y tristes .

46

Sonetos Hieroglificos.

Porque hauia de ser el galan mejor Lapidario que enamorado para conocer si hera Diamante, Amatiste, Topacio, o Zafiro, la piedra que en caso se la comediese por Diamante, si hera fina, o falsa? y ser Chilingue para traduzir en su lengua, la Hebrea, o saber oido por tradicion que lo [illegible] quiere dezir, porque me desamparable?

En Madrid vi un famoso soneto de Hieroglificos iluminadas las figuras en una gran tabla con sus propios colores. Y otro de estampa negra a la Reyna Doña Margarita, que por falta de Iluminacion, no se distinguian las figuras de un genero, porque los arboles parecian unos, y las monedas, y piedras, tenian la misma impropiedad. Esta Octaua mia pongo por exemplo para darme a entender mas claro en las cifras Hieroglificas.

Octaua en cifras Hieroglificas,

La Palma, que dà Amor a mi cuidado,
y al Coraçon que en Veros se desuela,
Es feruores cruel, y el ampeçado
De la Muerte, que de dolor le asuela,
Que siendo vuestro esclauo abalança do
A dar garrote a la esperança, espuela
De su fatiga soys, llamando a excessos
Las pasiones del alma y tristes ojos.

Estos versos Hieroglificos, Acrosticos, y de Ecos, y Laberintos, no pueden ser tan sonoros, y tersos, como los comunes: Mas por su ingenio se les puede suplir alguna aspereza.

https://www.flickr.com/photos/130418531@N02/16782188377/in/photolist-ryZ6Vt-Qbdz2s-xBDAPD-osGvgu-odevZf-osVbe3-b2F4k-osR1qv-rG1BcF-ot8gpv-xjqvRQ-wEavwc-xjqD3o-xB2Zk6-xjqoVA-xyHzRm-xBDnyF-xjqBou-wE1UH1-xyHr6L-osVboS-xjqwbh-wE1XvW-xBDvxg-xjwNyK-wEavWR-wEax2g-wE28Ss-xB38Vx-wEat5i-ot6Muw-wEamXZ-wE1Unb-xjqBuU-xjwQqa-ouT45p-xjwM8i-osQZSM-ot8g2M-ouT4SB-ouT4zH-or6o99-wEaxH6-xjwYpr-xjqBPS-xyHxUA-xB36sM-xjwTvD-xAdeTN-xjqyrC

The Basic Challenge

Pick a pair of keys such that:

- Messages encoded with one key can only be decoded with the other key
- Knowledge of the value of one key does not infer the value of the other key

https://www.flickr.com/photos/matthew_lancaster_2/16236820919/in/photolist-qJMXbg-k2DdCg-bzq2E4-pMr5jw-6dqcYA-dA2qD7-hQia7n-rKbWt5-8cGJjp-4LF4GB-aFRdWK-ozhoRq-crZRwJ-oxSYk7-8cGBVH-8cKWKQ-oRt26z-k2GHfS-pTut5d-8cGG9n-nkNjxY-diYG5E-pMAb9Q-dePaJq-cNyBTL-q9gPnG-qJqEK8-8cGCbT-8cGCjt-e19ouA-hvDKJv-otkJrQ-rPCDtZ-dzVTon-oo9QU2-bvEeLd-QdhGwY-pJGKSU-ommZQ5-ozgY5v-cy1Wiy-yoTFCD-y6zvq9-ovkUxR-y6EPqD-8cKVxw-yobvEZ-8cGDVV-xr9Zmo-y6yrNo

RSA

Select two large (> 256 bit) prime numbers, *p* and *q, then:*

$n = p.q$

$\phi(n) = (p-1).(q-1)$ *(the number of numbers that are relatively prime to n)*

Pick an *e* that is relatively prime to $\phi(n)$

The PUBLIC KEY is <e,n>

Pick a value for *d* such that $d.e = 1 \bmod \phi(n)$

The PRIVATE KEY is <d,n>

For any x, $x^{de} \equiv x \pmod{n}$

The Power of Primes

$$(m^e)^d = (m^d)^e \equiv m \pmod{n}$$

As long as *d* and *n* are relatively large, and *n* is the product of two large prime numbers, then finding the value of *d* when you already know the values of *e* and *n* is computationally expensive

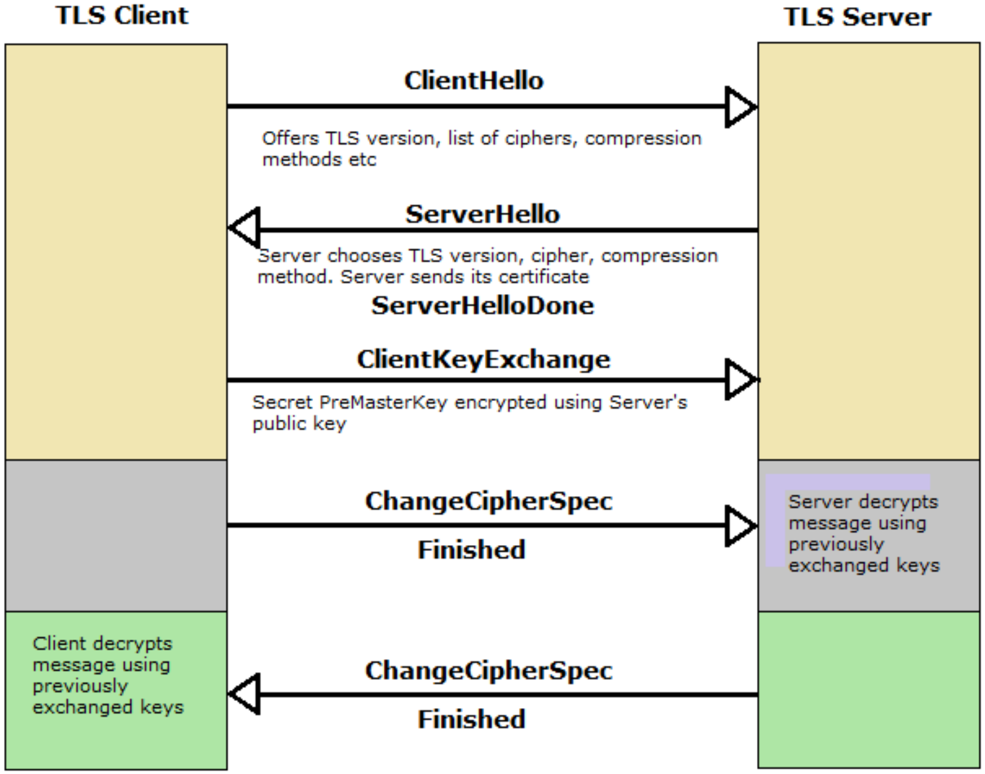
Now d.e = 1 mod ϕ(n)

So if you know ϕ(n), then you can calculate d

But ϕ(n) = (p-1).(q-1), where p.q = n

You need to find the prime factors of *n,* a very large composite number that is the product of two primes

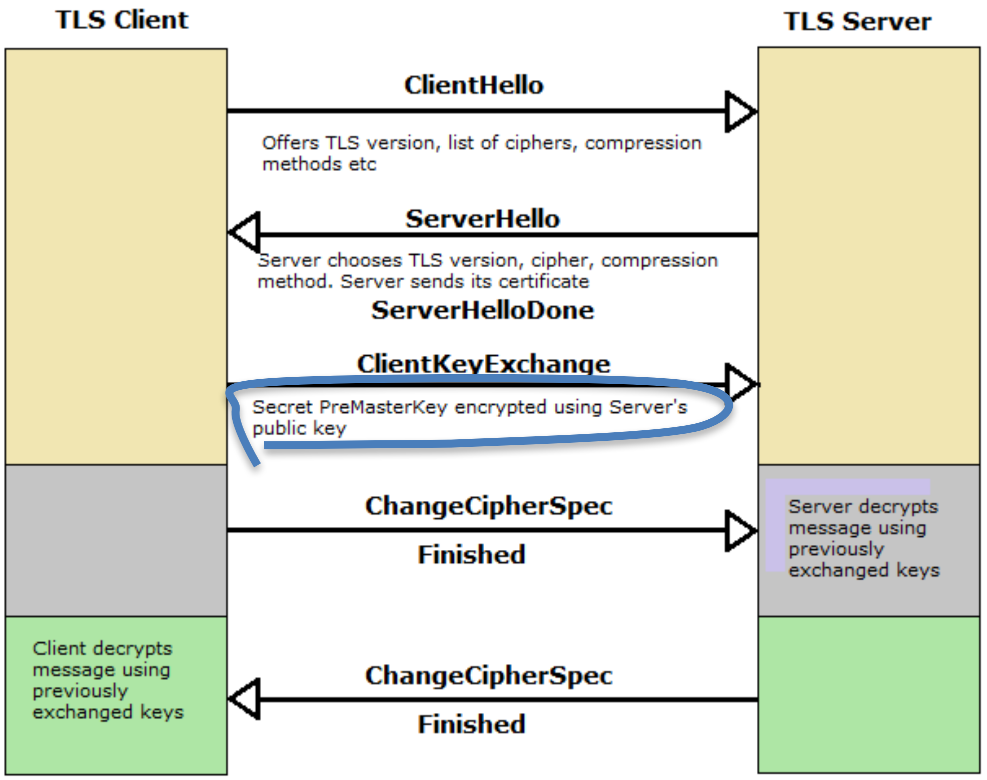
TLS Connections



TLS Client
TLS Server
ClientHello
Offers TLS version, list of ciphers, compression methods etc
ServerHello
Server chooses TLS version, cipher, compression method. Server sends its certificate
ServerHelloDone
ClientKeyExchange
Secret PreMasterKey encrypted using Server's public key
ChangeCipherSpec
Finished
Server decrypts message using previously exchanged keys
Client decrypts message using previously exchanged keys
ChangeCipherSpec
Finished

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

TLS Connections



TLS Client

TLS Server

ClientHello

Offers TLS version, list of ciphers, compression methods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

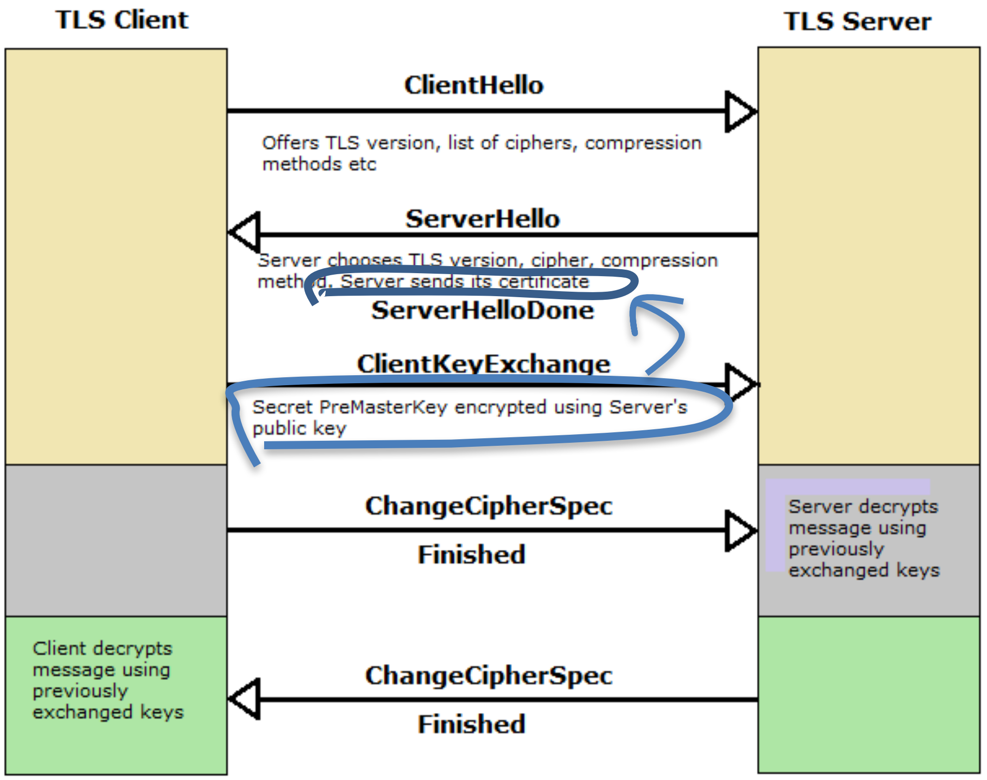
Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

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

TLS Connections



TLS Client

TLS Server

ClientHello

Offers TLS version, list of ciphers, compression methods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

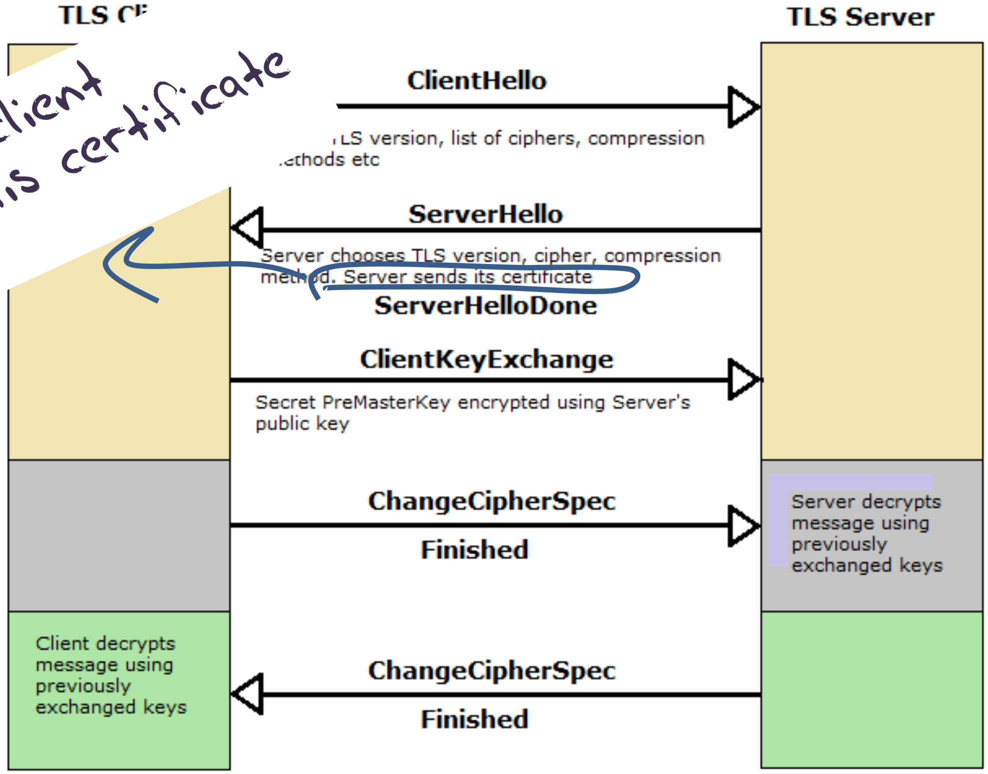
ChangeCipherSpec

Finished

Client decrypts message using previously exchanged keys

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

TLS Connections



TLS Cl

TLS Server

ClientHello

LS version, list of ciphers, compression ethods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

