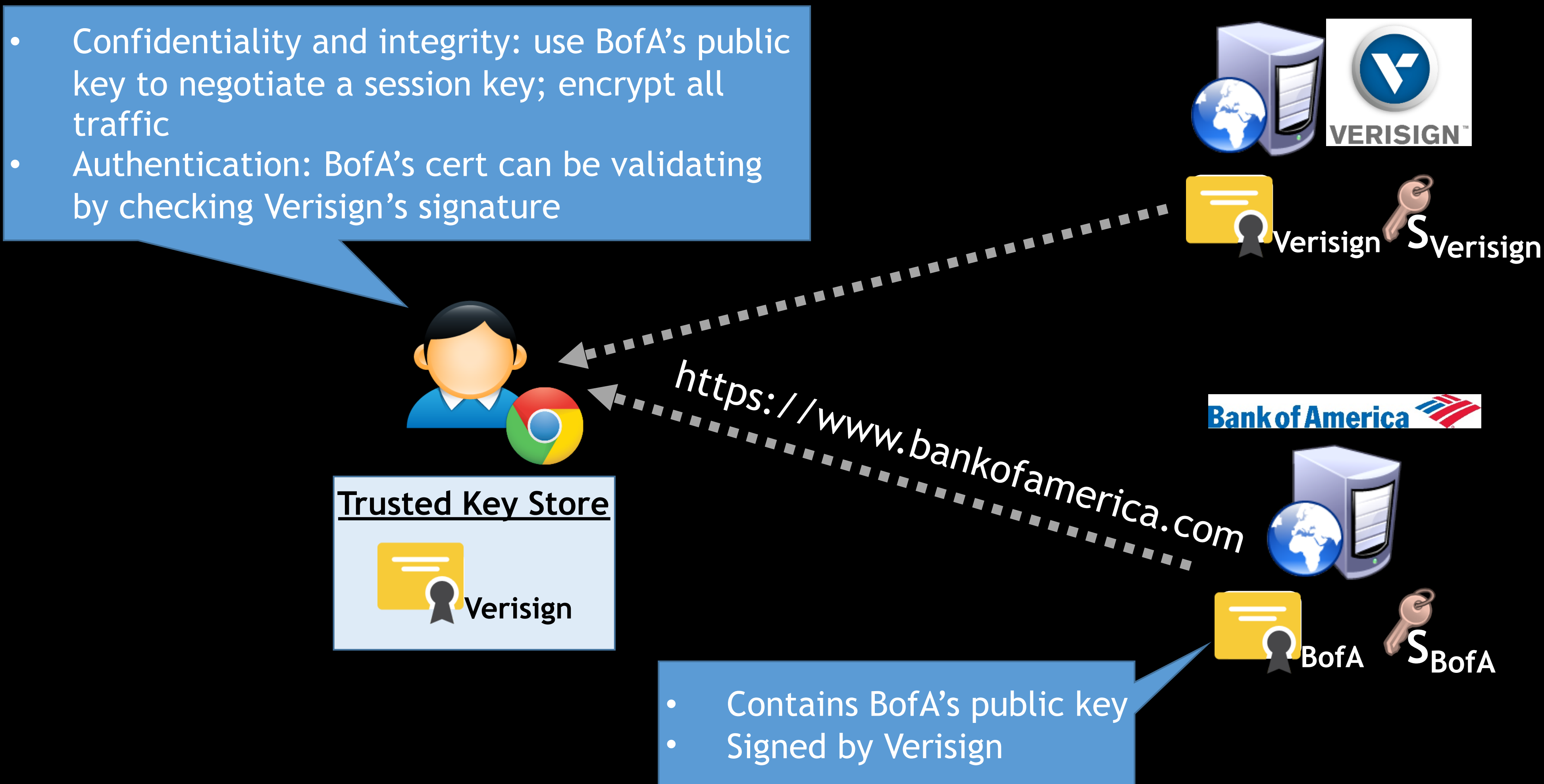
CSCI-759
Topics In Systems: Public Key Infrastructure and Network Security

Lecture 3: HTTPS

SSL/TLS

- Application-layer protocol for confidentiality, integrity, and authentication between clients and servers
  - Introduced by Netscape in 1995 as the Secure Sockets Layer (SSL)
  - Designed to encapsulate HTTP, hence HTTPS
- Transport Layer Security
  - Defined in an RFC in 1999
  - Supersedes SSL: SSL is known to be insecure and should not be used
- Sits between transport and application layers
  - Thus, applications must be TLS-aware

Goals of TLS


• Confidentiality and integrity: use BofA's public key to negotiate a session key; encrypt all traffic
• Authentication: BofA's cert can be validating by checking Verisign's signature
VERISIGN
Verisign
$S_{Verisign}$
https://www.bankofamerica.com
Bank of America
Trusted Key Store
Verisign
BofA
S_{BofA}
• Contains BofA's public key
• Signed by Verisign

Let's Talk about Certificates

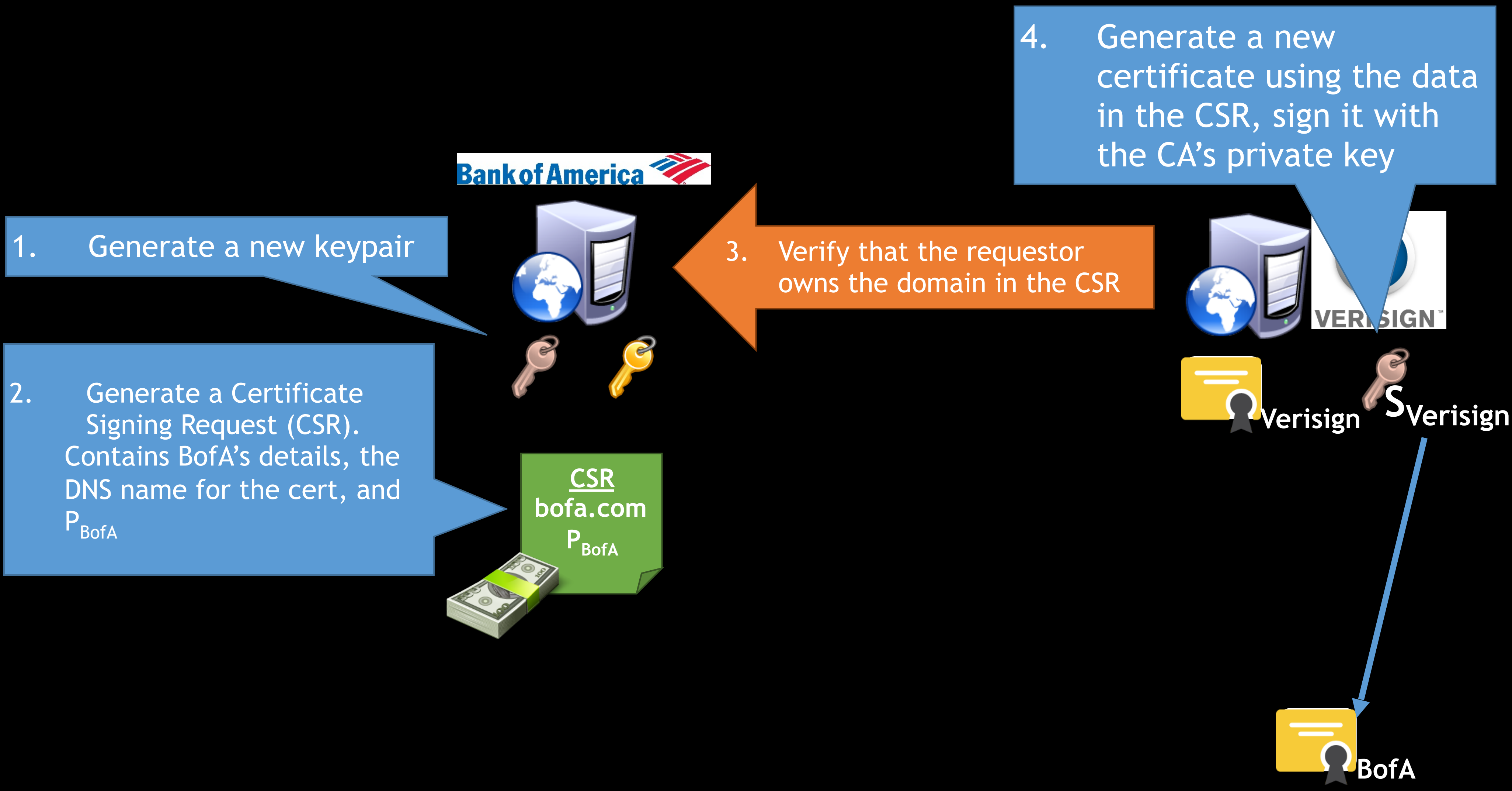
- Suppose you start a new website and you want TLS encryption
  - You need a certificate. How do you get one?
- Option 1: generate a certificate yourself
  - Use *openssl* to generate a new asymmetric keypair
  - Use *openssl* to generate a certificate that includes your new public key
- Problem?
  - Your new cert is *self-signed*, i.e. not signed by a trusted CA
  - Browsers cannot authenticate your cert to a trusted root CA
  - Users will be shown a scary security warning when they visit your site

Certificate Authorities

- Certificate Authorities (CAs) are the roots of trust in the TLS PKI
  - Symantec, Verisign, Thawte, Geotrust, Comodo, GlobalSign, Go Daddy, Digicert, Entrust, and hundreds of others
  - Issue signed certs on behalf of thir
- How do you become a CA?
  1. Create a self-signed root certifica
  2. Get all the major browser vendors to clude your cert with their software
  3. Keep your private key secret at all costs
- What is the key responsibility of being a CA?

> - Any CA can issue a cert for any domain!
> - **The only thing that stops me from buying a cert for *google.com* is a manual verification process**

Acquiring a Certificate



X.509 Certificate (Part 1)

Certificate:
Data:
Version: 3 (0x2)
Serial Number:
0c:00:93:10:d2:06:db:e3:37:55:35:80:11:8d:dc:87
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 Extended Validation Server CA
Validity
Not Before: Apr 8 00:00:00 2014 GMT
Not After : Apr 12 12:00:00 2016 GMT
Subject: businessCategory=Private Organization/1.3.6.1.4.1.311.60.2.1.3=US/1.3.6.1.4.1.311.60.2.1.2=Delaware/serialNumber=5157550/street=548 4th Street/postalCode=94107, C=US, ST=California, L=San Francisco, O=GitHub, Inc., CN=github.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b1:d4:dc:3c:af:fd:f3:4e:ed:c1:67:ad:e6:cb:

Issuer: who generated this cert? (usually a CA)

Certificates expire

Used for revocation

Github's public key

- Subject: who owns this cert?
- This is Github's certificate
- **Must be served from github.com**

X.509 Certificate (Part 2)

Additional DNS names that may serve this cert

X509v3 extensions:

X509v3 Subject Alternative Na

DNS:github.com, DNS:www.github.com

X509v3 CRL Distribution Points:

Full Name:

URI:http://crl3.digicert.com/sha2-ev-se

Full Name:

URI:http://crl4.digicert.com/sha2-ev-server-g1.crl

If this cert is revoked, it's serial will be in the lists at these URLS

X509v3 Certificate Policies:

Policy: 2.16.840.1.114412.2.1

CPS: https://www.digicert.com/CPS

Authority Information Access:

OCSP - URI:http://ocsp.digicert.com

This cert's revocation status may also be checked via OSCP

TLS Connection Establishment

Quick question

- TLS is based on the Transport Layer
  - The layer below domain name service (DNS)
- All message after TLS handshake encrypted
- If one server (with IP address) serves one domain name, it will be trivial
  - What about the server serving multiple domains (virtual hosting?)

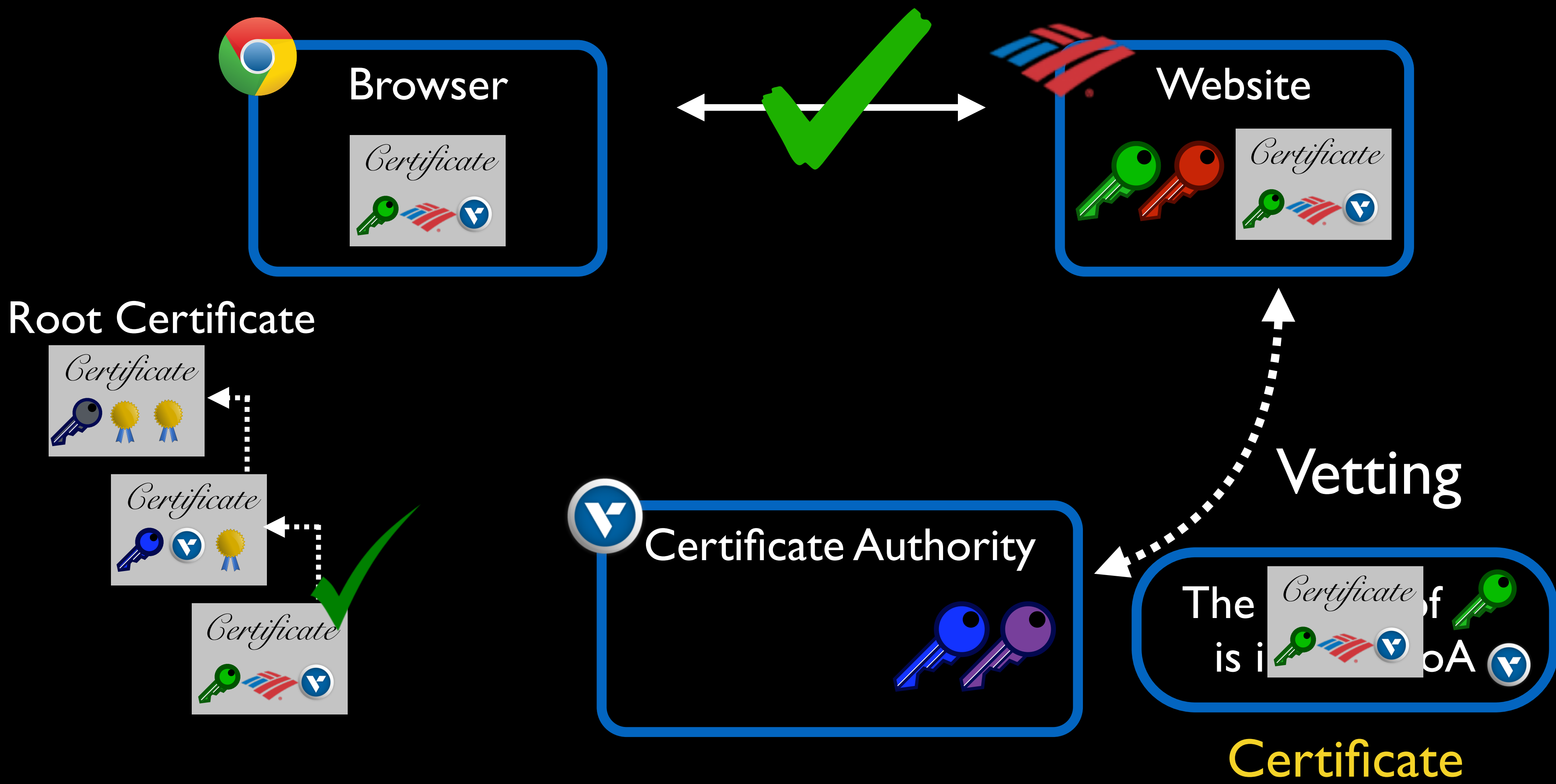
- SNI, DNS, ESNI, DNS-over-TLS, and so on.

