Eroding Trust & The CA Debacle

Jeremy Clark

Jeremy Clark
Concordia University

SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. *IEEE Symposium on Security and Privacy*

2013

Mirror City

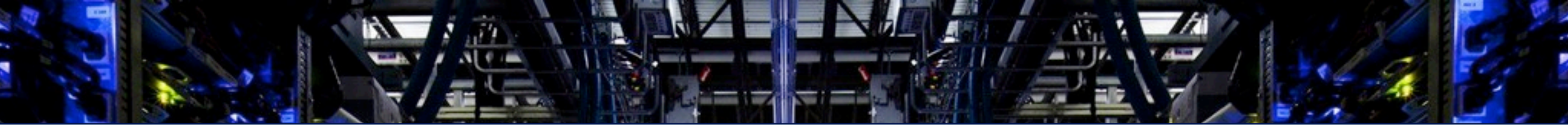
Shainblum Photo.com

Certificates for HTTPS

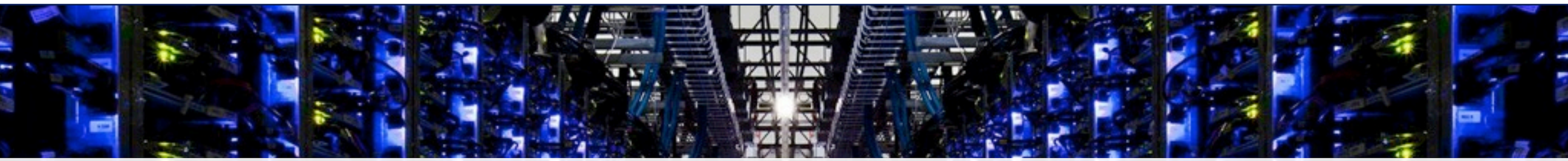
HTTPS (HTTP over SSL/TLS) design:

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

server is authenticated by a CA-issued & browser accepted certificate



Certificates for HTTPS

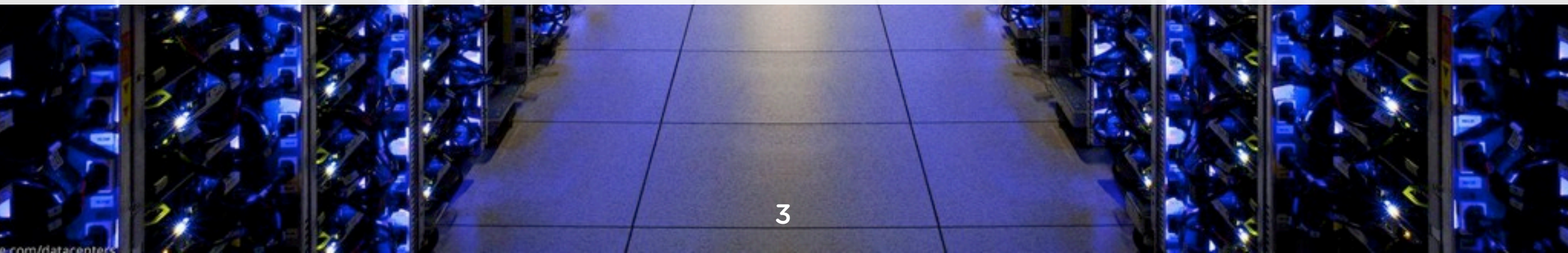


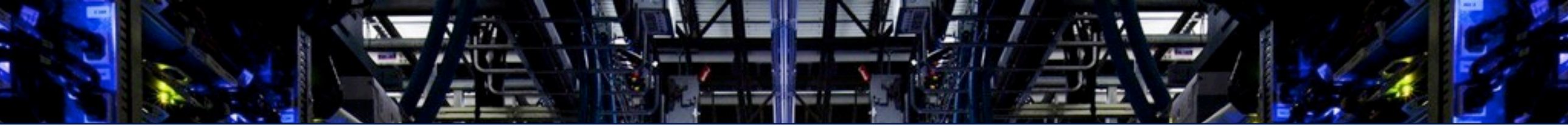
The essential problem:

CA-issued is no longer a high enough standard

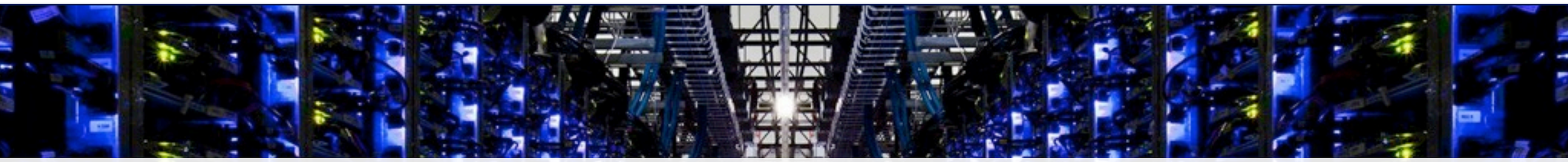
increase in CAs, increase in (known) breaches, decrease in baseline validation, lack of revocation

+ TLS protocol issues



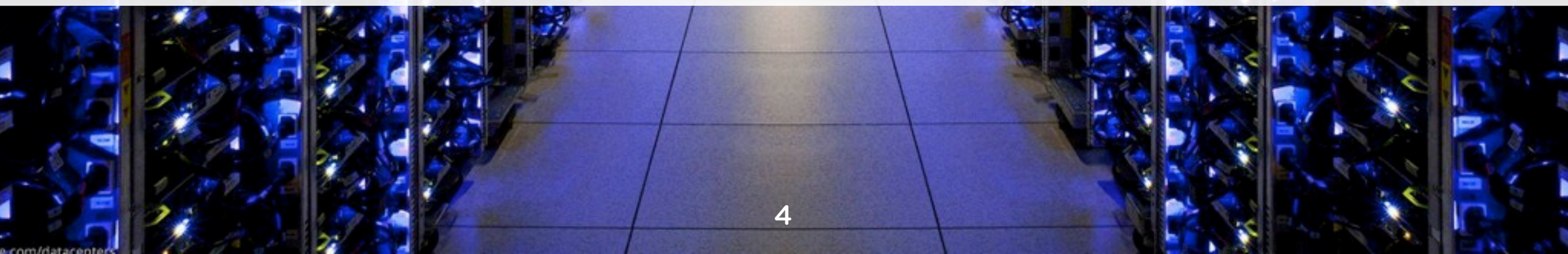


Agenda



Me: Primer on issues (~15 min)
You: Proposed Solutions (open for pitches)
Me: Sweep up of Solutions not Covered (~10 min)
You: General Discussion

Please interrupt and inject comments at any point



Prevent Fraudulent Certs:

Browser Preloads
CAge
CertLock
Certification Patrol
Convergence
DANE
Doublecheck
HPKP
MonkeySphere
Perspectives
Sovereign Keys
TACK

Detect Fraudulent Certs:

CAA
Certificate Transparency
TKI

Protect Login:

Channel ID (nee Origin Bound Certs)
DVCert

Secure Introduction:

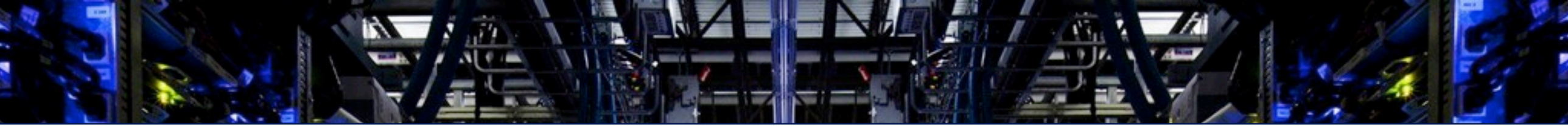
S-Links
YURLS

Prevent HTTP Downgrade:

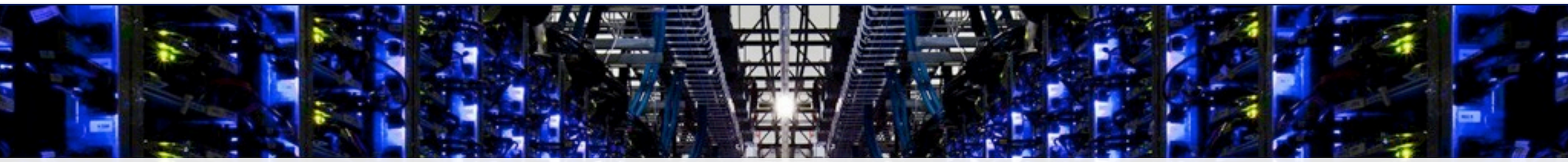
Browser Preloads
HSTS
SSLight

Improve Revocation:

Browser CRLs
OCSP Stapling
Short-lived Certificates

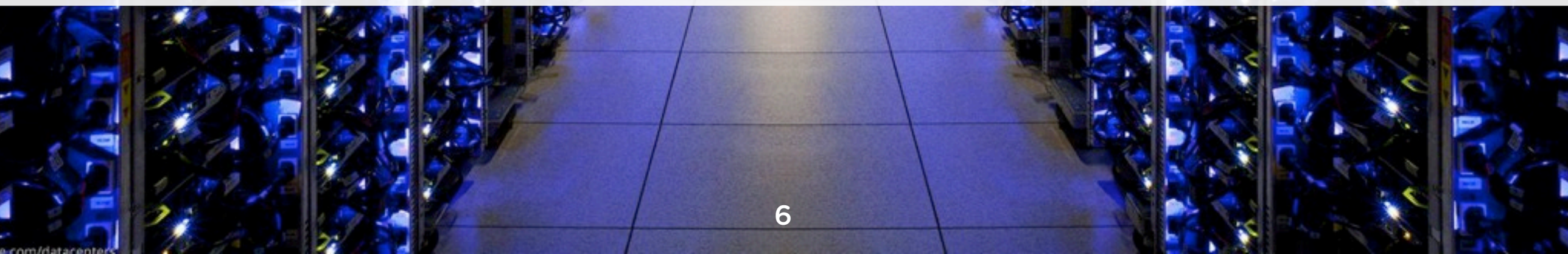


Agenda



Me: Primer on issues (~10 min)
You: Proposed Solutions (open for pitches)
Me: Sweep up of Solutions not Covered (~10 min)
You: General Discussion

Please interrupt and inject comments at any point



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, truncation, 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

Cryptographic & Protocol Issues

Version Downgrade Attacks:

TLS 1.0:
  RC4 (insecure), CBC (insecure)

TLS 1.2 [0.02%]:
  RC4 (insecure), CBC (secure?), GCM (secure?)

Cryptographic & Protocol Issues

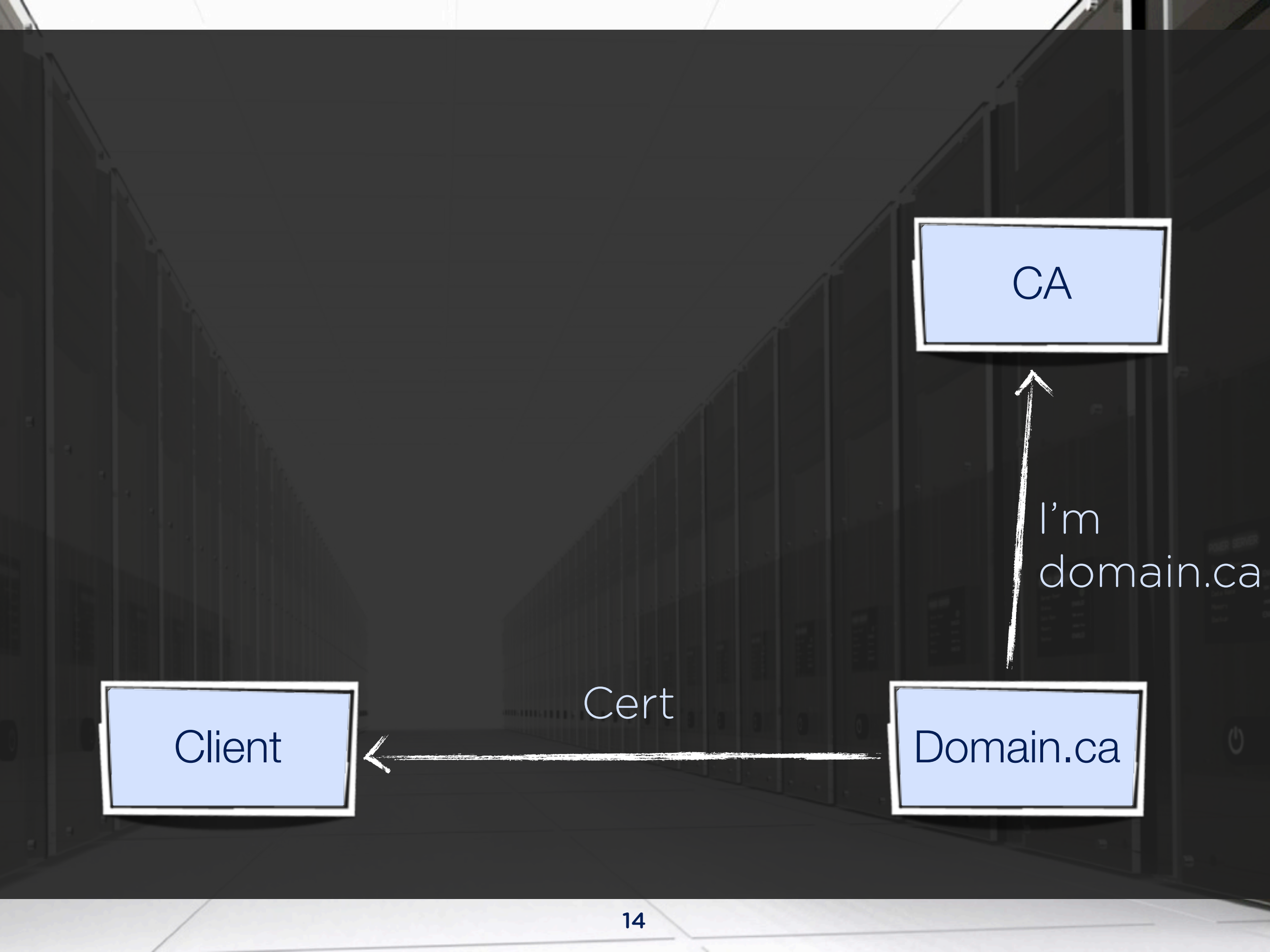
Version Downgrade Attacks:

TLS 1.0:
RC4 (insecure), CBC (insecure)

TLS 1.2 [0.02%]:
RC4 (insecure), CBC (secure?), GCM (secure?)

How to encourage upgrades?


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


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

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

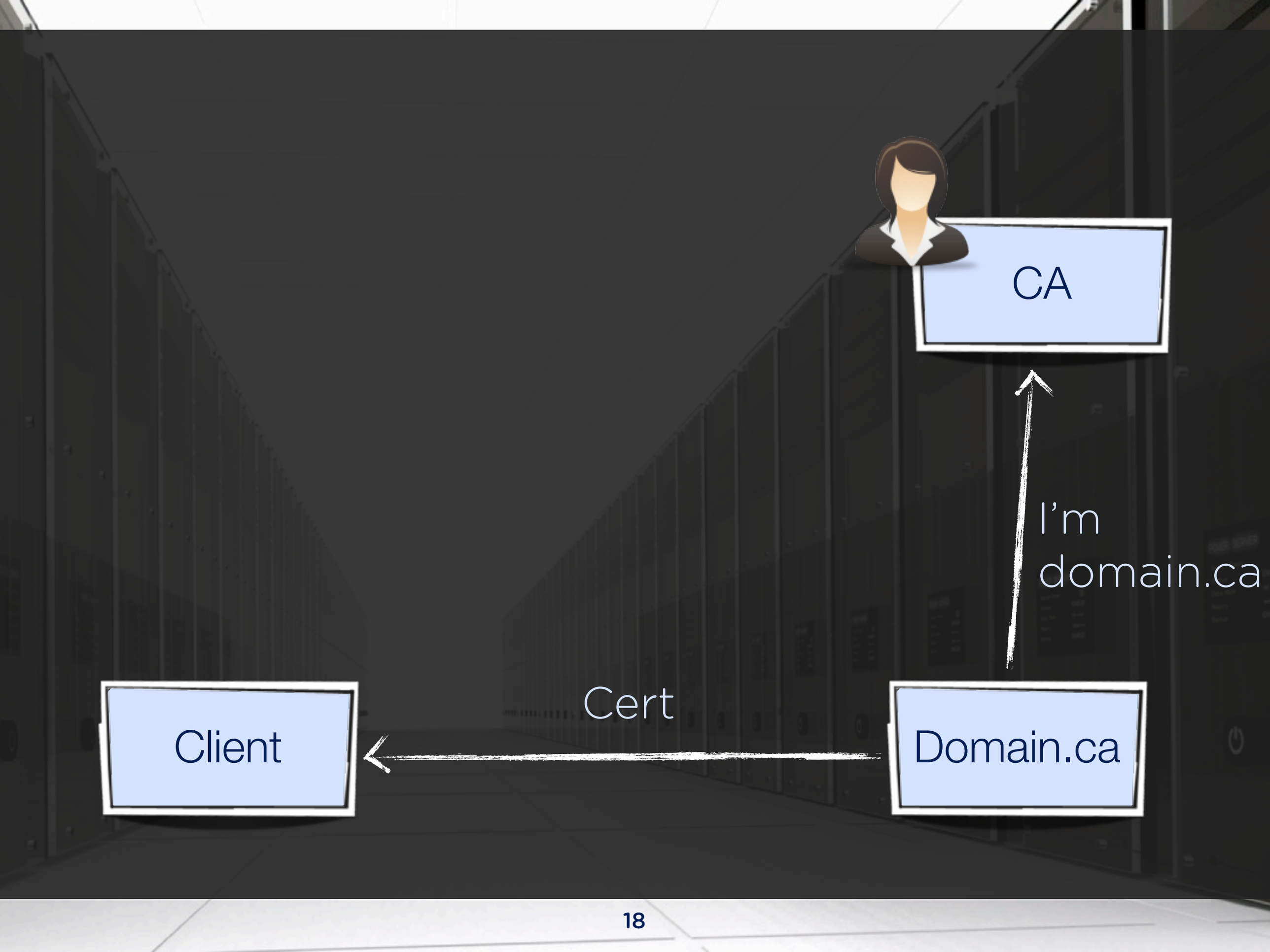
Any CA can issue an acceptable certificate for any site