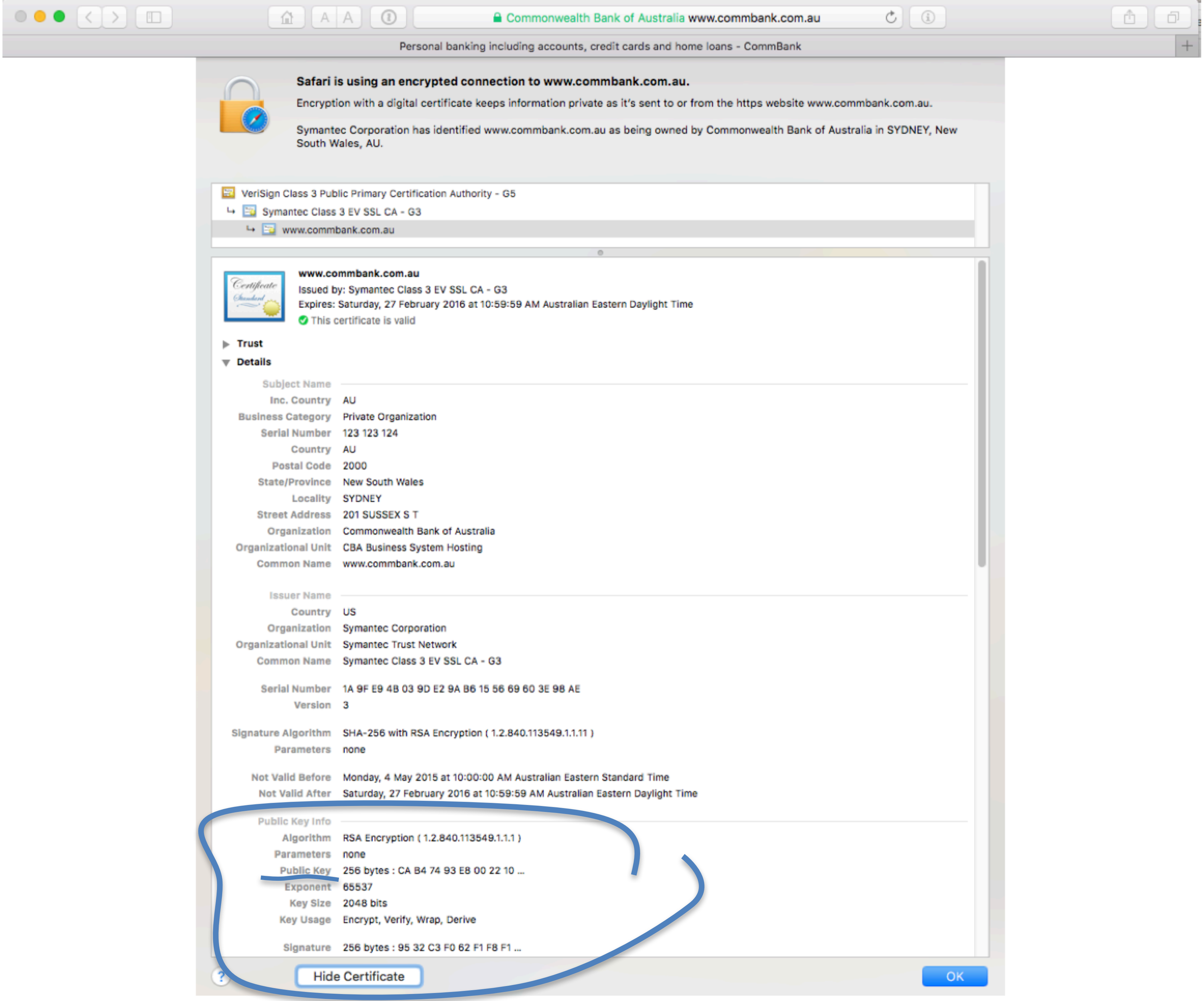
Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

How does the client "recognise" this certificate as valid?

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



Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank

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.

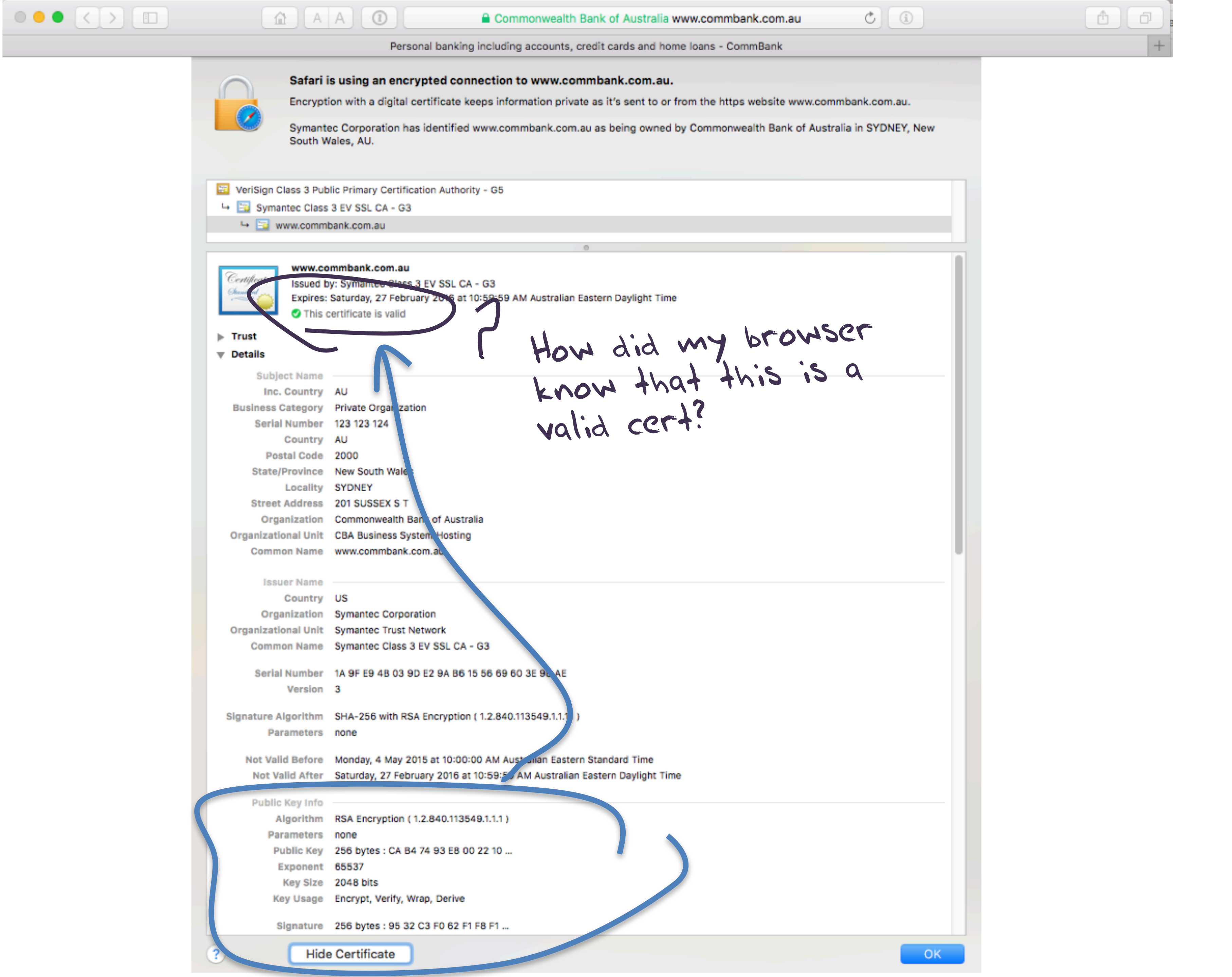
VeriSign Class 3 Public Primary Certification Authority - G5
↳ Symantec Class 3 EV SSL CA - G3
↳ www.commbank.com.au

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 ...



Commonwealth Bank of Australia www.commbank.com.au

Personal banking including accounts, credit cards and home loans - CommBank

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 9[illegible] AE
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.[illegible])
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?

Domain Name Certification

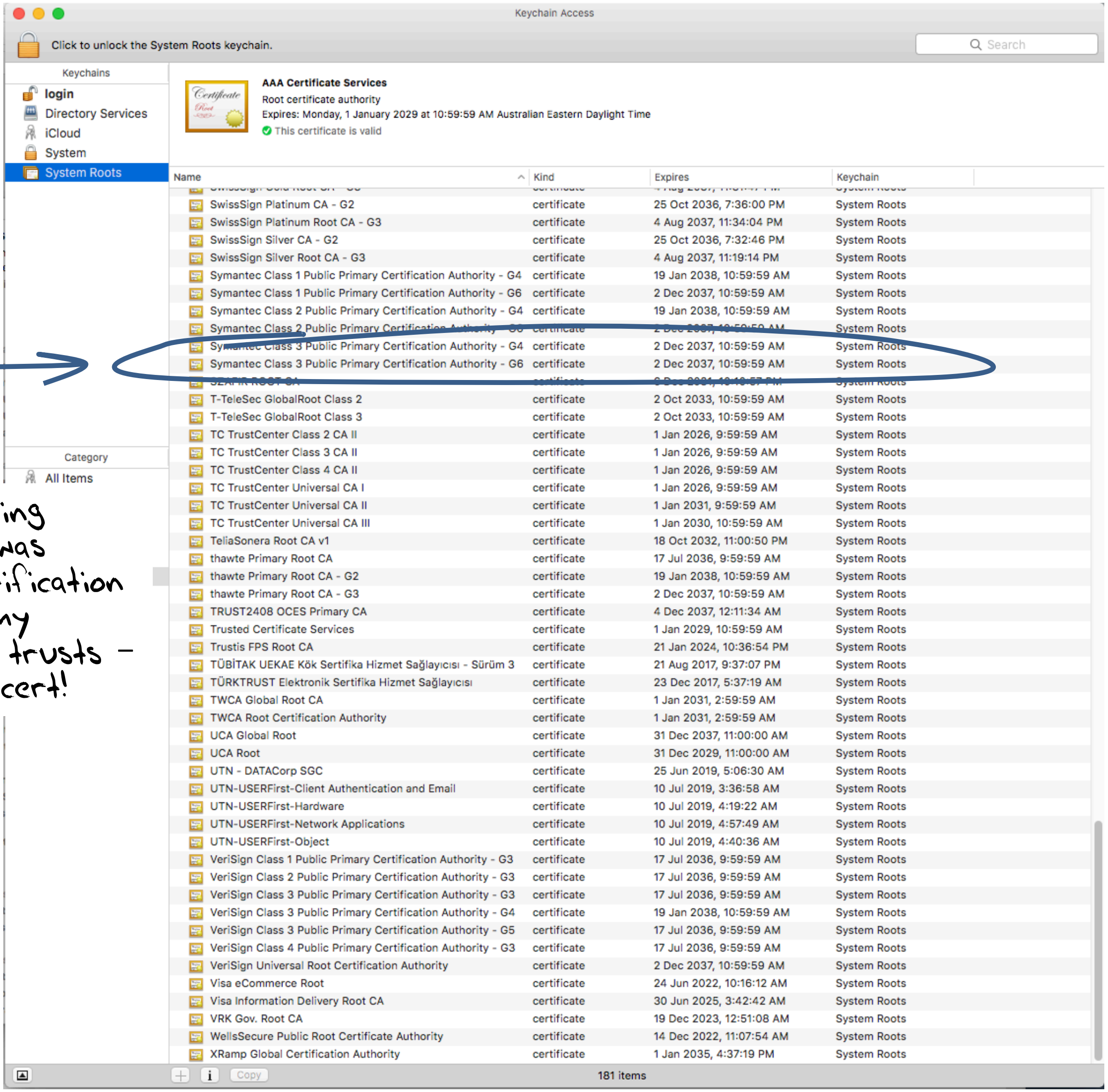
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a Certificate Signing Request to a company called "Symantec" (together with money)
- Symantec is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value (because it has been paid to do this!)
- So if I can associate this public key with a connection then I have a high degree of confidence that I've connected to the "real" www.commbank.com.au, as long as I am also prepared to trust Symantec, and their certificate issuance processes, and that the certificates that they issue are always genuine

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" (together with money)
- Symantec is 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 (because it has been paid to do this!)
- So if I can associate this public key with a connection then I have a high degree of confidence that I've connected to the "real" www.commbank.com.au, as long as I am also prepared to trust Symantec, and their certificate issuance processes, and that the certificates that they issue are always genuine

Why should i trust them?

Local Trust



Click to unlock the System Roots keychain.