TLS Authentication

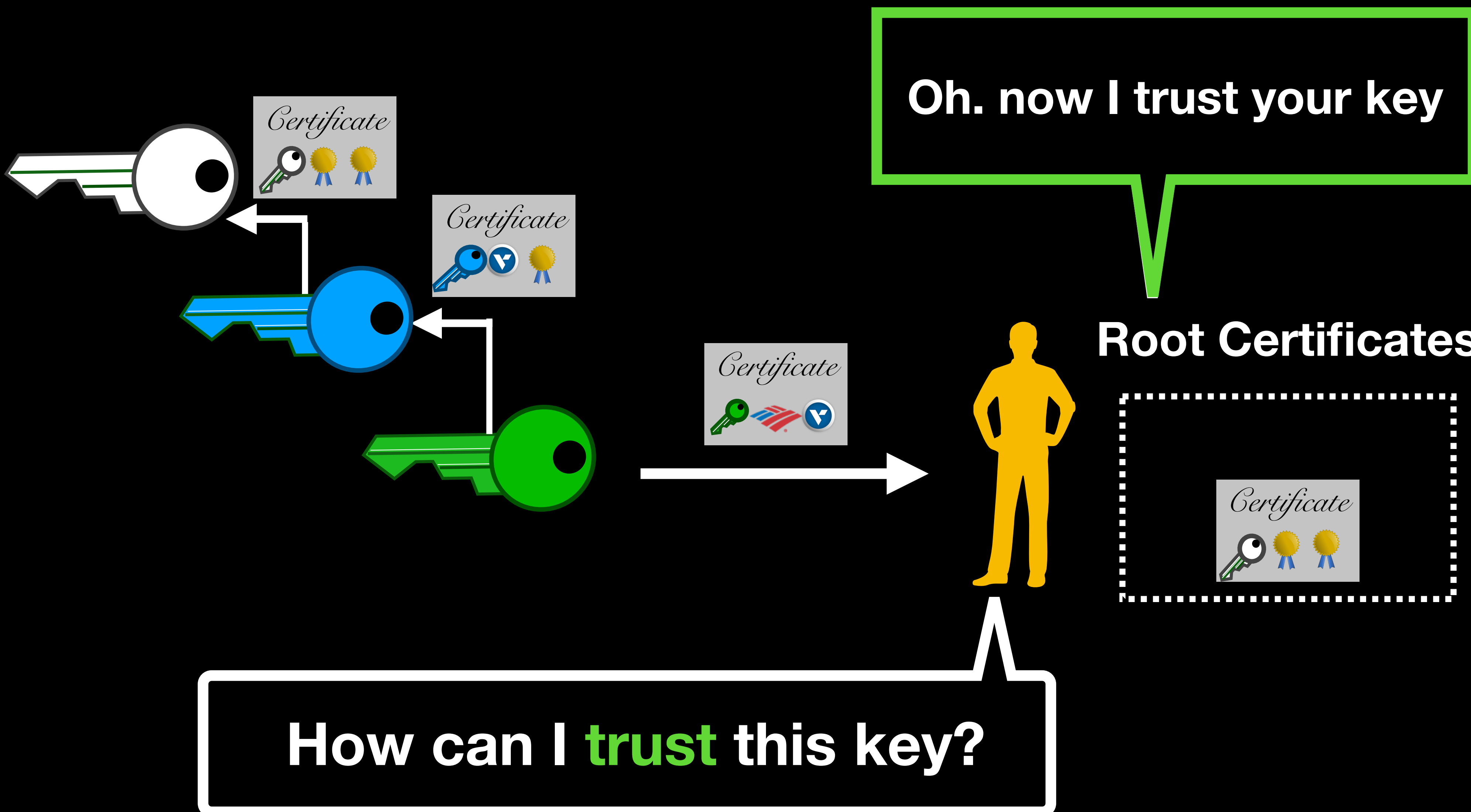
- During the TLS handshake, the client receives a certificate chain
  - Chain contains the server's cert, as well as the certs of the signing CA(s)
- The client must validate the certificate chain to establish trust
  - i.e. is this chain authentic, correct, cryptographically sound, etc.
- Client-side validation checks
  - Does the server's DNS name match the common name in the cert?
    - E.g. *example.com* cannot serve a cert with common name *google.com*
  - Are any certs in the chain expired?
  - Is the CA's signature cryptographically valid?

How HTTPS Works

How can users truly know with whom they are communicating?



HTTPS: Hierarchical PKI



X.509 Format

```
Version: 3 (0x2)
Serial Number:
  0e:77:76:8a:5d:07:f0:e5:79:59:ca:2a:9d:50:82:b5
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com,
        CN=DigiCert High Assurance EV CA-1
Validity
  Not Before: May 27 00:00:00 2011 GMT
  Not After : Jul 29 12:00:00 2013 GMT
Subject: C=US, ST=California, L=San Francisco,
         O=GitHub, Inc., CN=github.com
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
      Modulus:
        00:ed:d3:89:c3:5d:70:72:09:f3:33:4f:1a:72:74:
        d9:b6:5a:95:50:bb:68:61:9f:f7:fb:1f:19:e1:da:
```

X.509 Format

- Real world examples

CA Trustworthiness (1)

- A CA is essentially a trusted third party
  - Certificate signatures are attestations of authenticity for the server and (optionally) the client
  - Remember: trust is bad and should be minimized!
- If a CA mistakenly (or purposefully) signs a certificate for a domain and provides it to a malicious principal, TLS can be subverted
- Not only must we trust root CAs, but also intermediate CAs that have been delegated signing authority

CA Trustworthiness (2)

- Clearly, the CA secret key must be protected at all costs
  - Possession of the CA secret key grants adversaries the ability to sign any domain
  - Attractive target for adversaries
- Signatures should only be issued after verifying the identity of the requester
  - Also known as domain validation
  - Should be easy, right?

CA Failures

```
Issued to: Microsoft Corporation
Issued by: VeriSign Commercial Software Publishers CA
Valid from 1/29/2001 to 1/30/2002
Serial number is 1B51 90F7 3724 399C 9254 CD42 4637 996A

Issued to: Microsoft Corporation
Issued by: VeriSign Commercial Software Publishers CA
Valid from 1/30/2001 to 1/31/2002
Serial number is 750E 40FF 97F0 47ED F556 C708 4EB1 ABFD
```

- In 2001, VeriSign issued two executable signing certificates to someone claiming to be from Microsoft
  - Could be used to issue untrusted software updates

Comodo

Independent Iranian hacker claims responsibility for Comodo hack

Posts claiming to be from an Iranian hacker responsible for the Comodo hack …

by **Peter Bright** - Mar 28 2011, 11:15am EDT

65

```
Hello

I'm writing this to the world, so you'll know more about me..

At first I want to give some points, so you'll be sure I'm the hacker:

I hacked Comodo from InstantSSL.it, their CEO's e-mail address mfpenco@mfpenco.com
Their Comodo username/password was: user: gtadmin password: [trimmed]
Their DB name was: globaltrust and instantsslcms
```

The alleged hacker's claim of responsibility on pastebin.com

The hack that resulted in Comodo creating certificates for popular e-mail providers including Google Gmail, Yahoo Mail, and Microsoft Hotmail has been claimed as the work of an independent Iranian patriot. A post made to data sharing site pastebin.com by a person going by the handle "comodohacker" claimed responsibility for the hack and described details of the attack. A second post provided source code apparently reverse-engineered as one of the parts of the attack.

Diginotar

Another fraudulent certificate raises the same old questions about certificate authorities

For the second time this year, Iranian hackers have created a fraudulent …

by **Peter Bright** - Aug 29 2011, 11:12pm EDT

42

Earlier this year, an Iranian hacker broke into servers belonging to a reseller for certificate authority Comodo and issued himself a range of certificates for sites including Gmail, Hotmail, and Yahoo! Mail. With these certificates, he could eavesdrop on users of those mail providers, even if they use SSL to protect their mail sessions.

It's happened again. This time, Dutch certificate authority DigiNotar has issued a fraudulent certificate for google.com and all subdomains. As before, Gmail appears to be the target. The perpetrator also appears to be Iranian, with reports that the certificate has been used in the wild for man-in-the-middle attacks in that country. The certificate was issued on July 10th, and so could have been in use for several weeks prior to its discovery.

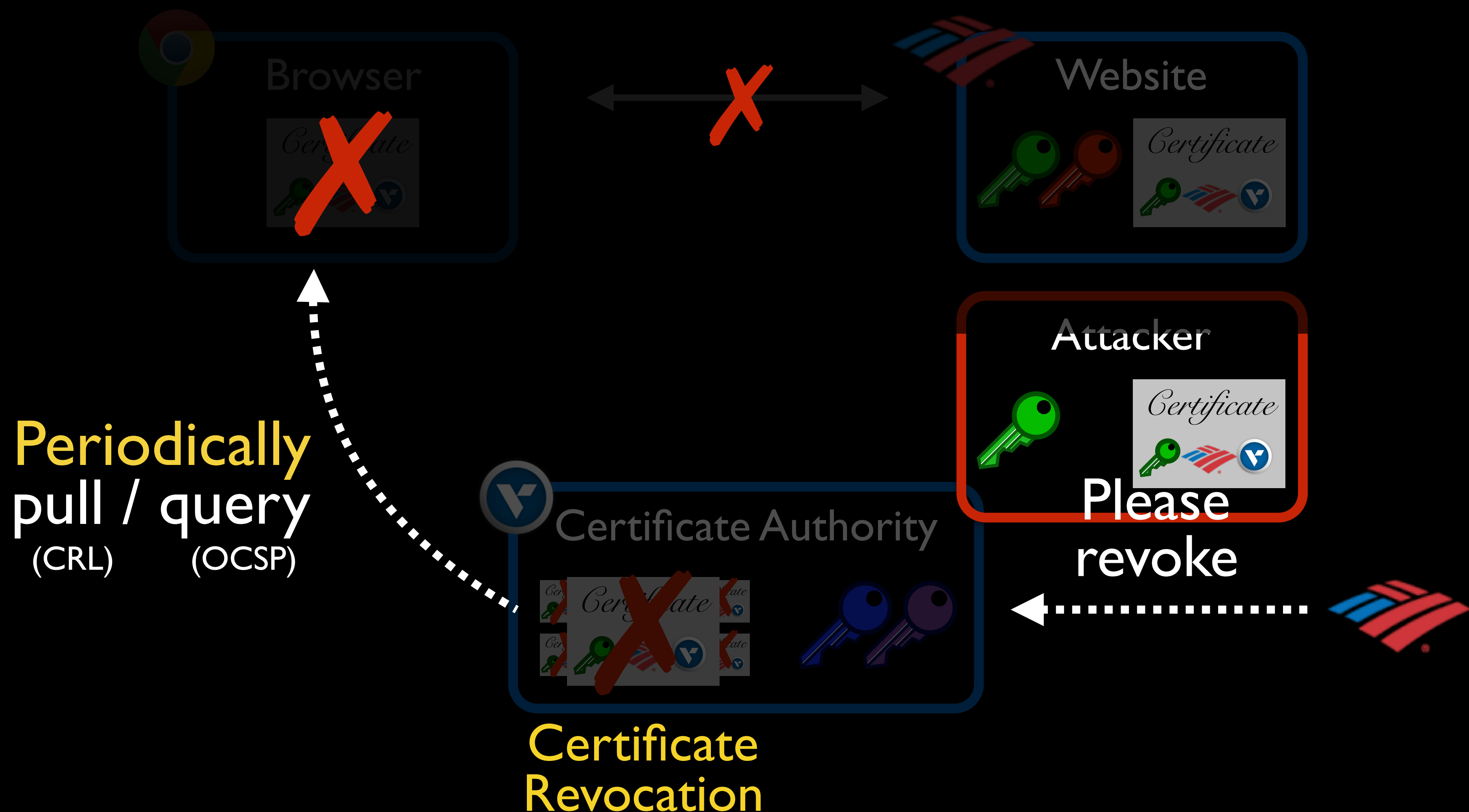
DigiNotar has revoked the certificate, which provides some protection to users (though many applications do not bother checking for revocations). However, the company has so far not disclosed how the certificate was issued in the first place, making it unclear that its integrity has been restored. As a result, Google and Mozilla have both made patches to Chrome and Firefox respectively that blacklist the entire certificate authority.

How to handle those situations?

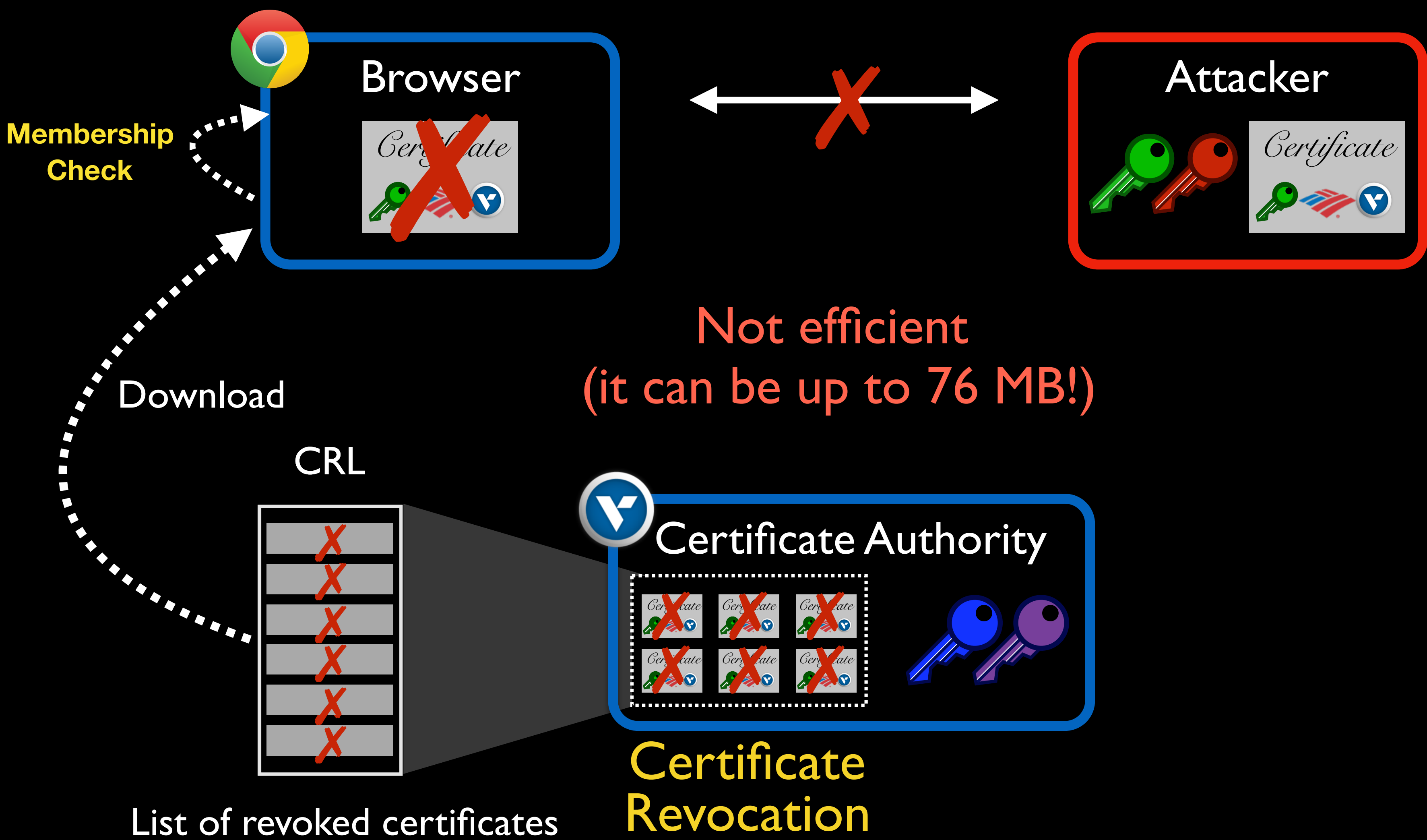
- A certificate has been mis-issued.
  - In the perspective of clients, the certificate seems legit
  - Still valid (not expired)
- Question:
  - How can we protect clients from accepting mis-issued certificates?
    - Revocation

