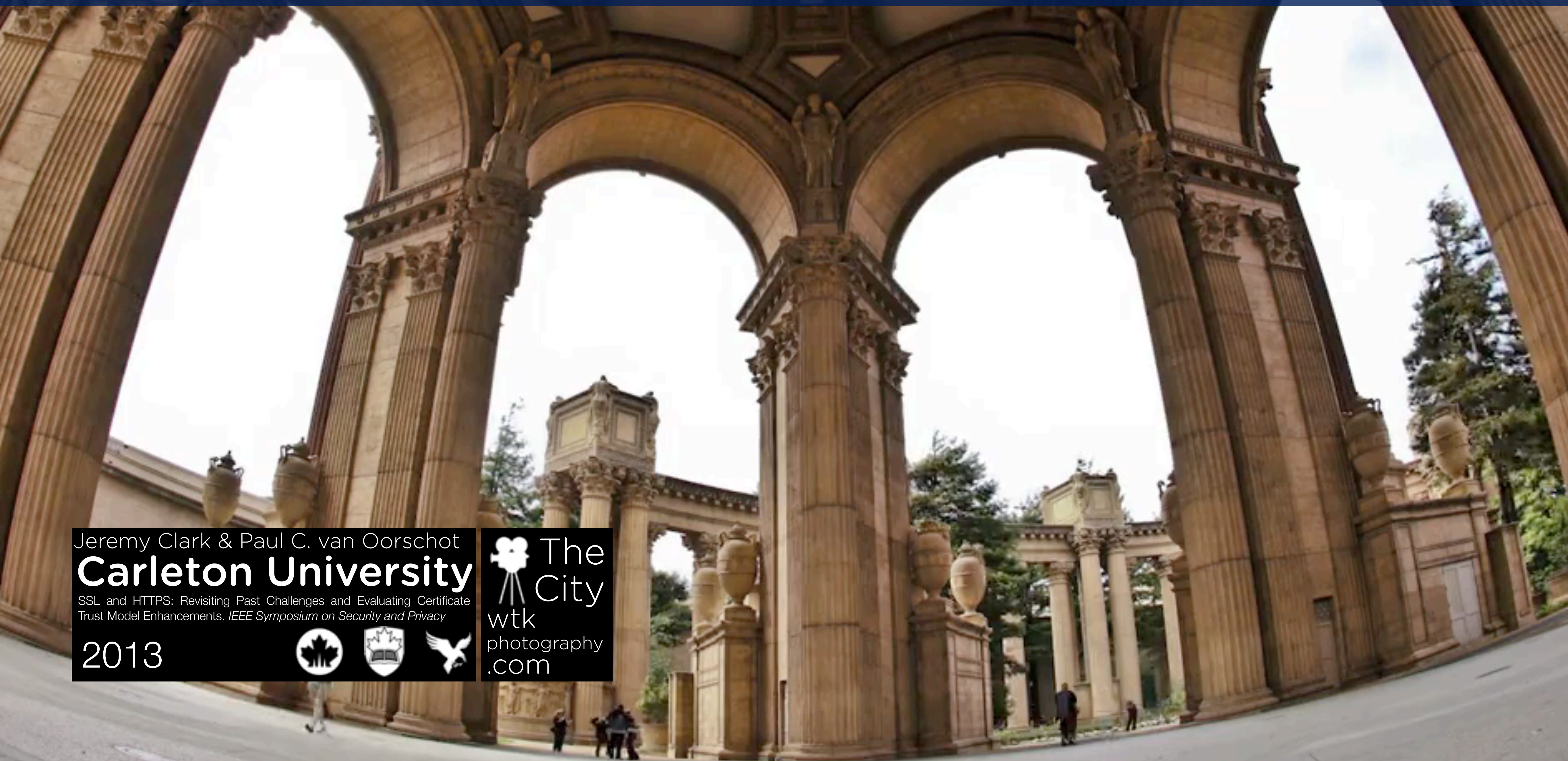
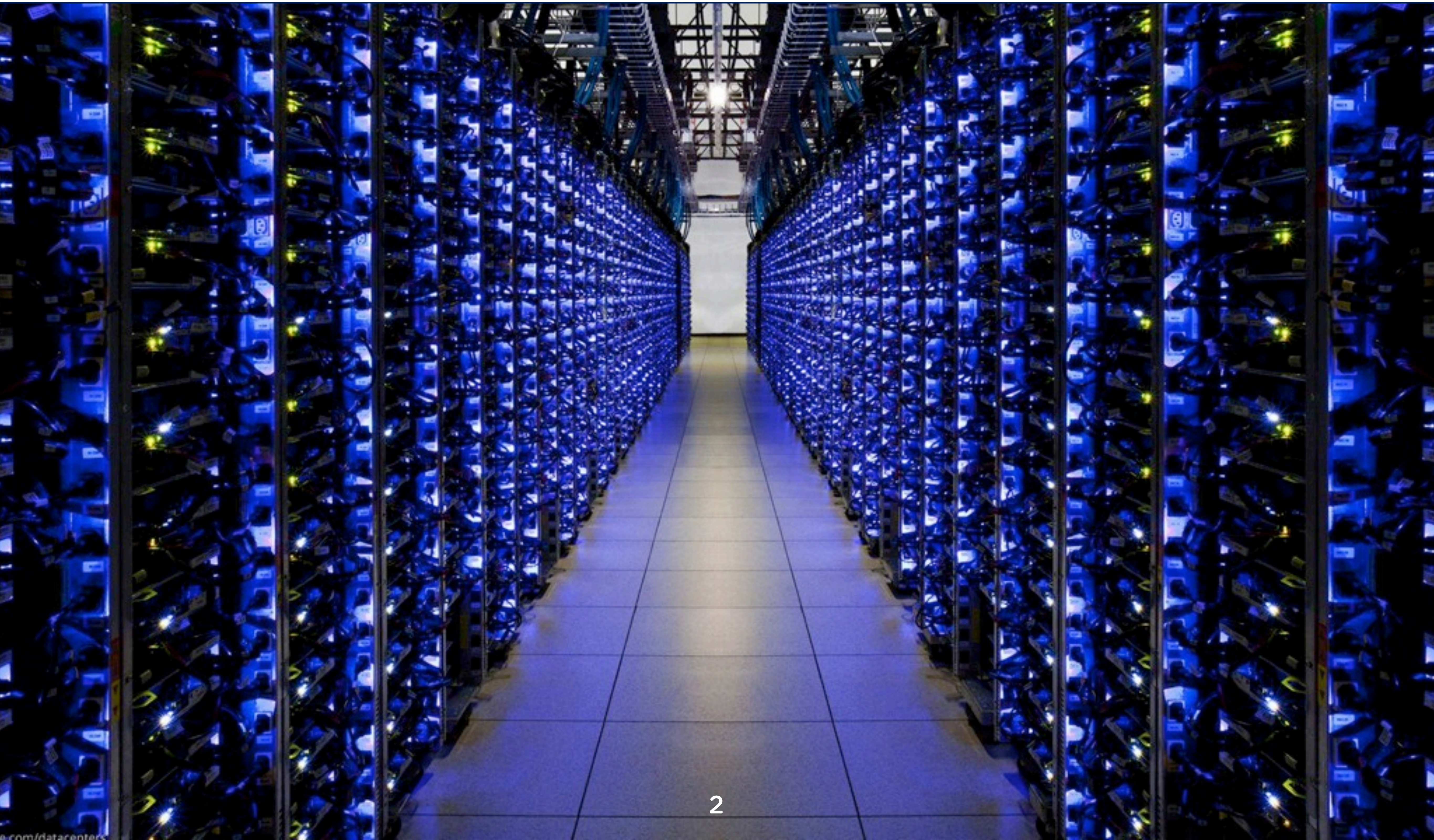
SSL and HTTPS:
Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements

Jeremy Clark & Paul C. van Oorschot



Jeremy Clark & Paul C. van Oorschot
Carleton University
SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. *IEEE Symposium on Security and Privacy*
2013

HTTPS: HTTP over SSL/TLS



HTTPS: HTTP over SSL/TLS

Secure web browsing:

- traffic flows are unmodified and confidential to everyone except the domain owner

HTTPS: HTTP over SSL/TLS

Scope of our work:

1) Cryptographic security of the protocol
2) The CA & browser trust model built around TLS
3) Enhancements to the model:
    a) Detect fake (browser accepted) certificates
    b) Prevent active downgrade to HTTP attacks
    c) Increase the reliability of revocation

Primitive	Detects MITM	Detects Local MITM	Protects Client Credential	Updatable Pins	Detects TLS Stripping	Affirms POST-to-HTTPS	Responsive Revocation	Intermediate CAs Visible	No New Trusted Entity	No New Traceability	Reduces Traceability	No New Auth'n Tokens	No Server-Side Changes	Deployable without DNSSEC	No Extra Communications	Internet Scalable	No False-Rejects	Status Signalled Completely	No New User Decisions
	Security Properties Offered								Evaluation of Impact on HTTPS										
	A				B		C		Security & Privacy				Deployability				Usability		
Key Pinning (Client History)	○	○	○						●	●		●	●	●	●	●			
Key Pinning (Server)	○	○	○						●	●				●	●	●	●		●
Key Pinning (Preloaded)	●	●	●	●					○	●		●	○	●	●		●	○	●
Key Pinning (DNS)	●	●	●	●					○	●	●		○		●	●	●	○	●
Multipath Probing		●		●								●	●	●		●		●	
Channel-bound Credentials			○						●	●		●		●	●	●	●	○	●
Credential-bound Channels			○						●	●		●		●	●	●	●	○	●
Key Agility/Manifest				●					●	●				●	●	●	●	●	●
HTTPS-only Pinning (Server)					○	○			●	●				●	●	●	●		●
HTTPS-only Pinning (Preloaded)				●	●	●			○	●		●	○	●	●		●	○	●
HTTPS-only Pinning (DNS)				●	●	●			○	●			○		●	●	●	○	●
Visual Cues for Secure POST						●			●	●		●	●	●		●		●	
Browser-stored CRL							●		○	●		●	●	●	●	●	●	●	●
Certificate Status Stapling							●		●	●	●			●	●	●	●	○	●
Short-lived Certificates							●		●	●	●	●		●	●	●	●	●	●
List of Active Certificates							●	●			●	●	●	●		●	●	●	●


November

THREAT LEVEL | cybersecurity

Google Discovers Fraudulent Digital Certificate Issued for Its Domain

BY KIM ZETTER 01.03.13 4:50 PM

Follow @KimZetter

Share 86
Tweet 191
+1 113
Share 30

...to: Alain Bachellier / Flickr

...updated 1.4.12; see below]

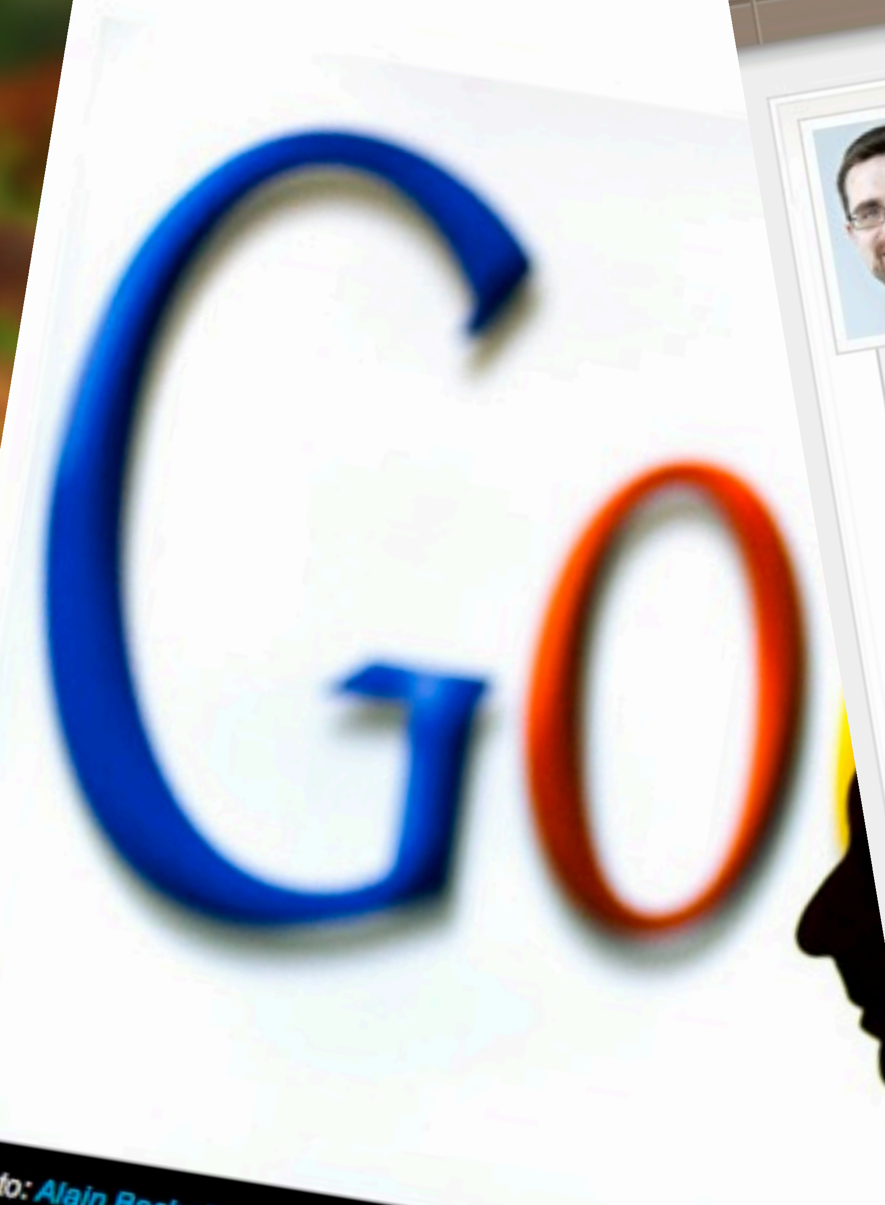
...asn't the only one sneaking around on G...
...ght trying to use a...