AAA Certificate Services
Root certificate authority
Expires: Monday, 1 January 2029 at 10:59:59 AM Australian Eastern Daylight Time
✓ This certificate is valid

Name	Kind	Expires	Keychain
SwissSign Gold Root CA - G3	certificate	4 Aug 2037, 11:31:47 PM	System Roots
SwissSign Platinum CA - G2	certificate	25 Oct 2036, 7:36:00 PM	System Roots
SwissSign Platinum Root CA - G3	certificate	4 Aug 2037, 11:34:04 PM	System Roots
SwissSign Silver CA - G2	certificate	25 Oct 2036, 7:32:46 PM	System Roots
SwissSign Silver Root CA - G3	certificate	4 Aug 2037, 11:19:14 PM	System Roots
Symantec Class 1 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
Symantec Class 1 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 2 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
Symantec Class 2 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 3 Public Primary Certification Authority - G4	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 3 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
SZAFIR ROOT CA	certificate	6 Dec 2031, 10:10:57 PM	System Roots
T-TeleSec GlobalRoot Class 2	certificate	2 Oct 2033, 10:59:59 AM	System Roots
T-TeleSec GlobalRoot Class 3	certificate	2 Oct 2033, 10:59:59 AM	System Roots
TC TrustCenter Class 2 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Class 3 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Class 4 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Universal CA I	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Universal CA II	certificate	1 Jan 2031, 9:59:59 AM	System Roots
TC TrustCenter Universal CA III	certificate	1 Jan 2030, 10:59:59 AM	System Roots
TeliaSonera Root CA v1	certificate	18 Oct 2032, 11:00:50 PM	System Roots
thawte Primary Root CA	certificate	17 Jul 2036, 9:59:59 AM	System Roots
thawte Primary Root CA - G2	certificate	19 Jan 2038, 10:59:59 AM	System Roots
thawte Primary Root CA - G3	certificate	2 Dec 2037, 10:59:59 AM	System Roots
TRUST2408 OCES Primary CA	certificate	4 Dec 2037, 12:11:34 AM	System Roots
Trusted Certificate Services	certificate	1 Jan 2029, 10:59:59 AM	System Roots
Trustis FPS Root CA	certificate	21 Jan 2024, 10:36:54 PM	System Roots
TÜBİTAK UEKAE Kök Sertifika Hizmet Sağlayıcısı - Sürüm 3	certificate	21 Aug 2017, 9:37:07 PM	System Roots
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı	certificate	23 Dec 2017, 5:37:19 AM	System Roots
TWCA Global Root CA	certificate	1 Jan 2031, 2:59:59 AM	System Roots
TWCA Root Certification Authority	certificate	1 Jan 2031, 2:59:59 AM	System Roots
UCA Global Root	certificate	31 Dec 2037, 11:00:00 AM	System Roots
UCA Root	certificate	31 Dec 2029, 11:00:00 AM	System Roots
UTN - DATACorp SGC	certificate	25 Jun 2019, 5:06:30 AM	System Roots
UTN-USERFirst-Client Authentication and Email	certificate	10 Jul 2019, 3:36:58 AM	System Roots
UTN-USERFirst-Hardware	certificate	10 Jul 2019, 4:19:22 AM	System Roots
UTN-USERFirst-Network Applications	certificate	10 Jul 2019, 4:57:49 AM	System Roots
UTN-USERFirst-Object	certificate	10 Jul 2019, 4:40:36 AM	System Roots
VeriSign Class 1 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 2 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G5	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 4 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Universal Root Certification Authority	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Visa eCommerce Root	certificate	24 Jun 2022, 10:16:12 AM	System Roots
Visa Information Delivery Root CA	certificate	30 Jun 2025, 3:42:42 AM	System Roots
VRK Gov. Root CA	certificate	19 Dec 2023, 12:51:08 AM	System Roots
WellsSecure Public Root Certificate Authority	certificate	14 Dec 2022, 11:07:54 AM	System Roots
XRamp Global Certification Authority	certificate	1 Jan 2035, 4:37:19 PM	System Roots

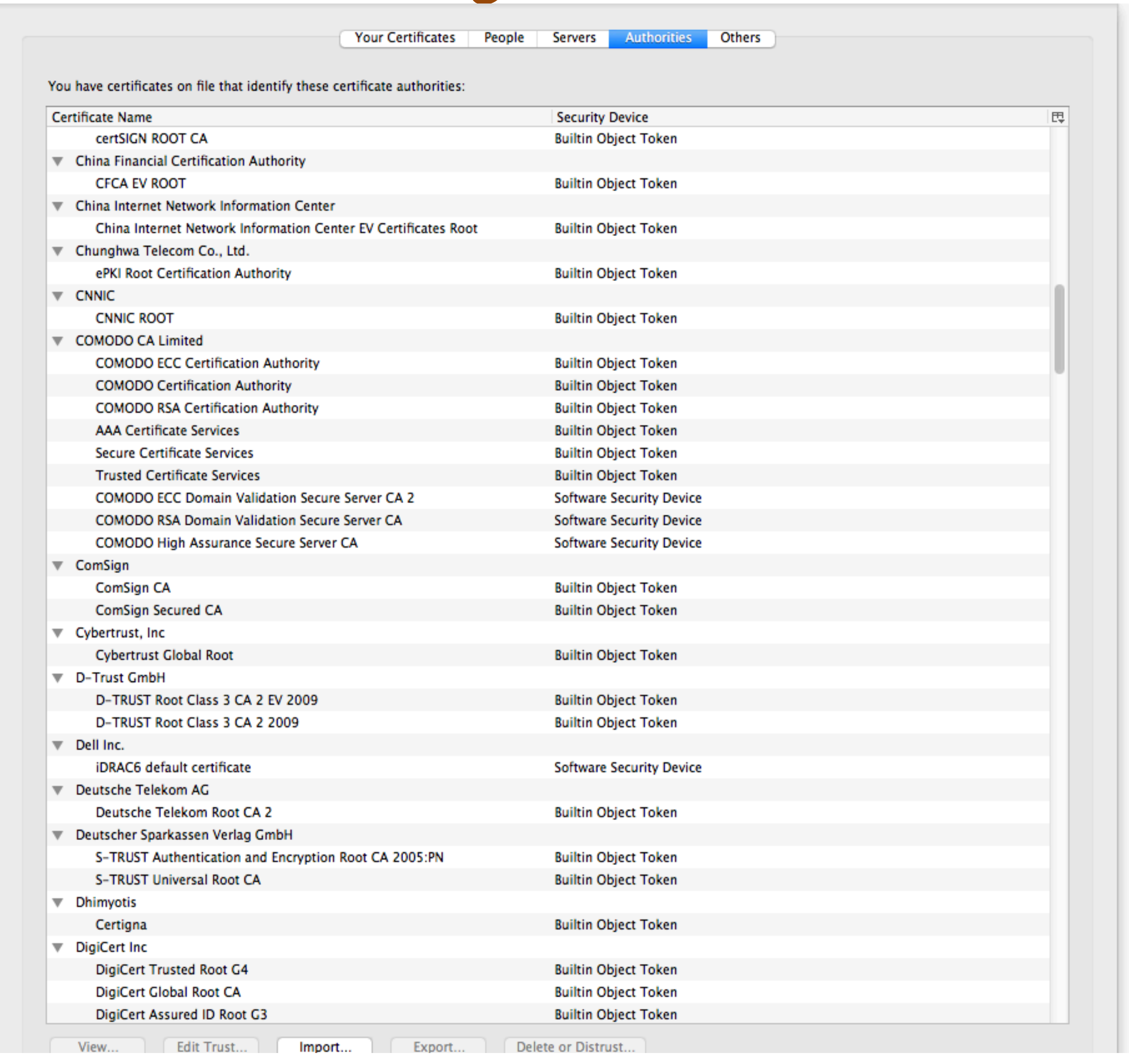
181 items

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?



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
certSIGN ROOT CA	Builtin Object Token
▼ China Financial Certification Authority	
CFCA EV ROOT	Builtin Object Token
▼ China Internet Network Information Center	
China Internet Network Information Center EV Certificates Root	Builtin Object Token
▼ Chunghwa Telecom Co., Ltd.	
ePKI Root Certification Authority	Builtin Object Token
▼ CNNIC	
CNNIC ROOT	Builtin Object Token
▼ COMODO CA Limited	
COMODO ECC Certification Authority	Builtin Object Token
COMODO Certification Authority	Builtin Object Token
COMODO RSA Certification Authority	Builtin Object Token
AAA Certificate Services	Builtin Object Token
Secure Certificate Services	Builtin Object Token
Trusted Certificate Services	Builtin Object Token
COMODO ECC Domain Validation Secure Server CA 2	Software Security Device
COMODO RSA Domain Validation Secure Server CA	Software Security Device
COMODO High Assurance Secure Server CA	Software Security Device
▼ ComSign	
ComSign CA	Builtin Object Token
ComSign Secured CA	Builtin Object Token
▼ Cybertrust, Inc	
Cybertrust Global Root	Builtin Object Token
▼ D-Trust GmbH	
D-TRUST Root Class 3 CA 2 EV 2009	Builtin Object Token
D-TRUST Root Class 3 CA 2 2009	Builtin Object Token
▼ Dell Inc.	
iDRAC6 default certificate	Software Security Device
▼ Deutsche Telekom AG	
Deutsche Telekom Root CA 2	Builtin Object Token
▼ Deutscher Sparkassen Verlag GmbH	
S-TRUST Authentication and Encryption Root CA 2005:PN	Builtin Object Token
S-TRUST Universal Root CA	Builtin Object Token
▼ Dhimyotis	
Certigna	Builtin Object Token
▼ DigiCert Inc	
DigiCert Trusted Root G4	Builtin Object Token
DigiCert Global Root CA	Builtin Object Token
DigiCert Assured ID Root G3	Builtin Object Token

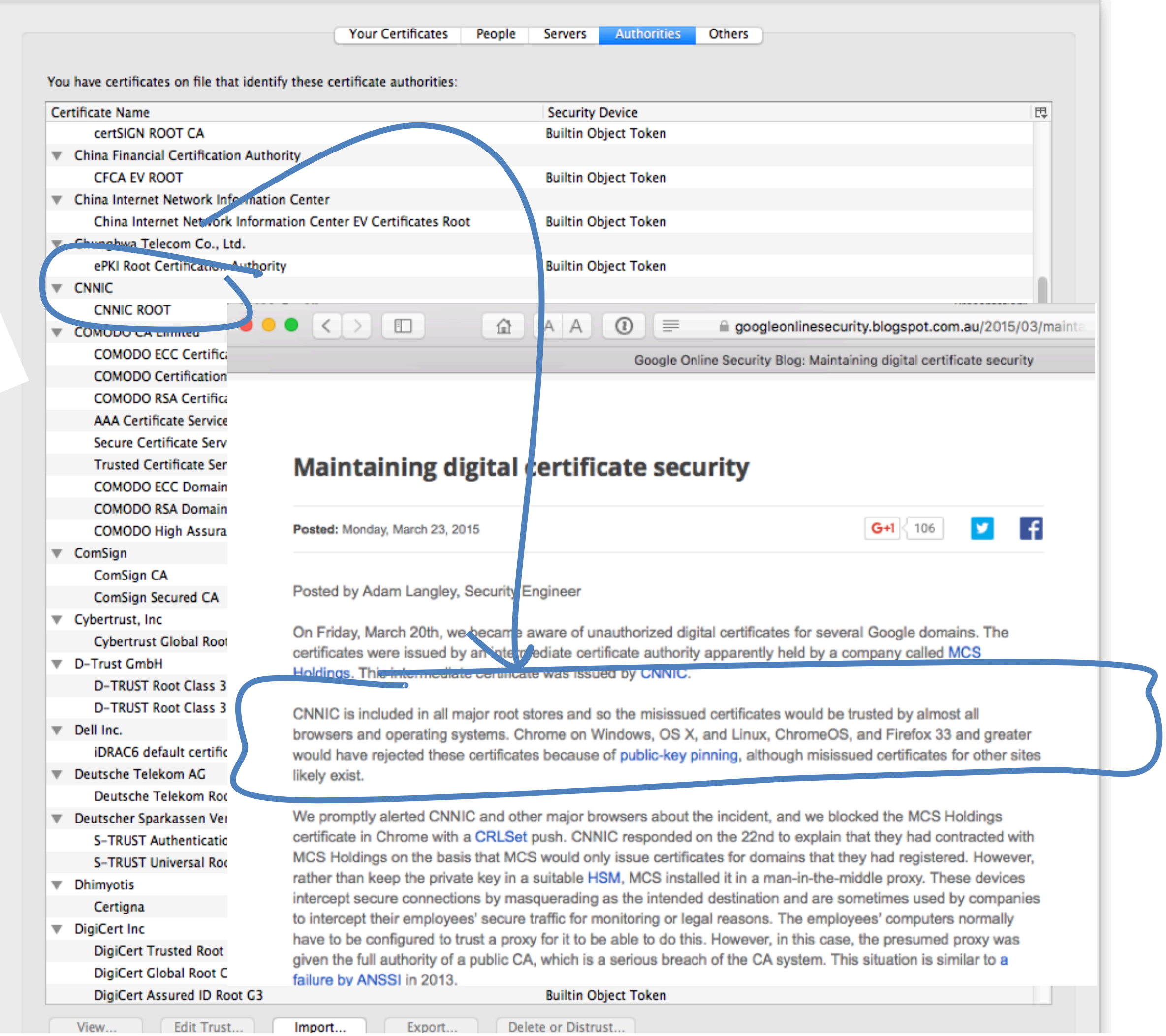
* cre·du·li·ty
/krəˈd(y)o͞olədē/
noun
a tendency to be too ready to believe that something is real or true.

Local Credulity

That’s a big list of people to Trust

Are they all trustable?

Evidently Not!



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
certSIGN ROOT CA	Builtin Object Token
▼ China Financial Certification Authority	
CFCA EV ROOT	Builtin Object Token
▼ China Internet Network Information Center	
China Internet Network Information Center EV Certificates Root	Builtin Object Token
▼ Chunghwa Telecom Co., Ltd.	
ePKI Root Certification Authority	Builtin Object Token
▼ CNNIC	
CNNIC ROOT	
▼ COMODO CA Limited	
COMODO ECC Certific	
COMODO Certification	
COMODO RSA Certific	
AAA Certificate Service	
Secure Certificate Serv	
Trusted Certificate Ser	
COMODO ECC Domain	
COMODO RSA Domain	
COMODO High Assura	
▼ ComSign	
ComSign CA	
ComSign Secured CA	
▼ Cybertrust, Inc	
Cybertrust Global Root	
▼ D-Trust GmbH	
D-TRUST Root Class 3	
D-TRUST Root Class 3	
▼ Dell Inc.	
iDRAC6 default certific	
▼ Deutsche Telekom AG	
Deutsche Telekom Roo	
▼ Deutscher Sparkassen Ver	
S-TRUST Authenticatio	
S-TRUST Universal Roo	
▼ Dhimyotis	
Certigna	
▼ DigiCert Inc	
DigiCert Trusted Root	
DigiCert Global Root C	
DigiCert Assured ID Root G3	Builtin Object Token

googleonlinesecurity.blogspot.com.au/2015/03/maint

Google Online Security Blog: Maintaining digital certificate security

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.