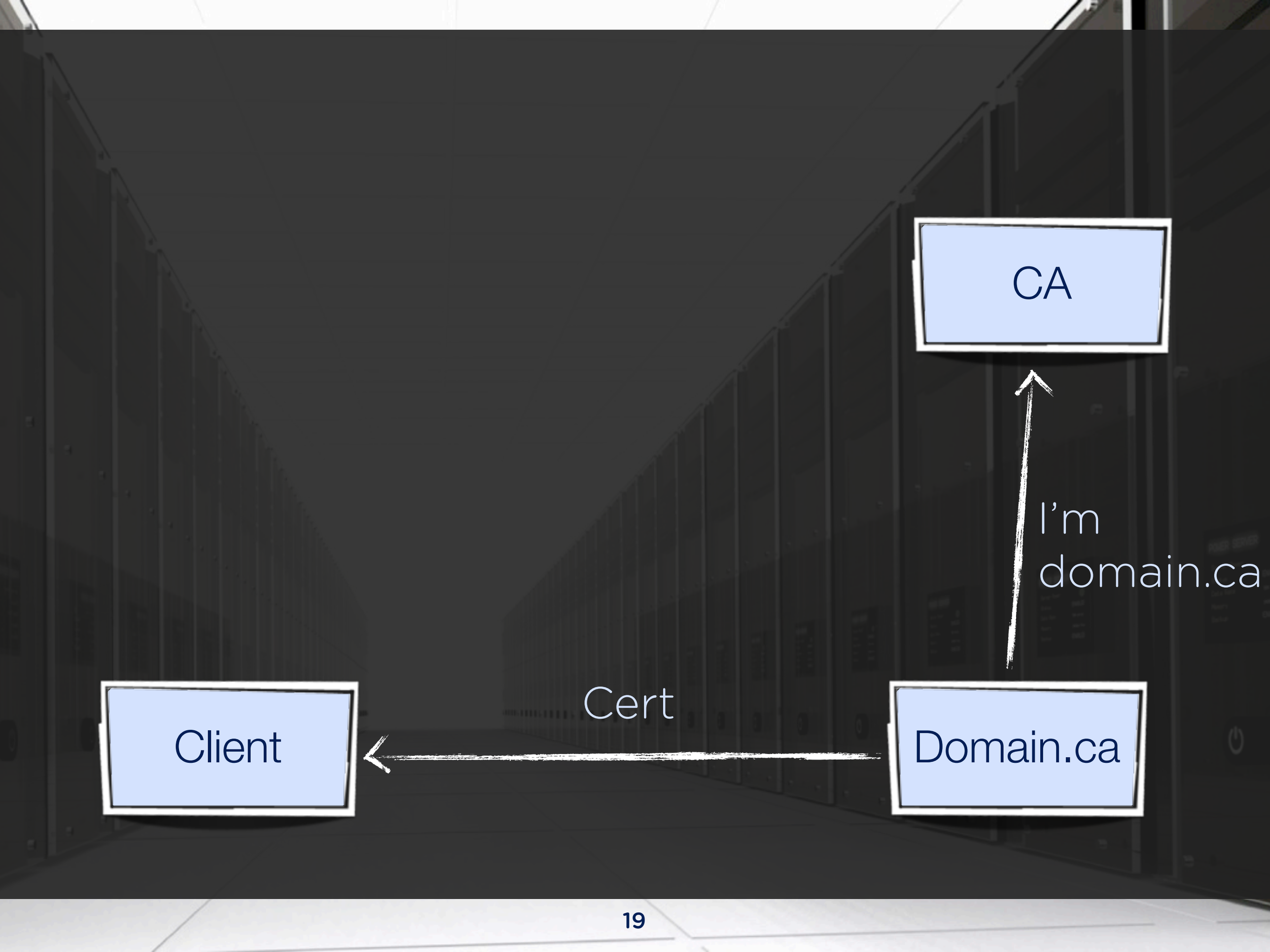
Certificate Authorities