Certificate revocation

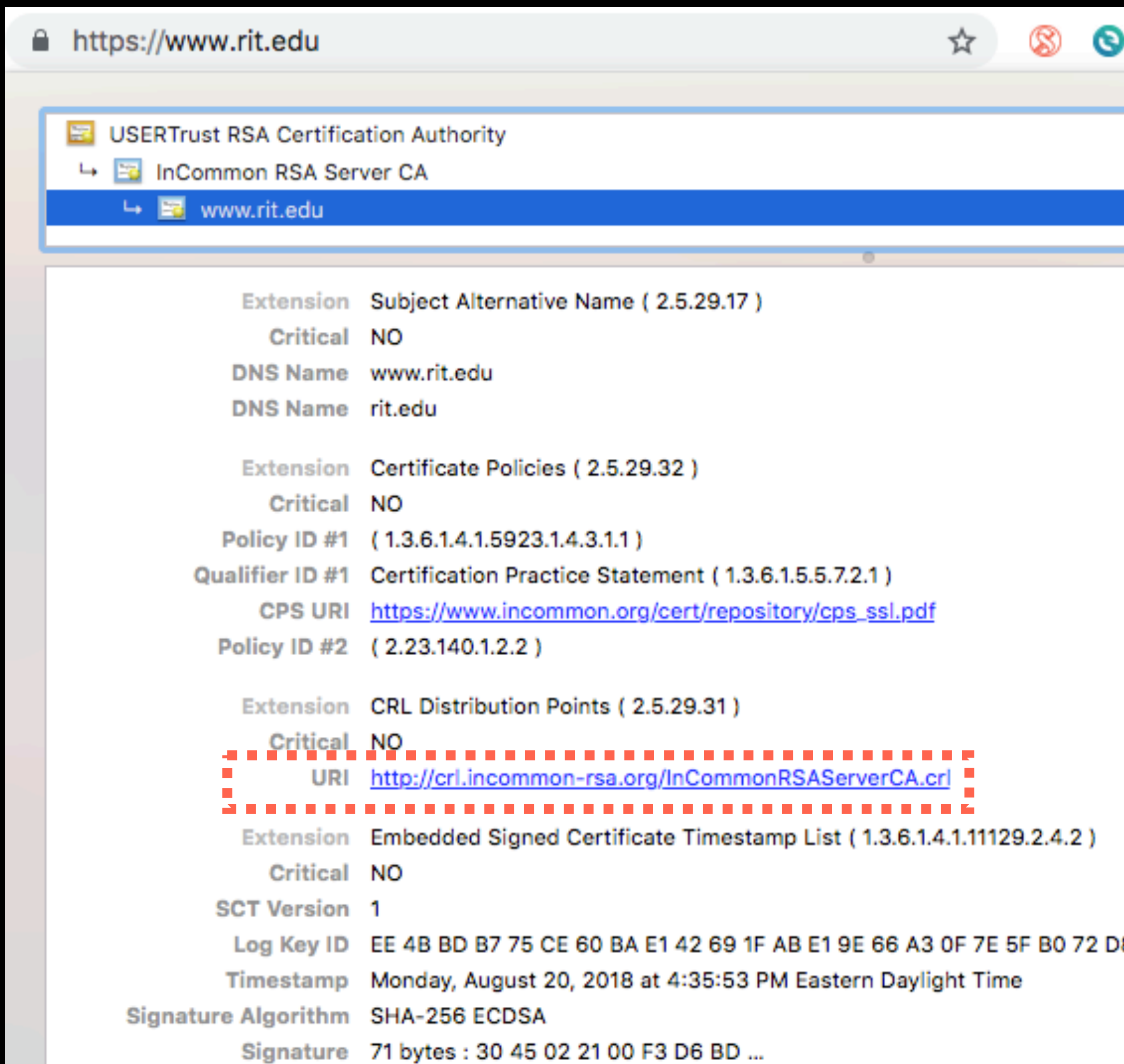
What happens when a certificate is no longer valid?



Revocation Check (1)
Certificate Revocation List

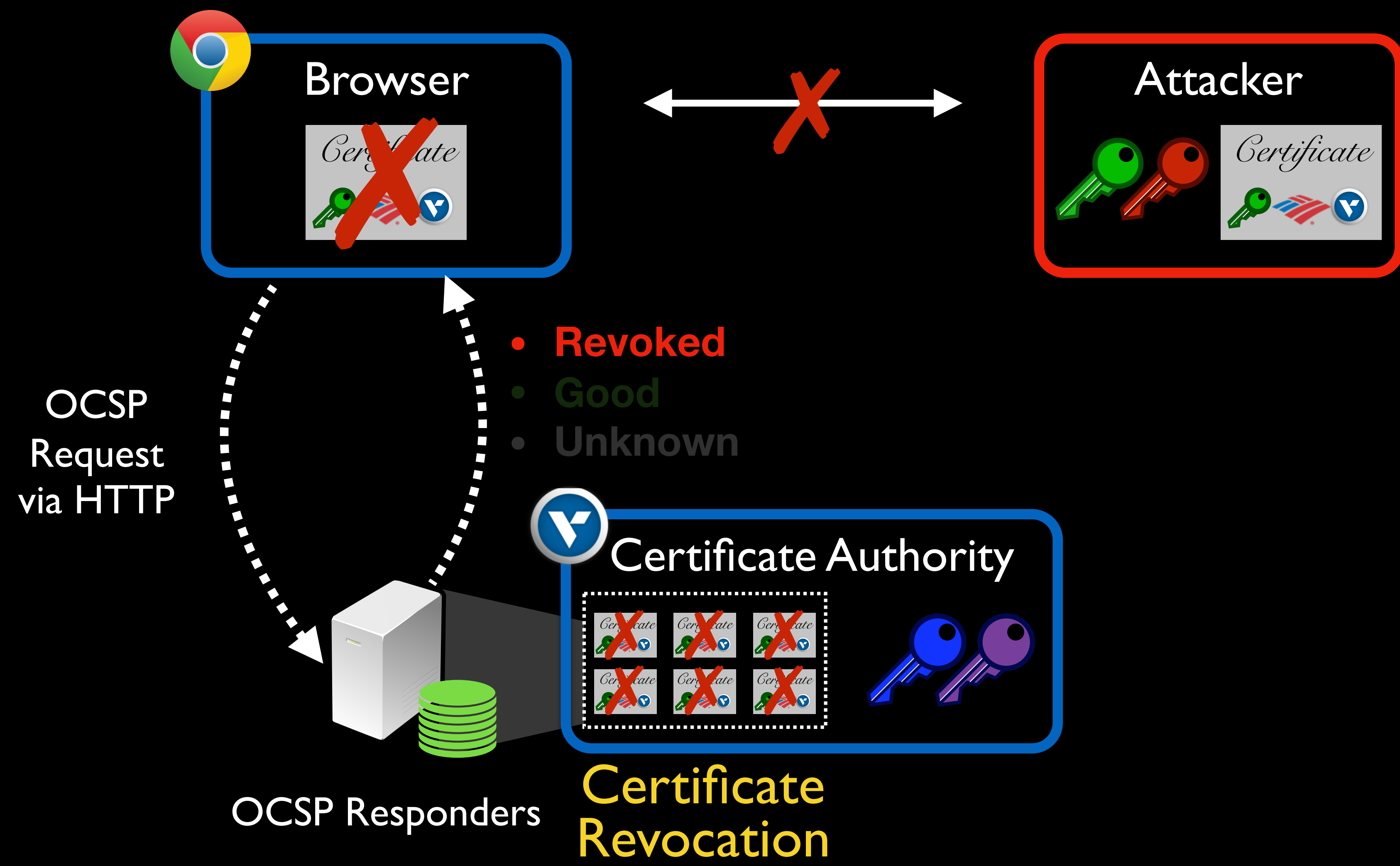


Revocation Check (1)
Certificate Revocation List

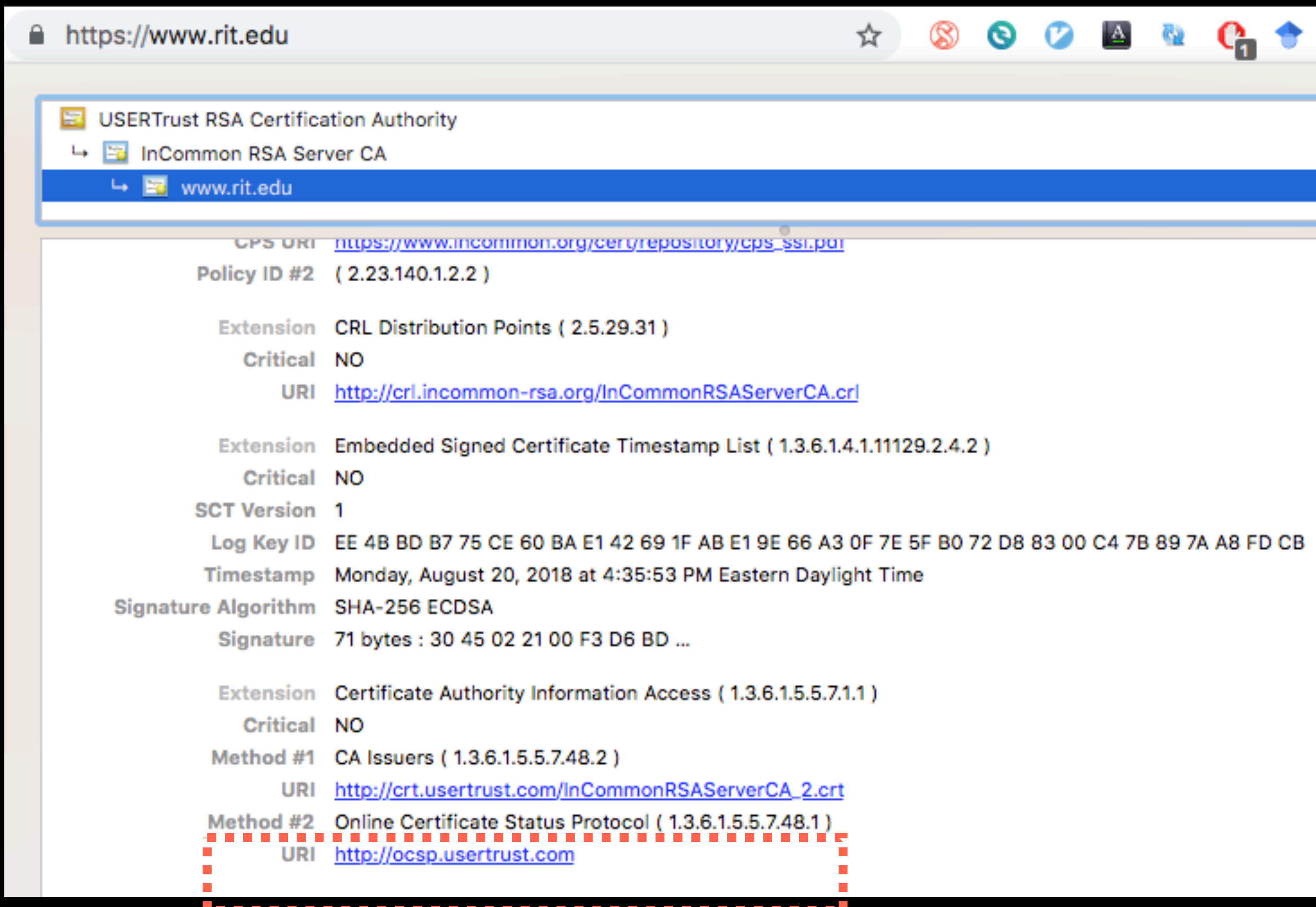



```
$ openssl crl -inform DER -text -noout -in InCommonRSAServerCA.crl
```