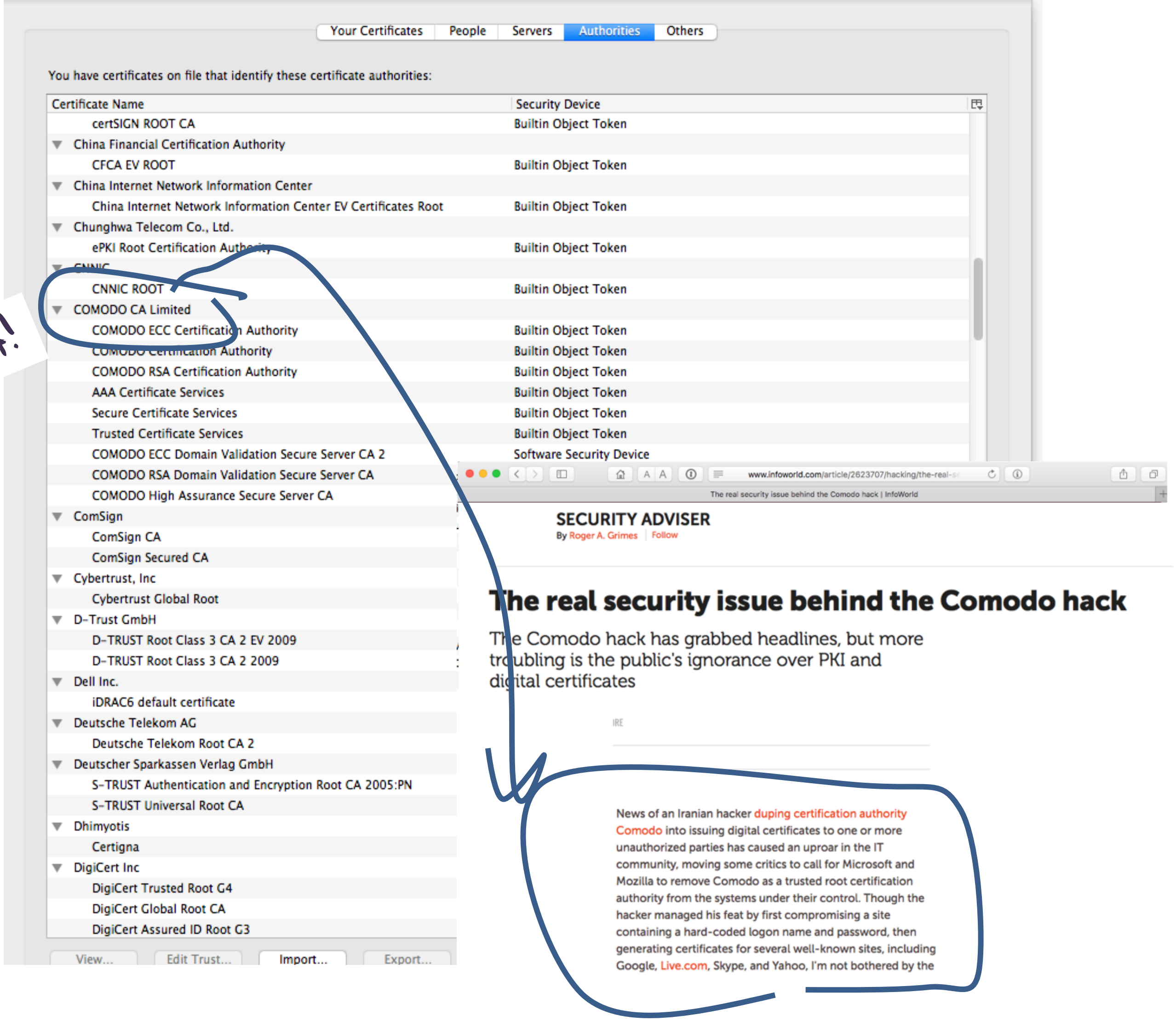
We promptly alerted CNNIC and other major browsers about the incident, and we blocked the MCS Holdings certificate in Chrome with a CRLSet push. CNNIC responded on the 22nd to explain that they had contracted with MCS Holdings on the basis that MCS would only issue certificates for domains that they 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 Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
certSIGN ROOT CA	Builtin Object Token
▼ China Financial Certification Authority	
CFCA EV ROOT	Builtin Object Token
▼ China Internet Network Information Center	
China Internet Network Information Center EV Certificates Root	Builtin Object Token
▼ Chunghwa Telecom Co., Ltd.	
ePKI Root Certification Authority	Builtin Object Token
▼ CNNIC	
CNNIC ROOT	Builtin Object Token
▼ COMODO CA Limited	
COMODO ECC Certification Authority	Builtin Object Token
COMODO Certification Authority	Builtin Object Token
COMODO RSA Certification Authority	Builtin Object Token
AAA Certificate Services	Builtin Object Token
Secure Certificate Services	Builtin Object Token
Trusted Certificate Services	Builtin Object Token
COMODO ECC Domain Validation Secure Server CA 2	Software Security Device
COMODO RSA Domain Validation Secure Server CA	
COMODO High Assurance Secure Server CA	
▼ ComSign	
ComSign CA	
ComSign Secured CA	
▼ Cybertrust, Inc	
Cybertrust Global Root	
▼ D-Trust GmbH	
D-TRUST Root Class 3 CA 2 EV 2009	
D-TRUST Root Class 3 CA 2 2009	
▼ Dell Inc.	
iDRAC6 default certificate	
▼ Deutsche Telekom AG	
Deutsche Telekom Root CA 2	
▼ Deutscher Sparkassen Verlag GmbH	
S-TRUST Authentication and Encryption Root CA 2005:PN	
S-TRUST Universal Root CA	
▼ Dhimyotis	
Certigna	
▼ DigiCert Inc	
DigiCert Trusted Root G4	
DigiCert Global Root CA	
DigiCert Assured ID Root G3	

www.infoworld.com/article/2623707/hacking/the-real-s

The real security issue behind the Comodo hack | InfoWorld

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

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 his feat 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

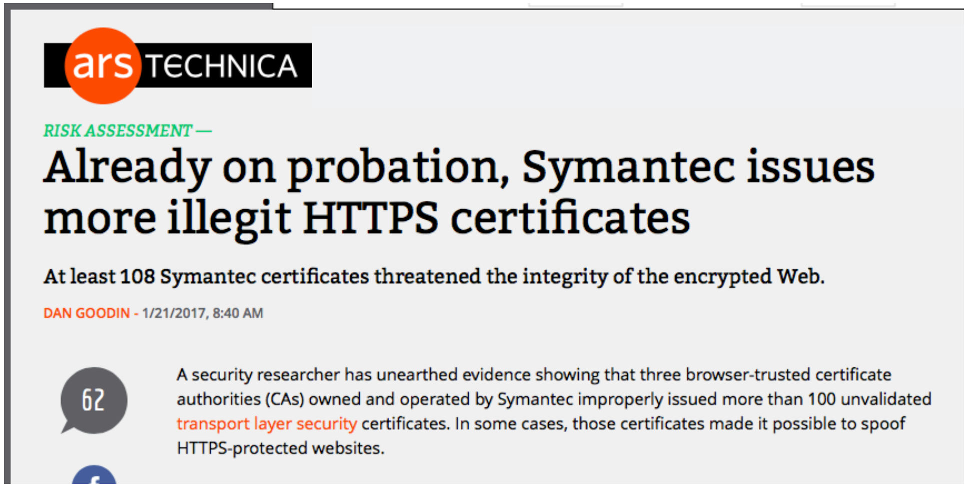
But my bank uses Symantec

as their Certificate Authority

And Symantec NEVER lie in the certificates they issue

Never?

Well, hardly ever



ars TECHNICA

RISK ASSESSMENT —

Already on probation, Symantec issues more illegit HTTPS certificates

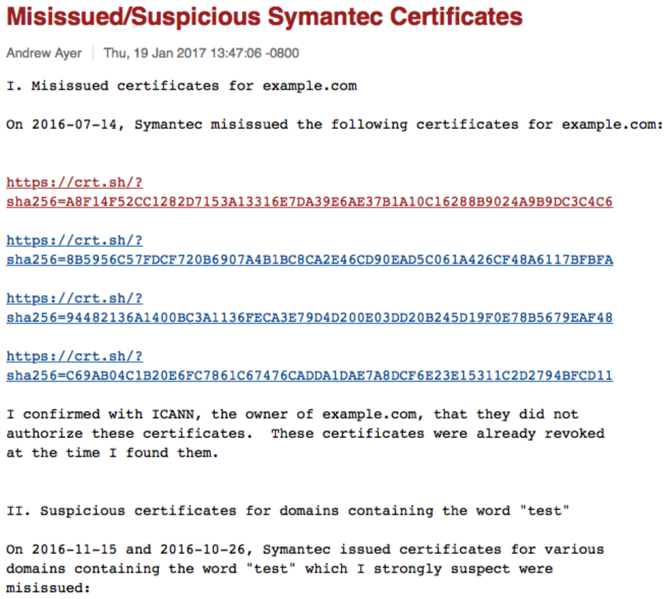
At least 108 Symantec certificates threatened the integrity of the encrypted Web.