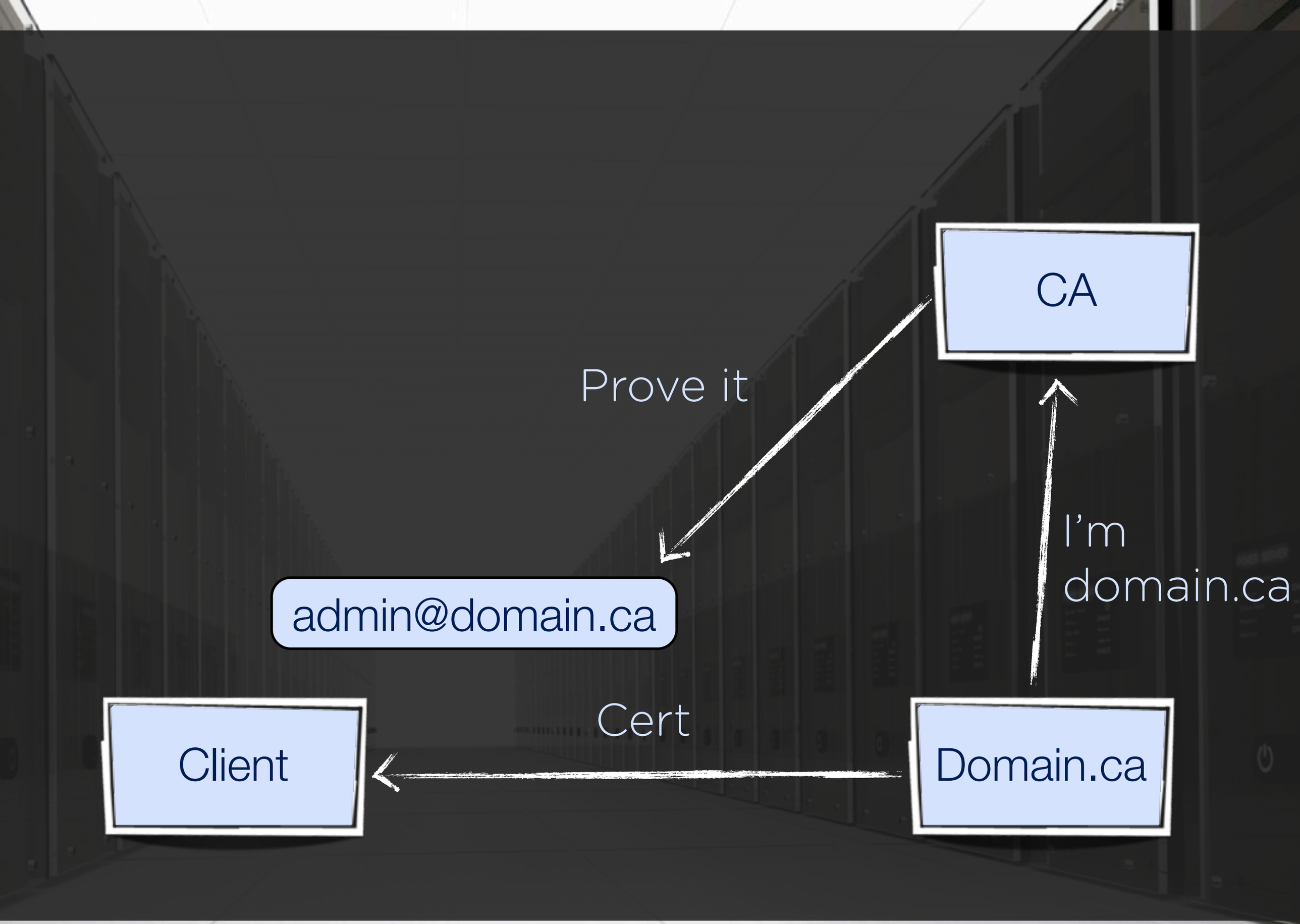
Any CA can issue an acceptable certificate for any site