THREAT LEVEL | cybersecurity

Google Discovers Fi
Certificate Issued fo

BY KIM ZETTER 01.03.13 4:50 PM

Follow @KimZetter



to: Alain Bachellier / Flickr

updated 1.4.12; see below]

vasn't the only one sneaking around on G
ght trying to use an

FREEDOM TO TINKER

research and expert commentary on digital technologies in public life

Turktrust Certificate Authority Errors Demonstrate The Risk of "Subordinate" Certificates

JANUARY 3, 2013 BY STEVE SCHULTZE

Update: More details have continued to come out, and I think that they generally support the less-paranoid version of events. There continues to be discussion on the **mozilla.dev.security.policy** list, Turktrust has **given more details**, and Mozilla has just opened up for public viewing their own **detailed internal response documentation** (including copies of all of the certs in question). None of this changes the fundamental riskiness of subordinate certificates, or the improvements that should be made to the CA system. It just means that in this case, the failure didn't progress to a full-blown meltdown.

Today, the public learned of another failure by a Certificate Authority–one of of companies that certifies SSL-encryption for our internet communications. (See the end of this post for a catalogue of our past writing on problems with this "CA" system.) This time, the company **Turktrust** was revealed to have **issued two *subordinate certificates*** (also known as "intermediate" certificates) to entities that should not have had them. Subordinate certificates are very powerful. They give the holder the ability to issue SSL certificates for any domain name as though they have control of the parent CA's "root" certificate. In this case, **Google discovered** that one of Turktrust's previously undisclosed subordinate certificates had issued SSL certificates for the domain **gmail.com**, and that these certificates had been used to intercept Gmail users' traffic... somewhere. This is where the details get foggy, but **Turktrust has begun to describe their version of events**.

There is a less paranoid and a more paranoid way of interpreting what happened.

According to Turktrust, they made some configuration errors in late 2011 that caused them to inadvertently hand out subordinate certificates to two entities–the Turkish government and a Turkish bank. They claimed that these users expected to receive ordinary SSL certificates that could be used to secure their own web sites, not what amounts to a master **"skeleton key" to the internet**. This assertion is somewhat supported by the fact that one of these certificates appears to have been used for precisely that until recently, at **www.ego.gov.tr** (which seems to be the web site for the ... Ankara or one of its offices, but I'm sure that Zeynep will clear this up in the comments). Turktrust says that ... ment to provide SSL-encrypted webmail–presumably for government employees ... dinate certificates as though they are ordinary SSL certific...). In any case, Turktrust says that a ... man-in-the


THREAT LEVEL
cybersecurity
Google Discovers Fr
Certificate Issued fo
BY KIM ZETTER 01.03.13 4:50 PM
Follow @KimZetter
updated 1.4.12; see below]
FREEDOM TO TINKER
research and expert commentary on digital technologies in public life
Turktrust Certificate Authority Errors Demonstrate The Risk of "Subordinate" Certificates
JANUARY 3, 2013 BY STEVE SCHULTZE
Update: More details have continued to come out, and I think that they generally support the less-paranoid version of events. There continues to be discussion on the mozilla.dev.security.policy list, Turktrust has given more details, and Mozilla has just opened up for public viewing their own detailed internal response documentation (including copies of all of the certs in question). None of this changes the fundamental riskiness of subordinate certificates, or the improvements that should be made to the CA system. It just means that in this case, the failure didn't progress to a full-blown meltdown.
Today, the public learned of another failure by a Certificate Authority–one of of companies that certifies SSL-encryption for our internet communications. (See the end of this post for a catalogue of our past writing on problems with this "CA" system.) This time, the company Turktrust was revealed to have issued two subordinate certificates (also known as
TURKTRUST

Nokia's Xpress Browser Decrypts Your HTTPS Data
t's come to light that Nokia's Xpress Browser—used on its Asha and
umia handsets—routes your secure and encrypted HTTPS data through
s servers and temporarily decrypts it.
aOm reports security research which reveals that Nokia's browser passes data to its servers—
hing odd there, it's a proxy browser designed to speed things up—but that, at stages, HTTPS
is decrypted. Nokia has gone as far as admitting that it's the case, but reassures consumers
it doesn't look at the data:
"Importantly, the proxy servers do not store the content of web pages
users or any information they enter into th
TINKER
ologies in public life
Errors Demonstrate The
es
hat they generally support the less-paranoid
ozilla.dev.security.policy list, Turktrust has
lic viewing their own detailed internal response
tion). None of this changes the fundamental
that should be made to the CA system. It just means
meltdown.
Authority–one of of companies that certifies SSL-encryption for
catalogue of our past writing on problems with this "CA"
o have issued two subordinate certificates (also known as
e had them. Subordinate certificates are very powerful. They give
main name as though they have control of the pa
Turktrust's previously undisclosed subordinate
ed them to inadvertent
. They claimed that these users
vn web sites, not what amounts to a
at could be used to secure th
ssertion is somewhat supported by the fact that one of these certificates
until recently, at www.ego.gov.tr (which seems to be the web site for the
t I'm sure that Zeynep will clear this up in the comments). Turktrust says that
to provide SSL-encrypted webmail–presumably for government employees
UST

Nokia's Xpress Browser D... Data

...t's come to light that Nokia's Xpress Browse... ...umia handsets—routes your secure and encry... ...s servers and temporarily decrypts it.

...aOm reports security research which reveals that Nokia's br... ...hing odd there, it's a proxy browser designed to speed things ... is decrypted. Nokia has gone as far as admitting that it's the ... it doesn't look at the data:

"Importantly, the proxy servers do not store the content of web ... users or any information they enter into th... ...HTTPS connecti...

ars technica

MAIN MENU MY STORIES: 0 FORUMS SUBSCRIBE VIDEO

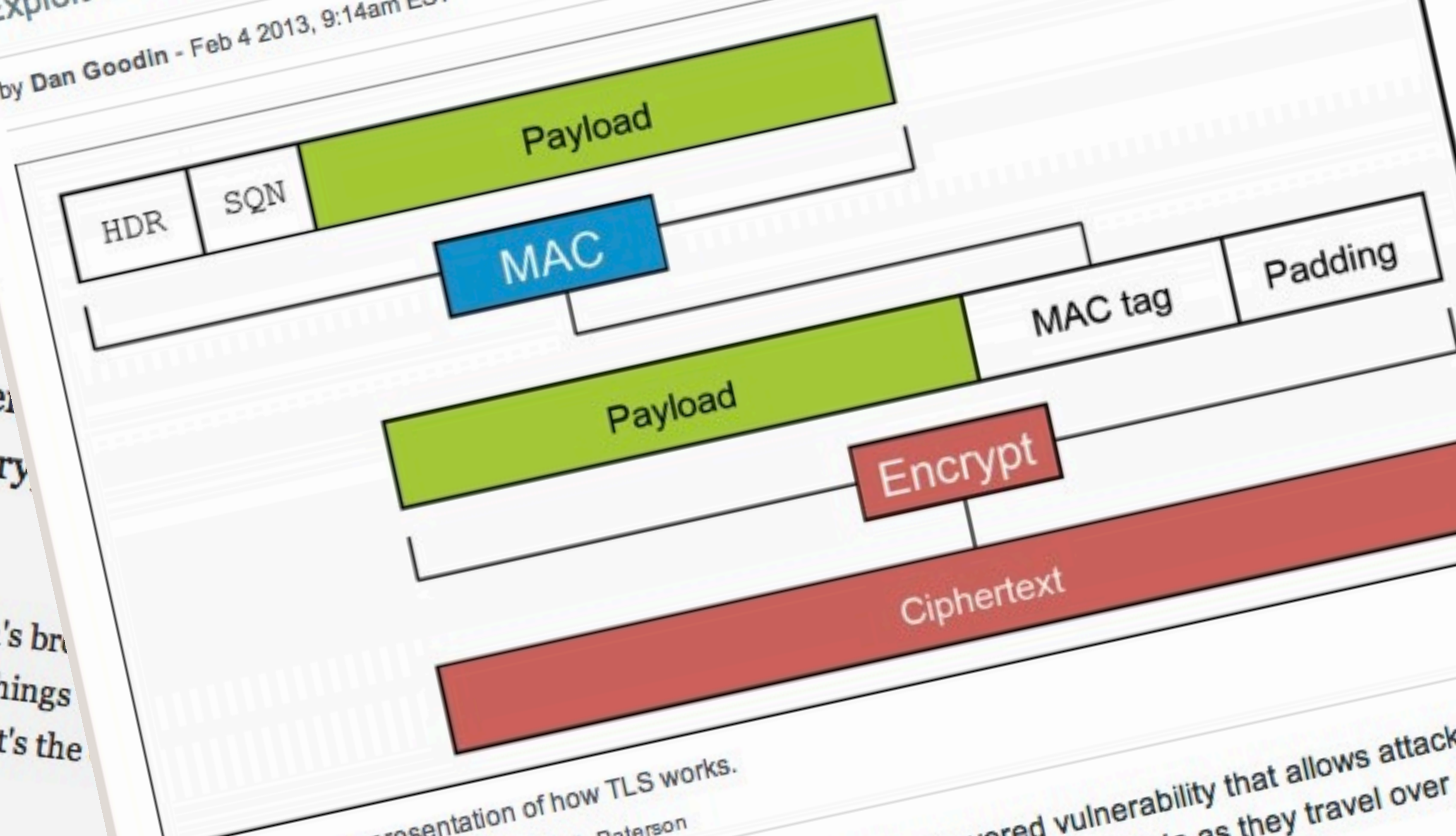
RISK ASSESSMENT / SECURITY & HACKTIVISM

"Lucky Thirteen" attack snarfs cookies protected by SSL encryption

Exploit is the latest to subvert crypto used to secure Web transactions.

by Dan Goodin - Feb 4 2013, 9:14am EST

HACKING PRIVACY 18



Enlarge / A representation of how TLS works.

Nadhem J. AlFardan and Kenneth G. Paterson

...are racing to patch a recently discovered vulnerability that allows attack... ...ion cookies and other encrypted data as they travel over... ...attackers to co...

ImperialViolet

Lucky Thirteen attack on TLS CBC (04 Feb 2013)

In an upcoming paper (made public this morning), Nadhem AlFardan and Kenny Paterson describe another method of performing Vaudenay's attack on CBC as used in TLS. Firstly I'd like to thank the researchers for notifying the various vendors ahead of time so that patches could be prepared: the disclosure process has gone very smoothly in this case. I couldn't have asked for anything more - they did everything right.

Vaudenay's attack requires an attacker to be able to detect when a CBC padding check has succeeded, even if the authentication check then fails. Authentication should always be applied after encryption to avoid this, but TLS famously did it the wrong way round with CBC mode.

Knowing whether a padding check succeeded reveals information about the decrypted plaintext. By tweaking the ciphertext over many trials, it's possible to progressively decrypt an unknown ciphertext. For details, see their paper, which does a better job of explaining it than I would.

Vaudenay first used the fact that these two situations (padding check failure and authenticator failure) resulted in different TLS alert values, although that didn't result in a practical attack because TLS errors are encrypted. Once that was corrected (by specifying that the same alert value should be sent in each case) the next year's paper used a timing-side channel: if the authentication check wasn't performed when the padding check failed then, by timing the server's response, an attacker could tell whether the server aborted processing early.

To try and remove that side-channel, TLS stacks perform the authentication check whether or not the padding check succeeds. However, here's what I commented in the Go TLS code:

"Note that we still have a timing side-channel in the MAC check, below. An attacker can align the record so that a correct padding will cause one less hash block to be calculated. Then they can iteratively decrypt a record by breaking each byte. However, our behavior matches OpenSSL, so we leak only as much as they do."

That pretty much sums up the new attack: the side-channel defenses that were hoped to be sufficient were found not to be (again). So the answer, this time I believe, is to make the processing rigorously constant-time. (Details below.)

As a practical matter, since a padding or authenticator check failure is fatal to a TLS connection, performing this attack requires a client to send the same plaintext secret on thousands of different connections to the same server. This isn't a trivial obstacle but it's possible to meet this requirement for cookies with a browser and bit of Javascript injected into any origin in the same session.

For DTLS the attack is much easier because a rejected record doesn't cause the connection to be destroyed and the same authors developed a method for amplifing timing attacks against DTLS in a previous paper

ImperialVi

Lucky Thirteen attack on TLS

In an upcoming paper (made pu
performing Vaudenay's attack on
vendors ahead of time so that pa
couldn't have asked for anything r

Vaudenay's attack requires an a
authentication check then fails. Au
it the wrong way round with CBC r

Knowing whether a padding chec
over many trials, it's possible to
better job of explaining it than I wo

Vaudenay first used the fact that
TLS alert values, although that di
(by specifying that the same alert
authentication check wasn't perfo
could tell whether the server abort

To try and remove that side-cha
succeeds. However, here's what I

"Note that we still have a timing s
padding will cause one less hash
However, our behavior matches C

That pretty much sums up the ne
(again). So the answer, this time I

As a practical matter, since a pa
requires a client to send the same
obstacle but it's possible to meet
the same session.

For DTLS the attack is much eas
authors developed a method for a

A Few Thoughts on Cryptographic Engineering

Some random thoughts about crypto. Notes from a course I teach. Pictures of my dachshunds.

Monday, February 4, 2013

Attack of the week: TLS timing oracles

Ever since I started writing this blog (and specifically, the posts on SSL/TLS) I've had a new experience: people come up to me and share clever attacks that they haven't made public yet.

This is pretty neat -- like being invited to join an exclusive club. Unfortunately, being in this club mostly sucks. That's because the first rule of 'TLS vulnerability club' is: *You don't talk about TLS vulnerability club*. Which takes all the fun out of it.