DAN GOODIN - 1/21/2017, 8:40 AM

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/

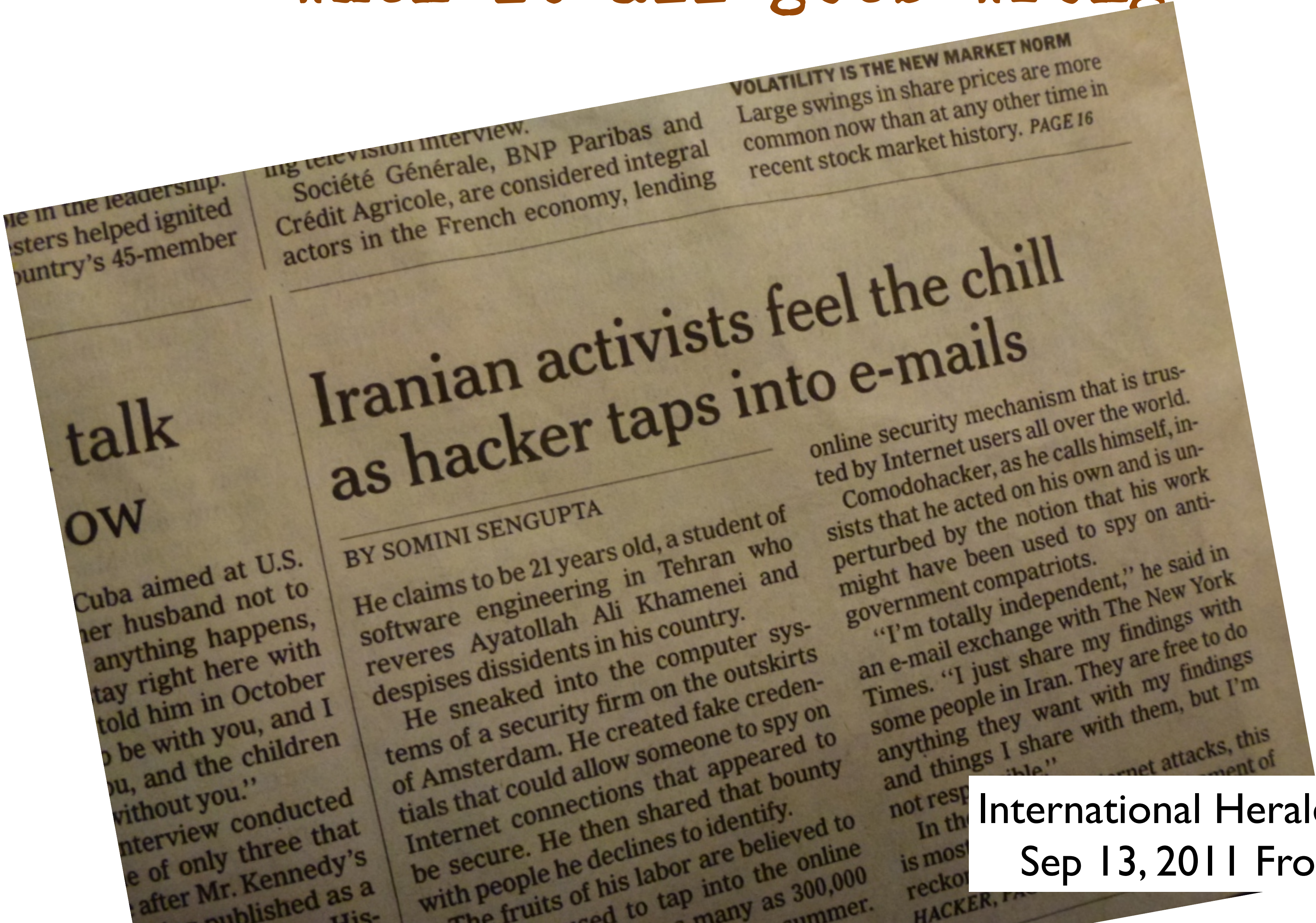


Misissued/Suspicious Symantec Certificates

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:
```

With unpleasant consequences when it all goes wrong



ne in the leadership. sters helped ignited ountry's 45-member

ing television interview. Société Générale, BNP Paribas and Crédit Agricole, are considered integral actors in the French economy, lending

VOLATILITY IS THE NEW MARKET NORM
Large swings in share prices are more common now than at any other time in recent stock market history. *PAGE 16*

talk ow

Cuba aimed at U.S. her husband not to anything happens, tay right here with told him in October be with you, and I u, and the children without you."

nterview conducted e of only three that after Mr. Kennedy's published as a His-

Iranian activists feel the chill as hacker taps into e-mails

BY SOMINI SENGUPTA

He claims to be 21 years old, a student of software engineering in Tehran who reveres Ayatollah Ali Khamenei and despises dissidents in his country.

He sneaked into the computer systems of a security firm on the outskirts of Amsterdam. He created fake credentials that could allow someone to spy on Internet connections that appeared to be secure. He then shared that bounty with people he declines to identify.

The fruits of his labor are believed to ed to tap into the online many as 300,000 summer.

online security mechanism that is trusted by Internet users all over the world.

Comodohacker, as he calls himself, insists that he acted on his own and is unperturbed by the notion that his work might have been used to spy on antigovernment compatriots.

"I'm totally independent," he said in an e-mail exchange with The New York Times. "I just share my findings with some people in Iran. They are free to do anything they want with my findings and things I share with them, but I'm not resp ible."

In th rnet attacks, this is most ment of reckon

HACKER, PA

International Herald Tribune
Sep 13, 2011 Front Page

What's going wrong here?

- The TLS handshake cannot specify WHICH CA should be used to validate the digital certificate
- That means that your browser may allow ANY CA to be used to validate a certificate

What's going wrong here?

- The TLS handshake cannot specify WHICH CA should be used t digital ce

WOW! That's awesomely bad!

- That means that your browser may allow ANY CA to be used to validate a certificate

What's going wrong here?

WOW! That's awesomely bad!

Here's a lock - it might be the lock on your front door for all i know.

The lock might LOOK secure, but don't worry - literally ANY key will open it!

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

See the EFF SSL observatory
http://www.eff.org/files/DefconSSLiverse.pdf

In a commercial environment

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

?

In a commercial environment

Where CA’s compete with each other for market share

And quality offers no protection

Than what ‘wins’ in the market?

Sustainable

Resilient

Secure

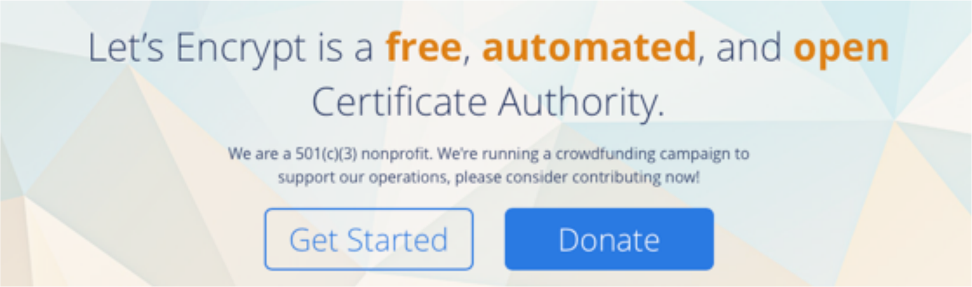
Privacy

Trusted

Cheap!

Where now?

Option A: Take all the money out of the system!



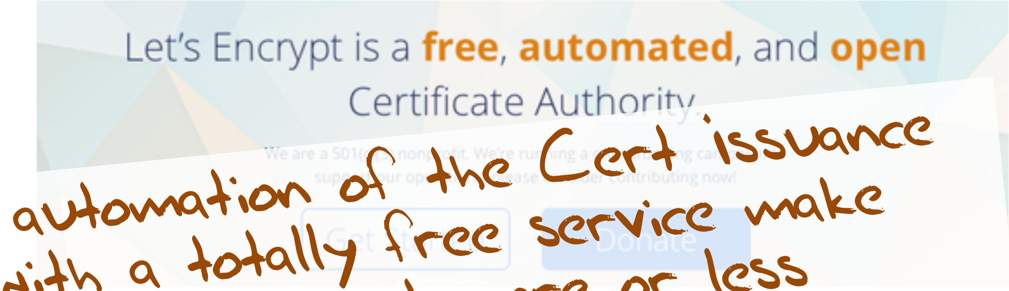
Let's Encrypt is a **free**, **automated**, and **open** Certificate Authority.

We are a 501(c)(3) nonprofit. We're running a crowdfunding campaign to support our operations, please consider contributing now!

www.letsencrypt.org

Where now?

Option A: Take all the money out of the system!



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

We are a 501(c)(3) nonprofit. We're running a crowdfunding campaign to support our operations. Please consider contributing now!

www.letsencrypt.org

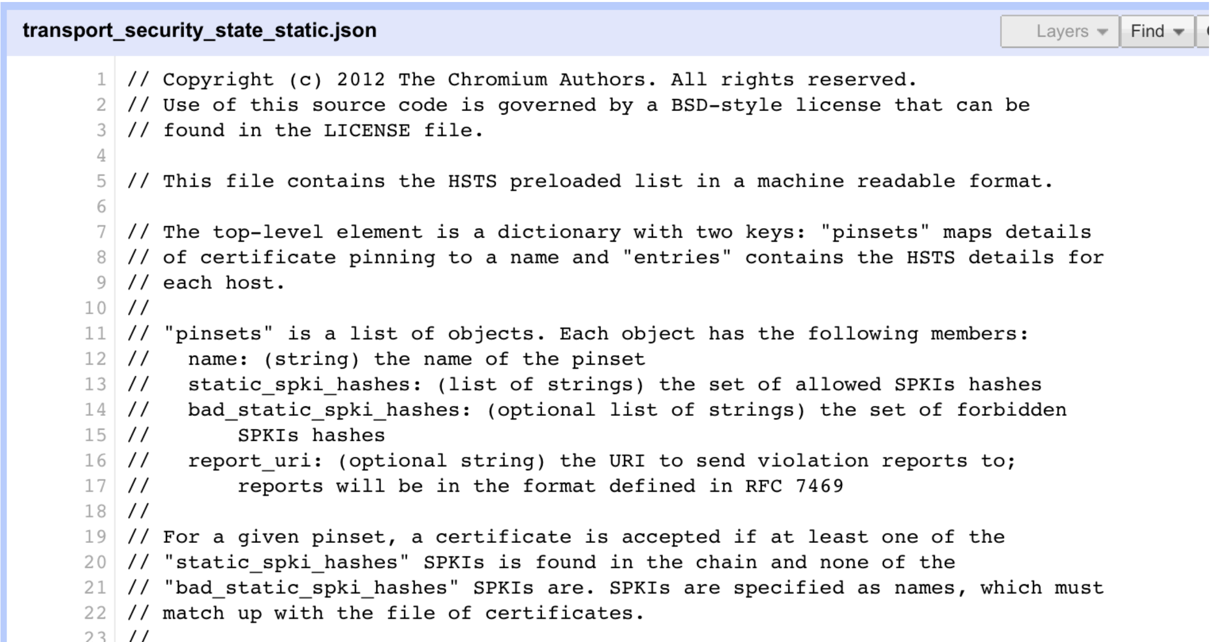
Will the automation of the Cert issuance coupled with a totally free service make the overall environment more or less secure?

We're probably going to find out real soon!

Where now?

Option B: White Listing and Pinning with HSTS

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



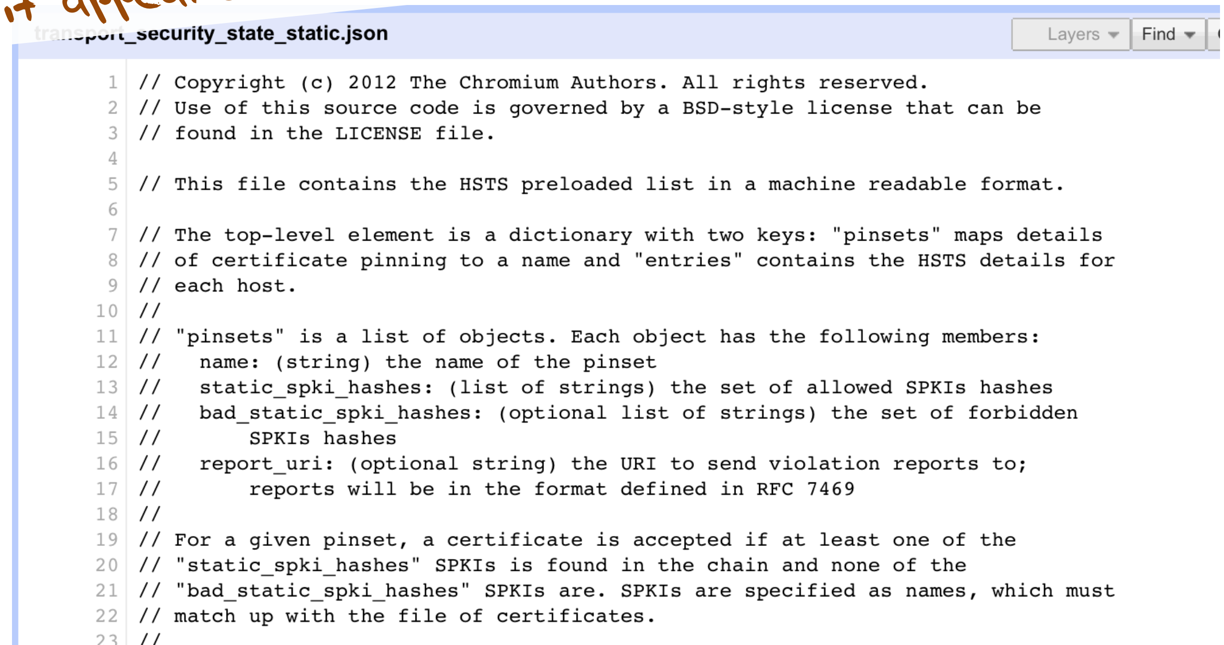
```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This file contains the HSTS preloaded list in a machine readable format.

// The top-level element is a dictionary with two keys: "pinsets" maps details
// of certificate pinning to a name and "entries" contains the HSTS details for
// each host.
//
// "pinsets" is a list of objects. Each object has the following members:
//   name: (string) the name of the pinset
//   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
//   bad_static_spki_hashes: (optional list of strings) the set of forbidden
//       SPKIs hashes
//   report_uri: (optional string) the URI to send violation reports to;
//       reports will be in the format defined in RFC 7469
//
// For a given pinset, a certificate is accepted if at least one of the
// "static_spki_hashes" SPKIs is found in the chain and none of the
// "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
// match up with the file of certificates.
//
```

Where now?

Option B: White Listing and Pinning with HSTS

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

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

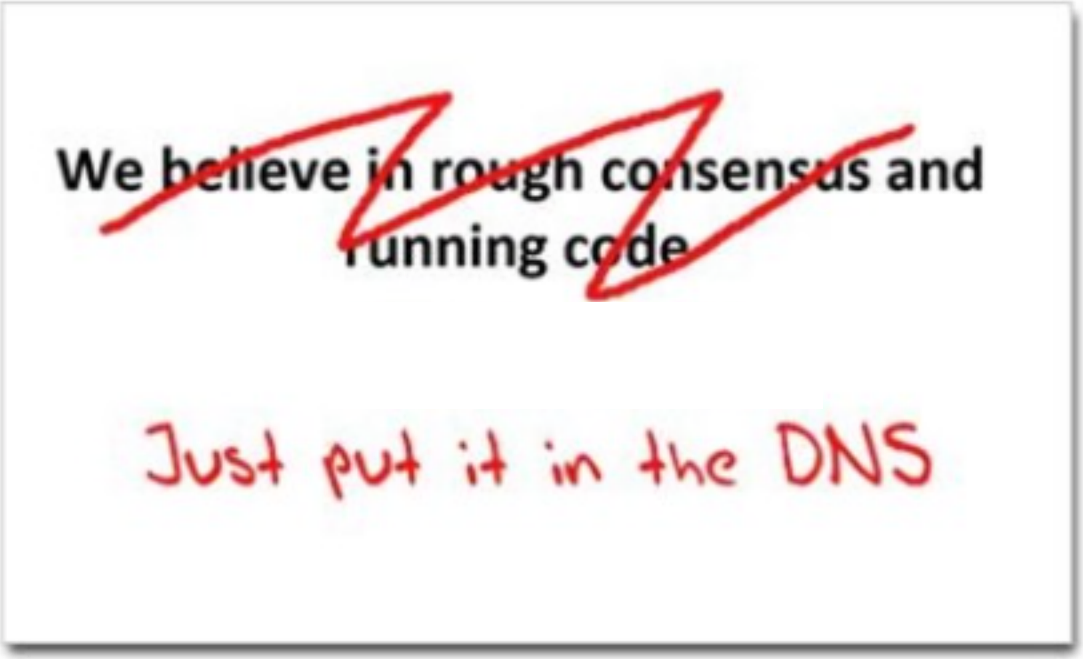


transport_security_state_static.json

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This file contains the HSTS preloaded list in a machine readable format.

// The top-level element is a dictionary with two keys: "pinsets" maps details
// of certificate pinning to a name and "entries" contains the HSTS details for
// each host.
//
// "pinsets" is a list of objects. Each object has the following members:
//   name: (string) the name of the pinset
//   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
//   bad_static_spki_hashes: (optional list of strings) the set of forbidden
//       SPKIs hashes
//   report_uri: (optional string) the URI to send violation reports to;
//       reports will be in the format defined in RFC 7469
//
// For a given pinset, a certificate is accepted if at least one of the
// "static_spki_hashes" SPKIs is found in the chain and none of the
// "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
// match up with the file of certificates.
//
```

Where now?

Option C: Use the DNS!



~~We believe in rough consensus and running code~~

Just put it in the DNS

cafepress.com/nxdomain

Seriously … just use the DNS Luke!*

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

* Ok, that was a cheap shot – my apologies to the Star Wars franchise!

Seriously

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

- Why not query the DNS for the HSTS record (pinning record)?

Seriously

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

- Why not query the DNS for the HSTS record?
- Why not query the DNS for the issuer CA?

Seriously

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

- Why not query the DNS for the HSTS record?
- Why not query the DNS for the issuer CA?
- Why not query the DNS for the hash of the domain name cert?

Seriously

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

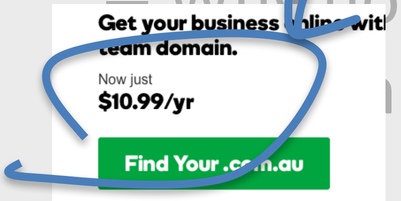
- Why not query the DNS for the HSTS record?
- Why not query the DNS for the issuer CA?
- Why not query the DNS for the hash of the domain name cert?
- Why not query the DNS for the hash of the domain name public key?

Seriously

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

- Why not quer r the HSTS record?
- Wh y the DNS for the issuer CA?
- hy not query the DNS for the hash of the domain name cert?
- Why not query the DNS for n name subject public key info?

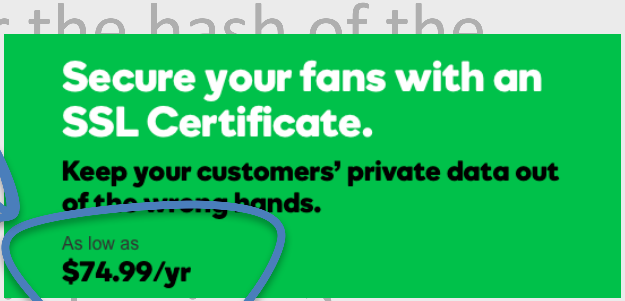
Who needs CA's anyway?



Get your business online with eam domain.