Revocation Check (2)
Online Certificate Status Protocol

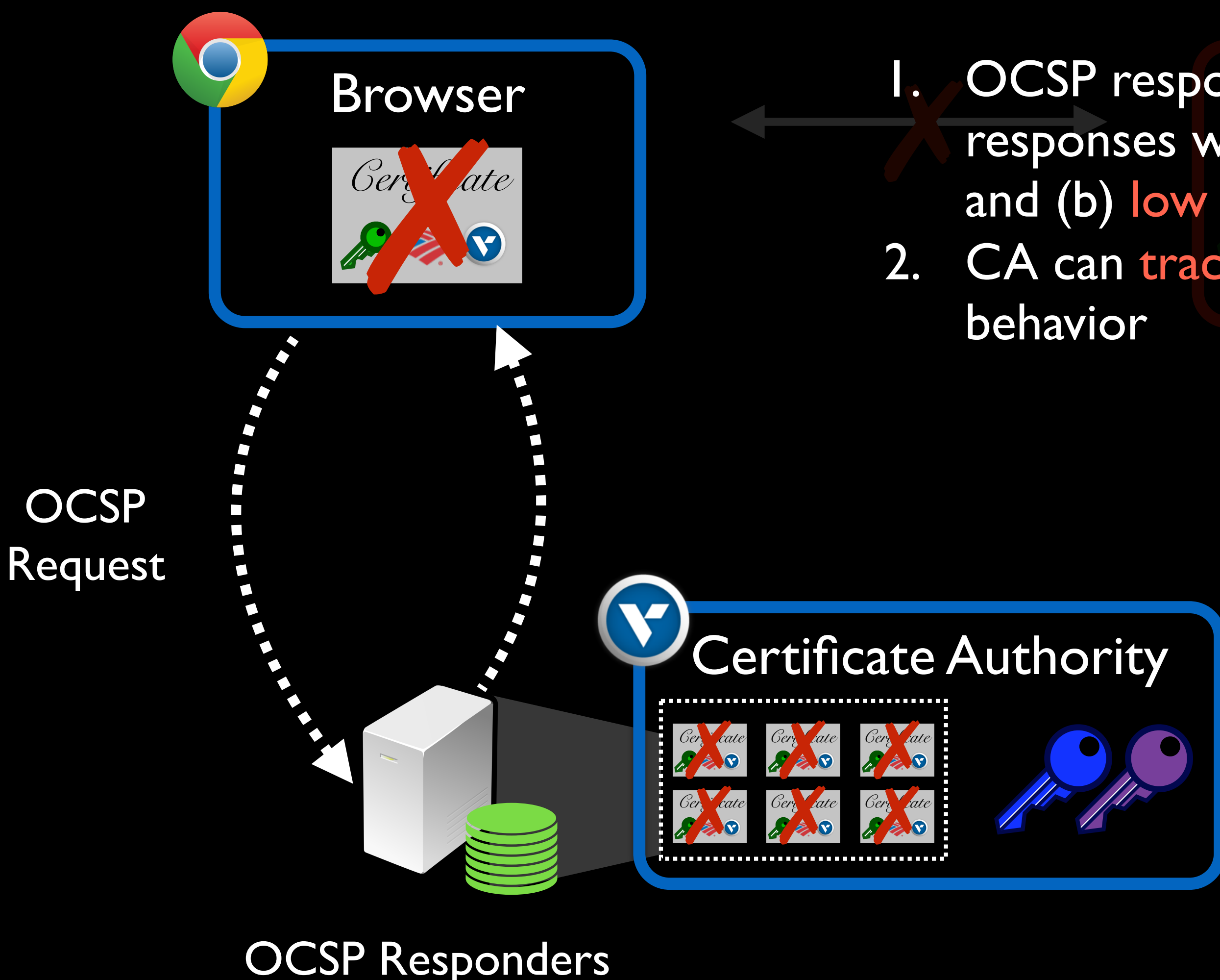


Revocation Check (2)
Online Certificate Status Protocol



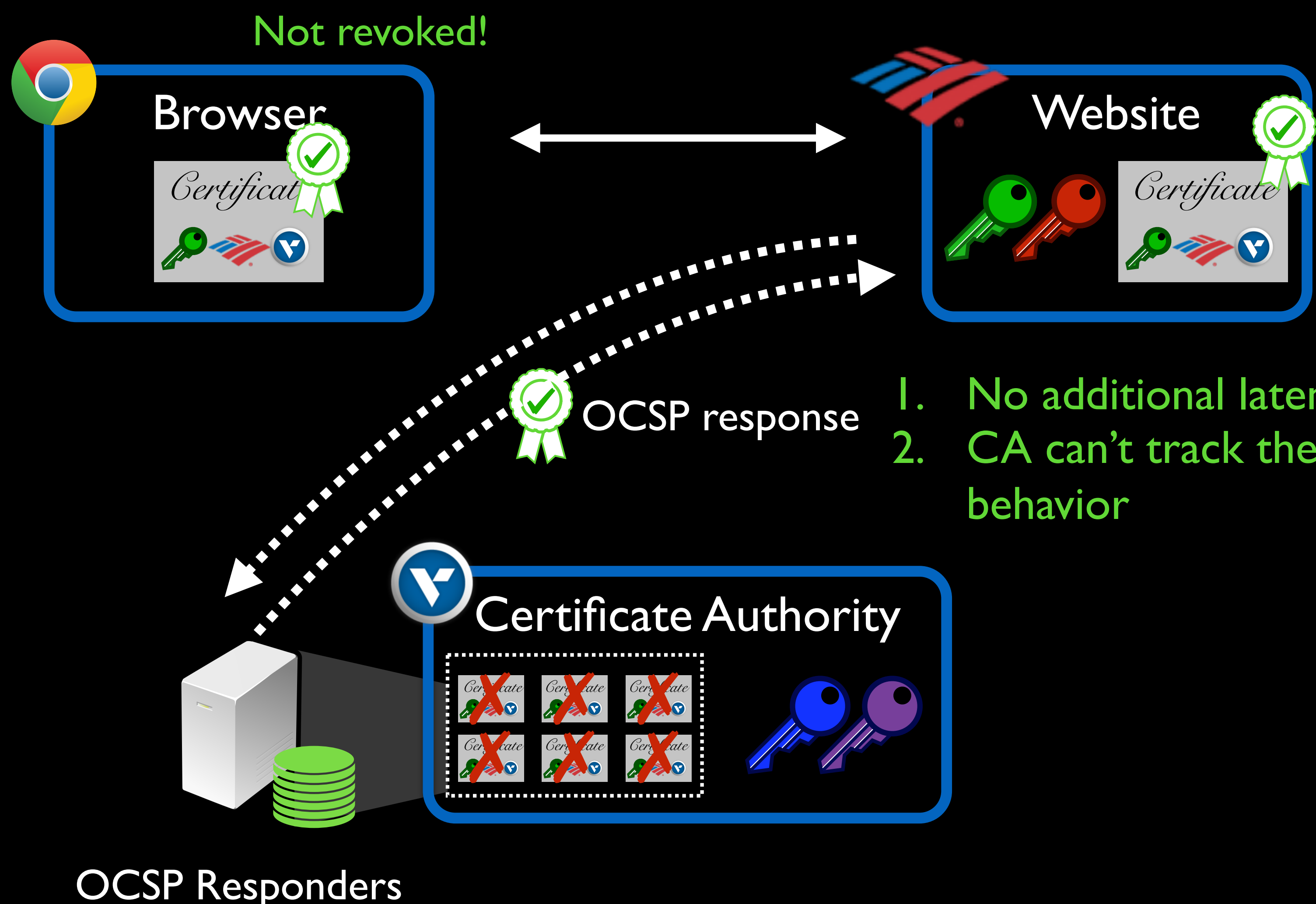

```
$ openssl ocsp -issuer cert.pem -serial
5226810331521645508876562747113126991 -url http://ocsp.usertrust.com
-header host ocsp.usertrust.com
```

Challenges of
Online Certificate Status Protocol



1. OCSP responders need to provide responses with (a) high availability and (b) low latency
2. CA can track users' browsing behavior

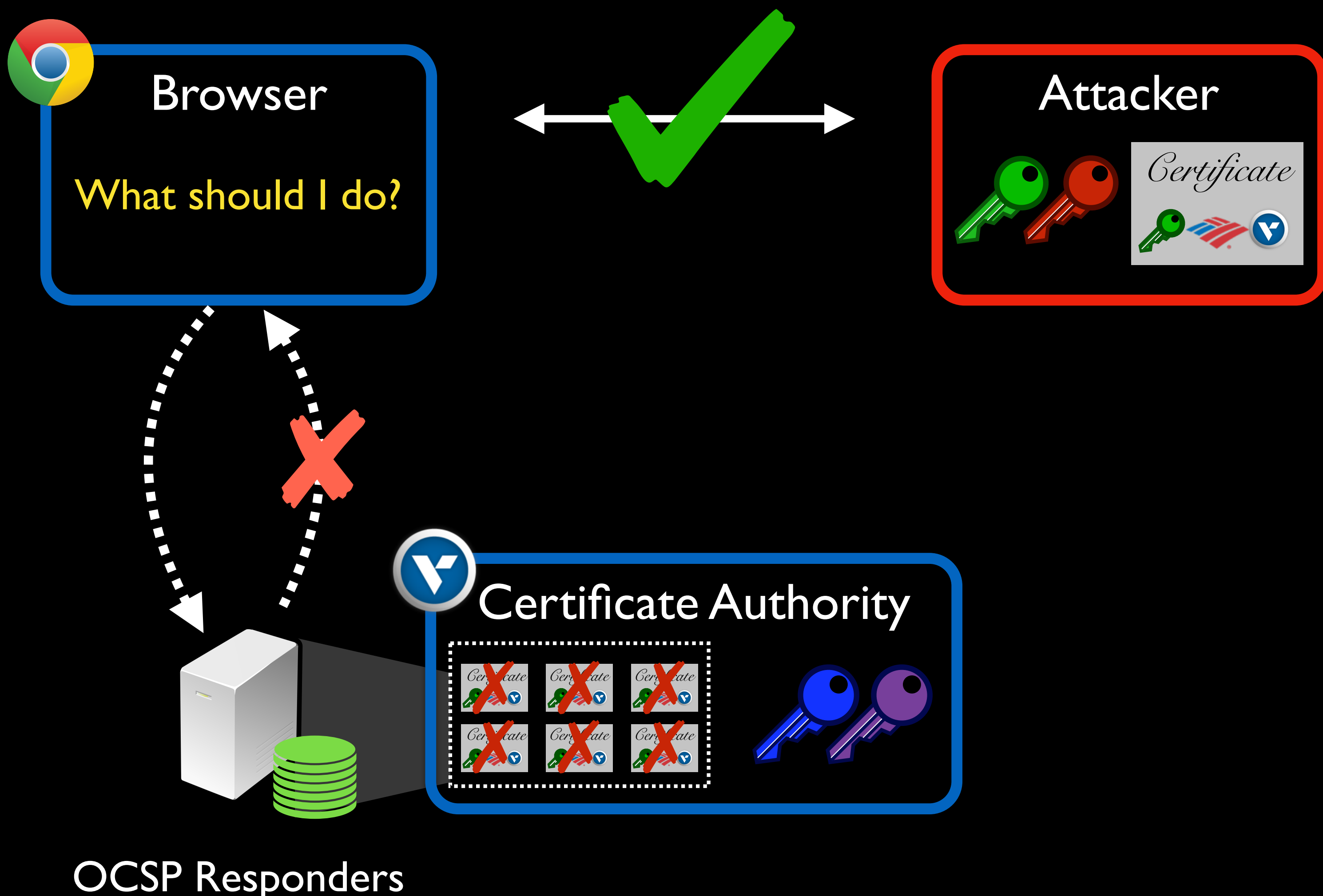
OCSP Stapling



1. No additional latency
2. CA can't track the browsing behavior

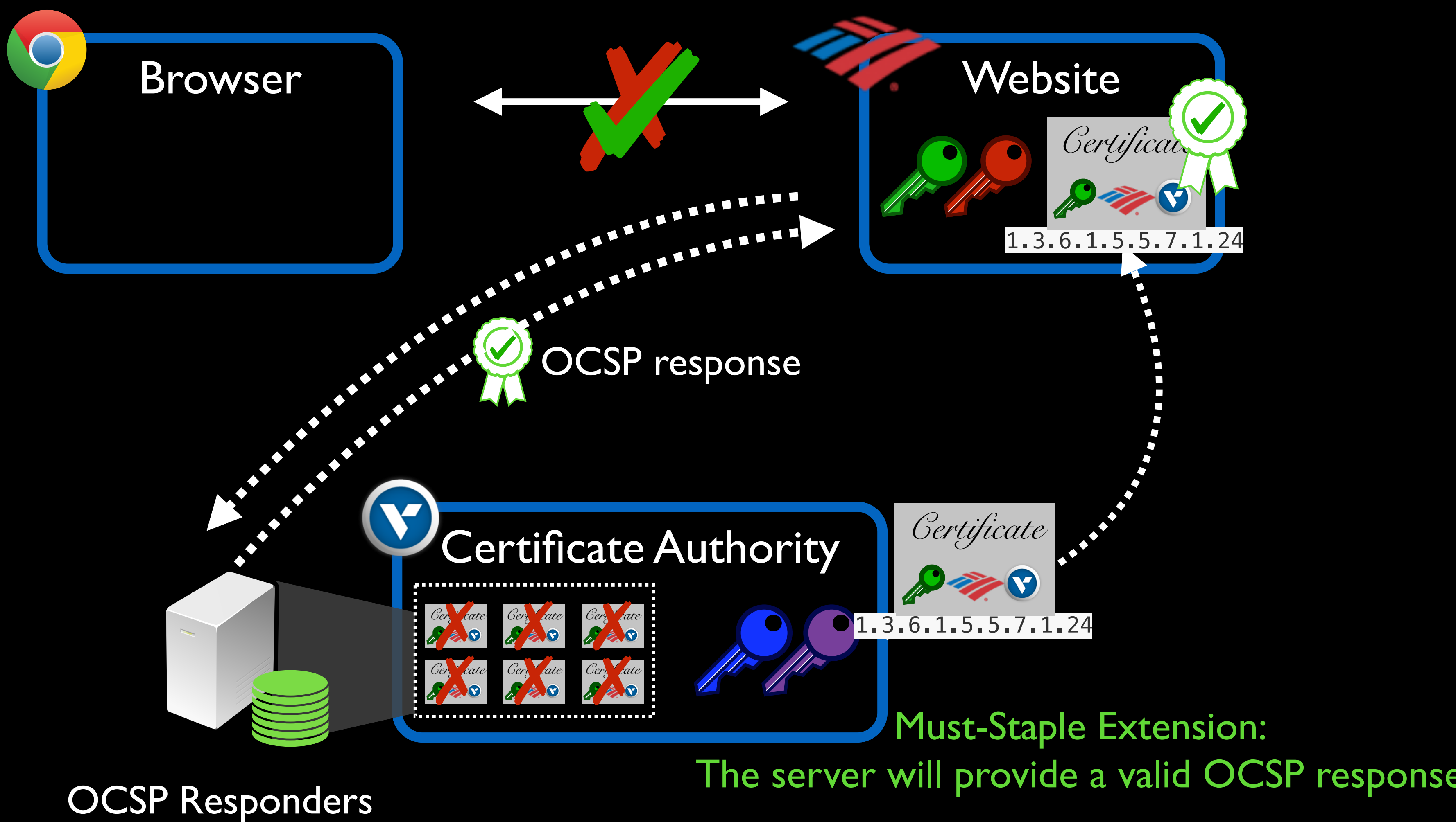
Challenges still remain:
Soft failure

Most clients will accept a certificate
even if they are unable to obtain revocation information