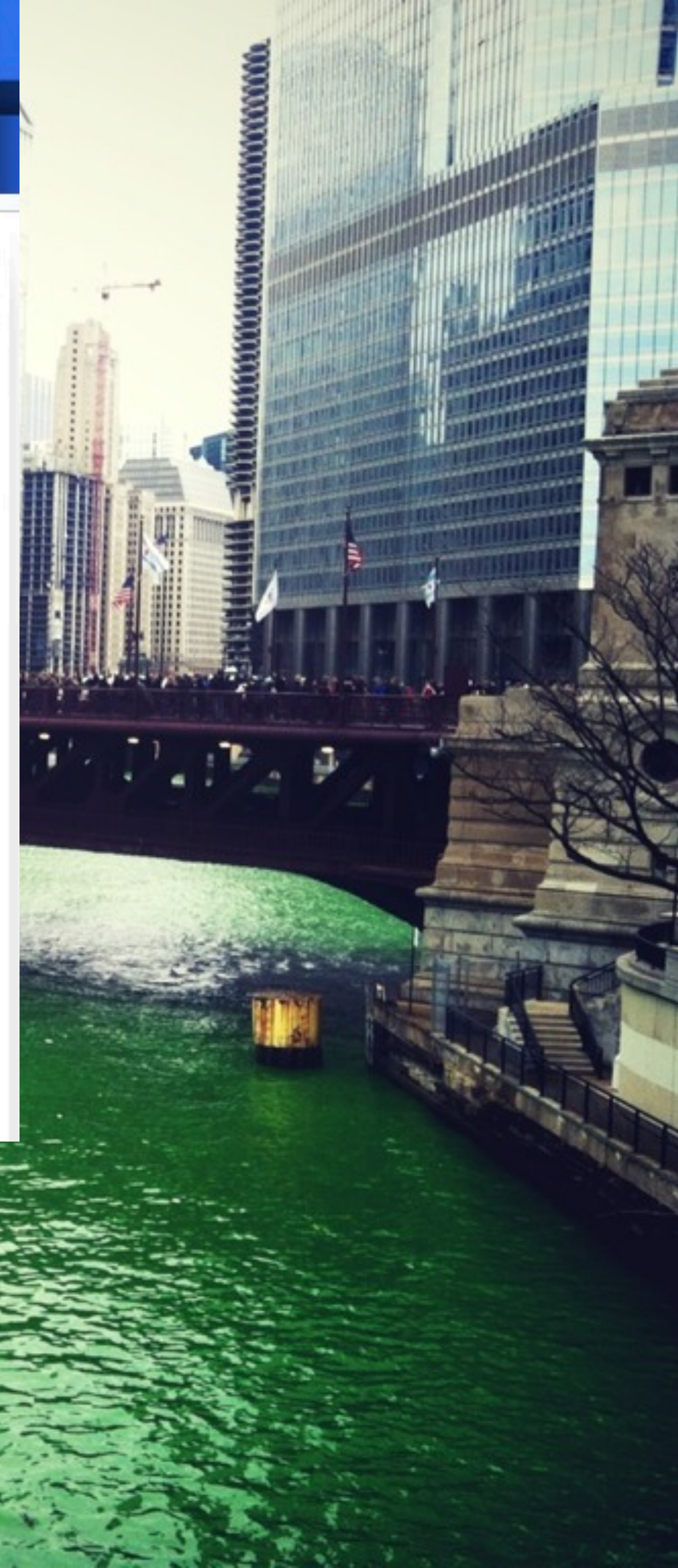
(Note that this is all for boring reasons -- stuff like responsible disclosure, publication and fact checking. Nobody is planning a revolution.)

Anyway, it's a huge relief that I'm finally free to tell you about a neat new TLS attack I learned about recently. The new result comes from Nadhem AlFardan and Kenny Paterson of Royal Holloway. Dubbed 'Lucky 13', it takes advantage of a very subtle bug in the way records are encrypted in the TLS protocol.

If you aren't into long crypto posts, here's the TL;DR:

> There is a subtle timing bug in the way that TLS data decryption works when using the (standard) CBC mode ciphersuite. Given the right set of circumstances, an attacker can use this to completely decrypt sensitive information, such as passwords and cookies.
>
> The attack is borderline practical if you're using the Datagram version of TLS (DTLS). It's more on the theoretical side if you're using standard TLS. However, with some clever engineering, that could change in the future. You should probably patch!

March



SECURITY LABS

‹ Previous Next ›

Posted by ivanr on Mar 19, 2013 5:32:41 AM

RC4 in TLS is Broken: Now What?

RC4 has long been considered problematic, but until very recently there was no known way to exploit the weaknesses. After the BEAST attack was disclosed in 2011, we—grudgingly—started using RC4 in order to avoid the vulnerable CBC suites in TLS 1.0 and earlier. This caused the usage of RC4 to increase, and some say that it now accounts for about 50% of all TLS traffic.

Last week, a group of researchers (Nadhem AlFardan, Dan Bernstein, Kenny Paterson, Bertram Poettering and Jacob Schuldt) announced significant advancements in the attacks against RC4, unveiling new weaknesses as well as new methods to exploit them. Matthew Green has a great overview on his blog, and here are the slides from the talk where the new issues were announced.

At the moment, the attack is not yet practical because it requires access to millions and possibly billions of copies of the same data encrypted using different keys. A browser would have to make that many connections to a server to give the attacker enough data. A possible exploitation path is to somehow instrument the browser to make a large number of connections, while a man in the middle is observing and recording the traffic.

We are still safe at the moment, but there is a tremendous incentive for researchers to improve the attacks on RC4, which means that we need to act swiftly.

SECURITY LA

Posted by ivanr on Mar 19, 2013 5:32:41 AM

RC4 in TLS is Broken: Now

RC4 has long been considered problematic, but until very recently
weaknesses. After the BEAST attack was disclosed in 2011, we
avoid the vulnerable CBC suites in TLS 1.0 and earlier. This caus
some say that it now accounts for about 50% of all TLS traffic.

Last week, a group of researchers (Nadhem AlFardan, Dan Bernst
Poettering and Jacob Schuldt) announced significant advancemer
new weaknesses as well as new methods to exploit them. Matthe
and here are the slides from the talk where the new issues were a

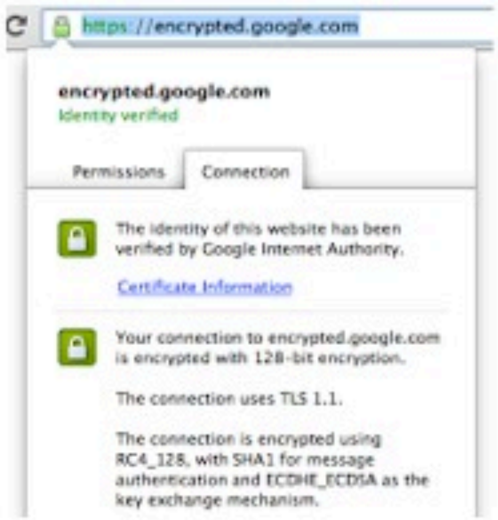
At the moment, the attack is not yet practical because it requires
copies of the same data encrypted using different keys. A browse
connections to a server to give the attacker enough data. A possil
instrument the browser to make a large number of connections, wl
recording the traffic.

We are still safe at the moment, but there is a tremendous incentive for researchers to improve the attacks on RC4, which means that we need to act swiftly.

Tuesday, March 12, 2013

Attack of the week: RC4 is kind of broken in TLS

Update: *I've added a link to a page at Royal Holloway describing the new attack.*



Listen, if you're using RC4 as your primary ciphersuite in SSL/TLS, now would be a great time to stop. Ok, thanks, are we all on the same page?

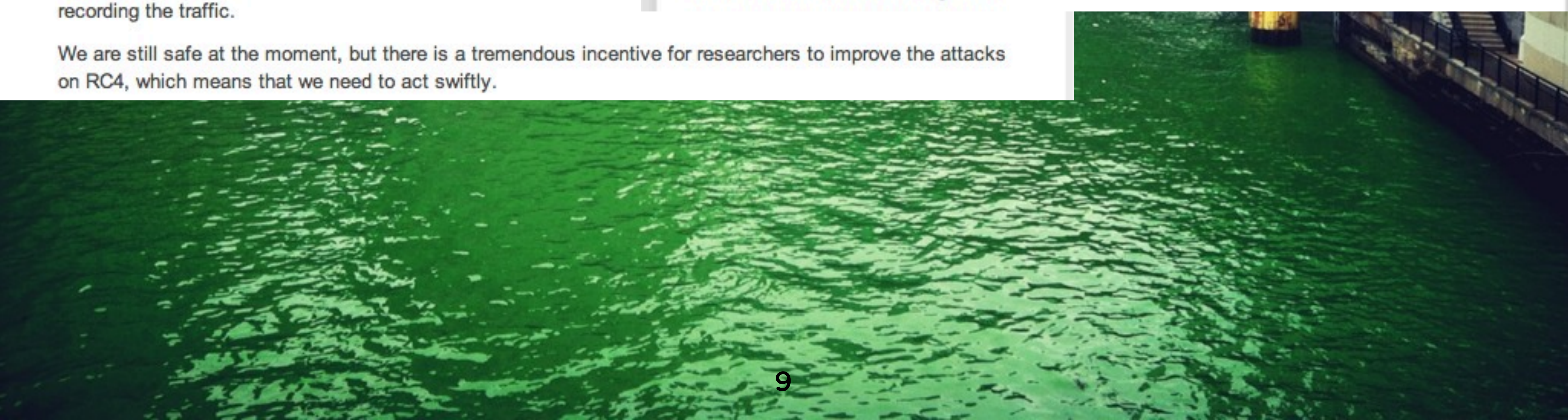
No?

I guess we need to talk about this a bit more. You see, these slides have been making the rounds since this morning. Unfortunately, they contain a long presentation aimed at cryptographers, and nobody much seems to understand the *real* result that's buried around slide 306 (!). I'd like to help.

Here's the short summary:

According to AlFardan, Bernstein, Paterson, Poettering and Schuldt (a team from Royal Holloway, Eindhoven and UIC) the RC4 ciphersuite used in SSL/TLS is broken. If you choose to use it -- as do a ridiculous number of major sites, including Google -- then it may be possible for a dedicated attacker to recover your authentication cookies. The current attack is just on the edge of feasibility, and could probably be improved for specific applications.

This is bad and needs to change soon.



SECURITY LA

Posted by ivanr on Mar 19, 2013 5:32:41 AM

RC4 in TLS is Broken: Now

Tuesday, March 12, 2013

Attack of the week: RC4 is kind of broken in TLS

***Update:** I've added a link to a page at Royal Holloway describing the new attack.*

Listen, if you're using RC4 as your primary ciphersuite in SSL/TLS, now would be a great time to stop. Ok, thanks, are we all on the same page?

No?

I guess we need to talk about this a bit more. You see, these ... is morning.
... on aimed at
... understand the
... 'd like to help.

https://encrypted.google.com
encrypted.google.com
Identity verified
Permissions
Connection
The identity of this website has been verified by Google Internet Authority.
Certificate Information
Your connection to encrypted.google.com is encrypted with 128-bit encryption.
The connection uses TLS 1.1.
The connection is encrypted using RC4_128, with SHA1 for message authentication and ECDHE_ECDSA as the key exchange mechanism.

Poettering and Schuldt (a team from Royal Holloway,
... ed in SSL/TLS is broken. If you choose to use it -- as do
Google -- then it may be possible for a dedicated
... ies. The current attack is just on the edge of feasibility,
applications.

The Register®

Data Centre Cloud Software Networks Security Policy Business Jobs Hardware Science Bootnotes

Print Like 493 Tweet 455 Alert

Firefox 'death sentence' threat to TeliaSonera over gov spy claims

Mozilla may snub telecom giant's new SSL certs

By **Gavin Clarke** • **Get more from this author**

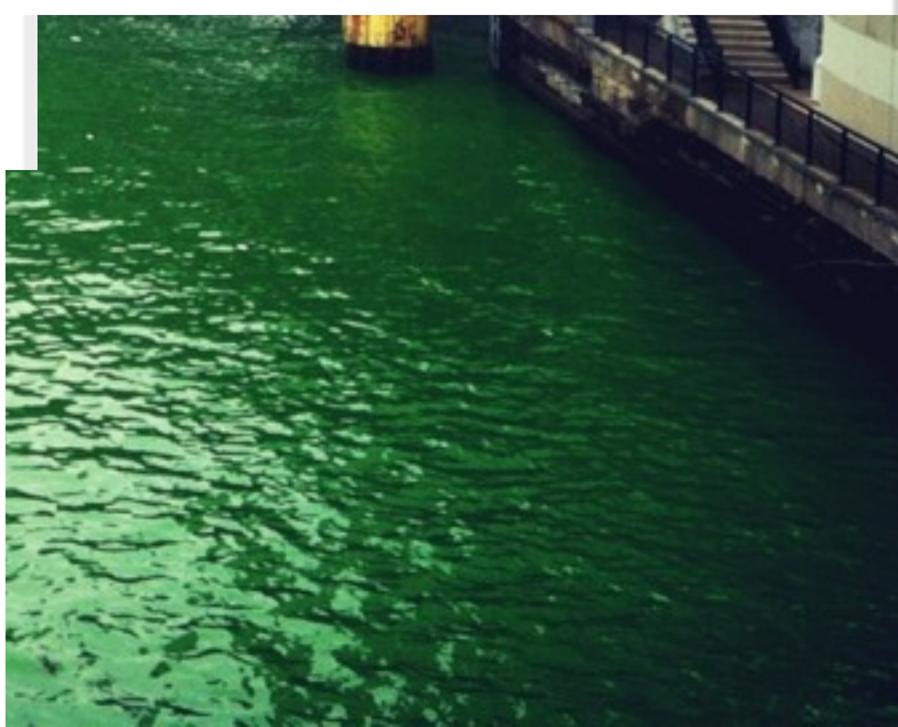
Posted in Security, 16th April 2013 10:19 GMT

Firefox-maker Mozilla could issue a "death sentence" to TeliaSonera's SSL business over allegations the telecoms giant sold Orwellian surveillance tech to dictators.

The punishment would be an embarrassing blow to the company: it would effectively cut off HTTPS-encrypted websites verified by TeliaSonera from Firefox users, who make up one-fifth of the planet's web surfers.

Crucially, it will be seen as a tough stance against corporations that trade with authoritarian states.

TeliaSonera, which has globe-spanning operations and sells SSL certificates to Nordic websites, asked Mozilla to include its new root certificate in Firefox's list of trusted Certificate Authorities (CAs).