Now just
$10.99/yr

Find Your .com.au



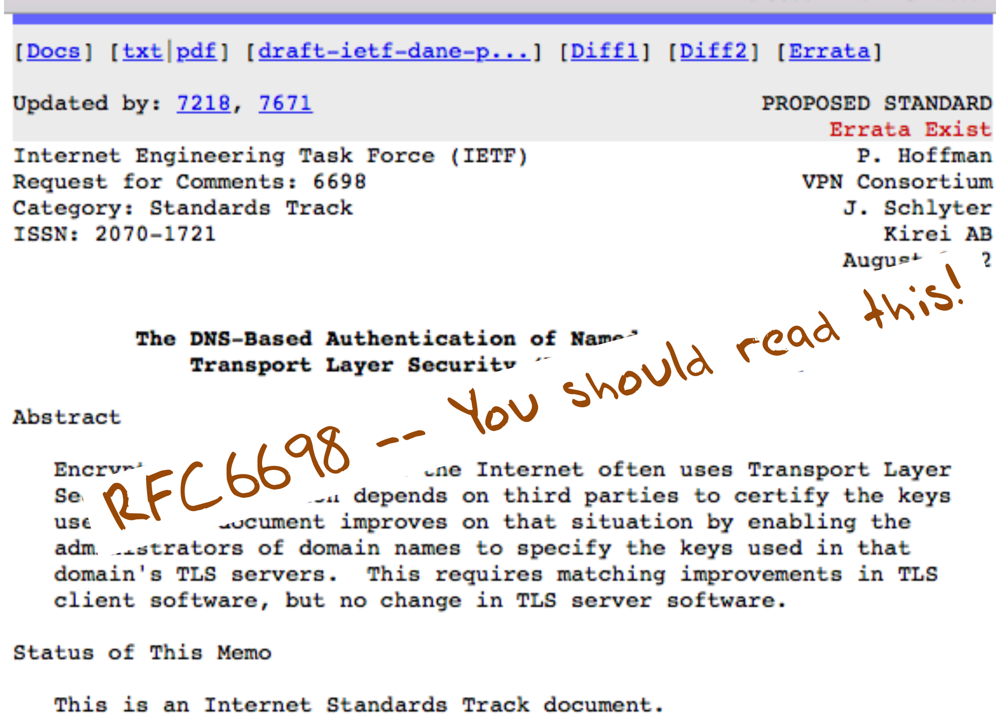
Secure your fans with an SSL Certificate.

Keep your customers' private data out of the wrong hands.

As low as
$74.99/yr

DANE

- Using the DNS to associated domain name public key certificates with domain name



[Docs] [txt|pdf] [draft-ietf-dane-p...] [Diff1] [Diff2] [Errata]

Updated by: 7218, 7671 PROPOSED STANDARD
Errata Exist

Internet Engineering Task Force (IETF) P. Hoffman
Request for Comments: 6698 VPN Consortium
Category: Standards Track J. Schlyter
ISSN: 2070-1721 Kirei AB
August 2

The DNS-Based Authentication of Name
Transport Layer Security

Abstract

Encry ne Internet often uses Transport Layer Se depends on third parties to certify the keys use document improves on that situation by enabling the adm strators of domain names to specify the keys used in that domain's TLS servers. This requires matching improvements in TLS client software, but no change in TLS server software.

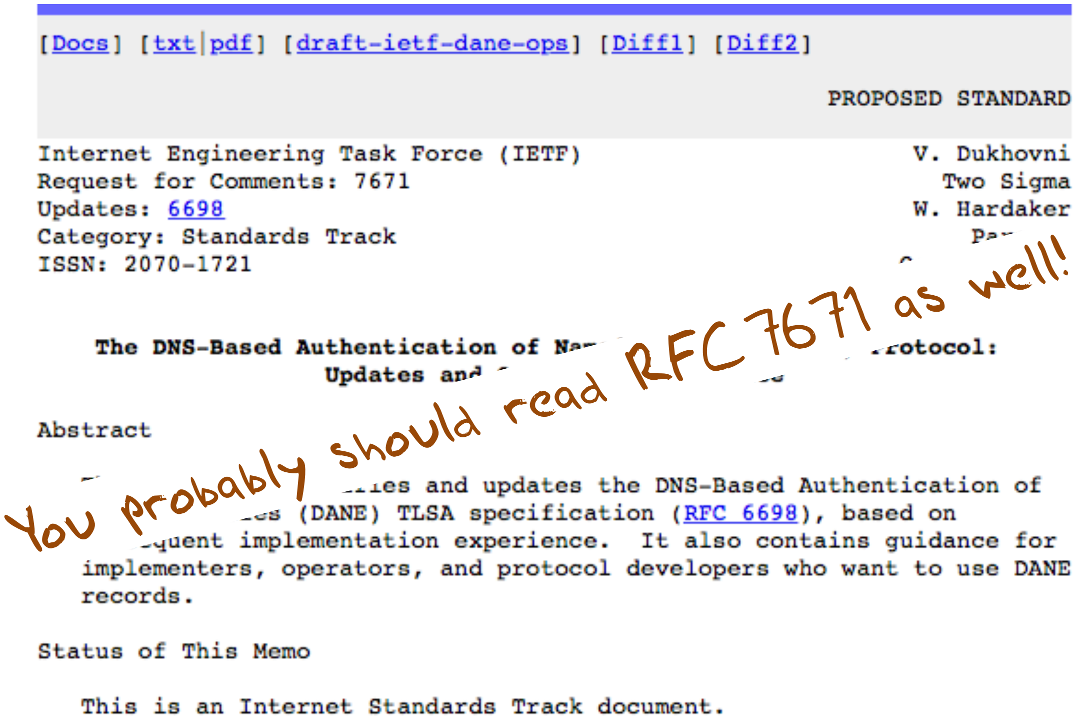
Status of This Memo

This is an Internet Standards Track document.

RFC6698 -- You should read this!

DANE

- Using the DNS to associated domain name public key certificates with domain name



[Docs] [txt|pdf] [draft-ietf-dane-ops] [Diff1] [Diff2]

PROPOSED STANDARD

Internet Engineering Task Force (IETF) V. Dukhovni
Request for Comments: 7671 Two Sigma
Updates: 6698 W. Hardaker
Category: Standards Track Pa
ISSN: 2070-1721

The DNS-Based Authentication of Na rotocol:
Updates and

Abstract

ies and updates the DNS-Based Authentication of es (DANE) TLSA specification (RFC 6698), based on quent implementation experience. It also contains guidance for implementers, operators, and protocol developers who want to use DANE records.

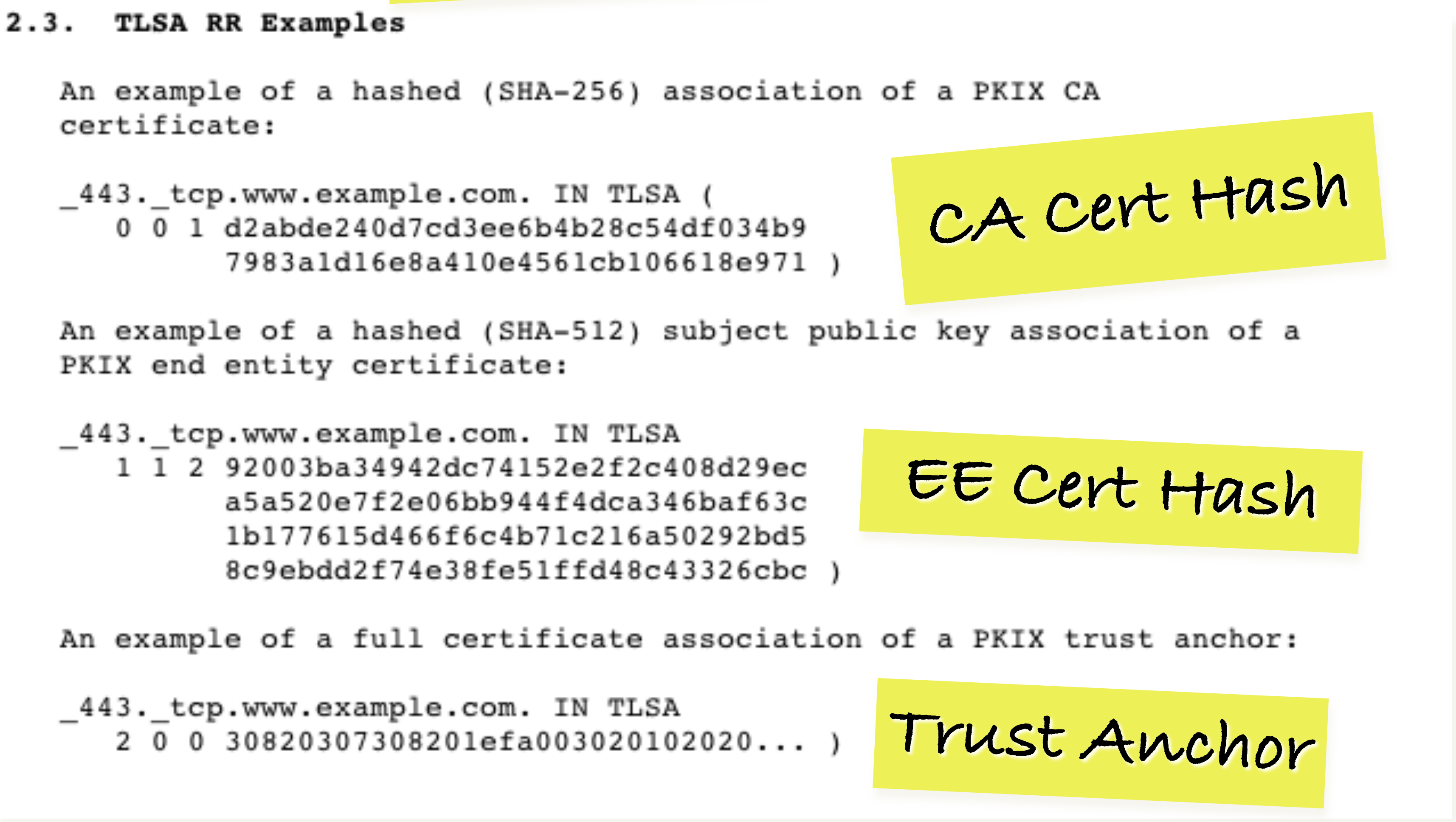
Status of This Memo

This is an Internet Standards Track document.

You probably should read RFC 7671 as well!

DANE

TLSA RR



2.3. TLSA RR Examples

An example of a hashed (SHA-256) association of a PKIX CA certificate:

```
_443._tcp.www.example.com. IN TLSA (
   0 0 1 d2abde240d7cd3ee6b4b28c54df034b9
         7983a1d16e8a410e4561cb106618e971 )
```

CA Cert Hash

An example of a hashed (SHA-512) subject public key association of a PKIX end entity certificate:

```
_443._tcp.www.example.com. IN TLSA
   1 1 2 92003ba34942dc74152e2f2c408d29ec
         a5a520e7f2e06bb944f4dca346baf63c
         1b177615d466f6c4b71c216a50292bd5
         8c9ebdd2f74e38fe51ffd48c43326cbc )
```

EE Cert Hash

An example of a full certificate association of a PKIX trust anchor:

```
_443._tcp.www.example.com. IN TLSA
   2 0 0 30820307308201efa003020102020... )
```

Trust Anchor

EECert TLSA record generation

```
; Convert the public key certificate to DER format
; Generate the SHA256 hash
; Add DNS gunk!

$ /usr/bin/openssl x509 -in /usr/local/etc/letsencrypt/live/www.dotnxdomain.net/cert.pem -outform DER |
/usr/bin/openssl sha256 |
cut -d ' ' -f 2 |
awk '{print ”_443._tcp.www.dotnxdomain.net  IN TLSA 3 0 1 " $1}'

_443._tcp.www.dotnxdomain.net. 899 IN     TLSA  3 0 1 D42101BCCE941D22E8E467C5D75E77EC4A7B8B7C9366C6A878CB4E15 7E602F17
```

```
$ dig +dnssec TLSA _443._tcp.www.dotnxdomain.net.

_443._tcp.www.dotnxdomain.net. 899 IN     TLSA  3 0 1 D42101BCCE941D22E8E467C5D75E77EC4A7B8B7C9366C6A878CB4E15 7E602F17
_443._tcp.www.dotnxdomain.net. 899 IN     RRSIG TLSA 13 5 900 20200724235900 20170122043100 56797 www.dotnxdomain.net.
dUYD1sMIpBc6RsUhturFzz5G8qX6oaDGRzaD/q6n+YJi2kqzDfWZls6F 3X1mXdpeQQYz52yOUOcdWvFRO9TQZQ==
```

SPKI TLSA record generation

```
; Generate the public key
; Convert it to DER format
; Generate the SHA256 hash
; Add DNS gunk!
```

```
$ /usr/bin/openssl x509 -in /usr/local/etc/letsencrypt/live/www.dotnxdomain.net/cert.pem -pubkey -noout |
openssl rsa -pubin -outform der |
/usr/bin/openssl sha256 |
cut -d ' ' -f 2 |
awk '{ print "_443._tcp.www.ndotnxdomain.net IN TLSA 3 1 1 " $1}’

_443._tcp.www.ndotnxdomain.net IN TLSA 3 1 1 df3a810d998cfddf8fa935ed33065ee27a67747366e2da40ddefef2b3a2032eb
```

TLS with DANE

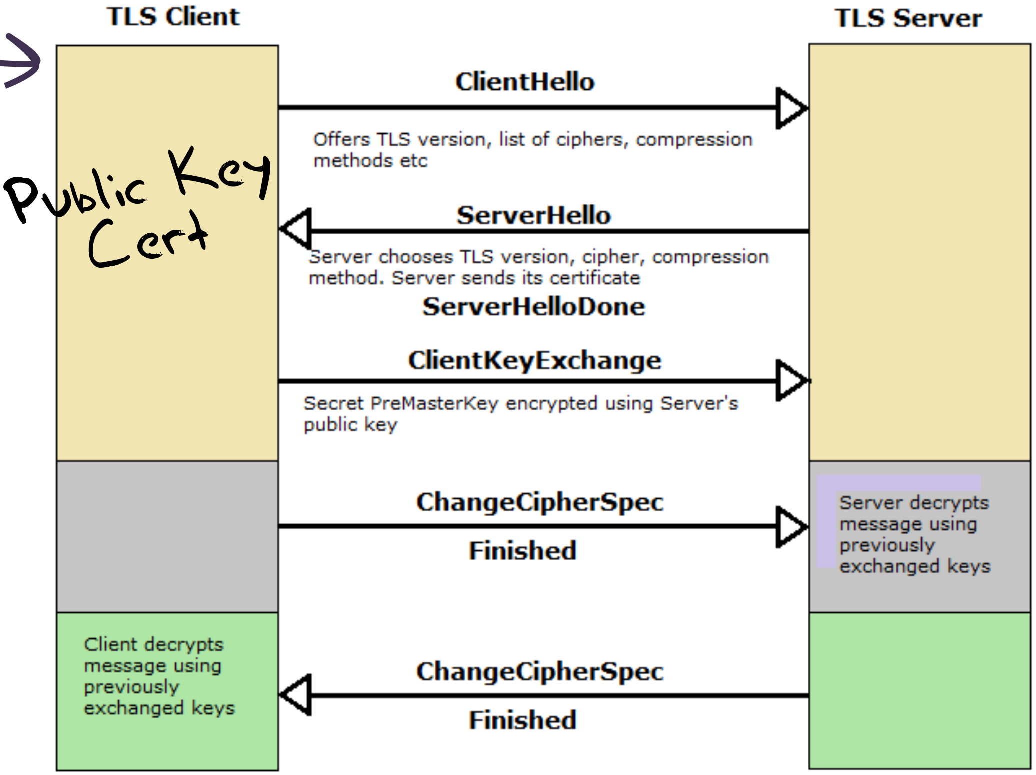
- Client receives server cert in Server Hello
  - *Client queries the DNS for the TLSA Resource Record of the domain name*
  - *Client validates the public key information in the presented certificate against the TLSA RR*
- Client performs Client Key exchange

TLS Connections

DNS Name

TLSA query

DNSSEC/TLSA Validator 2.2.0.2.1-signed
by CZ.NIC Labs
DNSSEC/TLSA Validator is a web browser add-on which allows you to check the existence and validity of DNSSEC records and TLSA records related to domain names.