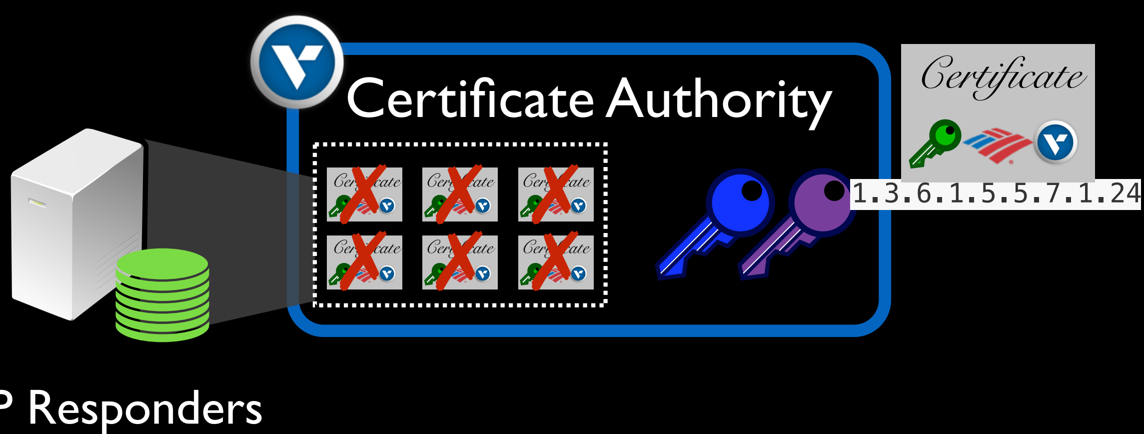
OCSP Must-Staple

- No additional latency
- No privacy issues
- No soft failure

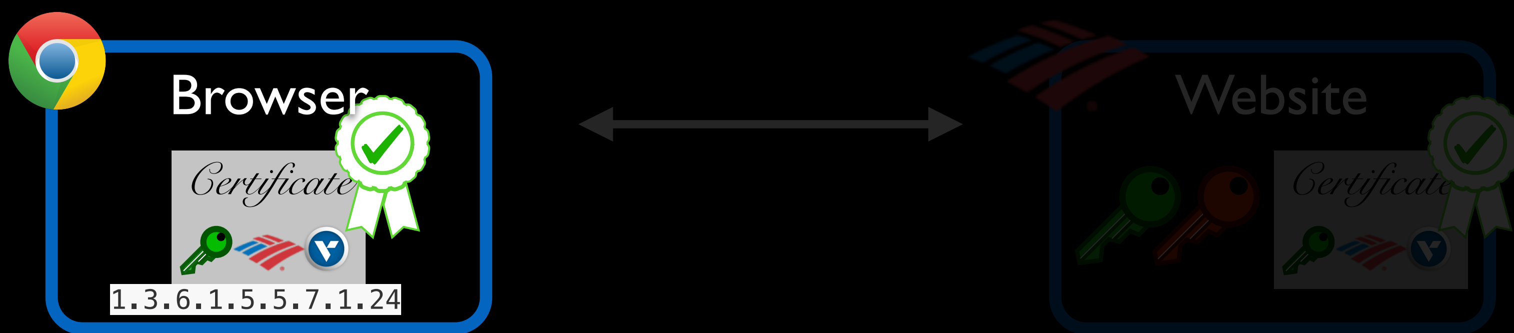


To support OCSP Must Staple
(1) CA

- Include the OCSP Must-Staple extension into certificates
- Run reliable/error-free OCSP responders

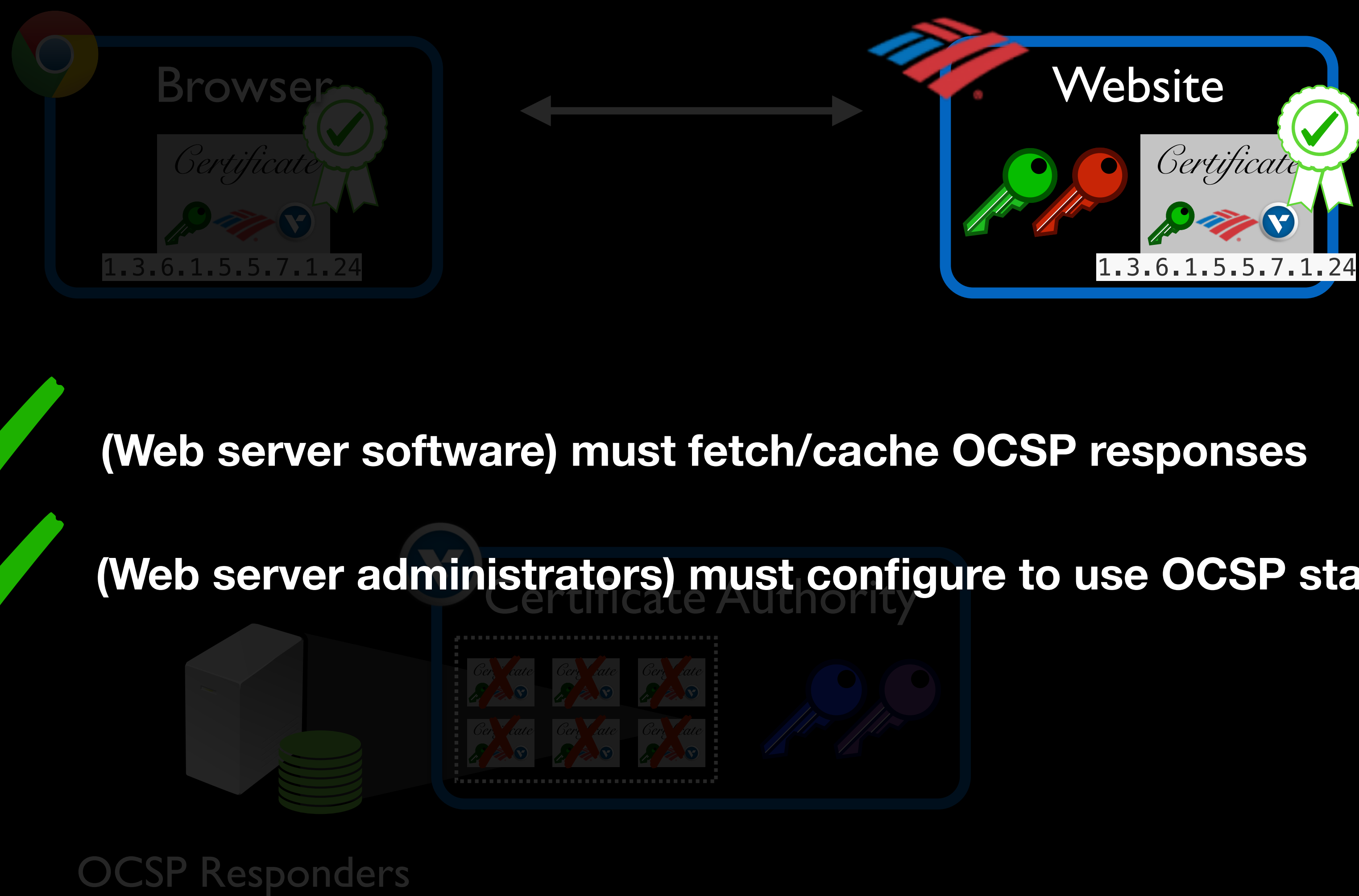


To support OCSP Must Staple (2) Clients



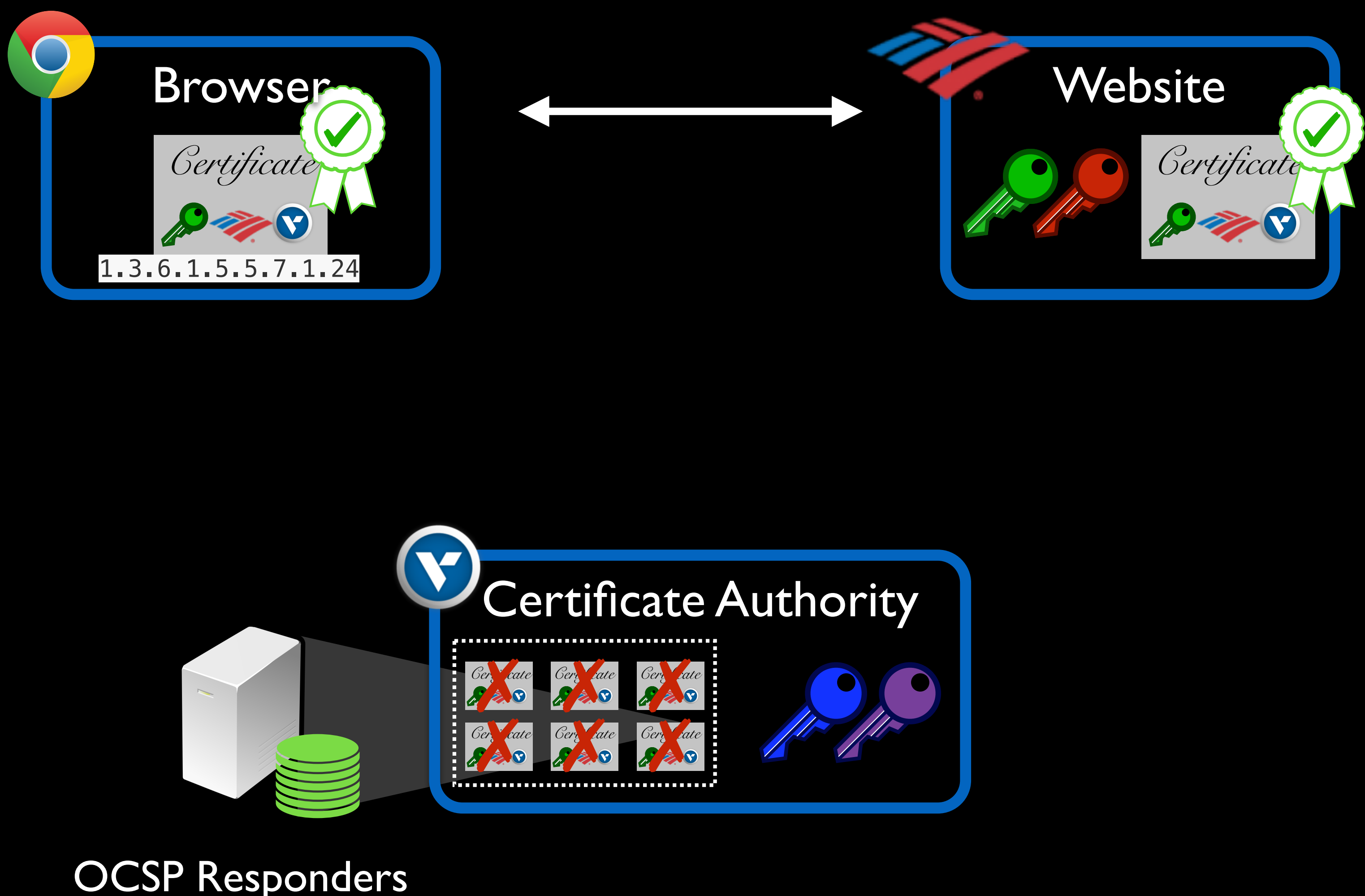
- Understand the OCSP Must-Staple extension in the certificate
- Present the Certificate Status Request (CSR) to the web servers
- Reject the certificate if they do not receive OCSP responses

To support OCSP Must Staple
(3) Web servers



- **(Web server software) must fetch/cache OCSP responses**
- **(Web server administrators) must configure to use OCSP stapling**

To support OCSP Must Staple



Is the Web Ready for OCSP Must-Staple?

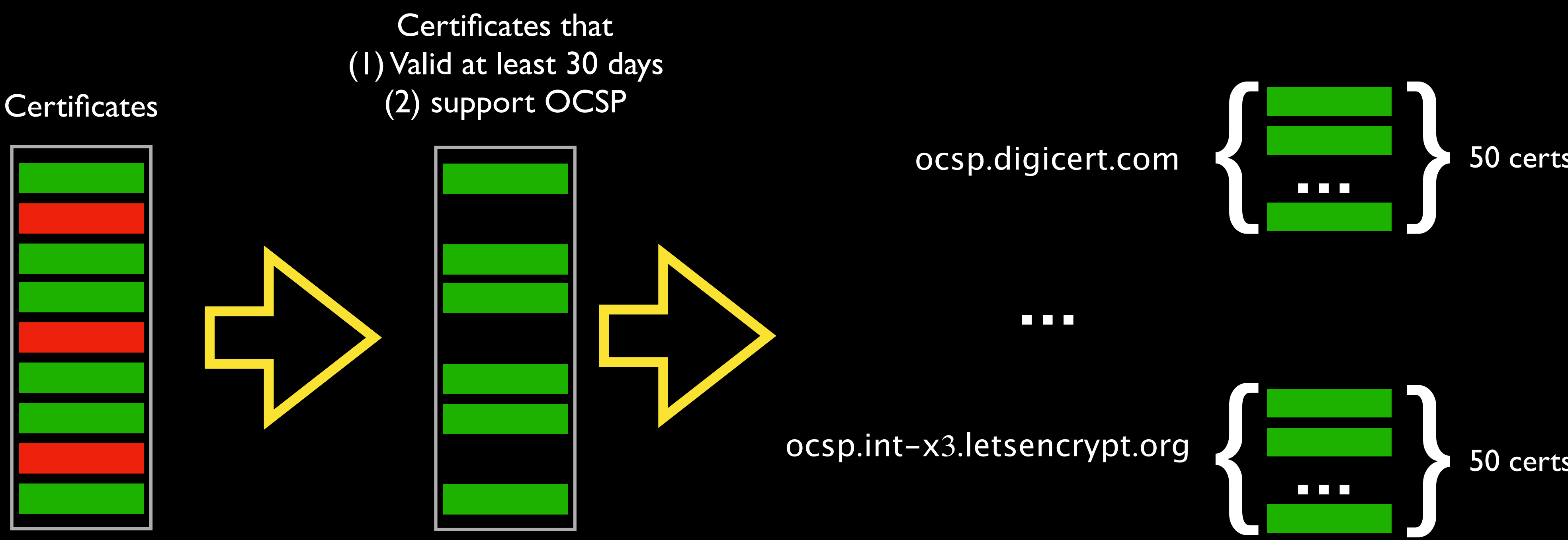
Certificate Authority
(OCSP Responder)