SECURITY LA

Posted by ivanr on Mar 19, 2013 5:32:41 AM

RC4 in TLS is Broken: Now

Tuesday, March 12, 2013

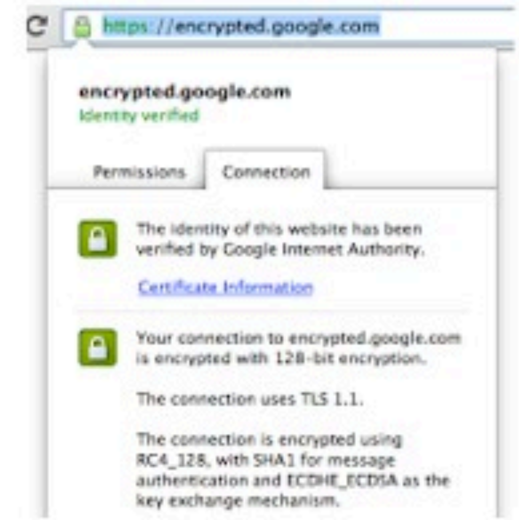
Attack of the week: RC4 is kind of broken in TLS

***Update:** I've added a link to a page at Royal Holloway describing the new attack.*

Listen, if you're using RC4 as your primary ciphersuite in SSL/TLS, now would be a great time to stop. Ok, thanks, are we all on the same page?

No?

I guess we need to talk about this a bit more. You see, these slides have been making the rounds since this morning. ... on aimed at ... understand the ... 'd like to help.



... Poettering and Schuldt (a team from Royal Holloway, ... ed in SSL/TLS is broken. If you choose to use it -- as do ... Google -- then it may be possible for a dedicated ... ies. The current attack is just on the edge of feasibility, ... applications.

The Register®

Data Centre | Cloud | Software | Networks | Security | Policy | Business | Jobs | Hardware | Science | Bootnotes

Print | Like 493 | Tweet 455 | Alert

Firefox 'death sentence' threat to TeliaSonera over gov spy claims

Mozilla may snub telecom giant's new SSL certs

By **Gavin Clarke • Get more from this author**

Posted in Security, 16th April 2013 10:19 GMT

Firefox-maker Mozilla could issue a "death sentence" ... the telecoms giant sold Orwellian surveillance tech to ...

The punishment would be an embarrassing blow to th... encrypted websites verified by TeliaSonera from Firef... web surfers.

Crucially, it will be seen as a tough stance against cor...

TeliaSonera, which has globe-spanning operations an... Mozilla to include its new root certificate in Firefox's lis...

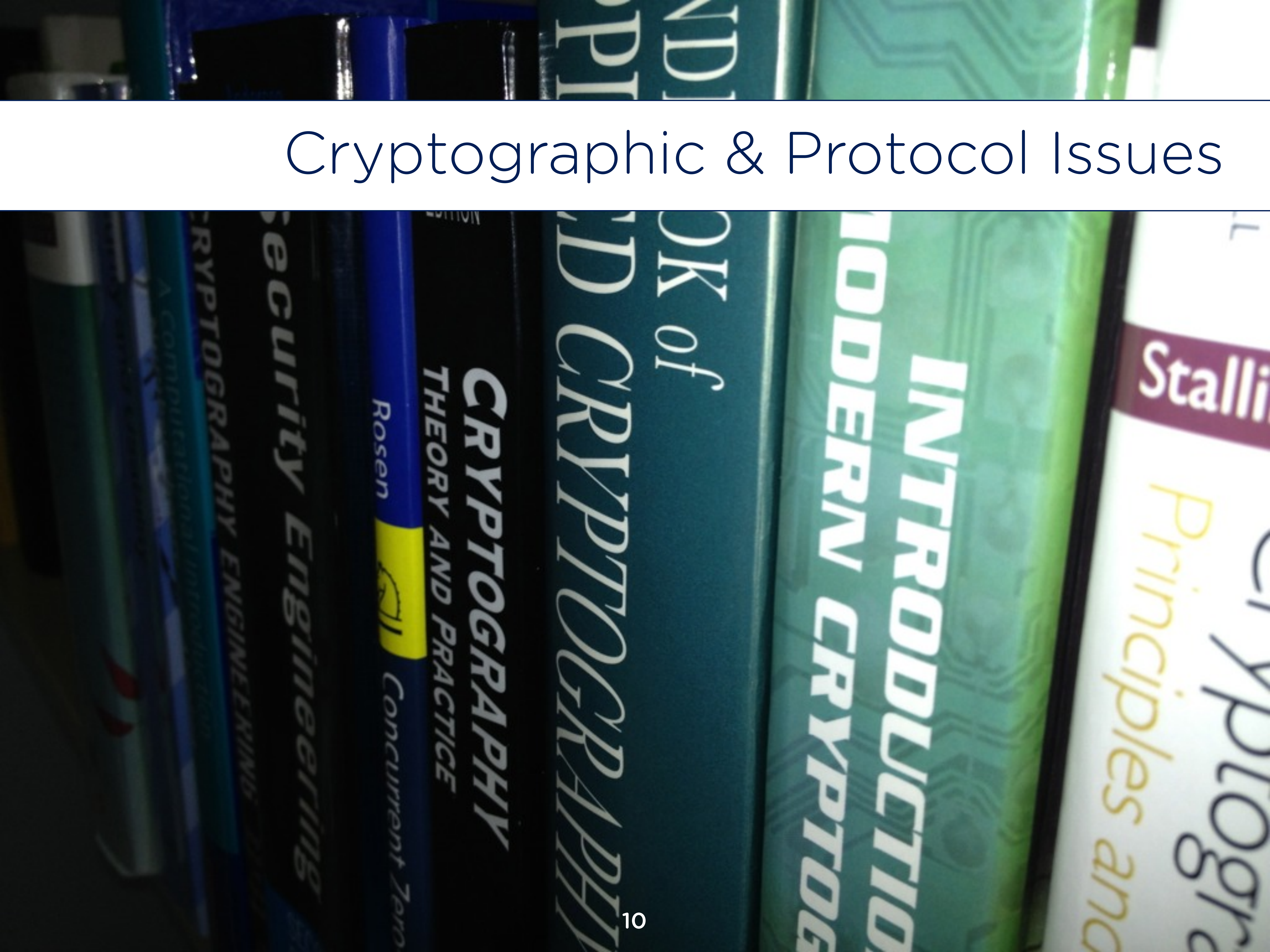
Mozilla Is Considering Revoking TeliaSonera Trust For Sales To Dictators

Posted by **Soulskill** on Tuesday April 16, 2013 @05:59PM
from the trust-must-be-deserved dept.

ndogg writes

> "Mozilla is considering pulling TeliaSonera from its list of root certificate SSL providers. They have asked for comments on this on their mailing list. They're concerned about the use of the certificates by those governments for spying on its citizens, particularly in Azerbaijan, Kazakhstan, Georgia, Uzbekistan and Tajikistan — where TeliaSonera operates subsidiaries or is heavily invested. Mozilla's concern is that TeliaSonera has possibly issued certificates that allow hardline government servers to masquerade as legitimate websites — so-called man-in-the-middle attacks — and decrypt web traffic. This alleged activity would contradict Mozilla's policy against 'knowingly issuing certificates without the knowledge of the entities whose information is referenced in the certificates.'"

Cryptographic & Protocol Issues



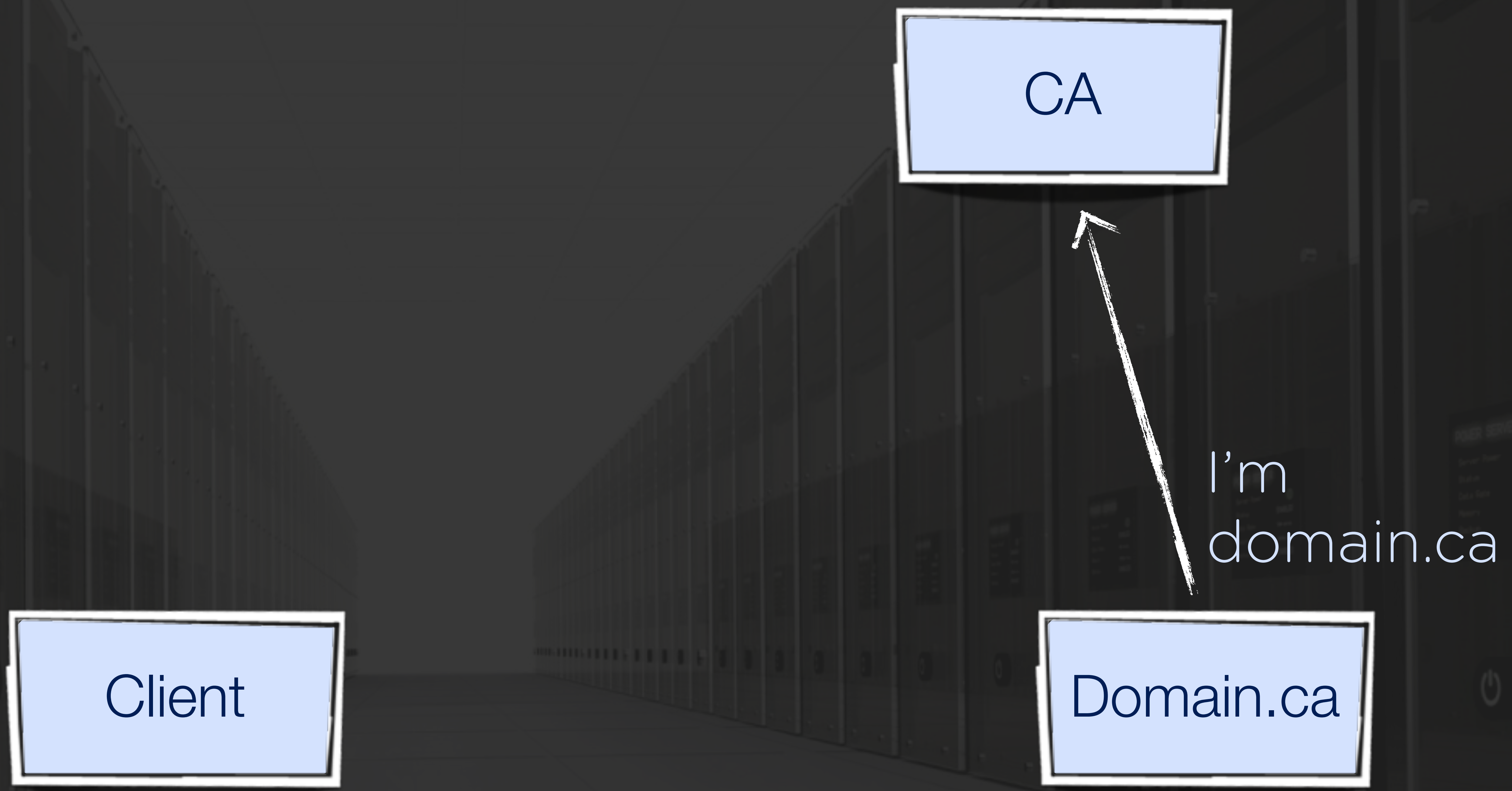
Cryptographic & Protocol Issues

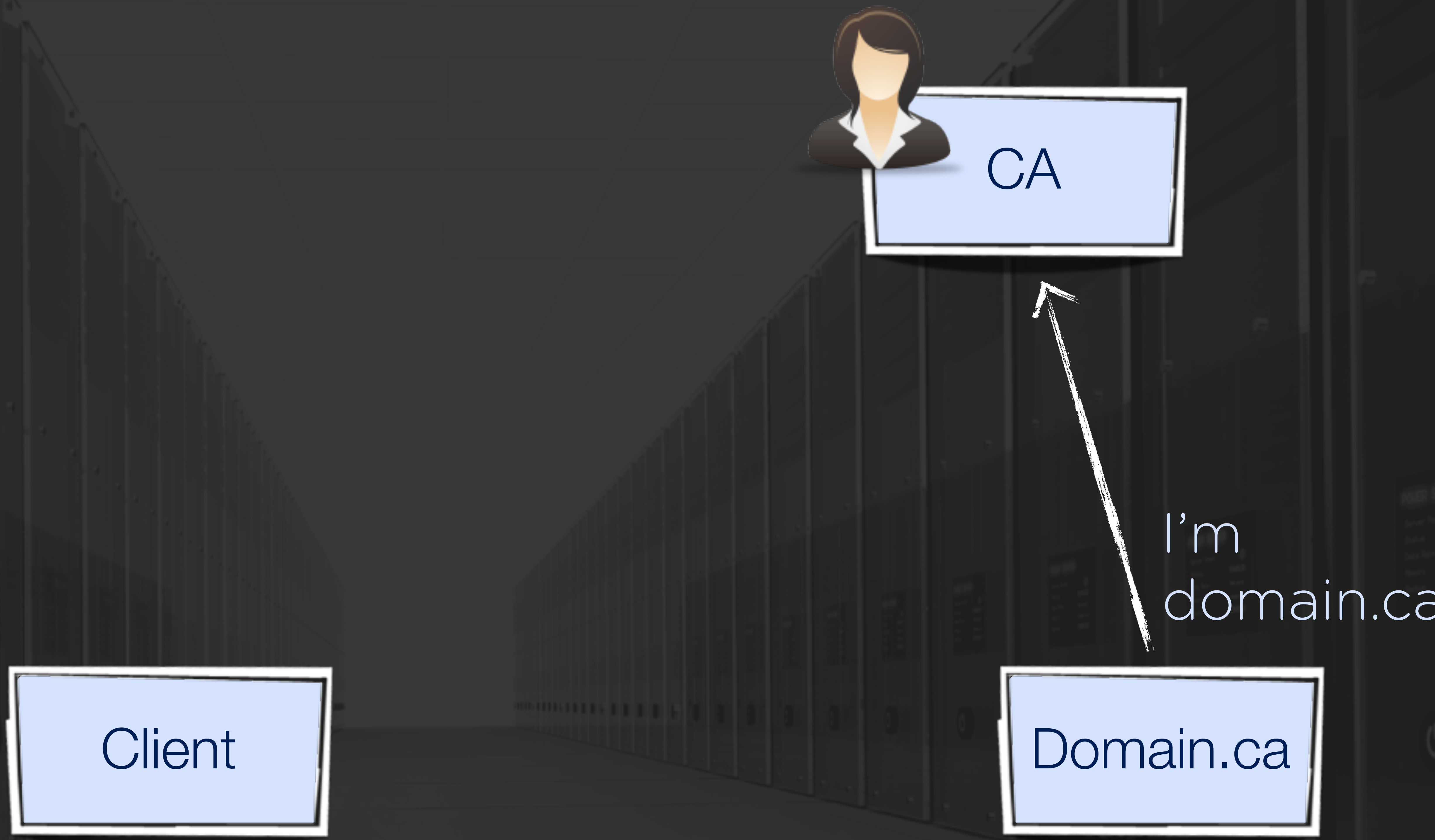
See Paper:

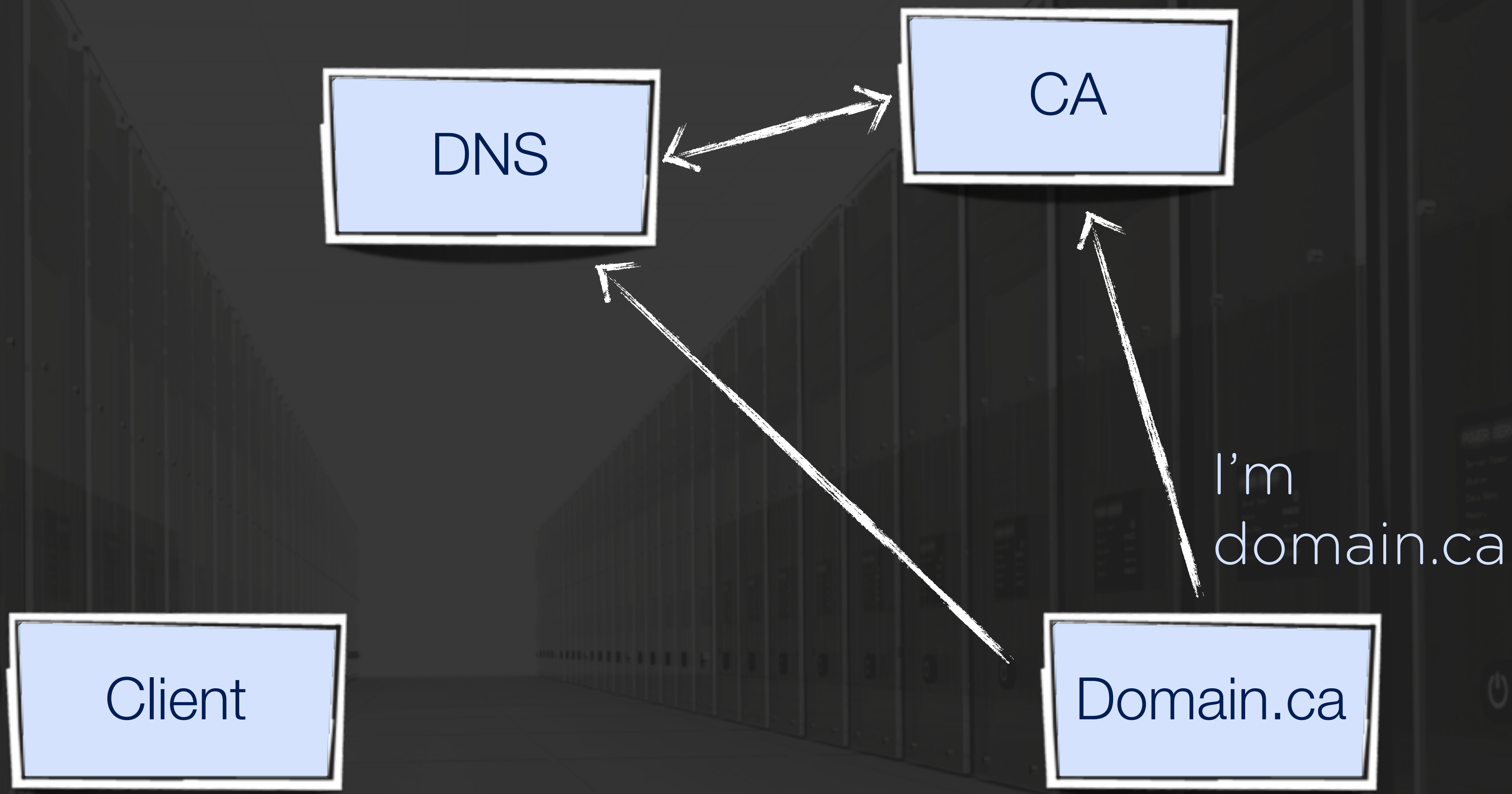
- Aging Primitives: MD2, MD5, RC4, weak keys
- Bad randomness: Netscape, Debian, embedded devs
- Timing Attacks: RSA, ECDSA
- Encryption Oracles: Predictable IVs, Compression
- Decryption Oracles: RSA encoding, CBC padding
- Protocol Flaws: Renegotiation
- Downgrade Attacks: version & ciphersuite

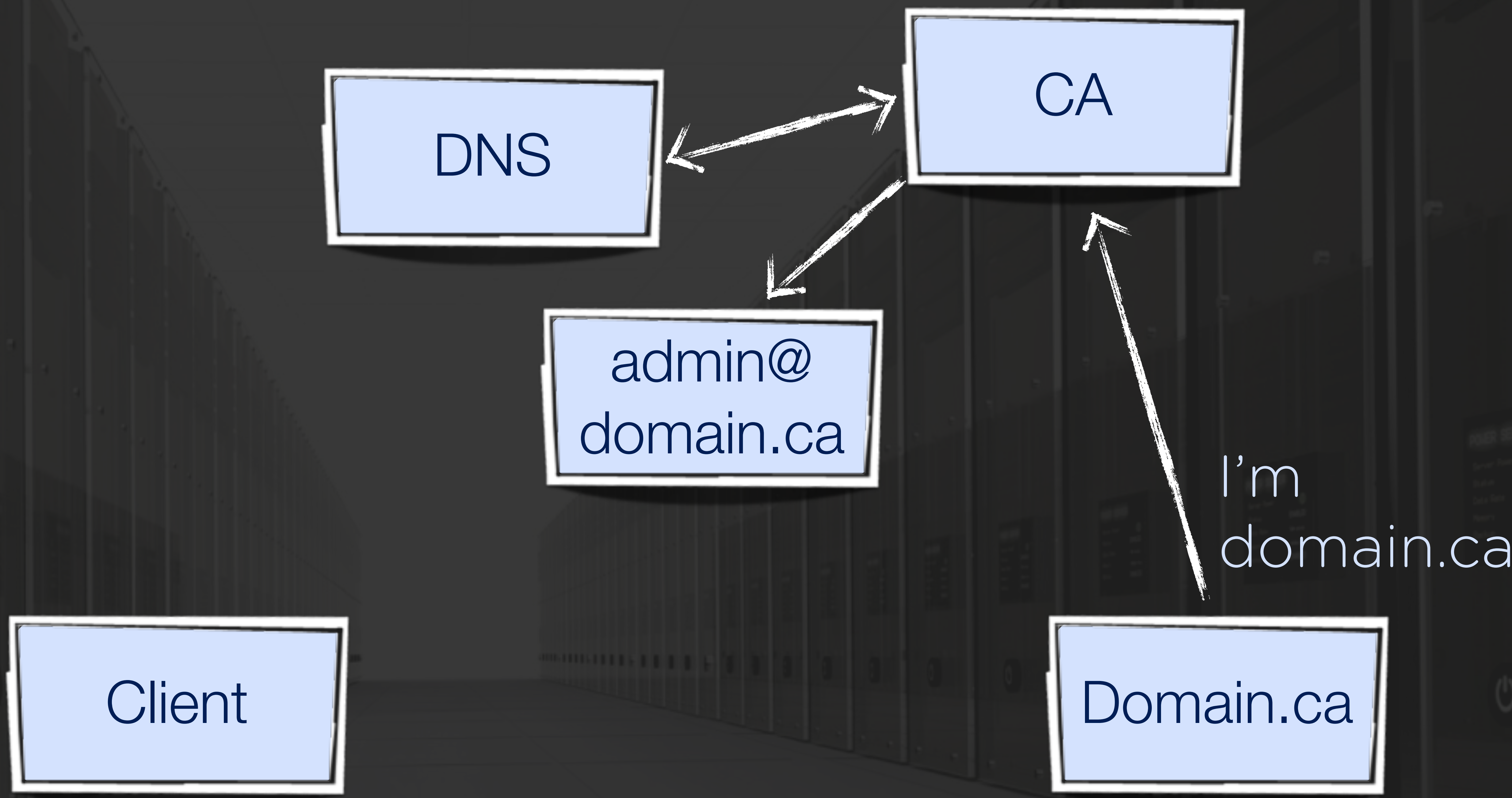
Server Authentication

$$\text{Client} \xleftarrow{\text{Sig}_{CA}(\text{Domain.ca}||\text{Key})} \text{Domain.ca}$$