TLS Client

TLS Server

Public Key Cert

ClientHello

Offers TLS version, list of ciphers, compression methods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

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

Just one problem…

- The DNS is full of liars and lies!
- And this can compromise the integrity of public key information embedded in the DNS
- Unless we fix the DNS we are no better off than before with these TLSA records!

Just one response…

- We need to allow users to validate DNS responses for themselves
- And for this we need a Secure DNS framework
- Which we have – and its called DNSSEC!

DNSSEC Interlocking Signatures

. (root)

. Key-Signing Key – signs over

. Zone-Signing Key – signs over

DS for .com (Key-Signing Key)

.com

.com Key-Signing Key – signs over

.com Zone-Signing Key – signs over

DS for example .com (Key-Signing Key)

.example.com

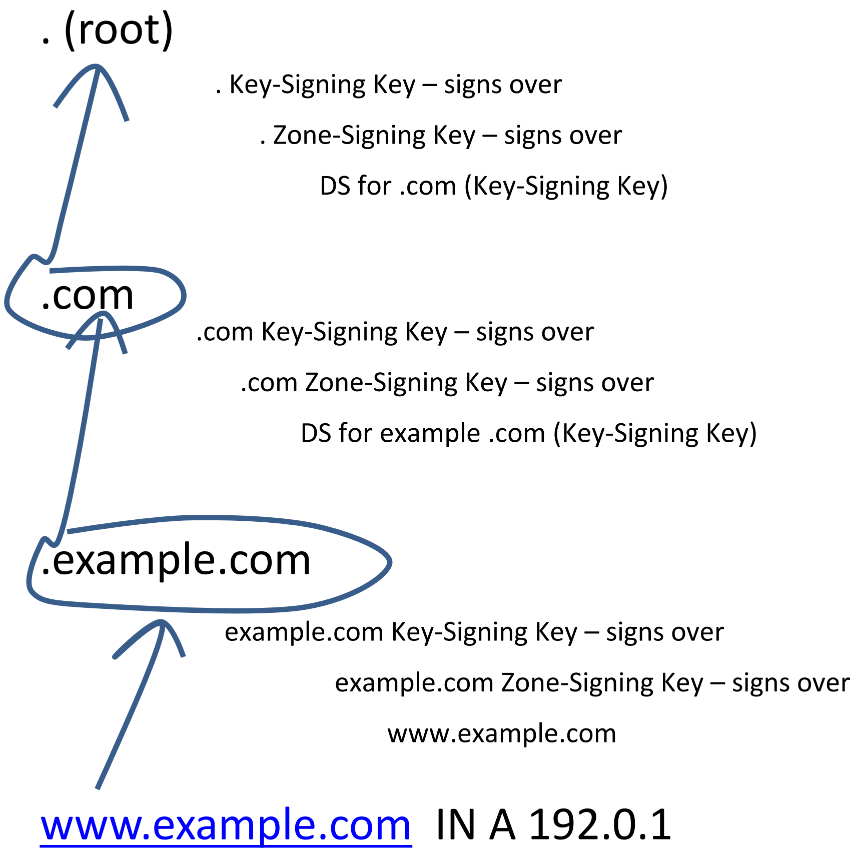
example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com

DNSSEC Interlocking Signatures



. (root)

. Key-Signing Key – signs over

. Zone-Signing Key – signs over

DS for .com (Key-Signing Key)

.com

.com Key-Signing Key – signs over

.com Zone-Signing Key – signs over

DS for example .com (Key-Signing Key)

.example.com

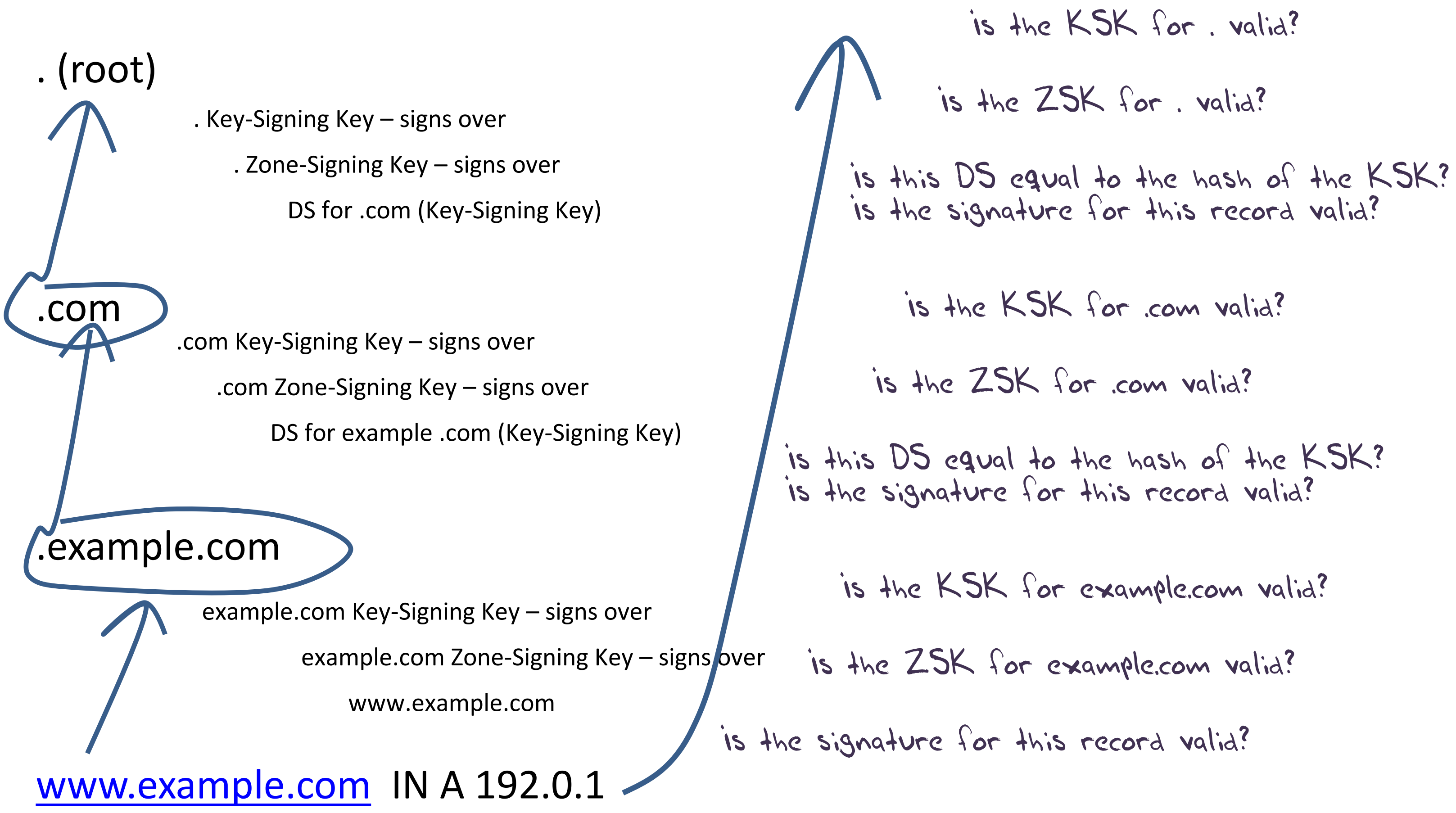
example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com IN A 192.0.1

DNSSEC Interlocking Signatures



. (root)

. Key-Signing Key – signs over

. Zone-Signing Key – signs over

DS for .com (Key-Signing Key)

.com

.com Key-Signing Key – signs over

.com Zone-Signing Key – signs over

DS for example .com (Key-Signing Key)

.example.com

example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com IN A 192.0.1

is the KSK for . valid?

is the ZSK for . valid?

is this DS equal to the hash of the KSK?
is the signature for this record valid?

is the KSK for .com valid?

is the ZSK for .com valid?

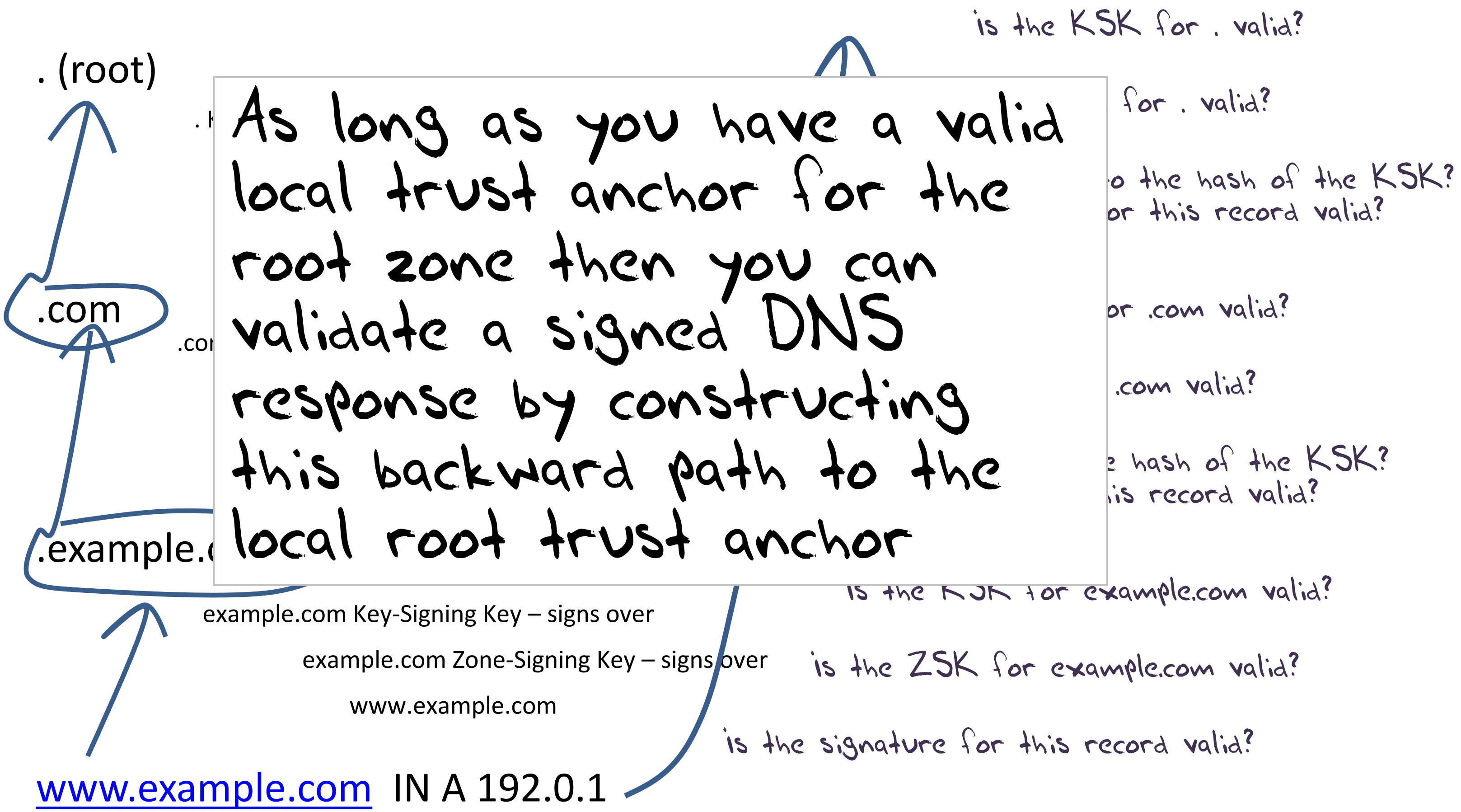
is this DS equal to the hash of the KSK?
is the signature for this record valid?

is the KSK for example.com valid?

is the ZSK for example.com valid?

is the signature for this record valid?

DNSSEC Interlocking Signatures


. (root)
. K
.com
.co
.example.
is the KSK for . valid?
for . valid?
o the hash of the KSK?
or this record valid?
or .com valid?
.com valid?
e hash of the KSK?
is record valid?
is the KSK for example.com valid?
example.com Key-Signing Key – signs over
example.com Zone-Signing Key – signs over
is the ZSK for example.com valid?
www.example.com
is the signature for this record valid?
www.example.com IN A 192.0.1

As long as you have a valid local trust anchor for the root zone then you can validate a signed DNS response by constructing this backward path to the local root trust anchor

DANE + DNSSEC

- Query the DNS for the TLSA record of the domain name and ask for the DNSSEC signature to be included in the response
- Validate the signature to ensure that you have an unbroken signature chain to the root trust point
- At this point you can accept the TLSA record as the authentic record, and set up a TLS session based on this data

But I can't see DANE in my browser!

Browser vendors appear to be dragging the chain on DANE support, often citing lack of DNSSEC deployment as the excuse de jour

DANE exists today as plug-ins rather than a core functionality

Cynically, one could observe that fast but insecure is the browser vendors’ current preference!