Website

Browser

- ✓ Availability
- ✓ Validity
- ✓ Consistency with CRL

Measuring OCSP Responders

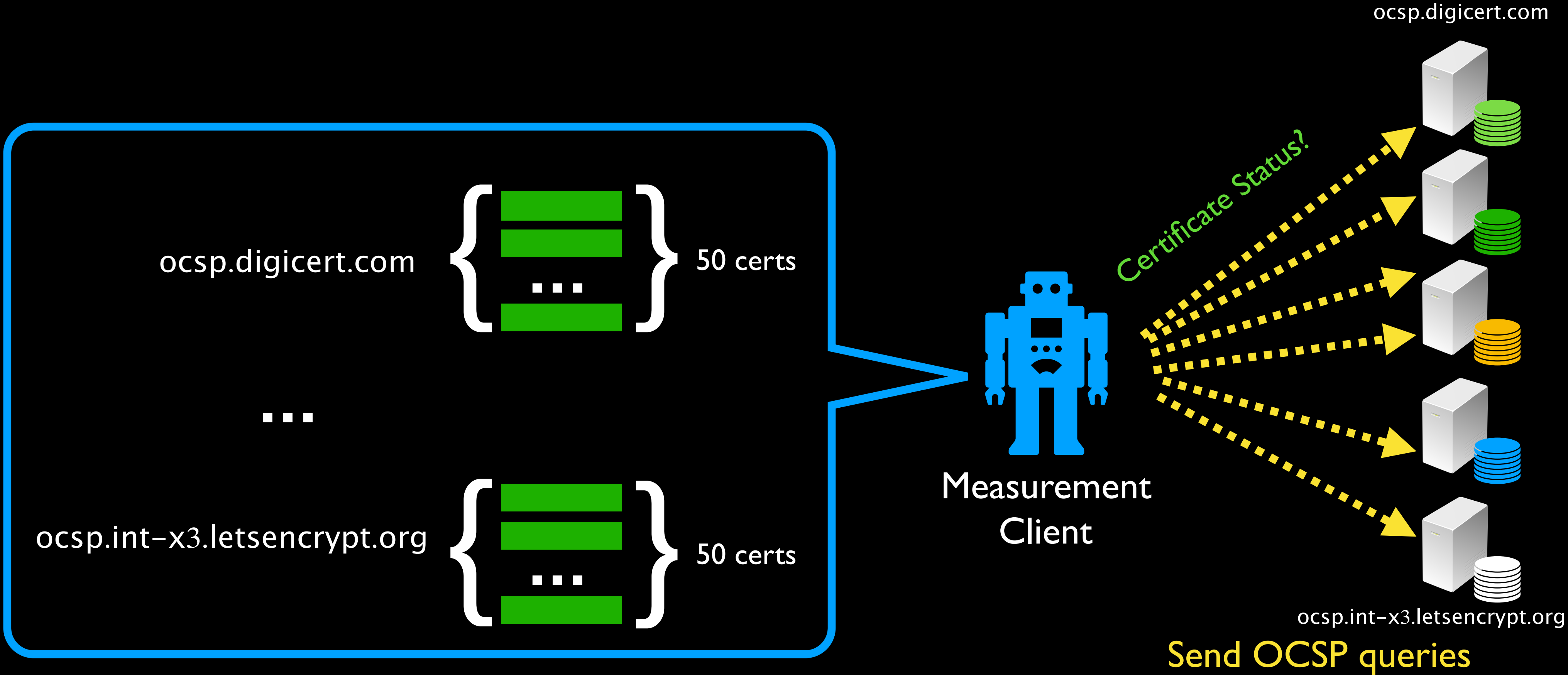


112 M certificates

77 M certificates

536 OCSP responders
with 14,634 certificates

Measuring OCSP Responders



Measurement

Oregon (US West)

Virginia (US East)

São Paulo (Brazil)

Paris (France)

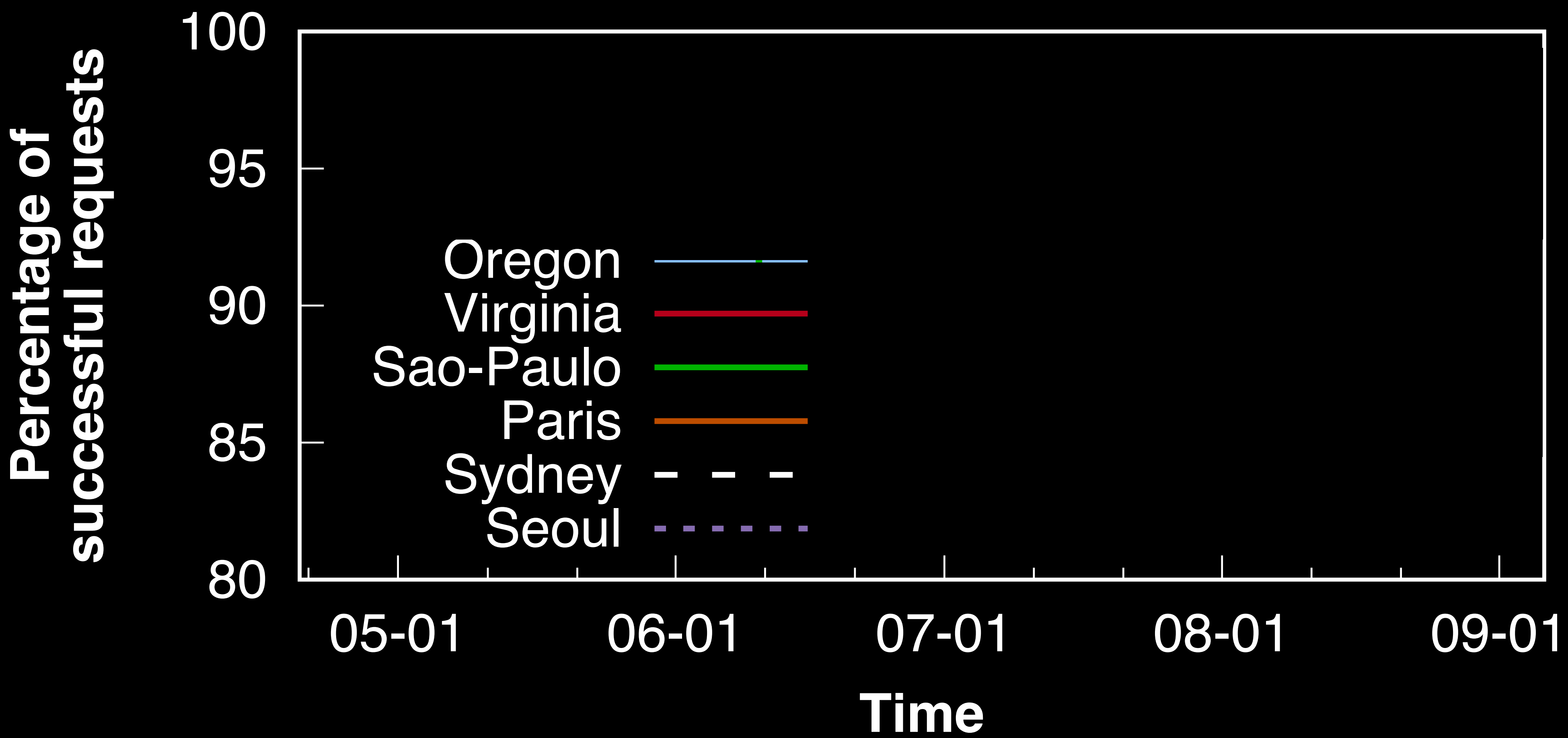
Sydney (Australia)

Seoul (Korea)

Scan them every hour
April 25, 2018 ~ September 4, 2018

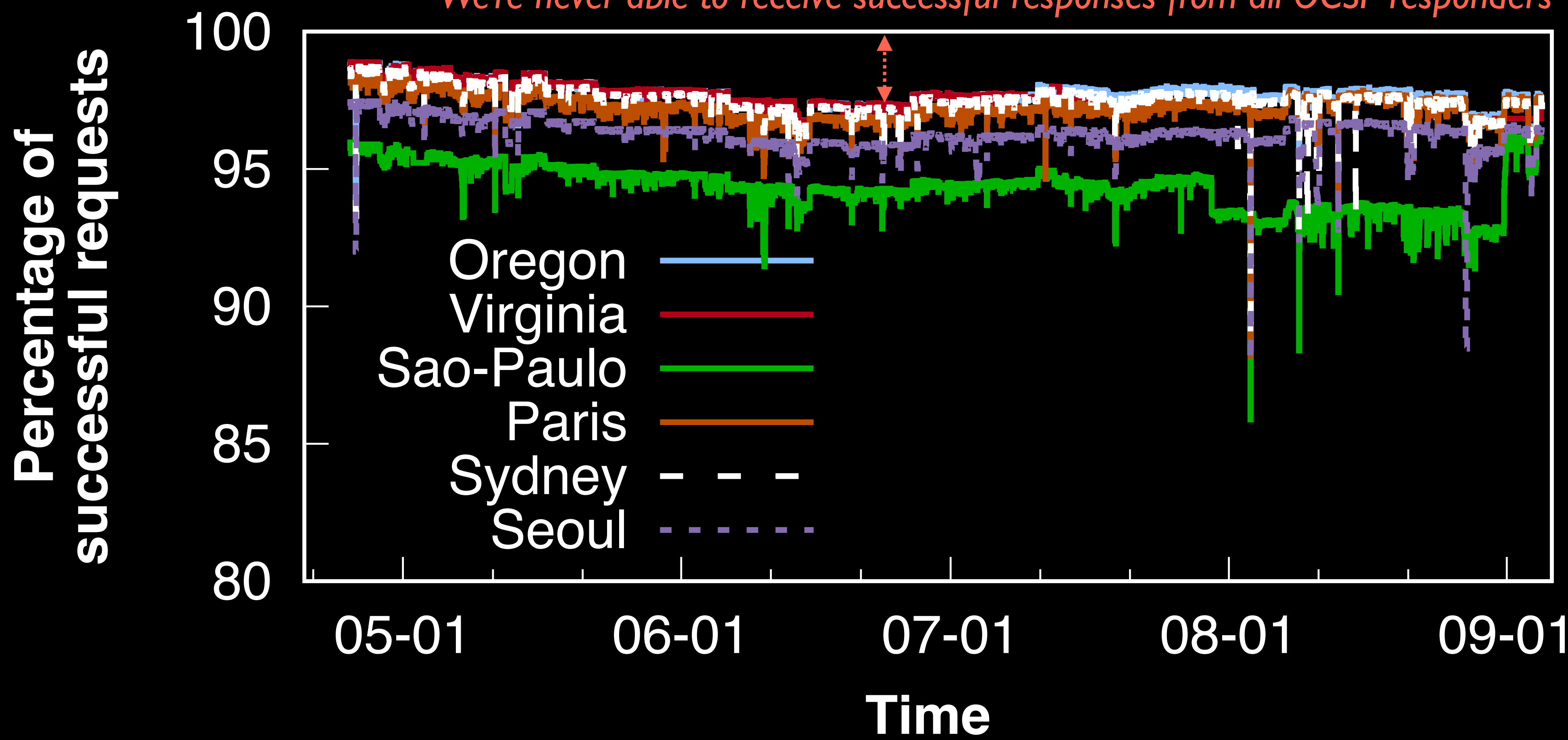
~ 46 M OCSP requests & responses

(1) Availability



(1) Availability Overview

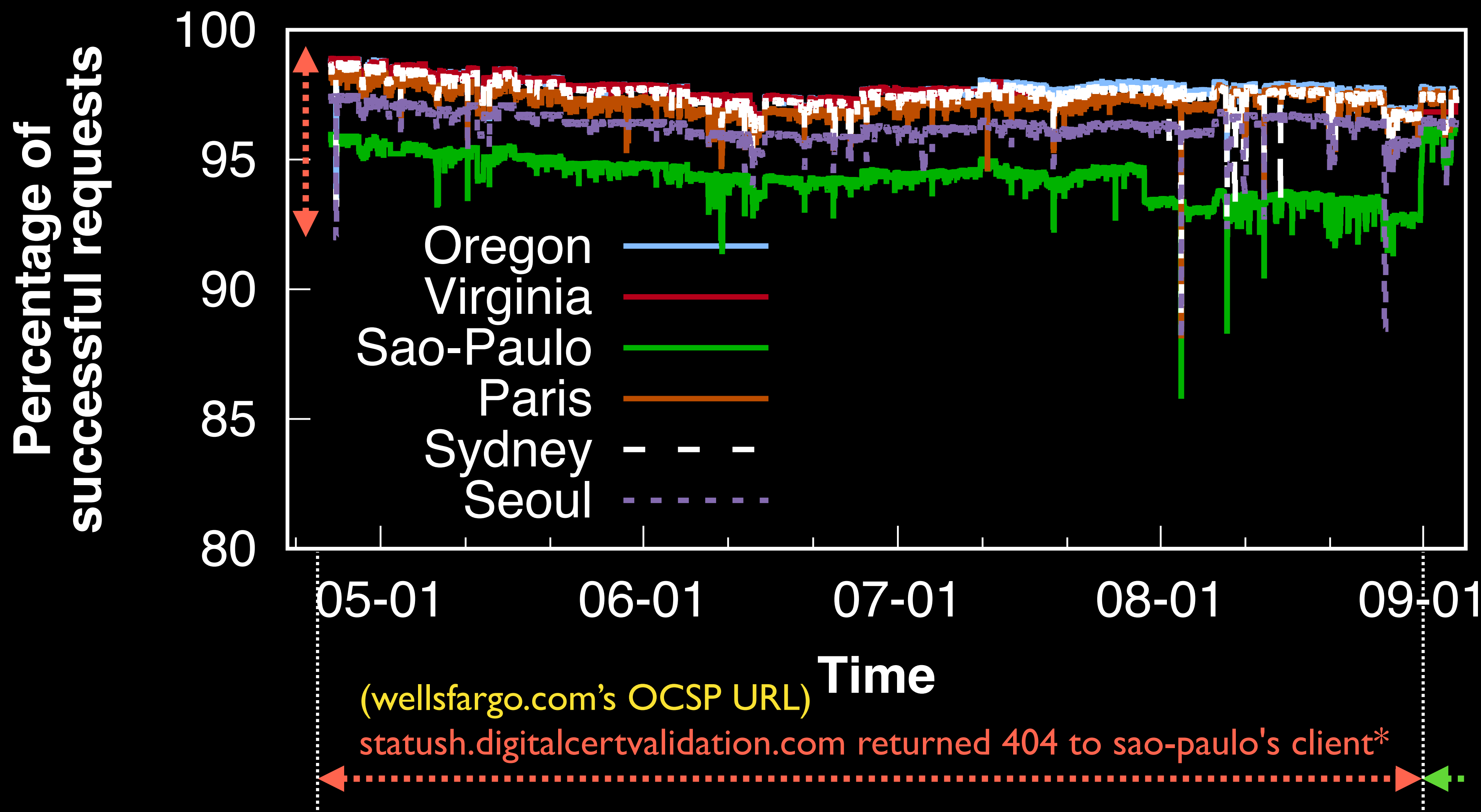
We're never able to receive successful responses from all OCSP responders



For 29 OCSP responders, there was at least one measurement client that was never able to make a successful request.