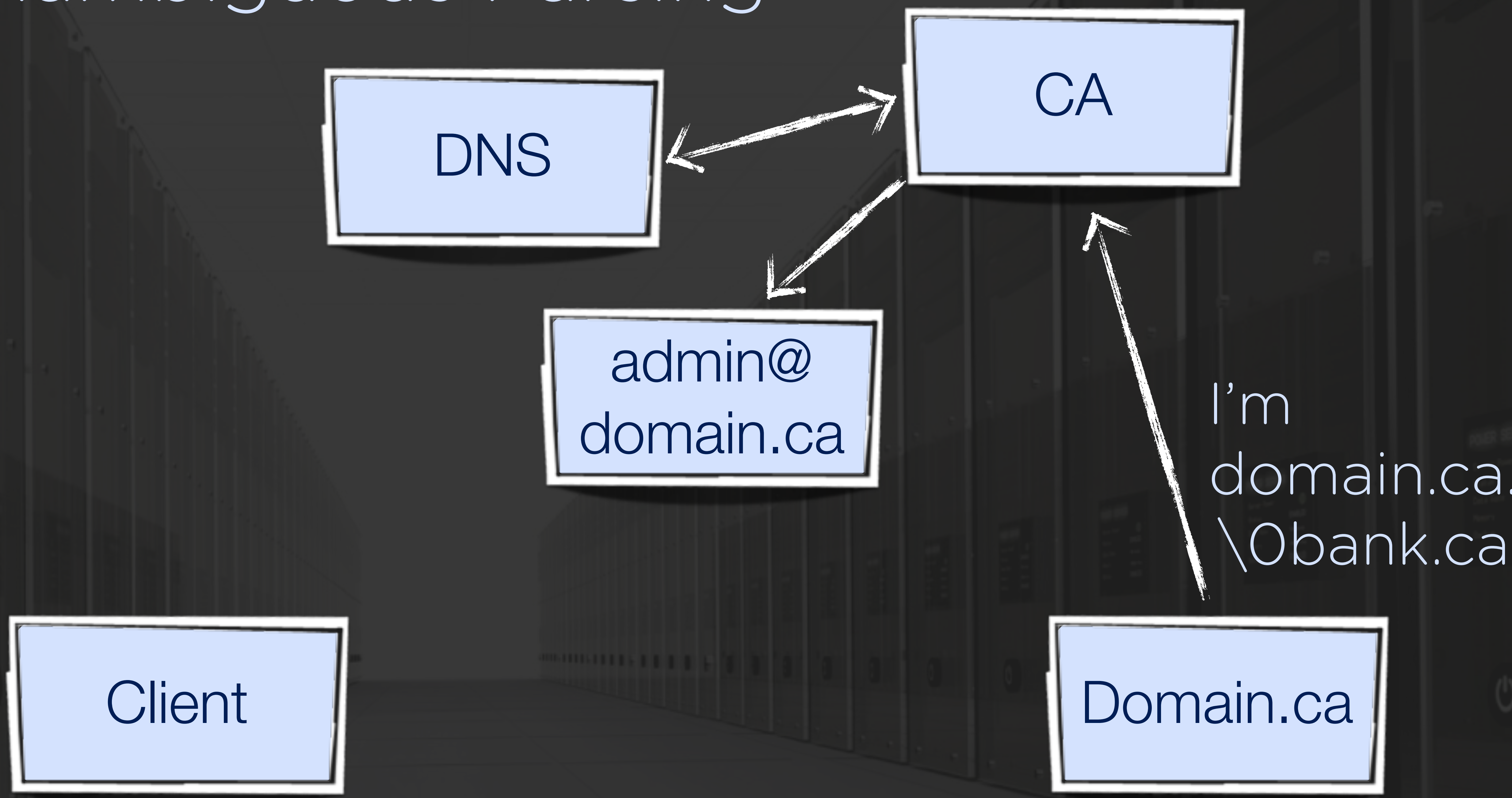

CA
I'm
domain.ca
Client
Domain.ca


CA
I'm domain.ca
Client
Domain.ca

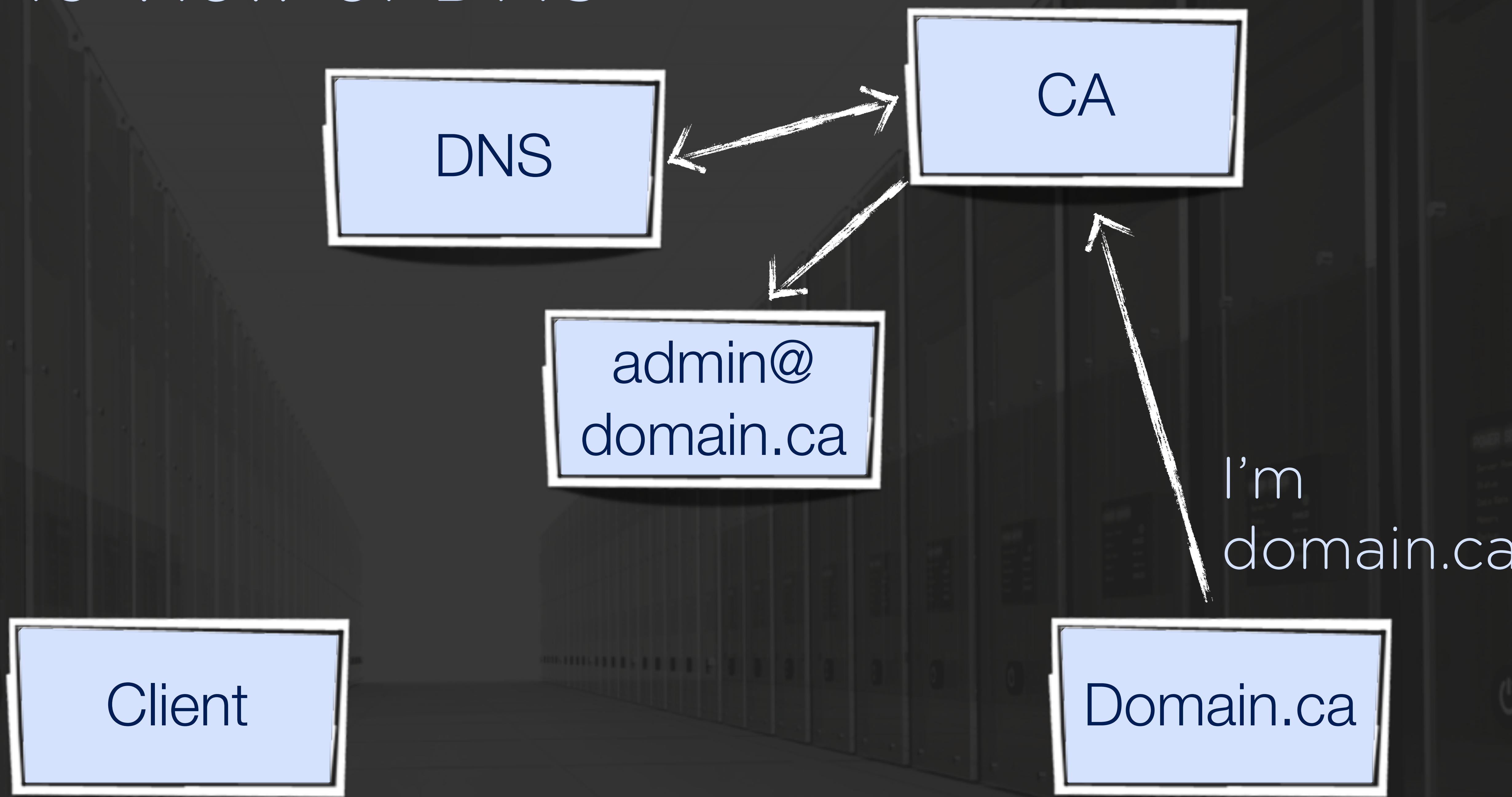

DNS
CA
Client
Domain.ca
I'm
domain.ca


DNS
CA
admin@
domain.ca
I'm
domain.ca
Client
Domain.ca

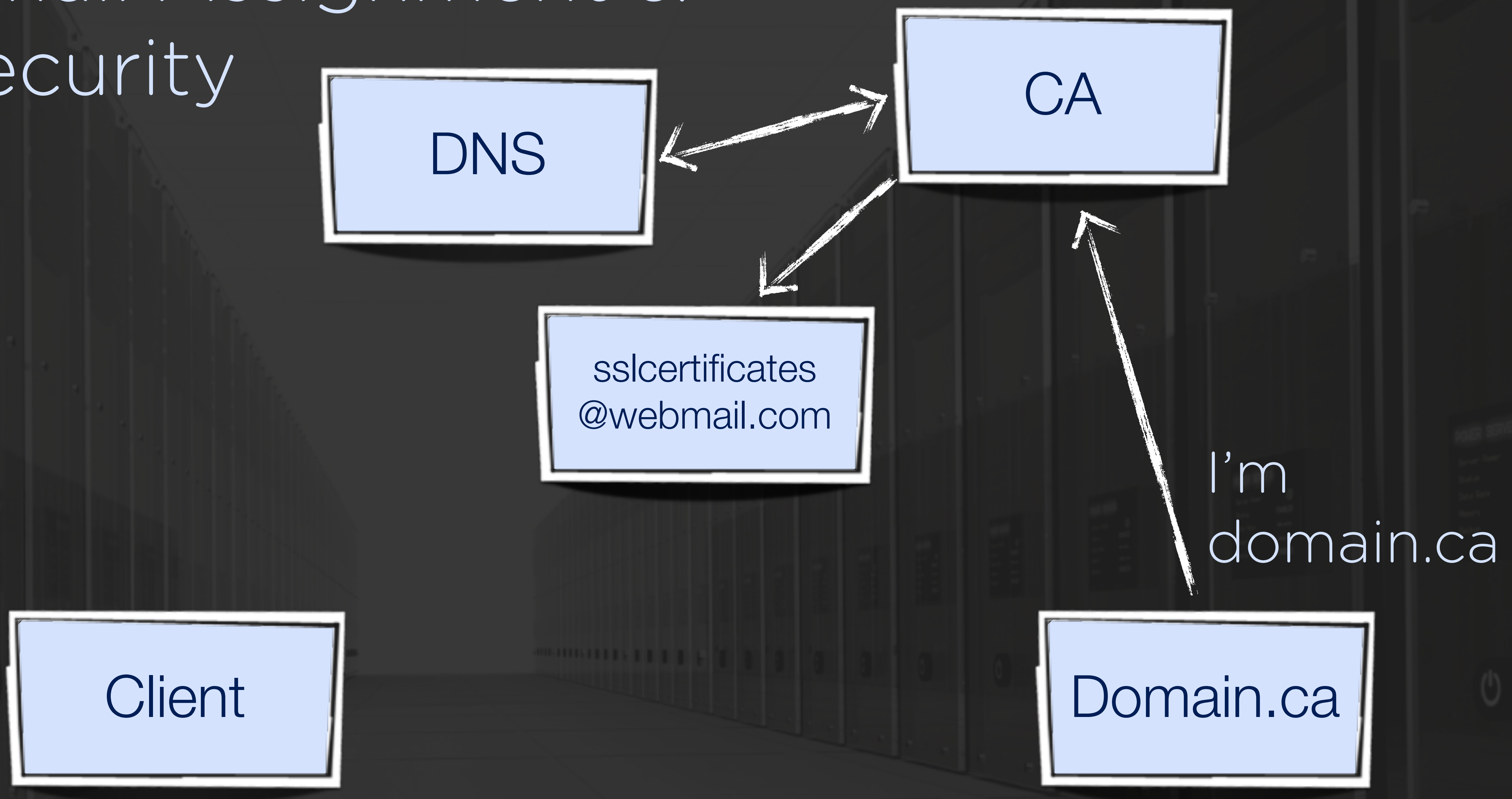
Unambiguous Parsing

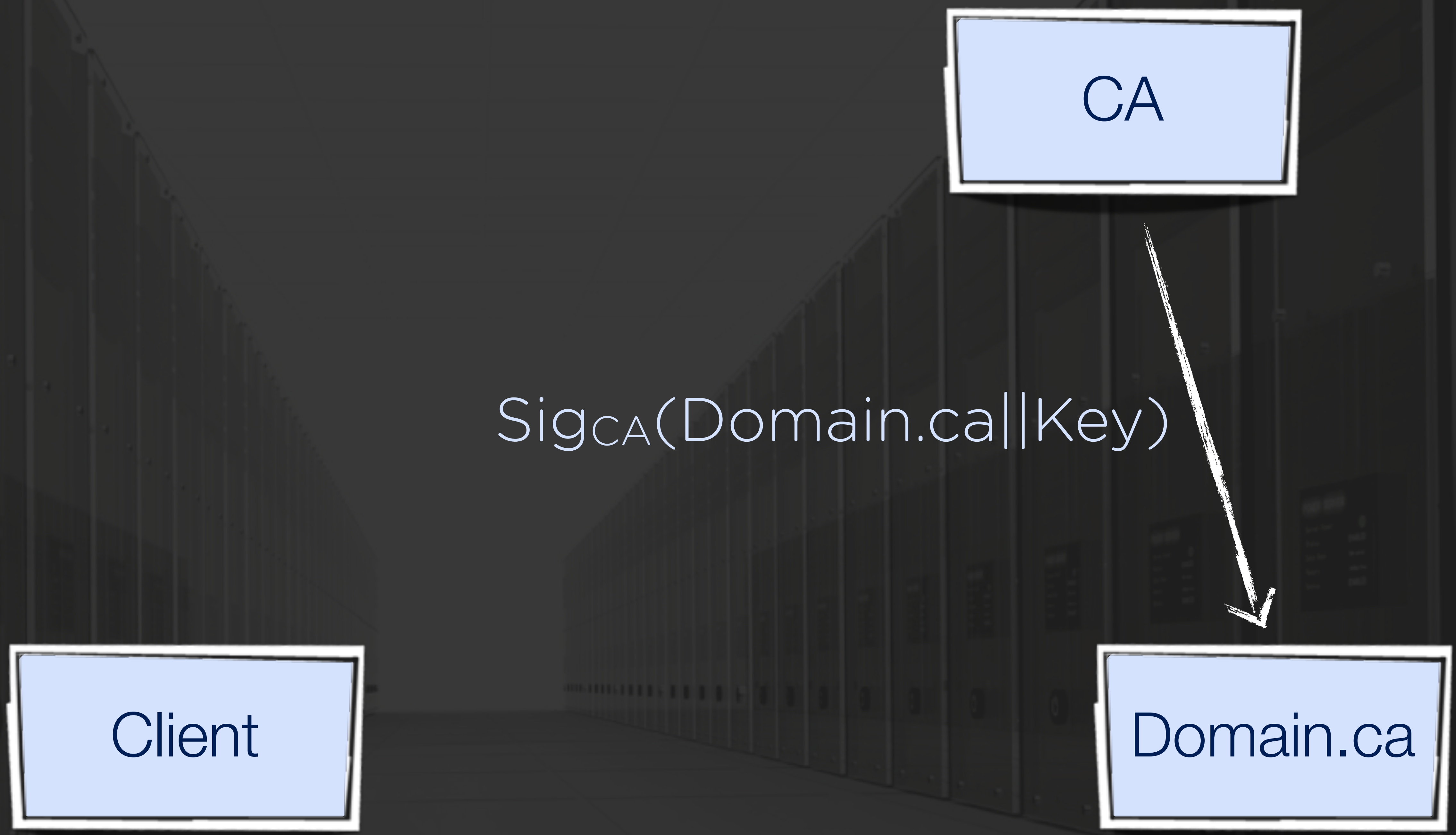


CA’s View of DNS

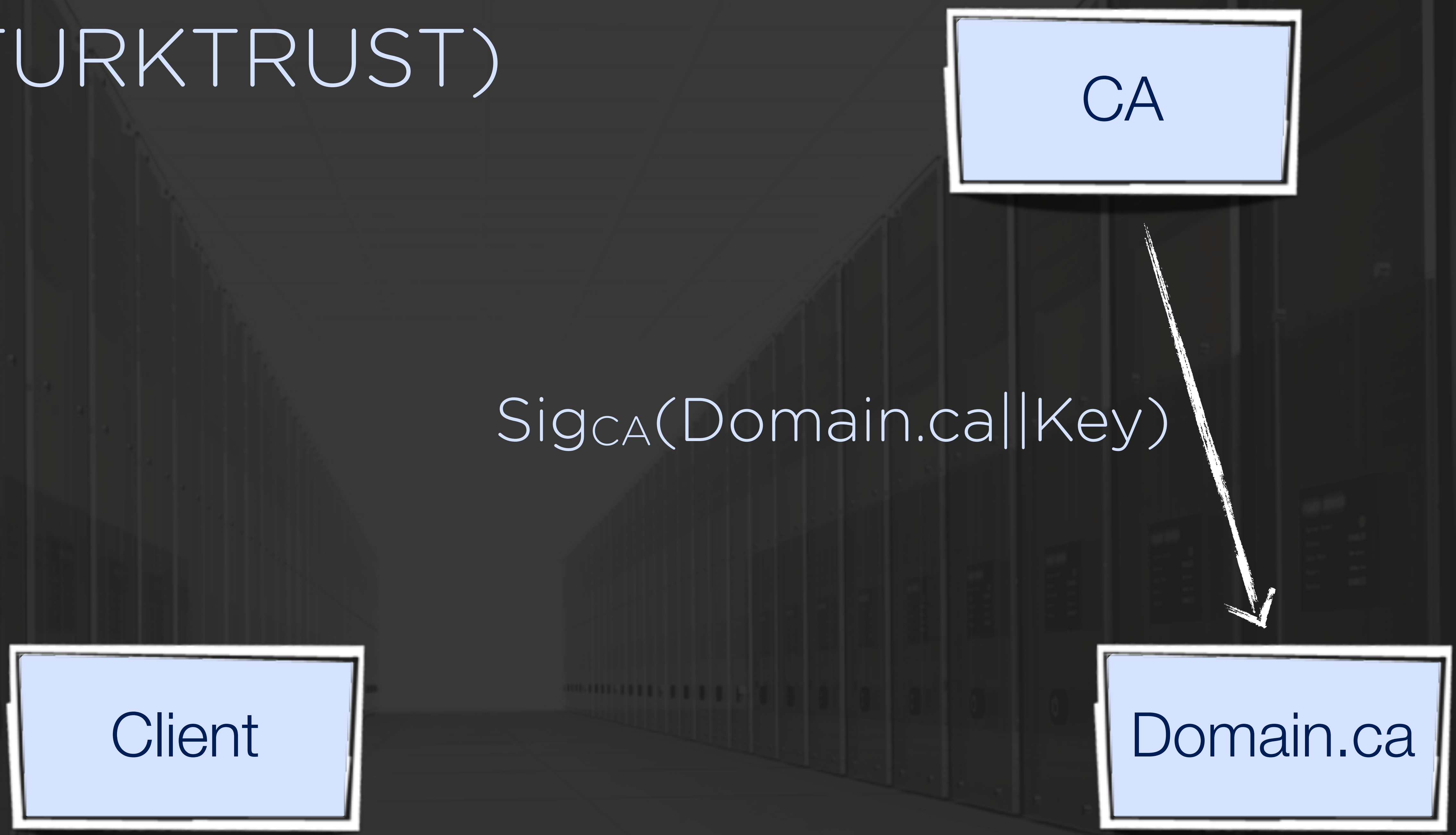


Email Assignment & Security




CA
$Sig_{CA}(Domain.ca||Key)$
Client
Domain.ca

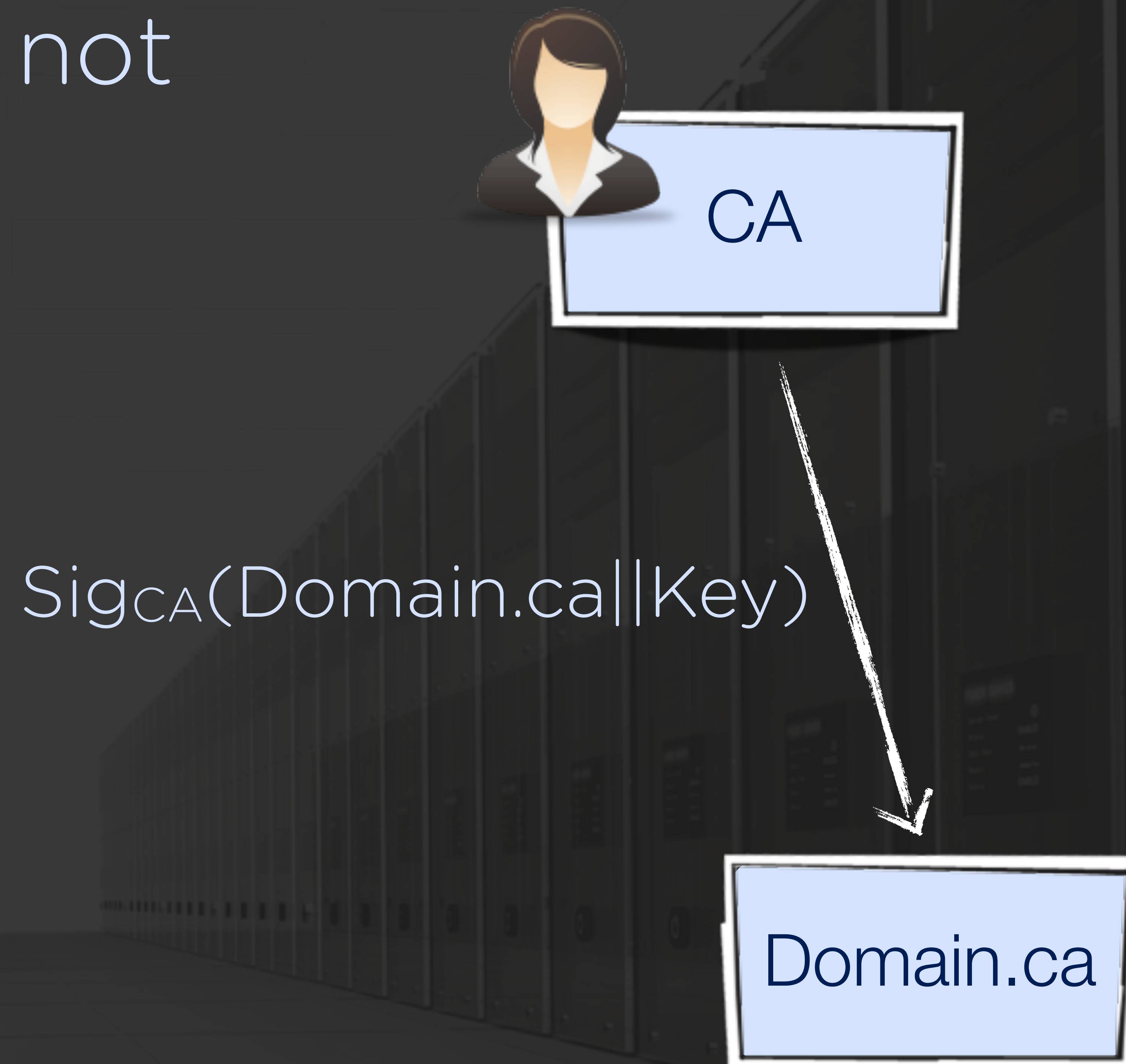
Certificate is a site cert (TURKTRUST)

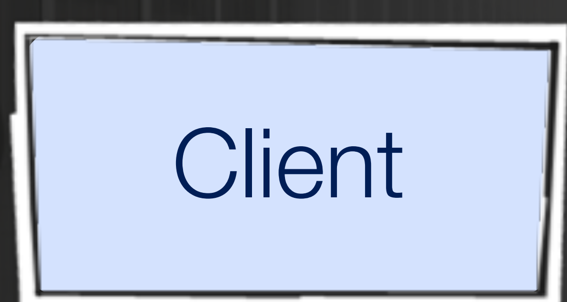


Certificate is a site cert (TURKTRUST)
& Browser checks this (IE and iOS)

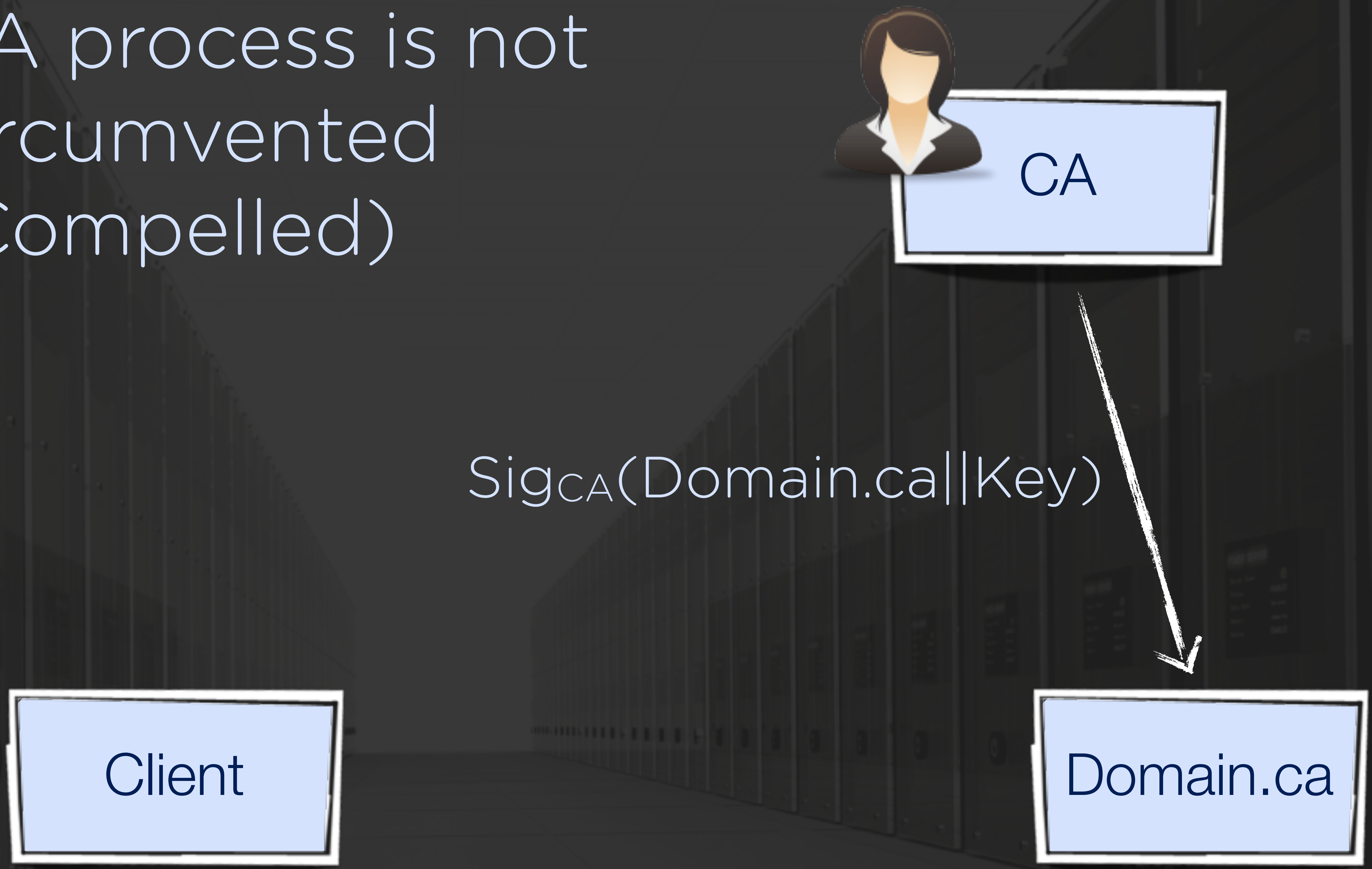
CA process is not circumvented (DigiNotar & Comodo)

CA process is not circumvented (Verisign)





CA process is not circumvented (Compelled)



Certificate Authorities

Pre-loaded into browser and/or OS

~150 root certificates from ~50 organizations

Roots certificates can authorize intermediate CAs

Hundreds of organizations have a CA cert

Certificate Authorities

Any CA can issue an acceptable certificate for any site

You Find a Bad Site Cert, Now What?

CA revokes the certificate

Revocation checking happens when receiving a certificate

Revocation checking is unreliable and fails open

Who Needs a Cert Anyways?

SSL Stripping: active adversary can strip out references to HTTPS sites and replace them with HTTP (POST-to-HTTPS)

Concede a Warning: Syria Telecom MITM on Facebook

Users tend to ignore security indicators, not understand warnings, and click through warnings they do understand

What to Do?

Detect or Prevent Fake Sites Certificate Attacks (This Talk)

Detect or Prevent SSL Stripping (See Paper)

Improve Revocation (See Paper)

Primitive	Security Properties Offered								Evaluation of Impact on HTTPS										
	A				B		C		Security & Privacy				Deployability				Usability		
	Detects MITM	Detects Local MITM	Protects Client Credential	Updatable Pins	Detects TLS Stripping	Affirms POST-to-HTTPS	Responsive Revocation	Intermediate CAs Visible	No New Trusted Entity	No New Traceability	Reduces Traceability	No New Auth'n Tokens	No Server-Side Changes	Deployable without DNSSEC	No Extra Communications	Internet Scalable	No False-Rejects	Status Signalled Completely	No New User Decisions
Key Pinning (Client History)	○	○	○						●	●		●	●	●	●	●			
Key Pinning (Server)	○	○	○						●	●				●	●	●	●		●
Key Pinning (Preloaded)	●	●	●	●					○	●		●	○	●	●		●	○	●
Key Pinning (DNS)	●	●	●	●					○	●	●		○		●	●	●	○	●
Multipath Probing		●		●								●	●	●		●		●	
Channel-bound Credentials			○						●	●		●		●	●	●	●	○	●
Credential-bound Channels			○						●	●		●		●	●	●	●	○	●
Key Agility/Manifest				●					●	●				●	●	●	●	●	●
HTTPS-only Pinning (Server)					○	○			●	●				●	●	●	●		●
HTTPS-only Pinning (Preloaded)				●	●	●			○	●		●	○	●	●		●	○	●
HTTPS-only Pinning (DNS)				●	●	●			○	●			○		●	●	●	○	●
Visual Cues for Secure POST						●			●	●		●	●	●		●		●	
Browser-stored CRL							●		○	●		●	●	●	●	●	●	●	●
Certificate Status Stapling							●		●	●	●			●	●	●	●	○	●
Short-lived Certificates							●		●	●	●	●		●	●	●	●	●	●
List of Active Certificates							●	●			●	●	●	●		●	●	●	●

Security

No New Trusted Entity