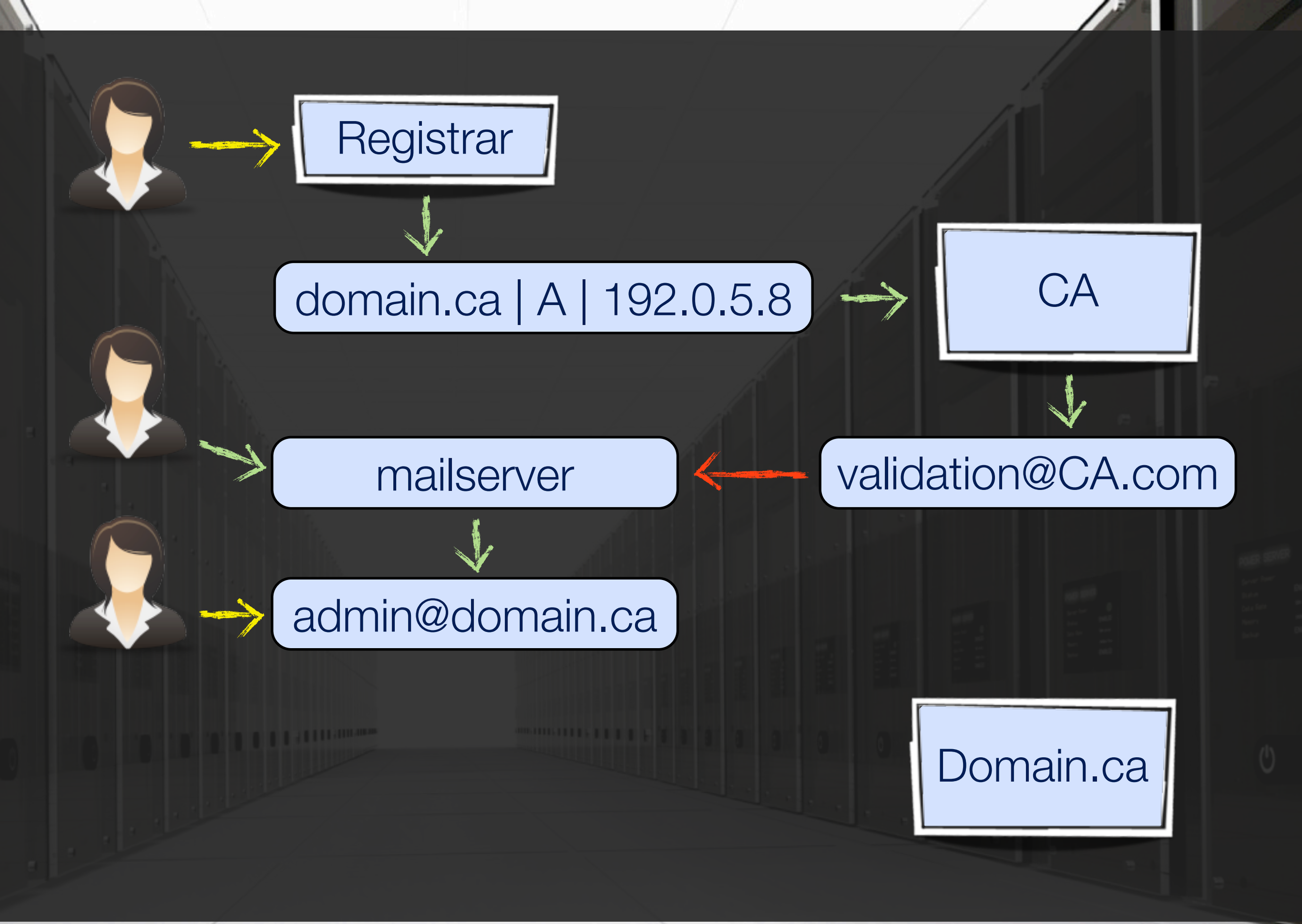
Reasonable to trust 1M sites automagically?

Should we have name constraints?