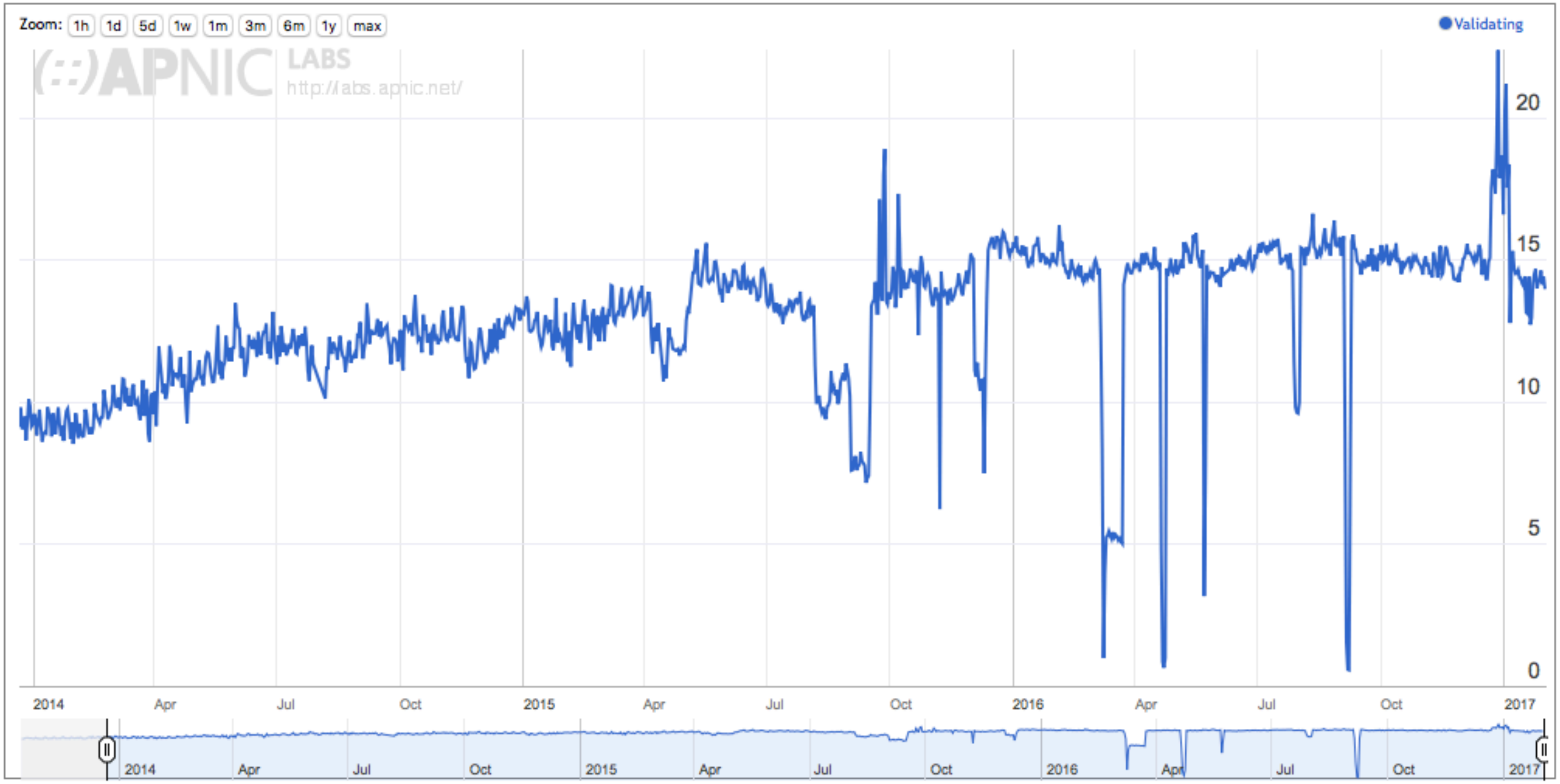
But is it really the case that DNSSEC deployment is lagging?

We need DNSSEC as well as DANE…

How much DNSSEC Validation is out there?

Do we do DNSSEC Validation?

Use of DNSSEC Validation for World (XA)



stats.labs.apnic.net/dnssec/XA

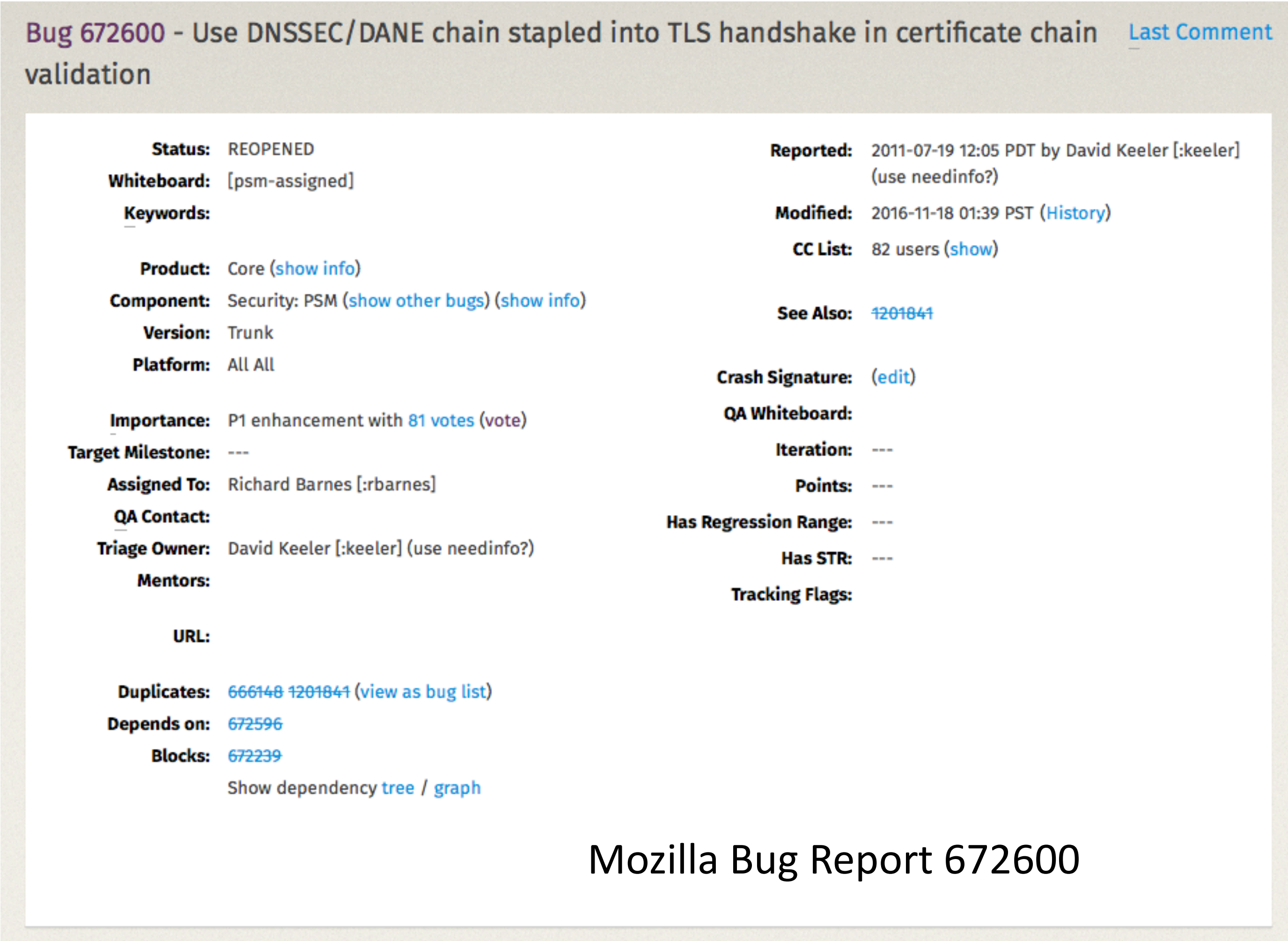
Or…

Look! No DNS!

- Server packages the server cert, TLSA record and the DNSSEC credential chain in a single bundle *
- Client receives bundle in Server Hello
  - *Client performs validation of TLSA Resource Record using the supplied DNSEC signatures plus the local DNS Root Trust Anchor without performing any additional DNS queries*
  - *Client validates the presented certificate against the TLSA RR*
- Client performs Client Key exchange

*** draft-shore-tls-dnssec-chain-extension**

Maybe browsers are ready adopt this approach to TLS + DANE



Bug 672600 - Use DNSSEC/DANE chain stapled into TLS handshake in certificate chain validation

Status: REOPENED
Whiteboard: [psm-assigned]
Keywords:

Product: Core (show info)
Component: Security: PSM (show other bugs) (show info)
Version: Trunk
Platform: All All

Importance: P1 enhancement with 81 votes (vote)
Target Milestone: ---
Assigned To: Richard Barnes [:rbarnes]
QA Contact:
Triage Owner: David Keeler [:keeler] (use needinfo?)
Mentors:

URL:

Duplicates: ~~666148~~ ~~1201841~~ (view as bug list)
Depends on: ~~672596~~
Blocks: ~~672239~~
Show dependency tree / graph

Reported: 2011-07-19 12:05 PDT by David Keeler [:keeler] (use needinfo?)
Modified: 2016-11-18 01:39 PST (History)
CC List: 82 users (show)

See Also: ~~1201841~~

Crash Signature: (edit)
QA Whiteboard:
Iteration: ---
Points: ---
Has Regression Range: ---
Has STR: ---
Tracking Flags:

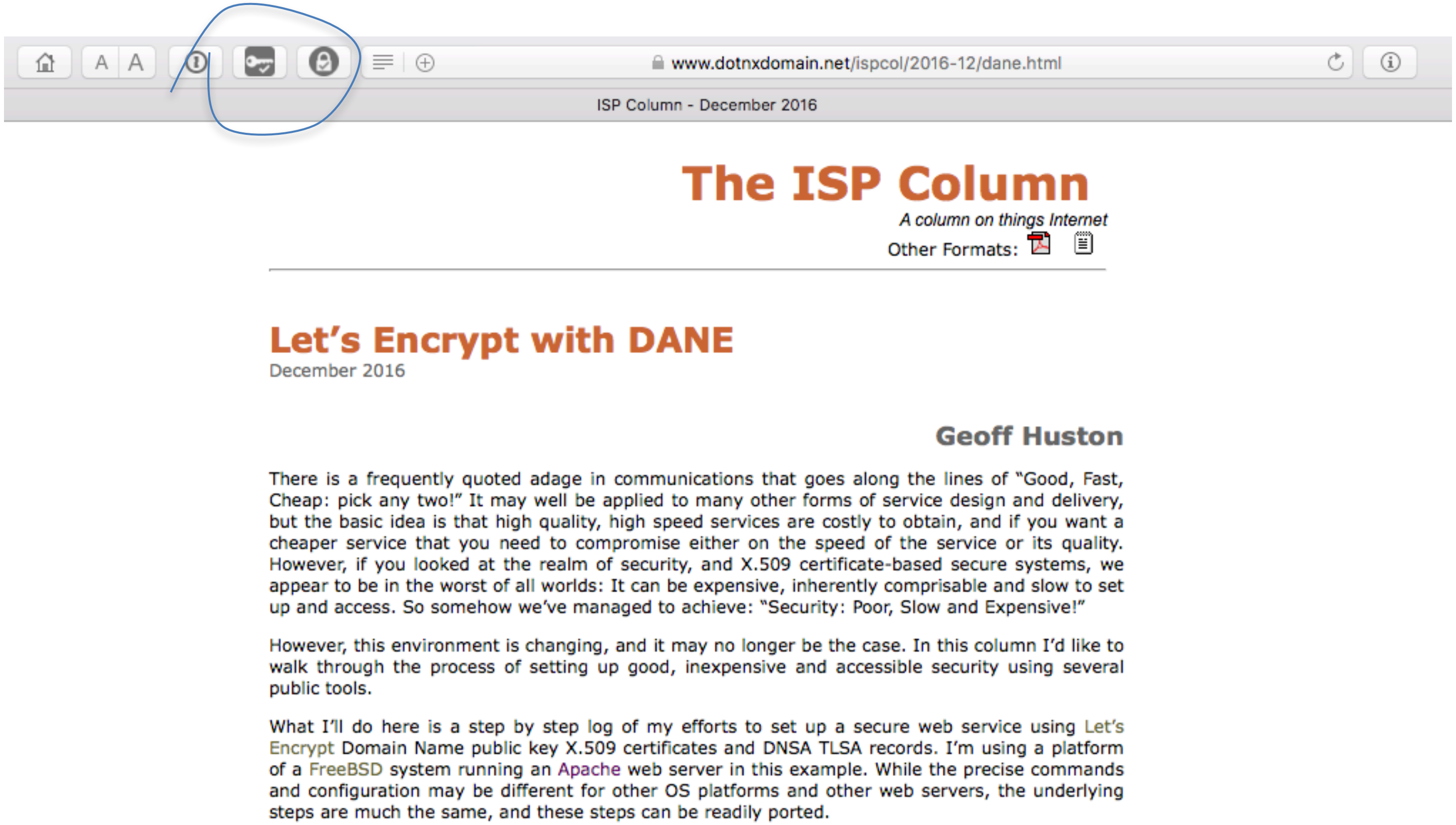
Mozilla Bug Report 672600

Where now?

We could do a **far** better job at Internet Security:

Publishing DNSSEC-signed zones
Publishing DANE TLSA records
Using DNSSEC-validating resolution
Using TLSA records to guide Key Exchange for TLS

Let's Do it!



www.dotnxdomain.net/ispcol/2016-12/dane.html

ISP Column - December 2016

The ISP Column

A column on things Internet

Other Formats:

Let’s Encrypt with DANE

December 2016

Geoff Huston

There is a frequently quoted adage in communications that goes along the lines of “Good, Fast, Cheap: pick any two!” It may well be applied to many other forms of service design and delivery, but the basic idea is that high quality, high speed services are costly to obtain, and if you want a cheaper service that you need to compromise either on the speed of the service or its quality. However, if you looked at the realm of security, and X.509 certificate-based secure systems, we appear to be in the worst of all worlds: It can be expensive, inherently comprisable and slow to set up and access. So somehow we’ve managed to achieve: “Security: Poor, Slow and Expensive!”

However, this environment is changing, and it may no longer be the case. In this column I’d like to walk through the process of setting up good, inexpensive and accessible security using several public tools.

What I’ll do here is a step by step log of my efforts to set up a secure web service using Let’s Encrypt Domain Name public key X.509 certificates and DNSA TLSA records. I’m using a platform of a FreeBSD system running an Apache web server in this example. While the precise commands and configuration may be different for other OS platforms and other web servers, the underlying steps are much the same, and these steps can be readily ported.

What Let’s Encrypt and DNSSEC offers is robust, affordable, accessible security without the current overheads of high priced vanity CA offerings

That's it!

Questions?