No New Auth’n Tokens

Privacy

No New Traceability
Reduces Traceability

Deployability

No Server-Side Changes
Deployable without DNSSEC
No Extra Communications
Internet Scalable

Usability

No False-Rejects
Status Signalled Completely
No New User Decisions

Primitive	*Security Properties Offered*								*Evaluation of Impact on HTTPS*										
	A				B		C		*Security & Privacy*				*Deployability*				*Usability*		
	Detects MITM	Detects Local MITM	Protects Client Credential	Updatable Pins	Detects TLS Stripping	Affirms POST-to-HTTPS	Responsive Revocation	Intermediate CAs Visible	No New Trusted Entity	No New Traceability	Reduces Traceability	No New Auth'n Tokens	No Server-Side Changes	Deployable without DNSSEC	No Extra Communications	Internet Scalable	No False-Rejects	Status Signalled Completely	No New User Decisions
Key Pinning (Client History)	○	○	○						●	●		●	●	●	●	●			
Key Pinning (Server)	○	○	○						●	●				●	●	●	●		●
Key Pinning (Preloaded)	●	●	●	●					○	●		●	○	●	●		●	○	●
Key Pinning (DNS)	●	●	●	●					○	●	●		○		●	●	●	○	●
Multipath Probing		●		●								●	●	●		●		●	
Channel-bound Credentials			○						●	●		●		●	●	●	●	○	●
Credential-bound Channels			○						●	●		●		●	●	●	●	○	●
Key Agility/Manifest				●					●	●				●	●	●	●	●	●
HTTPS-only Pinning (Server)					○	○			●	●				●	●	●	●		●
HTTPS-only Pinning (Preloaded)				●	●	●			○	●		●	○	●	●		●	○	●
HTTPS-only Pinning (DNS)				●	●	●			○	●			○		●	●	●	○	●
Visual Cues for Secure POST						●			●	●		●	●	●		●		●	
Browser-stored CRL							●		○	●		●	●	●	●	●	●	●	●
Certificate Status Stapling							●		●	●	●			●	●	●	●	○	●
Short-lived Certificates							●		●	●	●	●		●	●	●	●	●	●
List of Active Certificates							●	●			●	●	●	●		●	●	●	●

Pinning — Server Initiated

Send (via HTTP header or TLS handshake) the attributes about your certificate chain you want pinned.

Trust-on-first-use
Server-side changes
Self denial-of-service
No new authority

C. Evans, C. Palmer, & R. Sleevi
HPKP
Public key pinning extension for HTTP. *Web Security Working Group*. Internet-Draft. Intended Status: Standards Track. December 7, 2012
2012 Google

M. Marlinspike & T. Perrin
TACK
Trust assertions for certificate keys (TACK). *TLS Working Group*. Internet Draft. Intended status: Standards Track. January 7, 2013
2013

Pinning — Browser Preloads

Certificate attributes are pinned in a preloaded list, maintained by the browser vendor.

Resolves trust-on-first-use
Minimal server participation
Not scalable to millions of servers
Increases trust in your browser

Pinning — DNS

Certificate attributes are pinned in a DNS record for your domain and distributed with DNSSEC

Setting record scales to the internet
Distributing records: DNSSEC scalability debatable
Records could be stapled into TLS connection
Increased trust in DNS system
Could be used with self-issued certificates

P. Hoffman & J. Schlyter

DANE — TLSA

The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA. *Standards Track*. 2012.

RFC 6698

Notary — Multipath Probing

Third party notaries relay information about the certificate they see for a domain.

No server-side changes
Performance penalty and needs high reliability
A domain may have multiple certs (load-balancing)
Privacy issues
Trust agility: a pro or a con?