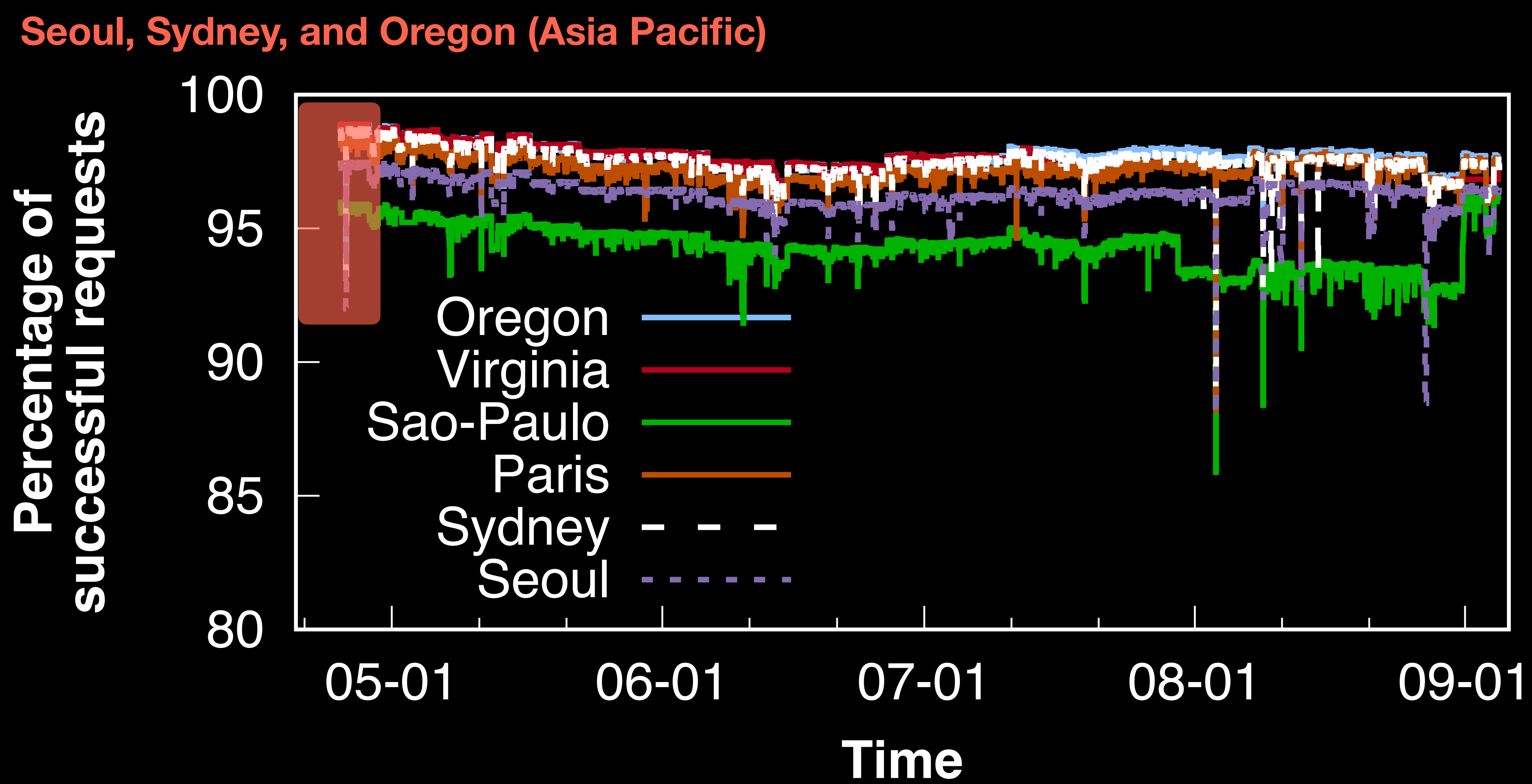
(16: DNS problem, 4:TCP connection errors, 8: HTTP problems, 1: HTTPS Error)

(1) Availability: Geographical Differences



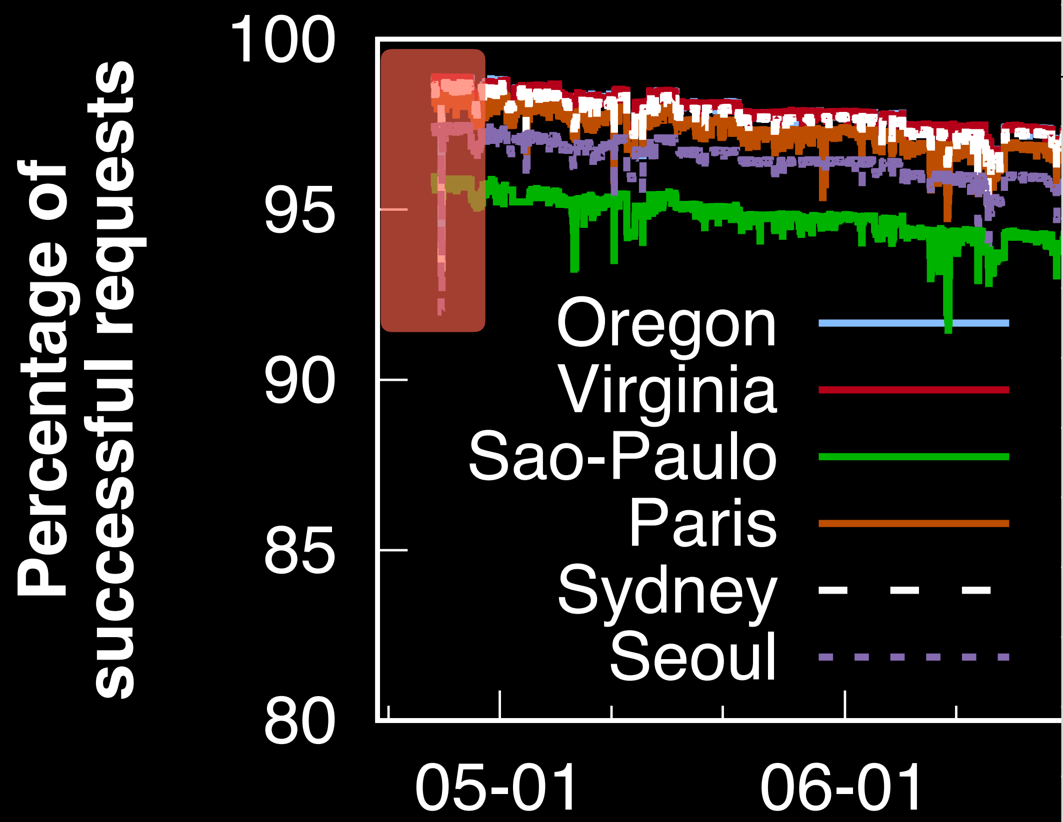
*After we contacted them on August 29th, the issue was fixed at 11pm August 31st.

(1) Availability: Transient Failure



(I) Availability:
Transient Failure (Case-Study)

Seoul, Sydney, and Oregon (Asia Pacific)



OCSP Server Name	DNS Records
ocsp.comodoca.com	
ocsp.comodoca4.com	
ocsp.gandi.net	CNAME: ocsp.comodoca.com
ocsp.globessl.com	CNAME: ocsp.comodoca.com
ocsp.incommon-ecc.org	CNAME: ocsp.comodoca.com
ocsp.incommon-igtf.org	NS: ns0.comododns.com.
ocsp.incommon-rsa.org	NS: ns0.comododns.com.
OCSP.intel.com	CNAME: ocsp.comodoca.com
ocsp.marketware.eu	CNAME: ocsp.comodoca.com
ocsp.netsolssl.com	CNAME: ocsp.comodoca.com
ocsp.register.com	CNAME: ocsp.comodoca.com
ocsp.securecore-ca.com	NS: ns0.comododns.com.
ocsp.sgssl.net.	NS: ns0.comododns.com.
ocsp.trustasiassl.com.	NS: ns0.comododns.com.
ocsp.trust-provider.com	CNAME: ocsp.comodoca.com
ocsp.usertrust.com	NS: ns0.comododns.com.