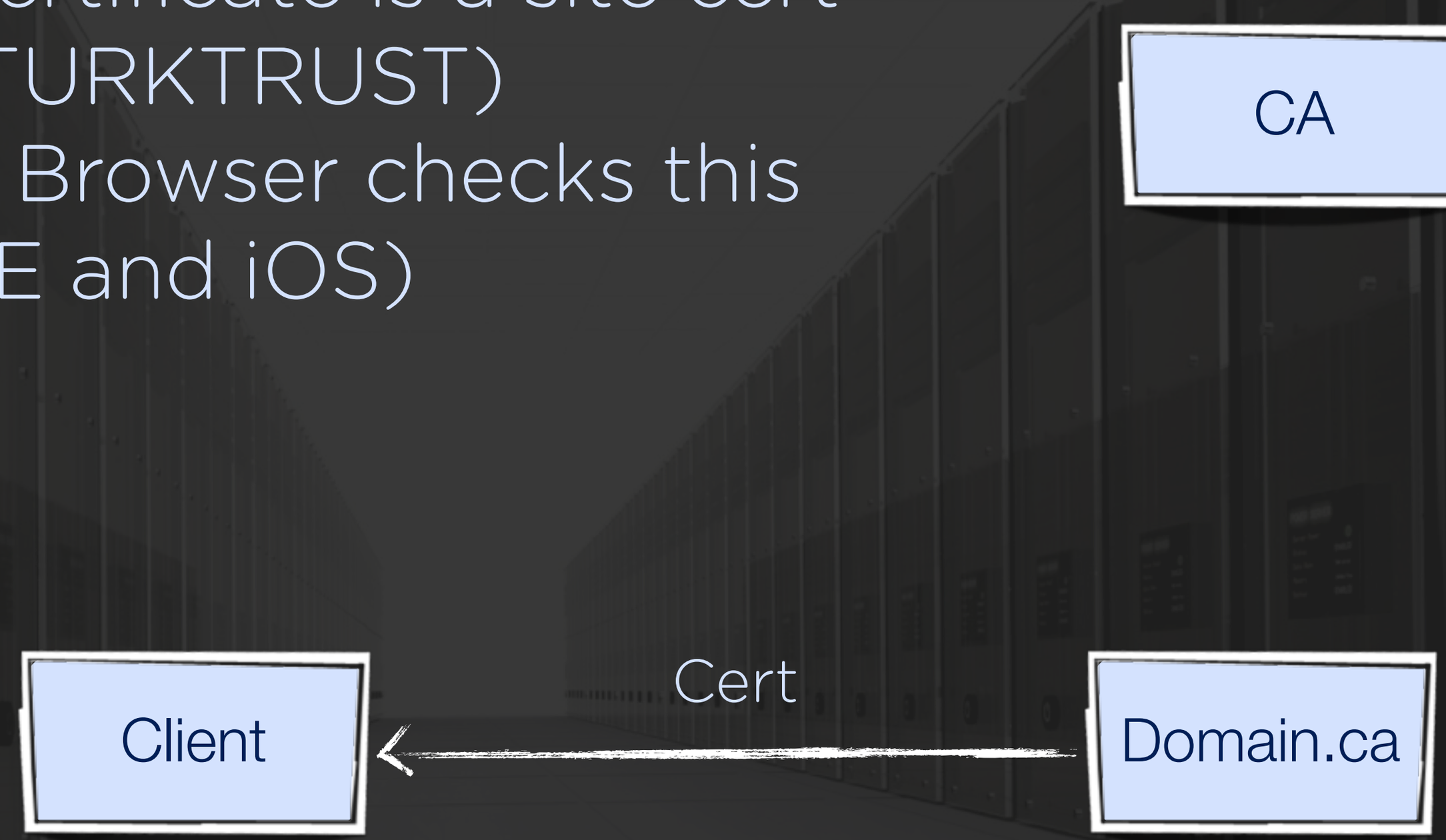

CA
I'm
domain.ca
Client
Cert
Domain.ca


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


CA
Prove it
I'm
domain.ca
admin@domain.ca
Client
Cert
Domain.ca


Registrar
domain.ca | A | 192.0.5.8
CA
mailserver
validation@CA.com
admin@domain.ca
Domain.ca

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



CA process is not circumvented (DigiNotar & Comodo) (OV: Verisign)

CA process is not circumvented (Compelled)

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?

Prevent Fraudulent Certs:

Browser Preloads
CAge
CertLock
Certification Patrol
Convergence
DANE
Doublecheck
HPKP
MonkeySphere
Perspectives
S-Links
Sovereign Keys
TACK
TKI
YURLS

Detect Fraudulent Certs:

CAA
Certificate Transparency

Protect Login:

Channel ID (nee Origin Bound Certs)
DVCert

Prevent HTTP Downgrade:

Browser Preloads
HSTS
SSLight

Improve Revocation:

Browser CRLs
OCSP Stapling
Short-lived 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

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?

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

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							●	●		●	●	●	●		●	●	●	●	●

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

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

No Server Side Changes

CT (Lookup)
Convergence
OCSP

CT (Stapled)
Certificate Patrol
S-Links

Preloads

Extra Communication

DANE (Lookup)

No Extra Communication

DANE (Stapled)
HSTS/HPKP/TACK

OCSP Stapling
Short-Lived Certs

Server Side Changes

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

Discussion

clark@ciise.concordia.ca
@PulpSpy