D. Wendlandt, D. G. Andersen, and A. Perrig
Perspectives
Perspectives: Improving SSH-style host authentication with multi-path probing. *USENIX Annual Tech*
2008

Moxy Marlinspike
Convergence
Convergence, Beta. SSL And The Future Of Authenticity. *BlackHat USA 2011.* convergence.io
2011

Notary — Log

Certificate authorities publish server certificates in an append-only log. Sites monitor the log for fraudulent certificates and report them for revocation

Detection instead of prevention
Increases visibility
Notary similarities: performance, tracing, etc.
Differences: one authority, sites can staple logs
Full CA opt-in
Relies on revocation

B. Laurie, A. Langley, & E. Kasper
Certificate Transparency
Certificate Transparency. *Network Working Group*. Internet-Draft. Intended Status: Experimental. April 18, 2013
2013 Google

Conclusions

The breadth of past and on-going issues with TLS is noteworthy

Sophistication of attacking the TLS protocol seems to have shifted interest to its trust infrastructure, which has on-going issues

No clear winner among enhancements: trade-offs


Questions?
clark@scs.carleton.ca
paulv@scs.carleton.ca
@PulpSpy

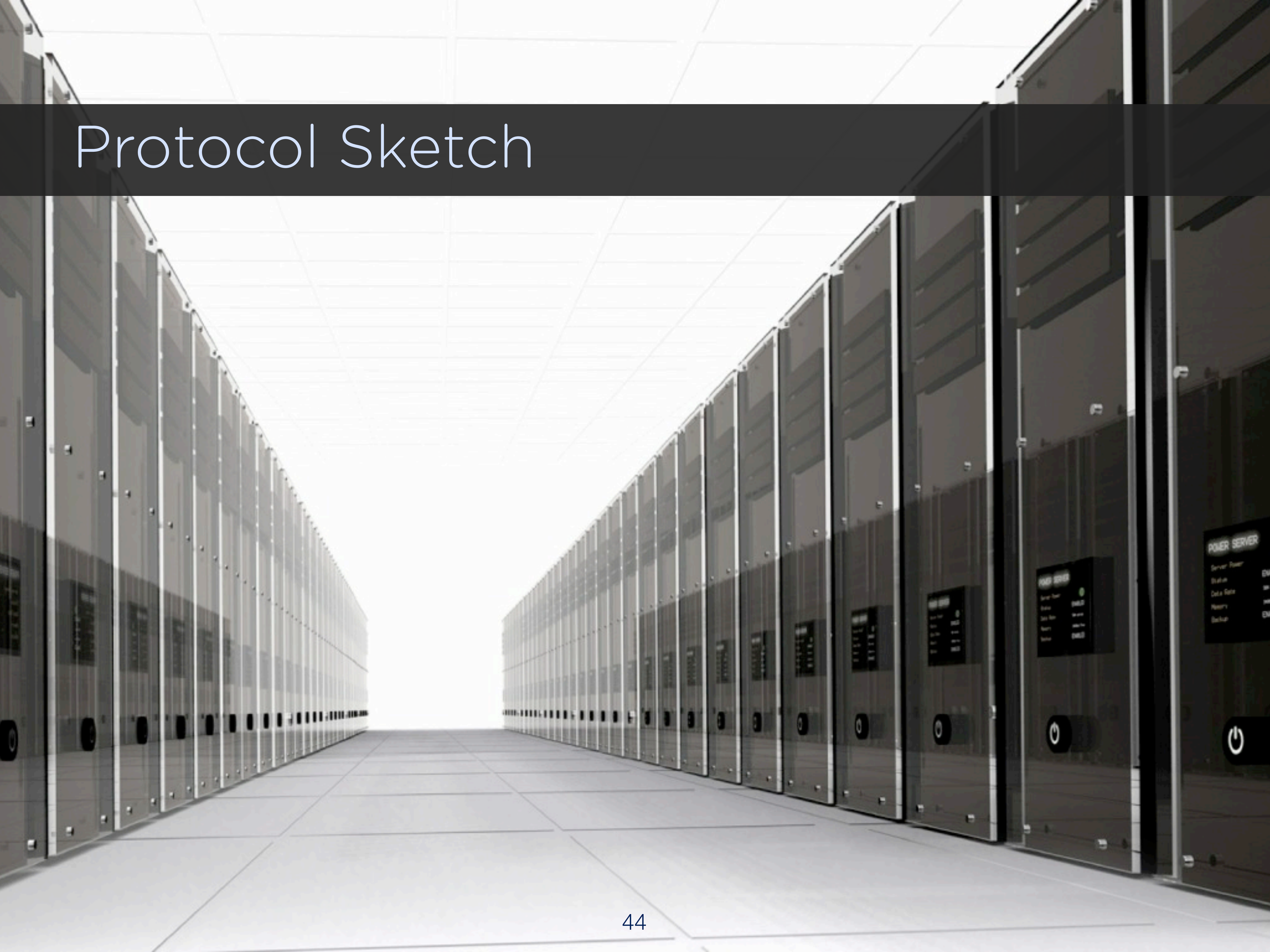
Extra Slides

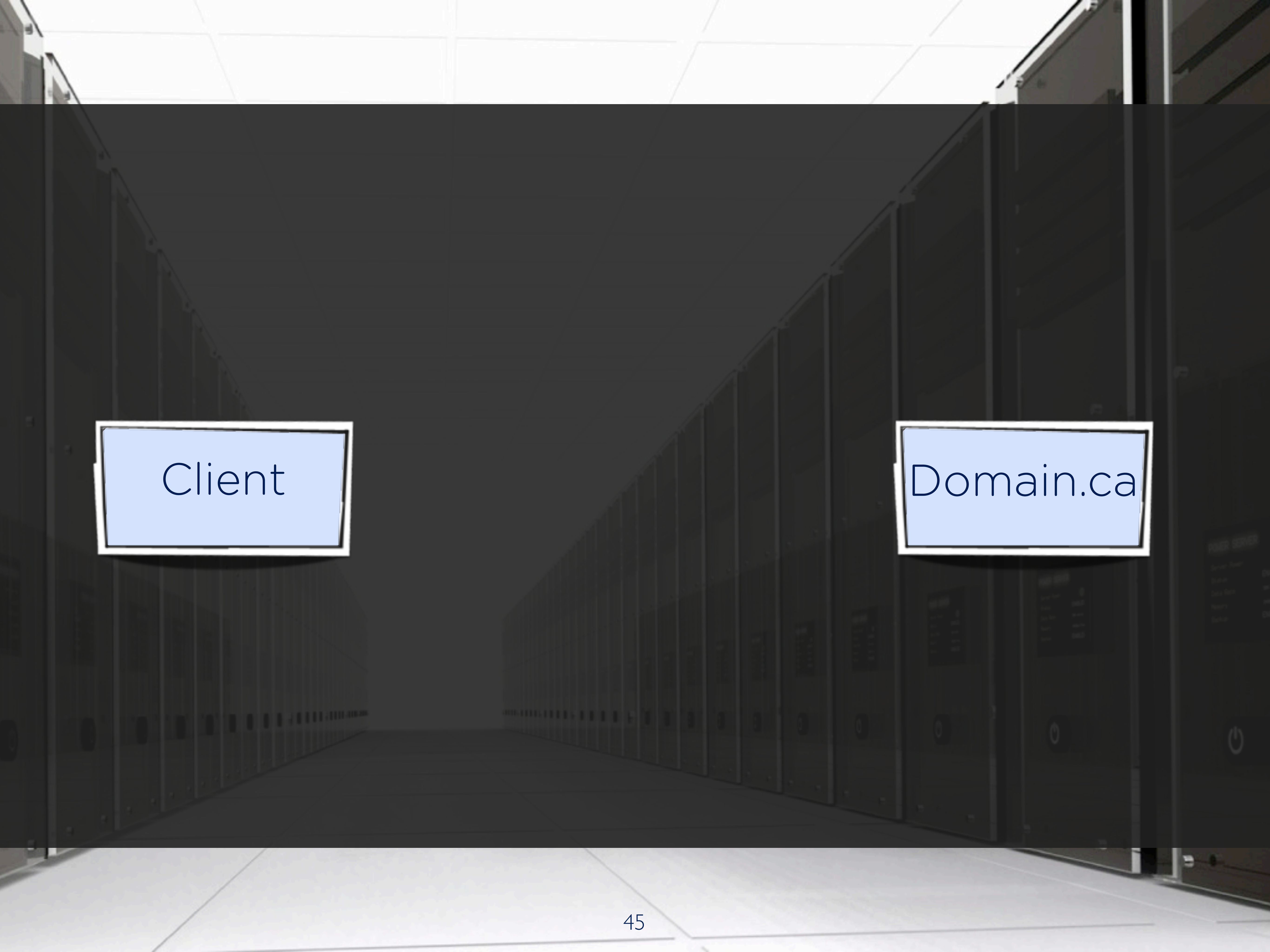
Will Your Anchor Hold?

Pre-loading the browser with pins (and HTTPS-only status and revocation information) will work in the short-term

In the long-term, DNS-pinning (e.g., DANE) is promising if DNSSEC is plausible, and Certificate Transparency is complimentary offering increased visibility

Protocol Sketch




Client
Domain.ca

Domain.ca's
Public Key

Client

Domain.ca

1) Client lists supported versions & ciphersuites
2) Server selects
3) Server sends public key

4a) Client chooses secret value and sends to server, encrypted with server's public key

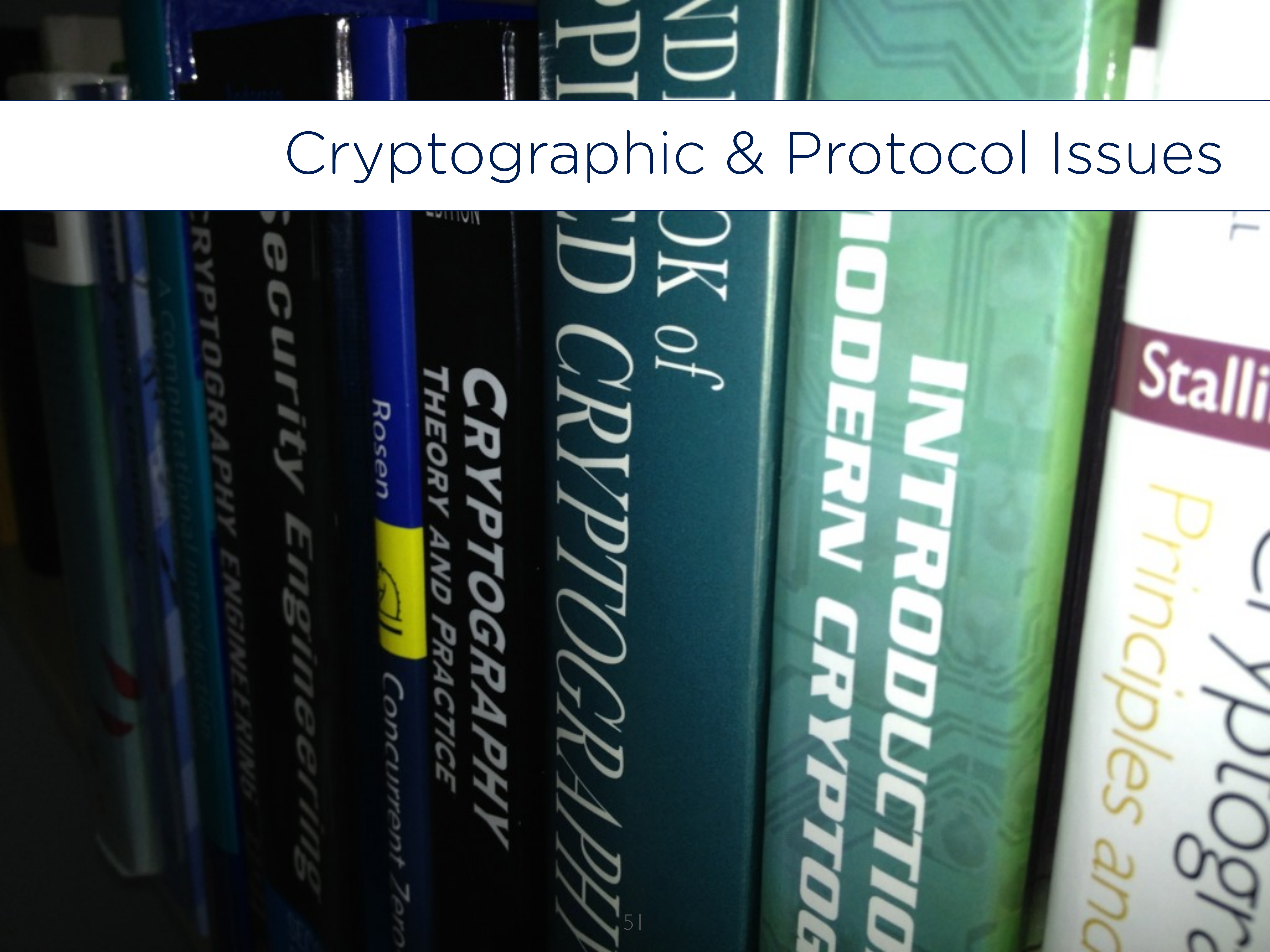
4b) Client and server use Diffie-Hellman to derive secret, and server signs values with its public key

5) Shared secret is extracted/expanded into encryption and MAC keys

6) Client MACs previous messages

7) Data is put into records, MACed, padded (if apl), and encrypted

Cryptographic & Protocol Issues



Cryptographic & Protocol Issues

Aging Primitives:
  MD2, MD5, RC4, weak keys (<112 bits equiv. sec.)

Implementation Flaws:
  Bad randomness: Netscape, Debian, embedded
  Timing Attacks: RSA encryption, ECDSA

Protocol Flaws:
  Renegotiation, version & ciphersuite downgrades

Cryptographic & Protocol Issues

An active adversary can use the server as a decryption oracle (adaptive CCA attacks):

1) RSA PKCS#1 v1.5 key transport:
   distinguish bad encoding from failed decryption

2) CBC mode data transport:
   distinguish bad padding from MAC failure
   MAC -> Pad -> Encrypt

Cryptographic & Protocol Issues

Malicious client-side code can use the client as an encryption oracle (adaptive CPA attacks):

1) CBC mode data transport:
   Initialization vectors are predictable

2) Block or stream cipher data transport:
   Compression is applied prior to encryption
   Length leaks semantic information