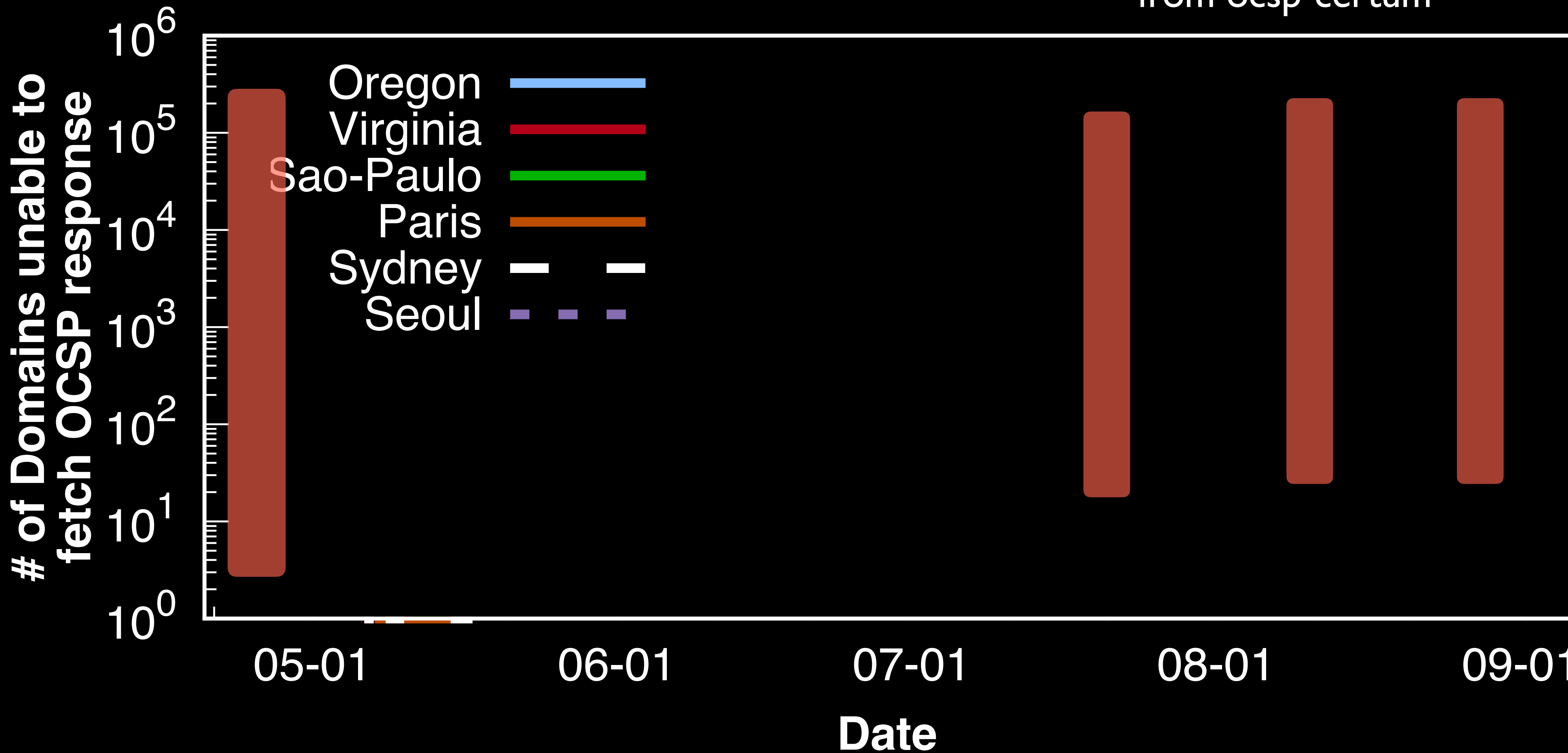
(1) Availability: Impact on the Web

Comodo
down for 2 hours

43 servers from wosign
5 servers from startssl

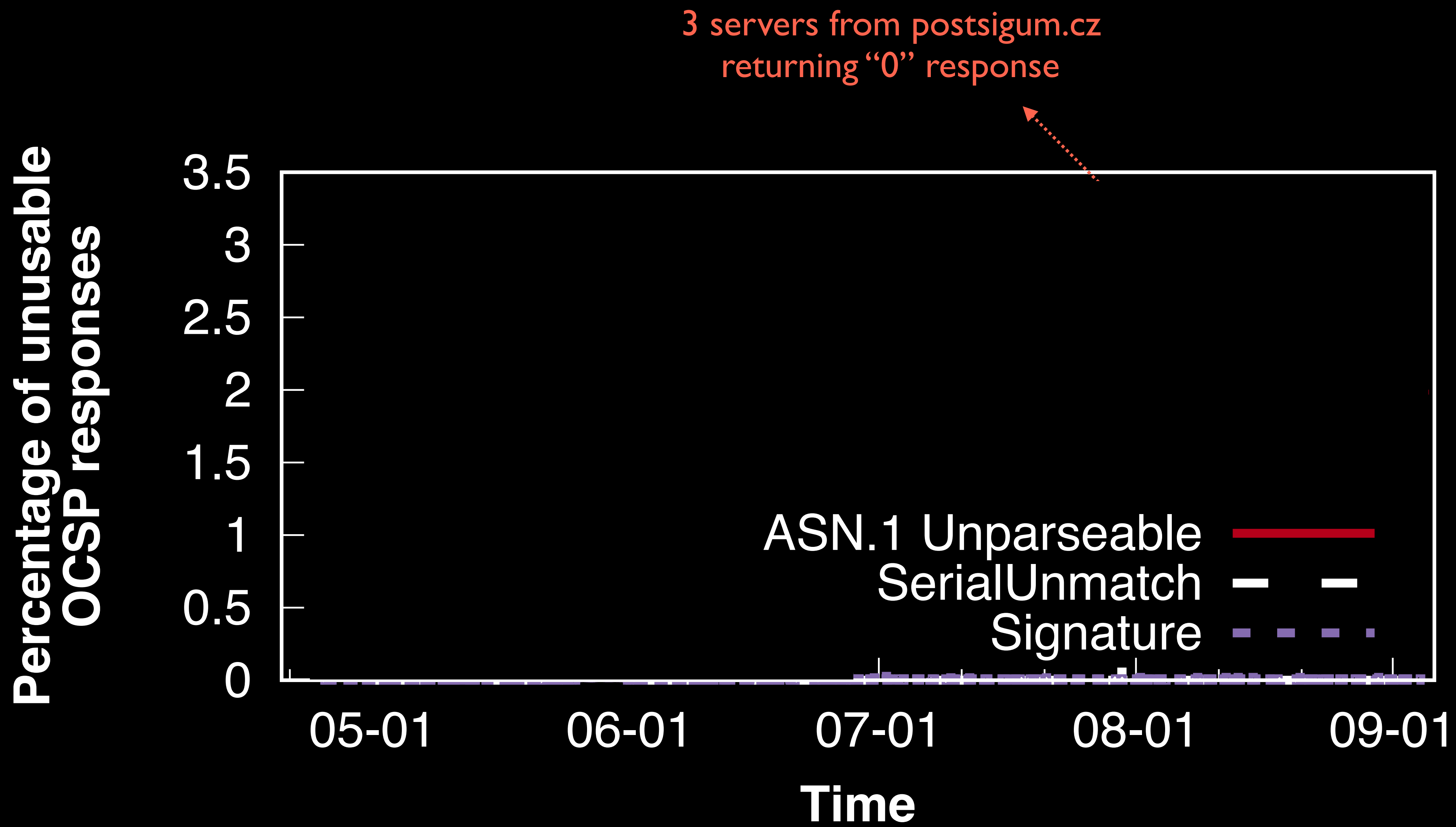
16 servers
from ocsp-certum

9 servers
from digicert



Availability OCSP responders are not fully reliable

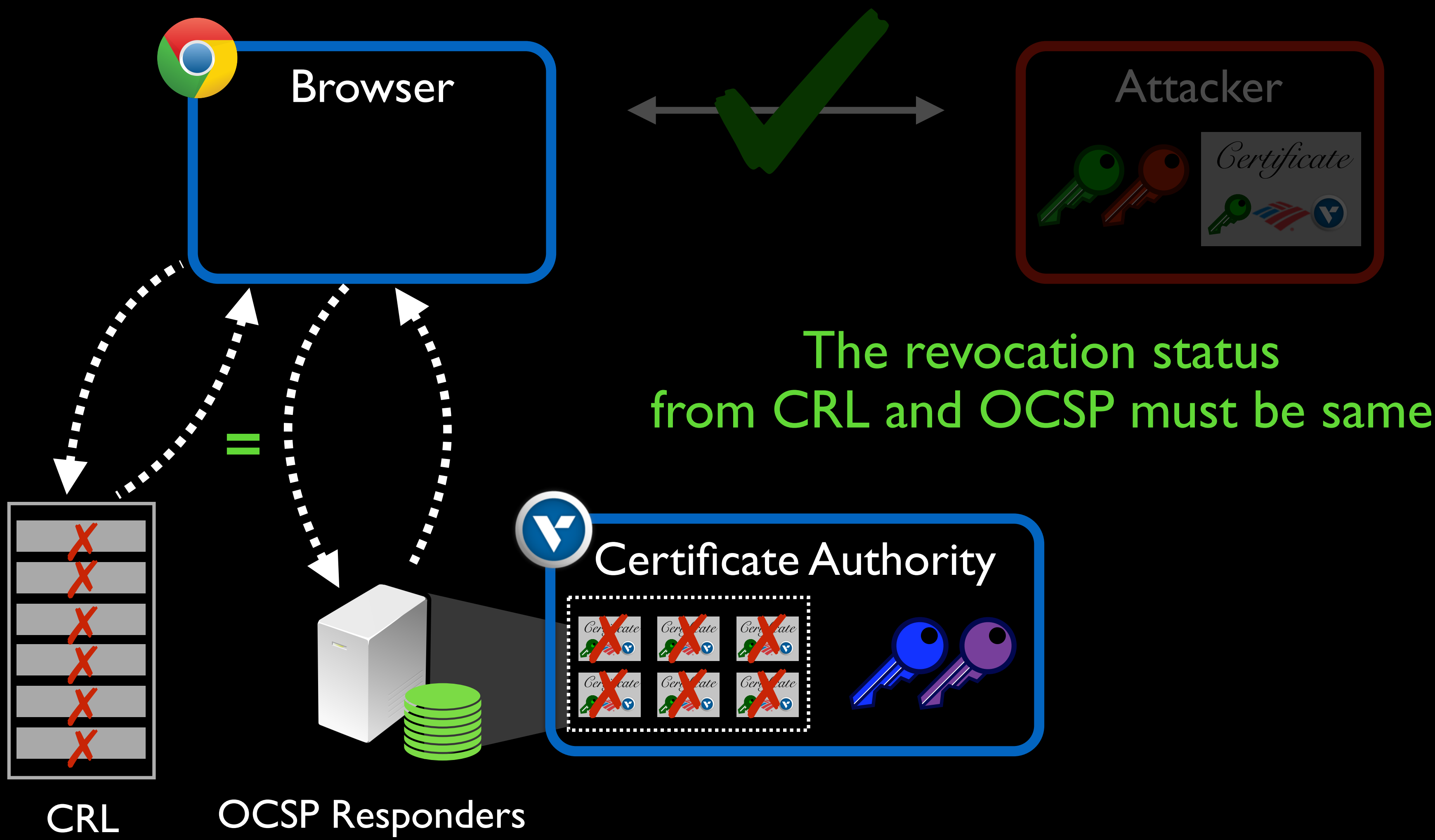
(2) Validity of the Response



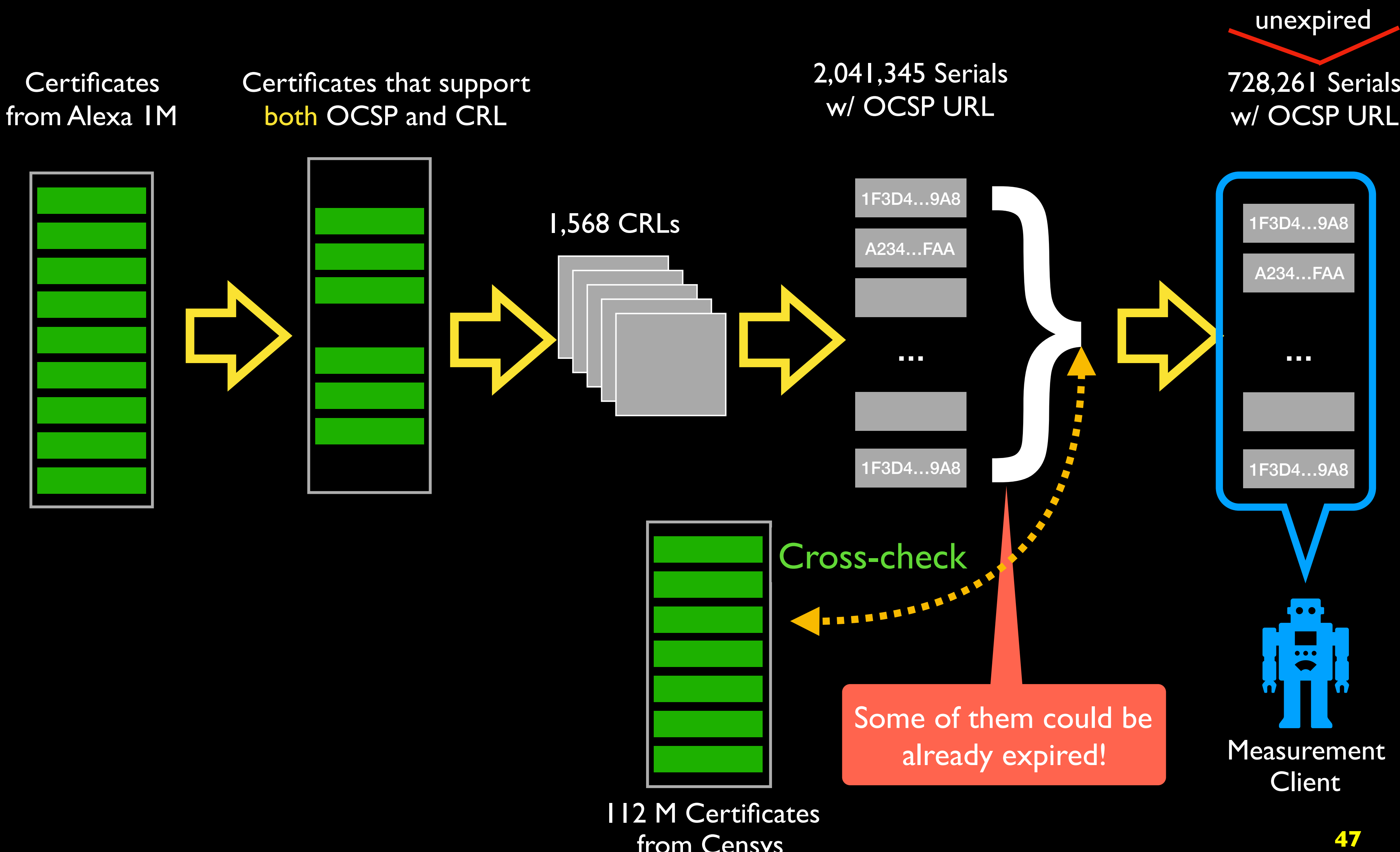
Validity

OCSP responses are (mostly) valid

(3) Consistency
OCSP vs. CRL



(3) Consistency
OCSP vs. CRL



(3) Consistency OCSP vs. CRL

OCSP URL	CRL	# of certificates where the OCSP response is		
		Unknown	Good	Revoked
ocsp.camerfirma.com	crl1.camerfirma.com/camerfirma_cserverii-2015.crl			
ocsp.quovadisglobal.com	crl.quovadisglobal.com/qvsslg3.crl			
ocsp.startssl.com	crl.startssl.com/sca-server1.crl			
ss.symcd.com	ss.symcb.com/ss.crl			
twcasslocsp.twca.com.tw/	sslserver.twca.com.tw/sslserver/securessl			
ocsp2.globalsign.com/gsalphasha2g2	crl2.alphassl.com/gs/gsalphasha2g2.crl			
ocsp.firmaprofesional.com	crl.firmaprofesional.com/infraestructura.crl			
…	…			

(3) Consistency
OCSP vs. CRL

OCSP URL	CRL	# of certificates where the OCSP response is		
		Unknown	Good	Revoked
ocsp.camerfirma.com	crl1.camerfirma.com/camerfirma_cserverii-2015.crl	0	7	369
ocsp.quovadisglobal.com	crl.quovadisglobal.com/qvsslg3.crl	0	1	514
ocsp.startssl.com	crl.startssl.com/sca-server1.crl	0	1	980
ss.symcd.com	ss.symcb.com/ss.crl	0	1	28,032
twcasslocsp.twca.com.tw/	sslserver.twca.com.tw/sslserver/securessl	0	1	122
ocsp2.globalsign.com/gsalphasha2g2	crl2.alphassl.com/gs/gsalphasha2g2.crl	**5,375**	0	0
ocsp.firmaprofesional.com	crl.firmaprofesional.com/infraestructura.crl	**11**	0	0
…	…	0	0	…

*"OCSP and PKI Management are two different platforms and are synchronized by means of some DDBB triggers that are failing in some circumstances. Meanwhile CRL management is easer and simple, OCSP should give information about any certificate serial number issued by *** and the amount of information transmitted between them. That's the source of this problem."*

Is the Web Ready for
OCSP Must-Staple?

Certificate authority

Web server

Browser

- ✓ Fetch and cache OCSP responses
- ✓ Handling errors

Web Server Methodology

(1) Performance — ? Prefetch OCSP response

(2) Caching — ? Cache OCSP response
? Respect nextUpdate* in cache

(3) Availability — ? Retain OCSP response on error

*Expiration date of a OCSP response

Web Server Administrator Result

	The Apache Software Foundation	NGINX
Prefetch OCSP response	✗	✗
Cache OCSP response	✓	✓
Respect nextUpdate in cache	✗	✓
Retain OCSP response on error	✗	✓

* Apache version 2.4.18 and Nginx version 1.13.12

Is the Web Ready for **OCSP Must-Staple?**

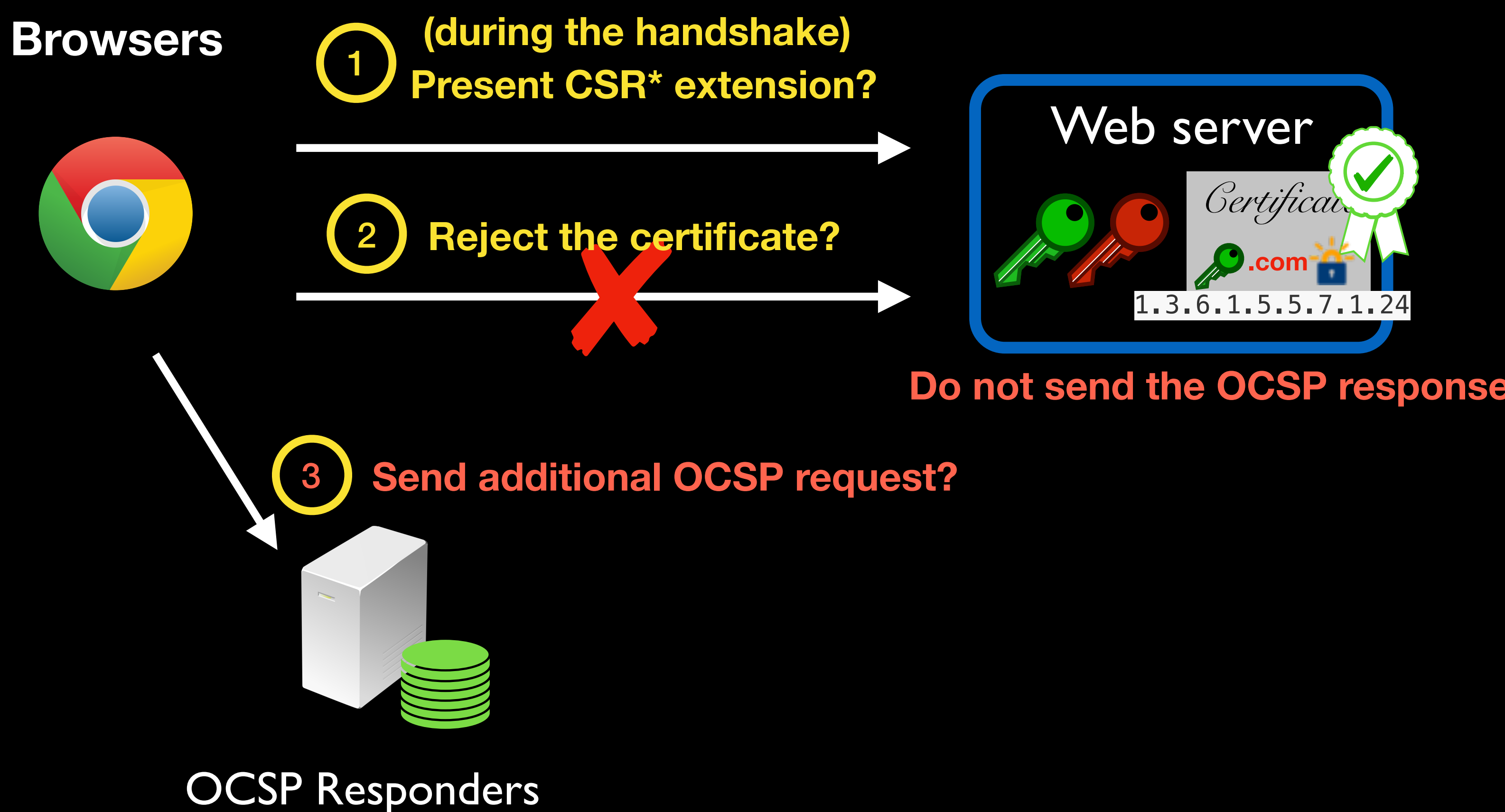
Certificate Authority

Website

Browser

- ✓ Understand the extension
- ✓ Present Certificate Status Request extension
- ✓ Reject the certificate if the response is not provided

Methodology



*CSR: Certificate Status Request

Methodology and Result

	Desktop Browsers (OS X, Linux, Windows)						Mobile Browsers			
	Chrome 66	Firefox 60	Opera	Safari	IE	Edge	Safari	Chrome	Firefox/ iOS	Firefox/ Android
Request OCSP Response	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Respect OCSP Must-Staple	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
Send own OCSP Request	✗	-	✗	✗	✗	✗	✗	✗	✗	-

Clients — Clients are largely not yet ready for OCSP Must-Staple

(the additional coding work necessary to support OCSP Must-Staple is likely not too significant)

*All tests were done on Ubuntu 16.04, Windows 10, OS X 10.12.6, iOS 11.3, and Android Oreo.

Conclusion

- Considering OCSP Must-Staple can operate only if each of the principals in the PKI performs correctly.
  - OCSP servers: not fully reliable
  - Web server softwares: not fully support
  - Browsers: not fully support

- But the bright side is
  - Only a few players need to take action to make it possible for web server administrators to begin enabling OCSP Must-staple
  - Much wider deployment of OCSP Must-Staple is an realistic and achievable goal

Some protocols

- HSTS (HTTP-STRICT-TRANSPORT-SECURITY)
  - "Strict-Transport-Security" Header
- HSTS-preloaded list
- HPKP (HTTP Public Key Pinning)
- SNI (Server Name Indication)
- Certificate Transparency